

DISS. ETH NO. 26842

**LOW-ALTITUDE CONTROL AND LOCAL  
RE-PLANNING STRATEGIES FOR SMALL  
FIXED-WING UAVs**

A thesis submitted to attain the degree of  
DOCTOR OF SCIENCES of ETH ZURICH  
(Dr. sc. ETH Zurich)

presented by

**THOMAS STASTNY**

B.Sc. Aerospace Engineering, University of Kansas  
M.Sc. Aerospace Engineering, University of Kansas

born on May 22, 1990  
citizen of the United States of America

accepted on the recommendation of

Prof. Dr. Roland Siegwart, Examiner  
Prof. Dr. Kostas Alexis, Co-examiner  
Prof. Dr. Lorenzo Marconi, Co-examiner

2020

Autonomous Systems Lab  
Department of Mechanical and Process Engineering  
ETH Zurich  
Switzerland

© 2020 Thomas Stastny. All rights reserved.



# Abstract

---

In recent years, small, easily manageable, operated, and maintained Unmanned Aerial Vehicles (UAVs) have become ubiquitous in an every-growing set of industrial, humanitarian, scientific, and commercial domains. For large-scale remote sensing and mapping applications, small fixed-wing platforms provide the advantages of longer range and higher speeds, with respect to their Micro-Aerial Vehicle (MAV) counterparts. However, today's fixed-wing UAVs are largely limited to a primitive set of basic waypoint following and pre-programmed tasks, with little awareness of the environment in which they fly, adaptability to changing conditions, or higher-level decision making capabilities. Particularly relevant to small fixed-wing mission profiles, is the required ability to operate safely near uncertain terrain while disturbed by possibly strong and turbulent wind fields. Enabling these activities entails the design of efficient, robust, and more adaptable motion planning and control algorithms which moreover adhere to the vehicle's restrictive dynamic flight envelope. The primary goal of this thesis is to develop practical control and local re-planning strategies for low-flying, small, fixed-wing UAVs with explicit awareness of these environmental hazards.

Part A of this thesis addresses the challenge of guiding small, low-speed fixed-wing aircraft in strong winds. Our first contribution in this part is the unique consideration of *excess winds*, i.e. wind speeds that exceed the vehicle's nominal airspeed, within a lateral-directional control law for fixed-wing UAVs. We develop a principled, nonlinear guidance law which guarantees convergence to a safe and stable vehicle configuration with respect to the wind field while preserving some tracking performance with respect to the path target. We then expand on this concept by including an energy efficient airspeed reference compensation logic, enabling not only mitigation, but also prevention or over-powering of excess winds which would otherwise cause the aircraft to "run-away". We emphasize heavily in this second iteration on field testing results, demonstrating track keeping errors of less than 1 m consistently maintained during gusting excess winds over various mountainous regions in Switzerland. The third component of this part revisits the *efficiency* of the airspeed references. A coupled approach to airspeed and heading reference commands is developed with a more principled consideration of airspeed reference minimization. The coupled method is compared against the previous decoupled approach in simulations showing both increased power-efficiency and tracking performance in static and dynamic winds.

Part B of this thesis operates as a road map to fast, environment-aware local re-planning for fixed-wing UAVs operating near terrain. We first delve into the practicalities of deploying a guidance level Nonlinear Model Predictive Controller

(NMPC) on a small fixed-wing platform. We develop new *control augment modeling* methods featuring reduced order models of the underlying low-level autopilot response and quasi-steady forces, simple parameter identification procedures, and open loop predictability on the order of tens of seconds, making our modeling approach suitable for long horizon NMPC. Through flight experiments, we demonstrate Dubins aircraft path segment tracking in three dimensions with wind speeds exceeding 50% of the vehicle’s airspeed, and further show a mock motor failure scenario. A particular focus is further spent on soft constraint formulation. The third component of this part reworks the developments of the first two towards our local re-planning formulation. Wind-aware reference trajectory generation is developed from the guidance logic in Part A for lateral-directional states, and vertical wind is included in the longitudinal guidance. Vision-based elevation mapping is utilized to provide a generalized 2.5D world representation to the aircraft. The map is bilinearly interpolated for height feedback, and we design an efficient ray casting approach for detection of forward (line of flight) and lateral occlusions. The occlusions are used to construct novel “relative” Euclidean Signed Distance Fields (RESDFs), which are a function of the relative velocity between the vehicle and obstacle. We further present a method of transforming the RESDFs into optimizable soft constraints for the objective function of the NMPC. A preliminary example of the full system acting to avoid an obstructing hillside is demonstrated in hardware-in-the-loop (HITL) simulation.

# Zusammenfassung

---

In den letzten Jahren sind kleine, leicht handhabbare, betriebene und gewartete Unbemannte Luftfahrzeuge (Engl. Unmanned Aerial Vehicles, UAVs) in einer ständig wachsenden Zahl von industriellen, humanitären, wissenschaftlichen und kommerziellen Bereichen allgegenwärtig geworden. Für gross angelegte Fernerkundungs- und Kartierungsanwendungen bieten kleine Starrflügel-Plattformen gegenüber ihren Pendants, den Micro-Aerial Vehicles (MAV), die Vorteile einer grösseren Reichweite und höherer Geschwindigkeiten. Die heutigen Starrflügel-UAVs beschränken sich jedoch weitgehend auf eine primitive Reihe grundlegender Aufgaben mit Wegpunktverfolgung und vorprogrammierten Aufgaben, mit limitiertem Bewusstsein für ihre Flugumgebung, geringer Anpassungsfähigkeit an sich ändernde Bedingungen, und nur rudimentärer Entscheidungsfähigkeit auf höherer Ebene. Besonders relevant für kleine Starrflügelmissionsprofile ist die erforderliche Fähigkeit, in unsicherem Gelände zuverlässig zu operieren, während sie durch möglicherweise starke und turbulente Windfelder gestört werden könnten. Um diese Aktivitäten zu ermöglichen, müssen effiziente, robuste und anpassungsfähigere Bewegungsplanungs- und Steuerungsalgorithmen entwickelt werden, die sich zudem an den restriktiven dynamischen Flugbereich des Luftfahrzeugs halten. Das primäre Ziel dieser Arbeit ist die Entwicklung praktischer Steuerungs- und lokaler Umplanungsstrategien für niedrig fliegende, kleine Starrflügel-UAVs mit explizitem Bewusstsein für Umweltgefahren.

Teil A dieser Arbeit befasst sich mit der Herausforderung, kleine, langsam fliegende Starrflügelflugzeuge bei starkem Wind zu steuern. Unser erster wissenschaftlicher Beitrag in diesem Teil ist die erstmalige Berücksichtigung Berücksichtigung von *überkritischen Winden*, d. h. die Nenngeschwindigkeit des Luftfahrzeugs überschreiten, im Rahmen eines Gesetzes zur Steuerung von Starrflügel-UAVs in Querrichtung. Wir entwickeln ein theoretisch fundiertes, nichtlineares Steuergesetz, das die Konvergenz zu einem sicheren und stabilen Systemzustand im Bezug auf das Windfeld garantiert und gleichzeitig die Leistungsfähigkeit in Bezug auf das Bahnziel beibehält. Wir erweitern dieses Konzept durch die Einbeziehung einer energieeffizienten Kompensationslogik für die Fluggeschwindigkeitsreferenzwerte, die nicht nur eine Entschärfung, sondern auch die Prävention oder das "Über-Powern" von Situationen mit überkritischem Wind ermöglicht, die sonst ein "Wegfliegen" des Flugzeugs verursachen würde. In dieser zweiten Iteration legen wir grossen Wert auf die Ergebnisse von Feldtests über verschiedenen Bergregionen der Schweiz, bei denen die Pfadabweichung selbst beim kurzzeitigen Auftreten von böigem Überwind durchgehend bei unter einem Meter gehalten werden konnte. Die dritte Komponente dieses Teils befasst sich erneut mit der Effizienz der Fluggeschwindigkeitsreferenzen.

Es wird ein gekoppelter Ansatz für Geschwindigkeits- und Kursreferenzkommandos entwickelt, wobei die Minimierung der Fluggeschwindigkeitsreferenzen theoretisch fundierter betrachtet wird. Die gekoppelte Methode wird in Simulationen mit dem vorherigen entkoppelten Ansatz verglichen, wobei sowohl eine erhöhte Effizienz als auch eine verbesserte Nachführleistung bei statischen und dynamischen Winden erkennbar wird.

Teil B dieser Arbeit dient als Fahrplan hin zu einer schnellen, umweltbewusste lokale Neuplanung von Flugtrajektorien unter Einbezug eines Umgebungsmodells, wie sie gerade für geländenah operierende UAVs von Bedeutung ist. Wir befassen uns zunächst mit den praktischen Möglichkeiten des Einsatzes eines Non-linear Model Predictive Controller (NMPC) für die Flugführung einer kleinen UAV Plattform. Wir entwickeln neue, durch Ansätze aus der Regelungstechnik erweiterte Modellierungsmethoden. Diese beinhalten reduzierte Modelle der grundlegenden Autopilotenreaktionen und quasistationären Kräfte, enthalten einfache Verfahren zur Parameteridentifikation, und erlauben eine Vorhersage des open-loop Flugführungsverhaltens des UAVs bis zu einem Zeithorizont von über 10 Sekunden. Dadurch sind diese Modellierungsmethoden auch für NMPCs mit langem Vorhersagehorizont nutzbar. Durch Flugexperimente demonstrieren wir die Verfolgung von Dubins-Flugbahnsegmenten in drei Dimensionen mit Windgeschwindigkeiten, die 50 Prozent der Fluggeschwindigkeit des UAVs übersteigen. Darüber hinaus zeigen wir ein Szenario mit einem nachgeahmten Motorausfall. Ein besonderer Schwerpunkt wird weiterhin auf die Formulierung weicher Randbedingung gelegt. Die dritte Komponente dieses Teils überarbeitet die Entwicklungen der ersten beiden im Hinblick auf unsere Methode zur lokalen Neuplanung von Trajektorien. Methoden zur Berechnung von Referenztrajektorien unter Beachtung eines Windmodells werden für die lateralen Trajektorienkomponenten basierend auf Teil A dieser Arbeit entwickelt, während für die longitudinale Flugführung ein Modell des vertikalen Winds einbezogen wird. Die auf Kamerabildern und computergestütztem Sehen basierende Höhenkartierung wird verwendet, um dem Flugzeug eine verallgemeinerte 2,5D-Weltdarstellung zu liefern. Die Karte wird zum Bezug exakter Höhendaten bilinear interpoliert. Ein effizienter Raycasting Ansatz zur Erkennung von vorwärtsgerichteten (Fluglinie) und seitlichen Okklusionen wird präsentiert. Die Okklusionen werden zur Konstruktion neuartiger "relativer" euklidischer Distanzfelder (s.g. Euclidean Signed Distance Fields or (R)ESDFs) verwendet, die eine Funktion der Relativgeschwindigkeit zwischen UAV und Hindernis sind. Wir stellen ferner eine Methode zur Umwandlung der RESDFs in optimierbare weiche Randbedingung für die Kostenfunktion der NMPC vor. Die Effektivität des Gesamtsystems zur Vermeidung von Hindernissen wird, beispielhaft für einen die Flugroute des UAVs obstruierenden Berghang, in einer Hardware-In-The-Loop (HITL) Simulation demonstriert.

# Acknowledgements

---

During the last years, it's been my pleasure to work with so many extremely motivated, talented, and genuine people. ASL is like nowhere else I've seen in academia, and I'm very grateful for the time I've been allowed to spend here.

I would like to first express my deep gratitude to Prof. Dr. Roland Siegwart for taking a chance on a random guy from Kansas City and giving me the opportunity to pursue a PhD in ASL, and further for always encouraging the fixed-wing team's "risky" research ideas, no matter where in the world they took us! I'd like to further thank Prof. Dr. Kostas Alexis for pushing me in the first year of my PhD and for remaining a reliable colleague and friend, even once we switched places across the ocean. Thanks to Prof. Dr. Lorenzo Marconi for agreeing to co-examine my thesis, as well as sharing your theoretical expertise for the windy control work even early on. I also cannot speak highly enough of Dr. Juan Nieto, your unending encouragement truly kept me going, especially towards the latter stages of thesis write-up.

To the "older" (..we're all still 25, right?) fixed-wing guys.. Dr. Philipp Oettershagen, Amir Melzer, Thomas Mantel, Timo Hinzmann – I've learned an incredible amount working with each of you, and would never have finished this PhD without your support and engagement. I've enjoyed all the time we spent in the field, even when watching planes fly in circles.. and more circles... and even more circles. To the new(er) fixed(and/or tilt-able)-wings, Florian Achermann, David Rohr, Dr. Nick Lawrance, it's been great working with you, and I believe with your new ideas and motivations the fixed-wing team is heading in exciting new directions. To Michael Hugentobler, thank you for getting me home safely from Greenland! A very special thanks to Jonas Langenegger, Tizian Steiger, and Yves Allensprach, for always safely landing the plane(s), no matter how many crazily windy mountain tops I took you to! And extra thanks specifically to Herr Rainer Lotz for following the team to the ends of the earth to pilot the AtlantikSolar. Dr. Igor Gilitschenski, your impenetrable positivity throughout even the lowest times of project integration weeks was outstanding. Dr. Guillaume Jouvét, the ideas and glaciological applications you've brought to ASL have sparked some of my favorite projects and challenges throughout my PhD, it was a pleasure working with you!

I also want to thank the many extremely dedicated and talented students I had the opportunity to supervise, with a special mention to Gian Heinrich, David Rohr, Silvan Fuhrer, Carl Olsson, Adyasha Dash, and Luca Furieri, who took on challenging thesis topics as well as my high expectations and exceeded them substantially. It was a pleasure working with *and* learning from each of you.

Sebastian Verling, I always enjoyed "talking shop" about nerdy aerospace stuff;

## Acknowledgements

---

thanks for posing so many thought provoking discussions, and of course the invites to Friday beer. Dr. Mina Kamel, it was a pleasure shooting back and forth ideas about UAV control, I always learned something new when we spoke. Dr. Helen Oleynikova, thank you for being a good friend and forcing me to take my eyes off of work on occasion. Dr. Dario Bellicoso, Thursday night jam sessions were my favorite two hours of the week. Vassilios Tsounis, Adyasha Dash, and Karen Bodie, thanks for going with me to concerts, out for coffee, beer, or food (*not mutually exclusive*), and keeping me sane!

I'm sure I've likely missed some people.. six years is a long time.. but thank you to everyone else who made this time great!

Last, I'm truly grateful for my friends and family back home, who put up with my distance and still ping in often to make sure I'm doing okay.

Summer, 2020

*Thomas Stastny*

## Financial Support

The research leading to these results has received funding from the European Commission project SHERPA (#600958) under the 7th Framework Programme, the Federal office armasuisse S+T under project number n°050-45 and the ETH Foundation Grant ETH-12 16-2 (Sun2Ice Project).

# Contents

---

|   |            |
|---|------------|
| <b>Abstract</b>   | <b>i</b>   |
| <b>Zusammenfassung</b>  | <b>iii</b> |
| <b>Acknowledgements</b>   | <b>v</b>   |
| <b>Preface</b>  | <b>1</b>   |
| <b>1 Introduction</b>   | <b>3</b>   |
| 1.1 Motivation . . . . .  | 3          |
| 1.2 Approach . . . . .  | 5          |
| <b>2 Contributions</b>  | <b>9</b>   |
| 2.1 Part A: Handling High Winds . . . . .   | 9          |
| 2.2 Part B: Model Predictive Control & Local Re-Planning . . . . .  | 11         |
| 2.3 List of Publications . . . . .  | 14         |
| 2.4 List of Supervised Students . . . . .   | 17         |
| <br>  |            |
| <b>A. HANDLING HIGH WINDS</b>   | <b>21</b>  |
| <br>  |            |
| <b>Paper I: Gone with the Wind: Nonlinear Guidance for Small Fixed-Wing Aircraft in Arbitrarily Strong Windfields</b> | <b>23</b>  |
| 1 Introduction . . . . .  | 24         |
| 2 Problem Definition . . . . .  | 25         |
| 3 The Lower Wind Case . . . . .   | 30         |
| 4 The Higher Wind Case . . . . .  | 34         |
| 5 Continuity . . . . .  | 39         |
| 6 Flight Results . . . . .  | 42         |
| 7 Conclusions . . . . .   | 42         |
| 8 Appendix: Stability Proof for High Wind Case . . . . .  | 43         |
| <br>  |            |
| <b>Paper II: On Flying Backwards: Preventing Run-Away of Small, Low-Speed, Fixed-Wing UAVs in Strong Winds</b>        | <b>49</b>  |
| 1 Introduction . . . . .  | 50         |
| 2 Bearing Feasibility . . . . .   | 52         |

|   |  |            |
|---|--|------------|
| 3   | Directional Guidance . . . . .   | 55         |
| 4   | Airspeed Reference Compensation . . . . .                                | 61         |
| 5   | Flight Experiments . . . . .   | 62         |
| 6   | Discussion & Future Work . . . . .                                       | 66         |
| <b>Paper III: Wind Fighting Efficiency Revisited: Coupling Airspeed and Heading Guidance for Fixed-wing UAVs</b>      |  | <b>71</b>  |
| 1   | Background & Scope . . . . .   | 72         |
| 2   | Windy Guidance Fundamentals . . . . .                                    | 72         |
| 3   | Coupling Heading and Airspeed References . . . . .                       | 76         |
| 4   | Simulation . . . . .   | 81         |
| 5   | Conclusions & Future Work . . . . .                                      | 85         |
| <br>  |  |            |
| <b>B. MODEL PREDICTIVE CONTROL &amp; LOCAL RE-PLANNING</b>  |  | <b>89</b>  |
| <br>  |  |            |
| <b>Paper IV: Nonlinear MPC for Fixed-Wing UAV Trajectory Tracking: Implementation and Flight Experiments</b>          |  | <b>91</b>  |
| 1   | Introduction . . . . .   | 92         |
| 2   | System Overview . . . . .  | 94         |
| 3   | Flight Dynamics & Identification . . . . .                               | 96         |
| 4   | Nonlinear Model Predictive Control . . . . .                             | 100        |
| 5   | Simulations & Flight Experiments . . . . .                               | 102        |
| 6   | Conclusions & Future Work . . . . .                                      | 106        |
| <br>  |  |            |
| <b>Paper V: Nonlinear Model Predictive Guidance for Fixed-Wing UAVs Using Identified Control Augmented Dynamics</b>   |  | <b>109</b> |
| 1   | Introduction . . . . .   | 110        |
| 2   | Control Augmented Modeling . . . . .                                     | 111        |
| 3   | System Identification . . . . .  | 115        |
| 4   | Control Formulation . . . . .  | 121        |
| 5   | Flight Experiments . . . . .   | 128        |
| 6   | Conclusions & Future Work . . . . .                                      | 134        |
| <br>  |  |            |
| <b>Paper VI: An Outlook on Environment-Aware Local Re-Planning for Safe Near-Terrain Operation of Fixed-wing UAVs</b> |  | <b>139</b> |
| 1   | Introduction . . . . .   | 140        |
| 2   | “Outsmarting” the Optimizer: Safe Objectives for Long Horizons . . . . . | 144        |
| 3   | Terrain Interpretation from Vision-based Elevation Maps . . . . .        | 153        |
| 4   | NMPC Problem Definition . . . . .  | 161        |
| 5   | Hardware-in-the-Loop Simulation . . . . .                                | 165        |
| 6   | Conclusions . . . . .  | 173        |
| <br>  |  |            |
| <b>9</b>  | <b>Conclusion &amp; Outlook</b>  | <b>177</b> |
| 9.1   | Contributions . . . . .  | 177        |



|                                |            |
|--------------------------------|------------|
| 9.2 Research Outlook . . . . . | 178        |
| <b>Bibliography</b>            | <b>180</b> |
| <b>Curriculum Vitae</b>        | <b>191</b> |



# Preface

---

This is a cumulative doctoral thesis and as such consists of the most relevant publications. In addition to the individual publications an overarching introduction is provided in Chapter 1. We start with explaining the relevance of this thesis, followed by the objectives and the approach taken to fulfill these. For each contributing publication we explain how it embeds into the overall goals of this thesis and highlight the relevance of the research work in Chapter 2. Furthermore, we show how each paper is related to our other publications. The publications are then grouped into two parts – Part A and Part B – and attached. An additional unpublished technical brief is included as a third paper in each part, which should be seen as an addendum to its preceding content, presenting some afterthoughts of the published work, as well as some newer preliminary, conceptual developments and an outlook on where the research specific to the respective part is headed. We close this thesis by a summary of the achievements and provide a cumulative outlook for future research directions in Chapter 9.



# Introduction

---

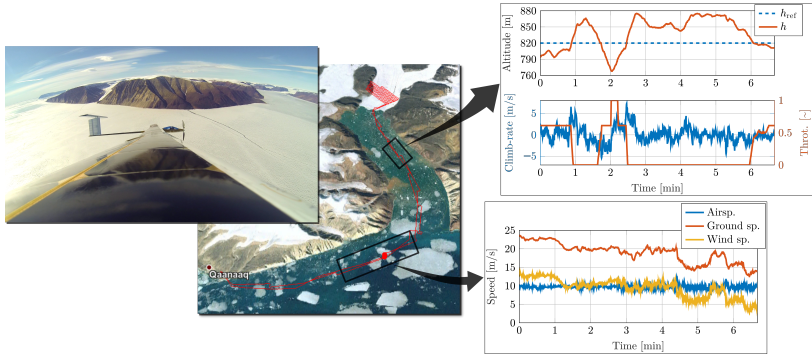
## 1.1 Motivation

In recent years, small, easily manageable, operated, and maintained Unmanned Aerial Vehicles (UAVs) have become ubiquitous in an every-growing set of industrial, humanitarian, scientific, and commercial domains. Widespread use of small UAVs for aerial photogrammetric mapping includes precision agriculture [69], disaster management [61], or some limited scientific applications in mountainous terrain [8]. For large-scale remote sensing and mapping applications, small fixed-wing platforms provide the advantages of longer range and higher speeds, with respect to their Micro-Aerial Vehicle (MAV) counterparts. However, today's fixed-wing UAVs are largely limited to a primitive set of basic waypoint following and pre-programmed tasks, with little awareness of the environment in which they fly, adaptability to changing conditions, or higher-level decision making capabilities.

The Autonomous Systems Lab (ASL) has pioneered the development and application of small-scale Low Altitude Long Endurance (LALE) solar-powered UAVs, such as AtlantikSolar. The platform has demonstrated perpetual endurance capabilities and holds an 81.5 hour flight endurance world record [60]. More recently, AtlantikSolar conducted remote sensing missions in the Arctic for glacier monitoring<sup>1</sup> where it experienced heavy rotors within the fjords, pushing the aircraft up to 50 meters above and below the commanded flight path; climb and descent rates reaching  $\pm 7 \text{ m s}^{-1}$  despite automatic throttle cuts and spoiler deployment and full throttle, respectively. Strong valley winds over the sea-ice, at times, exceeded the aircraft's maximum airspeed, though, fortunately, in a tail-wind orientation, see Fig. 1.1. We note that all severe wind conditions experienced were *not* forecast by meteorological services, nor is it currently possible to accurately predict low altitude, turbulent terrain induced flows. Especially important in the case of near terrain scanning

---

<sup>1</sup><https://sun2ice.ethz.ch/>



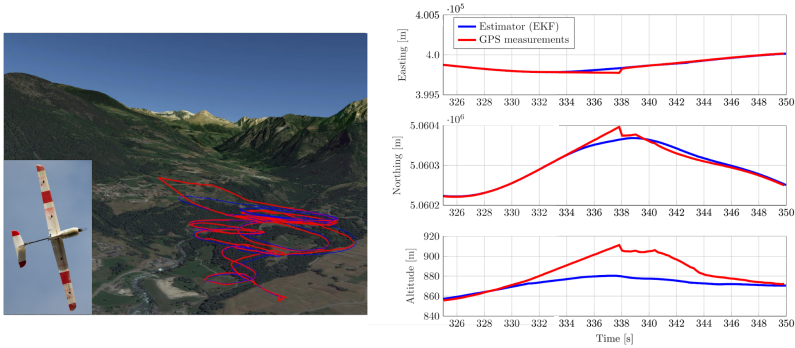
**Figure 1.1:** Influence of terrain-based wind fields on small UAVs. (Left) Atlantik-Solar en-route to Bowdoin Glacier, Northwest Greenland, over the Arctic sea-ice. (Center) Bowdoin scanning mission flight trajectory. Flight logs – heavy vertical up- and downdrafts causing large altitude deviations (Top-Right) and tail-wind greater than the vehicle’s maximum cruise speed (Bottom-Right).

missions, where mountain walls may lie on either side of the vehicle, slope winds and rotors are a paramount danger to the aircraft.

Winds are not the only concern for small UAVs operating in the wild. During field trials on a mountain side in Italy for the EU (FP7) Robotic Search and Rescue project SHERPA<sup>2</sup>, ASL’s fixed-wing research platform Techpod experienced perturbed GPS measurements in flight, see Fig. 1.2. While the multi-sensor Extended Kalman Filter (EKF) [42] kept the plane stable for enough time that the safety pilot could manually take-over in this incident, a longer GPS disturbance or even a GPS outage would have most probably led to loss of airframe. This incident illuminates the necessity for on-board perception which may inform the aircraft of its local surroundings in the event that global references are lost. In addition to perception, any scenario with unexpected, possibly abrupt, physical obstructions will require fast local re-planning (deviating from the global path set out for the mission) which still respects the particularly restricted vehicle dynamics and constraints of fixed-wing aircraft, e.g. minimum flight speeds, turn/climb rates, and aerodynamic stall.

To reach beyond today’s limited applications such as small-scale aerial photography, and to penetrate into applications that can be of pivotal societal and commercial use, UAVs need to be capable of fully autonomous beyond-visual-line-of-sight (BVLOS) operations. Safe flight in challenging terrain through detailed on-board knowledge of the UAV’s environment is a key ingredient to this next step

<sup>2</sup>SHERPA– Smart collaboration between Humans and ground-aerial Robots for imProving rescuing activities in Alpine environments (<http://www.sherpa-project.eu/sherpa/>)



**Figure 1.2:** Perturbed GPS measurements at a mountainside near Isollaz, Italy. While the multi-sensor EKF could save the plane in this incident, a longer GPS disturbance or even a GPS outage would have most probably led to a UAV loss.

in UAV technology. Towards addressing these issues, we focus our attention on two primary objectives:

1. Efficient guidance for small fixed-wing UAVs in strong, turbulent wind fields.
2. Fast, environment-aware local re-planning of vehicle trajectories with consideration of aircraft dynamics and constraints.

Particularly, we place a heavy emphasis on *field-able* algorithms, requiring lightweight computational solutions and consideration of the availability and quality of sensory inputs on-board small fixed-wing platforms as well as how each component will interface with the overall system. The goal of this thesis is, thus, to **develop practical control and local re-planning strategies for low-flying, small, fixed-wing UAVs with explicit awareness of environmental hazards.**

## 1.2 Approach

To accomplish our objectives, we group this thesis into two parts. Part A focuses on the design of lightweight control laws for efficient and safe flight in strong winds. Part B focuses first on the development of high-level model predictive controllers for fixed-wing UAVs considering vehicle dynamics and constraints, and second, on incorporating environmental awareness into the controllers including wind mitigation concepts from Part A and generalized terrain feedback from online vision-based elevation maps.

### Part A: Handling High Winds

Our aim in this part is the development of a lightweight control law capable of safely and efficiently guiding a fixed-wing aircraft to a provided reference while explicitly considering a potentially strong and variable local wind field in which it is flying. We address the issue in three successive stages:

**Directional Guidance** In this stage, the condition of *excess winds* is first addressed in the development of a nonlinear path following guidance law with the objective of, *ideally*, driving the vehicle to the reference position target while trading off *safely* orienting to minimize run-away from the track when winds do not allow convergence to the ideal objective. Command continuity is ensured during all state transitions while varying between the two objectives. The algorithm is programmed in C++ on a small flight controller and limited flight tests are conducted to demonstrate its functionality a small fixed-wing UAV.

**Airspeed Reference Compensation** While directionally guiding the aircraft with the controller from the first stage *mitigates* the effect of excess winds, in the case that extra power is available on the aircraft, airspeed references may be increased above the nominal value to further reduce and/or *prevent* run-away from the track. This stage expands the previous control law with a parallel airspeed reference compensation logic, incrementing the commanded airspeed as appropriate when the bearing would have otherwise been infeasible. Extensive field tests are performed in mountainous regions of Switzerland towards evaluating the functionality and effectiveness of the controller in strong, gusty winds.

**Coupling Heading and Airspeed References** In this final stage, we re-evaluate the *efficiency* of the commanded airspeed references in the prior, developing a principled approach to airspeed reference minimization, coupling the heading and airspeed guidance logic. A simulation study is conducted towards preliminarily evaluating efficiency and track keeping performance in static and dynamic winds, with respect to the decoupled version.

### Part B: Model Predictive Control & Local Re-planning

Our overall goal in this part is to provide small fixed-wing UAVs with the ability to safely fly near terrain with the ability to adapt to abrupt, unplanned obstructions along their mission plan. This objective involves tight coordination between on-board, real-time perception, mapping, planning, and control algorithms, especially considering the particularly restrictive dynamics and constraints of fixed-wing vehicles. We focus our efforts mostly on system modeling and motion planning, then end with a preliminary look into how on-board environmental perception may be interpreted and utilized directly in the control loop.



**Nonlinear Model Predictive Control for Fixed-Wing UAVs** Leveraging recent advances in auto-code generation of fast, efficient embedded nonlinear solvers, e.g. the ACADO Toolkit [30] or FORCES<sup>3</sup>, we design a guidance-level Nonlinear Model Predictive Controller (NMPC) for fixed-wing UAVs. A control augmented modeling approach is developed which takes a “middle road” between full classical identification of aircraft aerodynamics and kinematics-only model-free formulations by considering low-order models of the low-level autopilot’s attitude response dynamics and identifying quasi-steady, momentless aerodynamic and thrust models. The resulting models are easy to identify and predictive on the order of tens of seconds, allowing their utilization in longer NMPC horizons with the trust that optimized trajectories will be adequately tracked. An objective is formulated for simultaneous tracking of Dubins Aircraft path segments and airspeed setpoints while respecting angle of attack constraints. Flight experiments are conducted to evaluate Dubins aircraft path segment tracking in three dimensions with wind speeds exceeding 50 % of the vehicle’s airspeed, and further show how the controller handles a mock motor failure scenario.

**Environment-Aware Local Re-Planning** We approach the problem of fixed-wing terrain avoidance by re-evaluating and expanding the previously developed long-horizon NMPC, incorporating *generalized terrain feedback* to the objective function. A wide-baseline stereo vision solution suitable for fixed-wing UAVs [28] is used to provide real-time sensory input of the environment which is fused to the cells of a 2.5D elevation map, providing a scalable representation for large outdoor environments. With an elevation map being populated on the fly, we implement an efficient means of interpreting the grid through height look-ups and radial ray casting, constructing local “relative” Euclidean Signed Distance Field (RESDF), which considers not only the distance to terrain in the map, but also the relative approach velocity of the aircraft, only building distance fields where the aircraft is currently planning to fly. The RESDF is directly incorporated in the NMPC objective function, providing a *generalized* representation of obstacles which may adapt over time and space, without the need for any further shape abstraction (e.g. planes or spheres) or tedious collision checking through point clouds along trajectories. Wind-aware reference trajectory generation is designed using the developments in Part A. We preliminarily evaluate our full system in hardware-in-the-loop (HITL), forcing the vehicle to avoid an abruptly detected hillside obstructing the globally planned path.

---

<sup>3</sup><https://www.embotech.com/FORCES-Pro>



# Chapter 2

## Contributions

---

This chapter details the contributions of each of the papers presented as part of this cumulative thesis. We describe the context of the paper at the time of publication, the scientific contribution, and how it relates to the rest of the thesis and our other publications. Additionally, we provide a list of all authored and co-authored papers published over the course of our doctoral studies as well as a summary of supervised student theses and projects.

### 2.1 Part A: Handling High Winds

The following papers relate to the development of guidance strategies, robust to strong winds, for small, fixed-wing UAVs.

#### Paper I

Luca Furieri, Thomas Stastny, Lorenzo Marconi, Roland Siegwart, and Igor Gilitschenski, “Gone with the Wind: Nonlinear Guidance for Small Fixed-Wing Aircraft in Arbitrarily Strong Windfields”. In *American Control Conference*, 2017.

#### Context

Small, fixed-wing UAVs are especially susceptible to strong winds. However, until recent years, widespread deployment of automatic fixed-wing platforms was limited enough to where the idea that an aircraft may move negatively with respect to the ground was not considered. Winds were at most considered minor, low frequency disturbances in aeronautical literature to be rejected either by robust control techniques or simply feeding back inertial velocities to guidance loops, where the wind is implicitly contained within this measurement. Flight in more inclement weather was simply avoided.

With the goal of deploying small UAVs, e.g. within the context of the AtlantikSolar project<sup>1</sup>, for remote long-range and/or long-endurance missions, so came the inevitability that the aircraft would encounter a variety of environmental conditions, specifically high winds and gusts. To be able to use such systems safely and efficiently, it was necessary to consider these conditions directly at the guidance level of control, explicitly taking into account online wind estimates.

### Contribution

This paper developed, to our best knowledge, the first guidance law for fixed-wing aircraft considering the unique condition of *excess* winds, i.e. the case where wind speed rises above the vehicle's nominal airspeed. We design a principled, nonlinear guidance strategy guaranteeing convergence to a safe and stable vehicle orientation with respect to the wind, while still preserving some tracking performance with respect to the desired position target. We evaluate the control law in simulations and limited flight experiments, confirming the feasibility of the approach. Stability of the closed loop system is analyzed with a formal geometric argument for the excess wind case, and nonlinear phase portraits for the lower-wind case.

### Interrelations

This paper forms the base on which Papers II and III build and improve upon.

### Paper II

Thomas Stastny and Roland Siegart, "On Flying Backwards: Preventing Run-Away of Small, Low-Speed, Fixed-Wing UAVs in Strong Winds". In *IEEE International Conference on Robots and Intelligent Systems (IROS)*, 2019.

### Context

While the developed controller in Paper I showed promise in mitigating the effect of strong winds, in the case the aircraft may have remaining energy available, the airspeed reference could be increased above the nominal value to further reduce, or even prevent run-away from the track. Our aim in this paper was to design an efficient guidance strategy which returned the vehicle to the track.

### Contribution

In this work, an efficient airspeed compensation logic was developed for small, fixed-wing UAVs enabling either mitigation, prevention, or over-powering of excess winds. Enhancements to the directional guidance from Paper I were made including a reformulation of the "bearing feasibility" function, which indicates *how* feasible a given bearing command is in the present wind state, and new condition independent

---

<sup>1</sup><http://www.atlantiksolar.ethz.ch/>

tuning strategies. The combined parallel guidance laws were, at the time, the first example in literature of an algorithm considering both excess wind conditions as well as providing a means of fully preventing run-away from the track. The paper concludes with extensive flight experimentation in strong, turbulent winds in mountainous regions of Switzerland.

### **Interrelations**

This paper extended the baseline controller from Paper I, and is compared against in newer developments in Paper III.

## **Paper III**

Thomas Stastny, “Wind Fighting Efficiency Revisited: Coupling Airspeed and Heading Guidance for Fixed-wing UAVs”. In *Unpublished*, 2020.

### **Context**

This brief is an unpublished work written as a closing chapter to Part A. It includes some afterthoughts of the two published works, as well as some newer preliminary, conceptual developments and an outlook on where the research specific to this part is headed. The brief revisits the efficiency of the proposed airspeed compensation logic in Paper II, exploring how taking a coupled approach to airspeed and heading commands can not only reduce energy demands, but may further improve tracking performance in highly variable winds.

### **Contribution**

This brief developed a new coupled formulation for generating airspeed and heading commands in high winds, taking a more principled consideration of airspeed reference minimization. Simulation comparisons of the Paper II formulation and new coupled approach were provided for static and dynamic wind scenarios which indicate support for the hypothesis that the coupled approach both reduces energy demands and further improves tracking performance.

### **Interrelations**

This brief revisits the developments in Paper II, providing a conceptual development of a new coupled airspeed and heading command logic and analyzing it against the efficiency claims of Paper II. The formulation designed in this work is used in the development of wind-aware trajectory generation in Paper VI.

## **2.2 Part B: Model Predictive Control & Local Re-Planning**

The following papers relate to the design and implementation of nonlinear model predictive control strategies for fixed-wing UAVs.

### Paper IV

Thomas Stastny, Adyasha Dash, and Roland Siegwart, “Nonlinear MPC for Fixed-Wing UAV Trajectory Tracking: Implementation and Flight Experiments”. In *AIAA Guidance, Navigation, and Control (GNC) Conference*, 2017.

#### Context

At the time of this publication, real-time implementation of NMPCs on MAVs was just starting to appear, with newly improved solver runtimes available from C++ auto-code-generation frameworks like the ACADO toolkit [30]. However, deployment on fixed-wing platforms outside of simulation was non-existent. The early NMPC work on MAVs such as in [35] and [56] showed promise for generating control solutions to dynamic and complex trajectory tracking scenarios with consideration of the vehicle dynamics and constraints, and we wished to investigate if similar results could be achieved for fixed-wing platforms.

The majority of simulation-based fixed-wing literature on NMPC either used purely kinematic models, assuming lower-level loops would sufficiently track the higher-level commands, or assumed a full aerodynamic model of the vehicle, down to actuators, was available and implemented whole body control. Using kinematic models alone in the NMPC would neglect the underlying low-level control response, leading to poor tracking of high-level commands, and full classical system identification of aircraft either entails expensive hours in the wind tunnel, or potentially risky open-loop maneuvering to obtain data from flight logs. We instead investigated performing simple identification procedures for the stabilized low-level system and integrating the autopilot response dynamic into the model of the high-level NMPC as a means to ensure generated control solutions would be adequately tracked by the low-level autopilot and further reduce model complexity, with respect to full aerodynamic models, such that longer horizons could be achieved.

#### Contribution

This paper provides an approach for identification and modeling of the control augmented roll channel dynamics for a small fixed-wing UAV and their incorporation in a high-level, lateral-directional NMPC. A simulation comparison of different horizon lengths in strong wind is provided. Flight experiments were conducted for path following of Dubins Car segments, and were the first real-world flight demonstration of a NMPC on fixed-wing UAVs in literature.

#### Interrelations

This paper was the inspiration to continue with NMPC as the controller of choice for future local re-planning work in our subsequent papers V and VI. The implementation of the controller was further detailed and incorporated in a tutorial book chapter [37].

## Paper V

Thomas Stastny and Roland Siegwart, “Nonlinear Model Predictive Guidance for Fixed-Wing UAVs Using Identified Control Augmented Dynamics”. In *International Conference on Unmanned Aerial Systems (ICUAS)*, 2018.

### Context

In order to generate fixed-wing motion plans which perform more dynamic maneuvers in three dimensions, the model and objective function from Paper IV needed to be extended. However, longitudinal aircraft dynamics are quite dissimilar from the lateral-directional, and common OTS autopilots stabilize pitch, while a higher-level control loop generates pitch reference and throttle commands for airspeed and altitude stabilization. We developed new, coupled control augmented model, also considering quasi-steady aerodynamic and thrust forces and an objective function suitable for stabilizing airspeed, constraining angle of attack, and driving the vehicle to 3D Dubins Aircraft paths.

### Contribution

This paper develops an approach to modeling and identification of coupled lateral-directional and longitudinal control augmented dynamics and quasi-steady aerodynamic and thrust forces for a conventional fixed-wing platform with a widely available OTS autopilot in the loop, utilizing a standard sensor suite. The models are obtained using simple and safe parameter identification procedures, and achieve open loop predictability on the order of tens of seconds, justifying their utility in long horizon NMPC formulations. An objective function is developed for simultaneous airspeed stabilization, path following, and soft constraint handling (angle of attack). We validate the approach in several flight experiments including path following of helix and connected Dubins Aircraft segments in high winds as well as a motor failure scenario.

### Interrelations

This paper extends Paper IV to three dimensions. The developed approaches for control augmented modeling and the ROS implementation are further applied for airborne wind energy in [83]. Paper VI reformulates the developments of this work, reducing control augmented model complexity, improving thrust modeling, and incorporating new objectives for terrain avoidance.

## Paper VI

Thomas Stastny, Timo Hinzmam, and David Rohr, “An Outlook on Environment-Aware Local Re-Planning for Safe Near-Terrain Operation of Fixed-wing UAVs”. In *Unpublished*, 2020.

### Context

This brief is an unpublished work written as a closing chapter to Part B. It includes some afterthoughts of the two published works, as well as some newer preliminary, conceptual developments and a detailed outlook on where the research specific to this part is headed. In this brief, we showcase some ongoing work in its preliminary stages of development towards utilizing the developed control structures in Papers IV and V, reconceptualizing the work as an environment-aware local re-planner.

### Contribution

In this brief, we conceptually evaluate the potential of incorporating obstacles into the NMPC formulation in the form of *generalized terrain feedback*. Wind-aware reference trajectory generation is developed using the formulation presented in III. Real-time vision-based elevation maps are used to provide a 2.5D world representation to the aircraft, providing a “simulated” environment for the local replanner to optimize within. The map is bilinearly interpolated for height feedback, and we propose an efficient ray casting approach for detection of forward (line of flight) and lateral occlusions. Novel “relative” Euclidean Signed Distance Fields (RESDFs) are formulated as a function of the relative velocity between the vehicle and occlusion and a method to translate these RESDFs into optimizable soft constraints within the NMPC objective function is developed. A preliminary example of the full local re-planning system acting to avoid an obstructing hillside is demonstrated in hardware-in-the-loop (HITL) simulation, all while respecting soft angle of attack and airspeed constraints.

### Interrelations

This brief draws from the developments of Papers III, IV, and V, reconceptualizing the work as a unified environment-aware local re-planning formulation.

## 2.3 List of Publications

In the context of the author’s doctoral studies the following publications were achieved. They are sorted chronologically and grouped by publication type.

### Journal Papers

- D. Malyuta, C. Brommer, D. Hentzen, **T. Stastny**, R. Siegwart, and R. Brockers. “Long-duration Fully Autonomous Operation of Rotorcraft Unmanned Aerial Systems for Remote-sensing Data Acquisition”. *Journal of Field Robotics (JFR)*. Vol. 37(1). pp. 137–157. 2020.
- D. Rohr, **T. Stastny**, S. Verling, and R. Siegwart. “Attitude and Cruise Control of a VTOL Tiltwing UAV”. *IEEE Robotics and Automation Letters (RA-L)*. Vol. 4(3). pp. 2683–2690. 2019.



- T. Hinzmann, **T. Stastny**, C. Cadena, R. Siegwart, and I. Gilitschenski. “Free LSD: Prior-free Visual Landing Site Detection for Autonomous Planes”. *IEEE Robotics and Automation Letters (RA-L)*. Vol. 3(3). pp. 2545–2552. 2018.
- P. Oettershagen, **T. Stastny**, T. Hinzmann, K. Rudin, T. Mantel, A. Melzer, B. Wawrzacz, G. Hitz, and R. Siegwart. “Robotic Technologies for Solar-powered UAVs: Fully Autonomous Updraft-aware Aerial Sensing for Multiday Search-and-rescue Missions”. *Journal of Field Robotics (JFR)*. Vol. 35(4). pp. 612–640. 2018.
- P. Oettershagen, A. Melzer, Mantel, K. Rudin, **T. Stastny**, B. Wawrzacz, T. Hinzmann, S. Leutenegger, K. Alexis, and R. Siegwart. “Design of Small Hand-launched Solar-powered UAVs: From Concept Study to a Multi-day World Endurance Record Flight”. *Journal of Field Robotics (JFR)*. Vol. 34(7). pp. 1352–1377. 2017.

## Book Chapters

- M. Kamel, **T. Stastny**, K. Alexis, R. Siegwart. “Model Predictive Control for Trajectory Tracking of Unmanned Aerial Vehicles Using Robot Operating System”. *Robot Operating System (ROS), The Complete Reference (Volume 2)*. pp. 3–39. 2017.

## Peer Reviewed Conference Papers

- C. Olsson, S. Verling, **T. Stastny**, and R. Siegwart. “Full Envelope System Identification of a VTOL Tailsitter UAV”. *AIAA Guidance, Navigation, and Control (GNC) Conference*. 2021. *Accepted for publication*.
- M. Harms, N. Kaufmann, F. Rockenbauer, N. Lawrance, **T. Stastny**, and R. Siegwart. “Differential Sweep Attitude Control for Swept Wing UAVs”. *International Conference on Unmanned Aircraft Systems (ICUAS)*. 2020.
- **T. Stastny** and R. Siegwart. “On Flying Backwards: Preventing Run-away of Small, Low-speed, Fixed-wing UAVs in Strong Winds”. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2019.
- D. Hentzen, **T. Stastny**, R. Siegwart, and R. Brockers. “Disturbance Estimation and Rejection for High-Precision Multirotor Position Control”. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2019.
- **T. Stastny**, E. Ahbe, M. Dangel, and R. Siegwart. “Locally Power-optimal Nonlinear Model Predictive Control for Fixed-wing Airborne Wind Energy”. *American Control Conference (ACC)*. 2019.

- S. Fuhrer, S. Verling, **T. Stastny**, and R. Siegwart. “Fault-tolerant Flight Control of a VTOL Tailsitter UAV”. IEEE International Conference on Robotics and Automation (ICRA). 2019.
- J. Lee, T. Muskardin, C. Pacz, P. Oettershagen, **T. Stastny**, I. Sa, R. Siegwart, and K. Kondak. “Towards Autonomous Stratospheric Flight: A Generic Global System Identification Framework for Fixed-Wing Platforms”. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2018.
- **T. Stastny** and R. Siegwart. “Nonlinear Model Predictive Guidance for Fixed-wing UAVs Using Identified Control Augmented Dynamics”. International Conference on Unmanned Aircraft Systems (ICUAS). 2018.
- L. Furieri, **T. Stastny**, L. Marconi, R. Siegwart, and I. Gilitschenski. “Gone with the Wind: Nonlinear Guidance for Small Fixed-wing Aircraft in Arbitrarily Strong Windfields”. American Control Conference (ACC). 2017. **Best Paper Award**.
- S. Verling, **T. Stastny**, G. Bättig, K. Alexis, and R. Siegwart. “Model-based Transition Optimization for a VTOL Tailsitter”. IEEE International Conference on Robotics and Automation (ICRA). 2017.
- Y. Demitri, S. Verling, **T. Stastny**, A. Melzer, and R. Siegwart. “Model-based Wind Estimation for a Hovering VTOL Tailsitter UAV”. IEEE International Conference on Robotics and Automation (ICRA). 2017.
- **T. Stastny**, A. Dash, and R. Siegwart. “Nonlinear MPC for Fixed-wing UAV Trajectory Tracking: Implementation and Flight Experiments”. AIAA Guidance, Navigation, and Control(GNC) Conference. 2017.
- P. Oettershagen, A. Melzer, T. Mantel, K. Rudin, **T. Stastny**, B. Wawrzacz, T. Hinzmann, K. Alexis, and R. Siegwart. “Perpetual Flight with a Small Solar-powered UAV: Flight Results, Performance Analysis and Model Validation”. IEEE Aerospace Conference. 2016.
- T. Hinzmann, **T. Stastny**, G. Conte, P. Doherty, P. Rudol, M. Wzorek, I. Gilitschenski, E. Galceran, and R. Siegwart. “Collaborative 3D Reconstruction Using Heterogeneous UAVs: System and Experiments”. International Symposium on Experimental Robotics (ISER). 2016.
- P. Doherty, J. Kvarnström, P. Rudol, M. Wzorek, G. Conte, C. Berger, T. Hinzmann, **T. Stastny**. “A Collaborative Framework for 3D Mapping Using Unmanned Aerial Vehicles”. International Conference on Principles and Practice of Multi-Agent Systems. 2016.

- Oettershagen, **T. Stastny**, T. Mantel, A. Melzer, K. Rudin, P. Gohl, G. Agamennoni, K. Alexis, and R. Siegwart. “Long-Endurance Sensing and Mapping using a Hand-Launchable Solar-Powered UAV”. Field and Service Robotics (FSR). 2015.
- A. Vempati, G. Agamennoni, **T. Stastny**, and R. Siegwart. “Victim Detection from a Fixed-Wing UAV: Experimental Results”. International Symposium on Visual Computing (ISVC). 2015.

## Patents

- M. Arigoni, R. Simpson, S. Fuhrer, P. Beardsley, D. Mammolo, M. Burri, M. Bischoff, **T. Stastny**, L. Rodgers, D. Krummenacher, and R. Siegwart. “Vehicles Configured For Navigating Surface Transitions”. US Patent 10,464,620. 2019.

## 2.4 List of Supervised Students

Throughout the author’s doctoral studies a large effort was spent supervising students. For projects that contributed to a publication, a reference to the page of the corresponding citation in Section 2.3 is given.

### Master Thesis

*Master student, 6 months full time*

- Gian Heinrich (Spring 2020): “Learning to Stall: Using in-air Pressure Data to Identify, Characterize, and Control Fixed-Wing Aircraft Stall”
- Dario Panzuto (Spring 2019): “Development of a Drone-Released GPS Logging Unit for In Situ Monitoring of Fast-Flowing Glaciers”
- Michael Imobersteg (Spring 2019): “Autonomous Retraction Control of Rigid Wing Airborne Wind Energy Systems”
- Adrian Ruckli (Spring 2019): “A Deep Learning-Based Approach to Flexible-Wing Modeling for Wide-Baseline Stereo Vision on Fixed-Wing Unmanned Aerial Vehicles”
- David Rohr (Autumn 2018): “Tiltwing VTOL Flight-Control System” [p. 14]
- Manuel Dangel (Autumn 2018): “Nonlinear Model Predictive Control for Fixed-Wing Airborne Wind Energy” [p. 15]
- Silvan Fuhrer (Spring 2018): “Fault-Tolerant Flight Control of a VTOL Tailsitter UAV” [p. 16]

- Daniel Hentzen (Spring 2018): “Robust Precision Landing for Multirotor UAS” [p. 15]
- Matthias Epprecht (Spring 2018): “Bayesian Optimization Based Automatic Controller Tuning for Fixed-Wing UAVs”
- Fabian Bachmann (Spring 2018): “Airflow Sensing and Control for Small Fixed-Wing UAVs”
- Adyasha Dash (Autumn 2017): “Magnetometer Error Sources and Mitigation for Small UAVs”
- Danylo Malyuta (Spring 2017): “Guidance, Navigation, Control and Mission Logic for Quadrotor Full-cycle Autonomy” [p. 14]
- Jongseok Lee (Spring 2017): “Towards Autonomous Stratospheric Flight: A Generic Global System Identification Framework for Fixed-Wing Platforms” [p. 16]
- Carl Olsson (Spring 2017): “Full Envelope System Identification of a VTOL Tailsitter UAV” [p. 15]
- Youssef Demitri (Autumn 2016): “Robust Operation of a Hovering VTOL Tailsitter UAV in Wind”
- Luca Furieri (Autumn 2016): “Geometric Versus Model Predictive Control Based Guidance Algorithms for Fixed-Wing UAVs in the Presence of Very Strong Wind Fields” [p. 16]
- Pavel Vechersky (Autumn 2016): “Development of a Comprehensive, Hardware-in-the-loop Simulation Environment for Fixed-wing Unmanned Aircraft”
- Nicolas El Hayek (Autumn 2016): “Ridge Lift Exploitation for Small Unmanned Fixed-Wing Aircraft”
- Felix Renaut (Autumn 2015): “Vision-Based Autonomous Landing Site Detection for Fixed-Wing Unmanned Aerial Vehicles”
- Markus Thurnherr (Autumn 2014): “Model Predictive Control for Fixed-Wing UAVs Using LPV-Models”

### **Semester Project**

*Master student, 3-4 months part time*

- Ramon Flammer (Spring 2020): “Analysis of the Aerodynamic Effects on Coaxial Rotor Configurations”

- Matthias Studiger (Spring 2020): “Model Predictive Longitudinal Cruise Control of a VTOL Tilt-Wing UAV”
- Gian Heinrich (Spring 2019): “Optimizing Deep-stall Landing Maneuvers on Small Fixed-wing UAVs”
- Cla Galliard (Spring 2019): “Nonlinear Total Energy Control System for Longitudinal Control of an Aircraft”
- Eloi Roset (Spring 2019): “Wing Pressure Distribution Measurement for Small Fixed-Wing UAVs”
- Marko Maljkovic (Spring 2019): “Online Neural Network Based Model Identification of a Fixed-Wing UAV”
- Jasmin Fischli (Autumn 2018): “Online Neural Network Based System Identification for Fixed-wing UAVs”
- Silvan Fuhrer (Spring 2017): “Sampling Based Motion Planning for Fixed-Wing UAV System Identification”
- Youssef Demitri (Spring 2016): “Wind Estimation for a Hovering VTOL Tailsitter UAV” [p. 16]
- Léonard Schai (Spring 2016): “Design, Manufacturing and Calibration of a Miniature Vector Probe for Small, Low Speed, Unmanned Aircraft”
- Adyasha Dash (Spring 2016): “High Level Predictive Control of Fixed-Wing UAVs using Low Order System Models” [p. 16]
- Gregory Bättig (Spring 2016): “Transition Optimization for a VTOL Tailsitter” [p. 16]
- Samuel Dobmann (Spring 2016): “Design, Modeling, and Control of an Autonomous Underwater Glider”
- Ricardo Zurfluh (Spring 2015): “Robust Airspeed Sensing in Harsh Environmental Conditions”

## Bachelor Thesis

*Bachelor student, 3-4 months part time*

- Severin Laasch (Spring 2020): “Impact Compensation on Multicopters using Model Predictive Control”
- Marvin Harms & Noah Kaufmann (Spring 2019): “Differential Sweep Attitude Control for Fixed-Wing UAVs” [p. 15]

- Friedrich Rockenbauer (Spring 2019): “Cross-Validation of Simulation Models and Flight Test Data for Fixed-Wing UAVs”
- Marco Ruggia (Spring 2018): “Design and Analysis of Small Scale Angle of Attack Sensors”
- Jonas Peter (Spring 2018): “System Identification for an Airborne Wind Energy Aircraft”
- Adrian Schneebeli (Spring 2018): “Optimization of the Back-Transition Maneuver for a VTOL UAV”
- Lucas Streichenberg (Spring 2018): “Improved Hover Controller for an AWE System”
- Cla Galliard (Spring 2017): “Spherical Guidance and Control for an Airborne Wind Energy System”
- Gabriel König (Spring 2017): “Implementation and Validation of the Transition for a VTOL Airborne Wind Energy System”
- David Krummenacher (Spring 2015): “Floor to Wall Transitions for a Wall Racing Robot”
- Michael Arigoni (Spring 2015): “Control of a Wall Racing Robot for Agile Ground Maneuvers”

### **Studies on Mechatronics**

*Bachelor student, 1-2 months part time*

- Flurin Schwerzmann (Autumn 2017): “On the Suitability of the Three-Surface Configuration for Small UAVs”

### **Focus Project**

*6-8 Bachelor students, 1 year project full time*

- Dipper (Fall 2018-2019): <https://dipper.ethz.ch/index.html>
- ftero2 (Fall 2017-2018): <https://www.ftero.ch/>
- ftero (Fall 2016-2017): <https://www.ftero.ch/>
- VertiGo (Fall 2014-2015): <https://www.vertigoproject.ch/>

## **Part A**

### HANDLING HIGH WINDS





# Gone with the Wind: Nonlinear Guidance for Small Fixed-Wing Aircraft in Arbitrarily Strong Windfields

Luca Furieri, Thomas Stastny, Lorenzo Marconi, Roland Siegwart, and Igor Gilitschenski

## Abstract

The recent years have witnessed increased development of small, autonomous fixed-wing Unmanned Aerial Vehicles (UAVs). In order to unlock widespread applicability of these platforms, they need to be capable of operating under a variety of environmental conditions. Due to their small size, low weight, and low speeds, they require the capability of coping with wind speeds that are approaching or even faster than the nominal airspeed. In this paper we present a principled nonlinear guidance strategy, addressing this problem. More broadly, we propose a methodology for the high-level control of non-holonomic unicycle-like vehicles in the presence of strong flowfields (e.g. winds, underwater currents) which may outreach the maximum vehicle speed. The proposed strategy guarantees convergence to a safe and stable vehicle configuration with respect to the flowfield, while preserving some tracking performance with respect to the target path. Evaluations in simulations and a challenging real-world flight experiment in very windy conditions confirm the feasibility of the proposed guidance approach.

## 1 Introduction

In recent years, the use of small fixed-wing Unmanned Aerial Vehicles (UAVs) has steadily risen in a wide variety of applications due to increasing availability of open-source and user-friendly autopilots, e.g. Pixhawk Autopilot [68], and low-complexity operability, e.g. hand-launch. Fixed-wing UAVs have particular merit in long-range and/or long-endurance remote sensing applications. Research in ETH Zürich's Autonomous Systems Lab (ASL) has focused on Low-Altitude, Long-Endurance (LALE) solar-powered platforms capable of multi-day, payload-equipped flight [58], already demonstrating the utility of such small platforms in real-life humanitarian applications [57]. UAVs autonomously navigating large areas for long durations will inherently be exposed to a variety of environmental conditions, namely, high winds and gusts. With respect to larger and/or faster aircraft, wind speeds rarely reach a significant ratio of the vehicle airmass-relative speed. Conversely, wind speeds rising close to the vehicle maximum airspeed, and even surpassing it during gusts, is a frequent scenario when dealing with a small-sized, low-speed aircraft.

Usually in aeronautics, windfields are handled as an unknown low-frequency disturbance which may be dealt with either using robust control techniques, e.g. loop-shaping in low-level loops, or simply including integral action within guidance-level loops. In the case of LALE vehicles, maximizing flight time would further require the efficient use of throttle, thus limiting airspeed bandwidth. In order to be able to use such systems safely and efficiently in a wide range of missions and different environments, it is necessary to take care of such situations directly at the guidance level of control, explicitly taking into account online wind estimates.

A standard approach to mitigate the effect of wind on path following tasks is to exploit the measurements of the inertial ground speed of the aircraft, which inherently includes wind effects, see [65], [55]. Another approach is to take the wind explicitly into account, either by available wind measurements [49] [74] or by exploiting a disturbance observer, as in [44]. Another possibility is described in [5], where adaptive backstepping is used to get an estimate for the direction of the wind.

As to wind compensation techniques, a possible approach is vector fields [55] [3]. In [55], an approach based on vector fields is used to achieve asymptotic tracking of circular and straight-line paths in the presence of non negligible persisting wind disturbances: vector fields are proposed for specific curves (e.g. straight lines, circles). This requires switching the commands when the target path is defined as the union of different parts, which makes the algorithm less uniform and its implementation trickier. Tuning of vector fields is also known to be difficult, as highlighted in [3].

Another popular approach is based on nonlinear guidance. The strategy proposed in [66], utilises a *look-ahead* vector for improved tracking of upcoming paths, introducing a predictive effect. The law was extended in [10] to any 3D path in the non-windy case. Great advantages of this law are that it is easy and intuitive to tune, the magnitude of the guidance commands is always upper-bounded, and it has flexibility in the set of feasible initial conditions.

The main contribution of this paper is a simple, safe, and computationally efficient guidance strategy for navigation in *arbitrarily* strong windfields. To our knowledge, there is no existing guidance law directly considering the case of the wind speed being higher than the airspeed. The provided design strategy relies on the solution provided in [10] in absence of wind whose choice for the look-ahead vector will be properly modified in order to cope with arbitrarily strong wind speed.

**Notation.** We shall use the bold notation to denote vectors in  $\mathbb{R}^3$ . For a vector  $\mathbf{v} \in \mathbb{R}^3$ ,  $\hat{\mathbf{v}}$  denotes the associated versor and  $\|\mathbf{v}\|$  the euclidean norm. For two vectors  $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{R}^3$ , their scalar and cross products are respectively indicated by  $\mathbf{v}_1 \cdot \mathbf{v}_2 \in \mathbb{R}$  and  $\mathbf{v}_1 \times \mathbf{v}_2 \in \mathbb{R}^3$ .

## 2 Problem Definition

As we wish to extend the results obtained in [10], it is useful to define the same mathematical framework. To have a better insight, we will clearly define the control problem for each different scenario, and define a state-space nonlinear formulation. This will allow us to state a robust control problem, which will be useful for analysis in future work.

### 2.1 The Frenet-Serret framework for autonomous guidance

The position of the vehicle is denoted by  $\mathbf{r}_M$ , which is a vector of  $\mathbb{R}^3$  expressed with respect to an inertial reference frame denoted by  $\mathcal{F}_i$  and described by an orthonormal right-hand basis  $(\mathbf{i}, \mathbf{j}, \mathbf{k})$ . We assume that  $(\mathbf{i}, \mathbf{j})$  are co-planar with the flight plane, with  $\mathbf{k}$  orthogonal to such a plane. The emphasis of the work is on developing a controller able to cope with strong wind. The latter is a vector  $\mathbf{w} \in \mathbb{R}^3$  assumed to be constant and to lie on the flight plane, namely with zero component along  $\mathbf{k}$ . The vectors  $\mathbf{v}_G \in \mathbb{R}^3$  and  $\mathbf{a}_M \in \mathbb{R}^3$  in the plane  $(\mathbf{i}, \mathbf{j})$  denote the ground speed and acceleration of the vehicle, the dynamics of the latter is described by

$$\dot{\mathbf{r}}_M = \mathbf{v}_G, \quad \dot{\mathbf{v}}_G = \mathbf{a}_M. \quad (3.1)$$

Considering flight through a moving airmass,  $\mathbf{v}_G = \mathbf{v}_M + \mathbf{w}$ , in which  $\mathbf{v}_M$  is the vehicle airmass-relative speed (or airspeed). Note that, since  $\mathbf{w}$  is constant,  $\dot{\mathbf{v}}_G = \dot{\mathbf{v}}_M$ . The acceleration  $\mathbf{a}_M$  represents the control input.

From a geometric viewpoint, the *vehicle path* is defined as the union of each  $\mathbf{r}_M(t)$  for every time  $t$ . At each  $t \geq 0$  the vehicle path can be geometrically characterised in terms of the *unit tangent vector*, the *actual orientation*, the *tangential acceleration*, the *normal acceleration*, the *tangential acceleration*, the *unit normal vector* and the *curvature* of the vehicle path respectively defined as

$$\begin{aligned}
 \hat{\mathbf{T}}_G(t) &:= \frac{\mathbf{v}_G(t)}{\|\mathbf{v}_G(t)\|}, & \hat{\mathbf{T}}_M(t) &:= \frac{\mathbf{v}_M(t)}{\|\mathbf{v}_M(t)\|}, \\
 \mathbf{a}_M^T(t) &:= (\mathbf{a}_M(t) \cdot \hat{\mathbf{T}}_M(t)) \hat{\mathbf{T}}_M(t), \\
 \mathbf{a}_M^N(t) &:= (\hat{\mathbf{T}}_M(t) \times \mathbf{a}_M(t)) \times \hat{\mathbf{T}}_M(t), \\
 \hat{\mathbf{N}}_M(t) &:= \frac{\mathbf{a}_M^N(t)}{\|\mathbf{a}_M^N(t)\|}, & k_M(t) &:= \frac{\|\mathbf{a}_M(t)\|}{\|\mathbf{v}_G(t)\|^2}.
 \end{aligned} \tag{3.2}$$

We observe that the unit normal vector is defined only for values of the acceleration such that  $\|\mathbf{a}_M^N(t)\| \neq 0$ . Furthermore, all the previous vectors lie in the plane  $(\mathbf{i}, \mathbf{j})$ .

Having in mind the application to fixed-wing UAVs, we will consider the vehicle to be unicycle-like, i.e. its speed norm  $\|\mathbf{v}_M\|$  will remain unchanged in time and it will be then guided through normal acceleration commands  $\mathbf{a}_M^N$ . In other words, the control law for  $\mathbf{a}_M$  will be chosen in such a way that  $\mathbf{a}_M^T(t) \equiv 0$ . According to this, and by bearing in mind (3.2), (3.1) can be rewritten as

$$\dot{\mathbf{r}}_M(t) = v_M^* \hat{\mathbf{T}}_M(t) + \mathbf{w}(t), \quad v_M^* \dot{\hat{\mathbf{T}}}_M(t) = \mathbf{a}_M^N(t) \tag{3.3}$$

in which  $v_M^*$  denotes the (constant) value of  $\|\mathbf{v}_M\|$ .

Inspired by [10], the *desired (planar) path* is a continuously differentiable space curve in the plane spanned by  $(\mathbf{i}, \mathbf{j})$  represented by  $\mathbf{p}(l)$ ,  $l \in \mathbb{R}$ , with associated a *Frenet-Serret* frame composed of three orthonormal vectors  $(\hat{\mathbf{T}}_p(l), \hat{\mathbf{N}}_p(l), \hat{\mathbf{B}}_p(l))$ , a curvature  $\kappa_p(l)$  and a torsion  $\tau_p(l)$ . In the following we let  $s \in \mathbb{R}$  the arc length along the curve  $p(\cdot)$  defined as

$$s(l) = \int_{l_0}^l \left\| \frac{d\mathbf{p}(\ell)}{d\ell} \right\| d\ell.$$

The desired path is thus endowed with the *Frenet-Serret* dynamics given by

$$\begin{pmatrix} \hat{\mathbf{T}}_p'(s) \\ \hat{\mathbf{N}}_p'(s) \\ \hat{\mathbf{B}}_p'(s) \end{pmatrix} = \begin{pmatrix} 0 & \kappa_p(s) & 0 \\ -\kappa_p(s) & 0 & \tau_p(s) \\ 0 & -\tau_p(s) & 0 \end{pmatrix} \begin{pmatrix} \hat{\mathbf{T}}_p(s) \\ \hat{\mathbf{N}}_p(s) \\ \hat{\mathbf{B}}_p(s) \end{pmatrix} \tag{3.4}$$

in which we used the notation  $(\cdot)'$  to denote the derivative with respect to  $s$ . As in [10], we define the "footprint" of  $\mathbf{r}_M$  on  $\mathbf{p}$  at time  $t$  as the closest point of  $\mathbf{r}_M(t)$  on  $\mathbf{p}(l)$  defined as

$$\mathbf{r}_P(s(t)) := \arg \min_{\mathbf{r} \in \mathbf{p}} \|\mathbf{r}_M(t) - \mathbf{r}\|.$$

The point  $P$  on the desired path is identified by  $l_P$ , which is the value of the curve parameter  $l$  at the closest projection. The unit tangent vector, the unit normal vector, the unit binormal vector, the curvature and the torsion of the desired path at the point  $P$  will be indicated in the following as  $\hat{\mathbf{T}}_P := \hat{\mathbf{T}}_p(l_P)$ ,  $\hat{\mathbf{N}}_P := \hat{\mathbf{N}}_p(l_P)$ ,

$\hat{\mathbf{B}}_P := \hat{\mathbf{B}}_P(l_P)$ ,  $\kappa_P := \kappa_P(l_P)$  and  $\tau_P := \tau_P(l_P)$ . They are all functions of time through  $s(t)$ . By bearing in mind (3.4), it turns out that the vehicle dynamics induce a *Frenet-Serret* dynamics on the desired path which is given by

$$\begin{pmatrix} \dot{\hat{\mathbf{T}}}_P(t) \\ \dot{\hat{\mathbf{N}}}_P(t) \\ \dot{\hat{\mathbf{B}}}_P(t) \end{pmatrix} = \dot{s}(t) \begin{pmatrix} 0 & \kappa_P(t) & 0 \\ -\kappa_P(t) & 0 & \tau_P(t) \\ 0 & -\tau_P(t) & 0 \end{pmatrix} \begin{pmatrix} \hat{\mathbf{T}}_P(t) \\ \hat{\mathbf{N}}_P(t) \\ \hat{\mathbf{B}}_P(t) \end{pmatrix} \quad (3.5)$$

in which  $\dot{s}(t)$  can be easily computed as (see Lemma 1 and Appendix B in [10]).

$$\dot{s}(t) = \frac{(v_M^* \hat{\mathbf{T}}_M(t) + \mathbf{w}) \cdot \hat{\mathbf{T}}_P(t)}{1 + \kappa_P(t)[(\mathbf{r}_P(t) - \mathbf{r}_M(t)) \cdot \hat{\mathbf{N}}_P(t)]}.$$

The (ideal) desired control objective is to asymptotically steer the position of the vehicle  $\mathbf{r}_M(t)$  to the footprint  $\mathbf{r}_P(s(t))$  by also aligning the unitary tangent vectors  $\hat{\mathbf{T}}_G(t)$  and  $\hat{\mathbf{T}}_P(t)$  and their curvature. To this end it is worth introducing an error  $\mathbf{e}(t)$  defined as

$$\mathbf{e}(t) := \mathbf{r}_P(t) - \mathbf{r}_M(t)$$

and to rewrite the relevant dynamics in error coordinates. In this respect, by considering the system dynamics (3.1), the Frenet-Serret dynamics (3.5), it is simple to obtain (for compactness we omit the arguments  $t$ )

$$\begin{aligned} \dot{\mathbf{e}} &= -(\mathbf{v}_G \cdot \hat{\mathbf{T}}_P) \left( \frac{\kappa_P(\mathbf{e} \cdot \hat{\mathbf{N}}_P)}{1 + \kappa_P(\mathbf{e} \cdot \hat{\mathbf{N}}_P)} \hat{\mathbf{T}}_P + \hat{\mathbf{N}}_P \right) \\ \dot{\hat{\mathbf{T}}}_P &= \frac{\kappa_P(\mathbf{v}_G \cdot \hat{\mathbf{T}}_P)}{1 + \kappa_P(\mathbf{e} \cdot \hat{\mathbf{N}}_P)} \hat{\mathbf{N}}_P \\ \dot{\hat{\mathbf{N}}}_P &= \frac{(\mathbf{v}_G \cdot \hat{\mathbf{T}}_P)}{1 + \kappa_P(\mathbf{e} \cdot \hat{\mathbf{N}}_P)} (\tau_P \hat{\mathbf{B}}_P - \kappa_P \hat{\mathbf{T}}_P) \\ \dot{\hat{\mathbf{B}}}_P &= \frac{-\tau_P(\mathbf{v}_G \cdot \hat{\mathbf{T}}_P)}{1 + \kappa_P(\mathbf{e} \cdot \hat{\mathbf{N}}_P)} \hat{\mathbf{N}}_P \\ v_M^* \dot{\hat{\mathbf{T}}}_M &= \mathbf{a}_M^N \end{aligned} \quad (3.6)$$

with the ground speed  $\mathbf{v}_G$  that is a function of  $\hat{\mathbf{T}}_M$  and  $\mathbf{w}$  according to

$$\mathbf{v}_G = v_M^* \hat{\mathbf{T}}_M + \mathbf{w}.$$

This is a system with state  $(\mathbf{e}, \hat{\mathbf{T}}_P, \hat{\mathbf{N}}_P, \hat{\mathbf{B}}_P, \hat{\mathbf{T}}_M)$  with control input  $\mathbf{a}_M$  (to be chosen so that  $\mathbf{a}_M^T \equiv 0$ ) subject to the wind disturbance  $\mathbf{w}$ . Note that for planar paths,  $\tau_P = 0$ .

In the paper, similarly to [10], the acceleration command will be chosen as

$$\mathbf{a}_M^N = (\mathbf{v}_M \times \mathbf{u}) \times \mathbf{v}_M \quad (3.7)$$

with  $\mathbf{u} \in \mathbb{R}^3$  an auxiliary input to be chosen. Note that this choice guarantees that  $\mathbf{a}_M^T(t) \equiv 0$  for all possible choices of  $\mathbf{u}$ . The degree-of-freedom for the problem is then the choice of the control input  $\mathbf{u}$  to accomplish control goals.

Motivated by [87], the choice of  $\mathbf{u}$  presented in the paper relies on the so-called *look-ahead vector*, denoted by  $\hat{\mathbf{L}}$ , which represents the *desired ground speed direction* for the vehicle. The latter will be taken as a function of the system state and of the wind, according to the objective conditions in which the vehicle operates.

## 2.2 Feasibility Cone and Control Objective Formulation

Although the ideal control objective is to steer the error  $\mathbf{e}(t)$  asymptotically to zero by also aligning the unitary tangent vectors  $\hat{\mathbf{T}}_G(t)$  and  $\hat{\mathbf{T}}_P(t)$  and their curvature, the presence of “strong” wind could make this ideal objective infeasible. For this reason we set two objectives that will be targeted according to the wind conditions.

**Ideal Tracking Objective.** Ideally, the control input  $\mathbf{u}$  must be chosen so that the following asymptotic objective is fulfilled

$$\begin{cases} \lim_{t \rightarrow \infty} \mathbf{e}(t) = 0 \\ \lim_{t \rightarrow \infty} (\hat{\mathbf{T}}_G(t) - \hat{\mathbf{T}}_P(t)) = 0 \\ \lim_{t \rightarrow \infty} \left( \frac{d\hat{\mathbf{T}}_G(t)}{dt} - \frac{d\hat{\mathbf{T}}_P(t)}{dt} \right) = 0 \end{cases} \quad (3.8)$$

namely position, ground speed orientation, and ground speed curvature of the vehicle converge to the path ones.

**Safety Objective.** When strong wind does not allow to achieve the ideal objective, the degraded safety objective consists of controlling the vehicle in such a way that the vehicle acceleration is asymptotically set to zero, the ground speed value is asymptotically minimized (by pointing the nose the vehicle against wind) and the vehicle nose asymptotically points to  $\mathbf{P}$ , namely

$$\begin{cases} \lim_{t \rightarrow \infty} \mathbf{a}_M^N(t) = 0 \\ \lim_{t \rightarrow \infty} \hat{\mathbf{T}}_M(t) = -\hat{\mathbf{w}} \\ \lim_{t \rightarrow \infty} \hat{\mathbf{e}}(t) = -\hat{\mathbf{w}}. \end{cases} \quad (3.9)$$

The targeted configuration, in particular, is the one in which the vehicle goes away with the wind, by minimizing the ground speed (safety objective), and minimizing the distance to the closest point on the path. Note that this objective makes sense for *finite-length* paths: the infinite-length linear path case is briefly discussed in Section 8 (Appendix). Ideal or degraded objectives are set according to the fulfillment of a “feasibility condition” by the look ahead vector. More precisely, with

$w^* := \|\mathbf{w}\|$  the wind strength, let  $\beta$  be defined as

$$\beta := \begin{cases} \arcsin \frac{v_M^*}{w^*} & w^* \geq v_M^* \\ \pi & w^* < v_M^* \end{cases} \quad (3.10)$$

Then, we define the ‘‘feasibility cone’’  $\mathcal{C}$  as the cone with apex centered at the vehicle position  $\mathbf{r}_M$ , main axis given by  $\mathbf{w}$  and with aperture angle  $2\beta$  (see Fig. 3.6). Notice that this becomes the entire plane when  $w^* < v_M^*$ , i.e.  $\beta = \pi$ . All desired ground speed vectors that lie in the cone can be indeed enforced by appropriately choosing the control input  $\mathbf{u}$ . This fact, and the fact that the look ahead vector represents the desired ground speed direction for the vehicle, motivates the fact of considering the ideal tracking objective feasible at a certain time  $t$  if it’s possible to shape the look ahead vector  $\hat{\mathbf{L}}(t)$  so that it lies in  $\mathcal{C}$ . More specifically, if

$$\lambda = \arccos \hat{\mathbf{w}} \cdot \hat{\mathbf{L}}(t) < \beta. \quad (3.11)$$

Otherwise, the ideal tracking objective is said infeasible at time  $t$ . The control objectives are set consequently, and we shape the control input separately for each subcase according to the following scheme:

$$\mathbf{u} = \begin{cases} \mathbf{u}_{\text{slow}} & w^* \leq v_M^* \\ \mathbf{u}_{\text{fast},1} & w^* > v_M^*, \lambda \leq \beta \\ \mathbf{u}_{\text{fast},2} & w^* > v_M^*, \lambda > \beta \end{cases} \quad (3.12)$$

In Sections 3, 4, we show how to design the control input  $\mathbf{u}$  as in (3.12) such that if the ideal tracking objective is feasible then (3.8) is achieved, otherwise the Safety Objective is enforced.

## 2.3 The Nominal Solution in Absence of Wind in [10]

In this section we briefly present the solution chosen in [10] for the look-ahead vector in absence of wind, as it represents the basis for developing the windy solution. A graphical sketch showing the notation is provided in Fig. 3.1.

The authors in [10] proposed the control law

$$\mathbf{u} = k\hat{\mathbf{L}} \quad (3.13)$$

in which  $k$  is a design parameter chosen so that  $k > \max_{P \in \mathcal{P}(t)} k_P$  and  $\hat{\mathbf{L}}$  is the look-ahead vector chosen as

$$\hat{\mathbf{L}} = \cos(\theta_L(\|\mathbf{d}\|))\hat{\mathbf{d}} + \sin(\theta_L(\|\mathbf{d}\|))\hat{\mathbf{T}}_P \quad (3.14)$$

where  $\mathbf{d} = \mathbf{e} + d_{\text{shift}}\hat{\mathbf{N}}_P = (\|\mathbf{e}\| + d_{\text{shift}})\hat{\mathbf{N}}_P$  is the *radially shifted distance*,  $\theta_L(\|\mathbf{d}\|)$

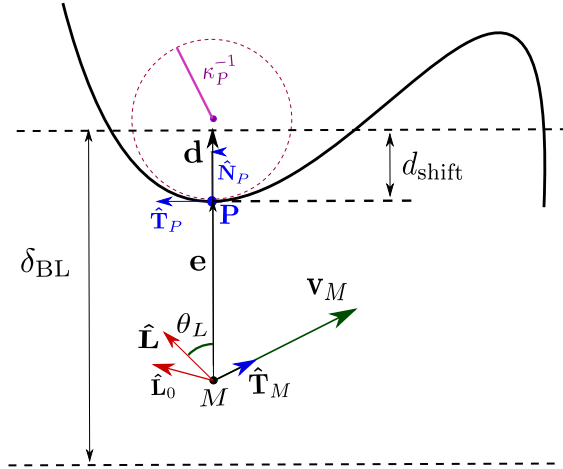


Figure 3.1: Sketch of the nominal solution of [10]

is the function

$$\theta_L(\|\mathbf{d}\|) = \frac{\pi}{2} \sqrt{1 - \text{sat}\left(\frac{\|\mathbf{d}\|}{\delta_{BL}}\right)} \quad (3.15)$$

in which  $\delta_{BL}$  is a *boundary layer parameter* and the parameter  $d_{\text{shift}}$  is chosen as  $d_{\text{shift}} = [1 - (\frac{2}{\pi} \arccos(\frac{|k_P|}{k})^2)]\delta_{BL}$ . As shown in [10], this choice guarantees a progressive and smooth steering of the vehicle along the path.

Instrumental for the next results, we also introduce the look-ahead vector computed on the error  $\mathbf{e}$  instead of the radially shifted distance  $\mathbf{d}$ , that is

$$\hat{\mathbf{L}}_0 := \hat{\mathbf{L}}|_{\mathbf{d}=\mathbf{e}}. \quad (3.16)$$

### 3 The Lower Wind Case

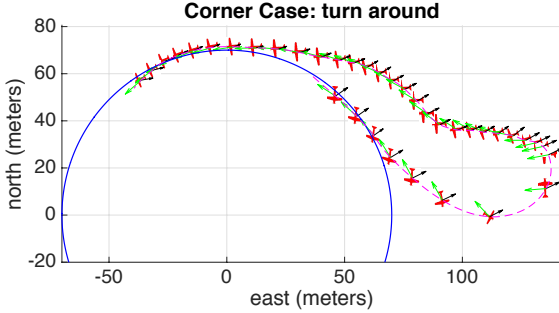
In this section, we consider the slower wind case, i.e.  $w^* < v_M^*$ . Here we design  $\mathbf{u}_{\text{slow}}$  as in (3.12).

#### 3.1 Previous solutions and their weaknesses

A simple and commonly used approach to achieve path convergence with any wind, similar to that shown in [66], is to apply the normal acceleration command  $\mathbf{a}_N^M = k(\mathbf{v}_G \times \hat{\mathbf{L}}) \times \mathbf{v}_G$  such that  $\mathbf{v}_G$  will eventually be aligned with the look-ahead



vector  $\hat{\mathbf{L}}$ . Though it should be noted that this acceleration command, defined perpendicular to the ground speed vector, is actually applied to the aircraft body-axis; a notable discrepancy for smaller/slower systems. This approach also presents non-easily predictable behaviours: as an example, it could happen what is shown in Fig. 3.2.



**Figure 3.2:** Initially, the ground speed is almost aligned with the look-ahead vector, hence the aircraft is not commanded to change its attitude and gets carried away by the wind. The aircraft is forced to perform a complete turn around to get back on track.

### 3.2 Proposed strategy

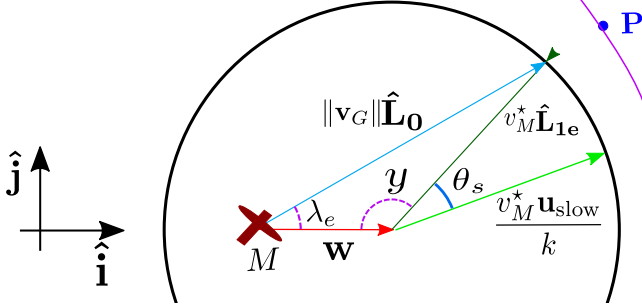
Here the goal is to find the control input  $\mathbf{u}_{\text{slow}}$  that satisfies the requirements in (3.8). We first find a *basic* control input, called  $\mathbf{u}_e$ , and improve on that to obtain  $\mathbf{u}_{\text{slow}}$ . To this end, we are going to reason in steady state, i.e.

$$\begin{cases} \mathbf{e} = 0 \\ \hat{\mathbf{T}}_{\mathbf{G}} = \hat{\mathbf{T}}_{\mathbf{P}} \\ \frac{d\hat{\mathbf{T}}_{\mathbf{G}}}{dt} = \frac{d\hat{\mathbf{T}}_{\mathbf{P}}}{dt} \end{cases} \quad (3.17)$$

#### Initial control input

Here we are going to satisfy the first two requirements in (3.8). It is useful to consider the geometry of the problem shown in Fig. 3.3 and introduce the following angles, using basic trigonometric relations

$$\begin{cases} \lambda_e = \arccos \hat{\mathbf{w}} \cdot \hat{\mathbf{L}}_0 \\ y = \arccos -\hat{\mathbf{w}} \cdot \hat{\mathbf{L}}_{1e} = \pi - \lambda_e - \arcsin \left( \frac{w^* \sin(\lambda_e)}{v_M^*} \right) \end{cases} \quad (3.18)$$



**Figure 3.3:**  $\hat{\mathbf{L}}_0$  is the desired direction for  $\mathbf{v}_G$ .  $\hat{\mathbf{L}}_{1e}$  is the target aircraft orientation.  $\hat{\mathbf{u}}_{\text{slow}}$  is the applied control input.  $\lambda_e$  and  $y$  are defined in (3.18) and  $\theta_s$  is defined in (3.28)

where  $\hat{\mathbf{L}}_{1e}$  is an unknown target orientation for the aircraft to be computed. It should be noted that these angles are not defined in case  $\mathbf{w} = 0$ .

We now aim to satisfy the first two requirements stated in (3.8) through the choice of a basic control input

$$\mathbf{u}_e = k\hat{\mathbf{L}}_{1e} \quad (3.19)$$

To find such a command, we assumed to already be at the Position/Orientation steady state condition. Since we assume to be on the path with the desired orientation, then  $k(\mathbf{v}_M \times \hat{\mathbf{L}}_{1e}) \times \mathbf{v}_M = 0$ , meaning that  $\hat{\mathbf{T}}_M = \hat{\mathbf{L}}_{1e}$  ( $\hat{\mathbf{T}}_M = -\hat{\mathbf{L}}_{1e}$  would be an unstable equilibrium, as shown in [10]).

The natural choice for the desired ground speed direction is  $\hat{\mathbf{L}}_0$ , as it was defined in (3.16). Note that  $\hat{\mathbf{L}}_0|_{\mathbf{e}=0} = \hat{\mathbf{T}}_P$ .

We need to find the desired direction  $\hat{\mathbf{L}}_{1e}$  for the aircraft by solving the geometry shown in Fig. 3.3, which means solving the following equation in  $\hat{\mathbf{L}}_{1e}$ :

$$\frac{\mathbf{w} + v_M^* \hat{\mathbf{L}}_{1e}}{\|\mathbf{w} + v_M^* \hat{\mathbf{L}}_{1e}\|} = \hat{\mathbf{L}}_0 \quad (3.20)$$

The solution, in terms of the angles defined in (3.18), is

$$\begin{cases} \hat{\mathbf{L}}_{1e} = \text{sign}([\hat{\mathbf{w}} \times \hat{\mathbf{L}}_0] \cdot \mathbf{k}) \text{rot}(-\hat{\mathbf{w}}, -y) & w^* > 0 \\ \hat{\mathbf{L}}_{1e} = \hat{\mathbf{L}}_0 & w^* = 0 \end{cases} \quad (3.21)$$

where  $\text{rot}(a, \theta)$  is the function that rotates vector  $a \in \mathbb{R}^3$  by angle  $\theta \in \mathbb{R}$  around the vertical axis  $\mathbf{k}$ . The basic  $\mathbf{u}_e$  will be improved in Section 3.2 to obtain curvature convergence.

### Improvement of the control input to satisfy the curvature convergence requirement

In order to satisfy the curvature convergence requirement, we need to force the correct amount of steady-state centripetal acceleration to the aircraft. This can be done by firstly reasoning on what additional acceleration should be imposed to the  $\mathbf{v}_G$  vector (which we call  $\|\mathbf{a}_{N_{\text{res}}}^G\|$ ), then by mapping to the actual aircraft control input. The function  $\|\mathbf{a}_{N_{\text{res}}}^G\|(\cdot)$  should satisfy the steady-state curvature requirement, i.e.:

$$\begin{aligned} \|\mathbf{a}_{N_{\text{res}}}^G\|_{|\mathbf{e}=0} &= \|k_P\| \|\mathbf{v}_G\|^2 = \\ &= k \|\mathbf{v}_G\|^2 \|(\hat{\mathbf{T}}_P \times \hat{\mathbf{L}}_{|\mathbf{d}|=\mathbf{d}_{\text{shift}}}) \times \hat{\mathbf{T}}_P\| \end{aligned} \quad (3.22)$$

In addition to that, when mapped to the actual control input, it should preserve convergence. Inspired by [10] and (3.22), we claim that the following function is a suitable choice:

$$\|\mathbf{a}_{N_{\text{res}}}^G\| = k \|\mathbf{v}_G\|^2 \|(\hat{\mathbf{L}}_0 \times \hat{\mathbf{L}}) \times \hat{\mathbf{L}}_0\| \quad (3.23)$$

with  $\hat{\mathbf{L}}$  and  $\hat{\mathbf{L}}_0$  as defined in Section 2.3. Indeed, (3.22) is satisfied by definition ( $\hat{\mathbf{L}}_0|_{\mathbf{e}=0} = \hat{\mathbf{T}}_P$ ,  $\mathbf{d}|_{\mathbf{e}=0} = \mathbf{d}_{\text{shift}}$ ), and convergence appears to be preserved. **Mapping to the control input.** We show that additional centripetal acceleration for the aircraft can be achieved by rotation of the basic control input  $\mathbf{u}_e$  (3.19) through a properly shaped angle function  $\theta_s(\cdot)$ . Notice indeed that, for  $\theta_s^* \in \mathbb{R}$ :

$$[(\mathbf{v}_M \times \text{rot}(\mathbf{u}_e, \theta_s^*)) \times \mathbf{v}_M]_{\text{angle}(\mathbf{v}_M, \mathbf{u}_e)=0} = k v_M^{*2} \sin(\theta_s^*) \quad (3.24)$$

where we assumed the vehicle direction to coincide with  $\mathbf{u}_e$ . Since  $\|\mathbf{a}_{N_{\text{res}}}^G\|$  is applied to the ground-speed vector  $\mathbf{v}_G$ , and remembering that for any normal acceleration it holds  $\mathbf{a}_N = \bar{\Omega} \times \mathbf{V}$ , where  $\bar{\Omega}$  is the angular speed vector and  $\mathbf{V}$  is the linear speed vector, then it holds through derivation w.r.t. time of  $\lambda_e$  defined in (3.18) (which indicates the  $\mathbf{v}_G$  orientation):

$$\dot{\lambda}_e = \frac{\|\mathbf{a}_{N_{\text{res}}}^G\|}{\|\mathbf{v}_G\|} \text{sign}(\kappa_P) \quad (3.25)$$

Still assuming that  $\text{angle}(\mathbf{v}_M, \mathbf{u}_e) = 0$ , noticing that angle  $y$  defined in (3.18) indicates the aircraft body-axis to which we apply the control input, and considering equation (3.24), it holds:

$$\dot{y} = \frac{d}{dt} (\text{angle}(\mathbf{v}_M, \mathbf{w})|_{\text{angle}(\mathbf{v}_M, \mathbf{u}_e)=0}) = k v_M^* \sin(\theta_s) \quad (3.26)$$

Also, deriving the second equation of (3.18) w.r.t time, it holds:

$$\dot{y} = -\dot{\lambda}_e - \frac{w^* \cos(\lambda_e) \dot{\lambda}_e}{v_M^* \sqrt{1 - \left(\frac{w^* \sin(\lambda_e)}{v_M^*}\right)^2}} \quad (3.27)$$

Hence plugging (3.23) into (3.25) and (3.25) into (3.27), we can compare equations (3.26) and (3.27) to obtain:

$$\frac{\theta_s}{\text{sign}(\kappa_P)} = \arcsin \left[ \text{sat} \left( -\frac{\|\mathbf{v}_G\| \|(\hat{\mathbf{L}}_0 \times \hat{\mathbf{L}}) \times \hat{\mathbf{L}}_0\|}{v_M^*} \left( 1 + \frac{w^* \cos \lambda_e}{\sqrt{v_M^{*2} - (w^* \sin \lambda_e)^2}} \right) \right) \right] \quad (3.28)$$

Where the saturation function bounds the argument between -1 and 1: this is needed because of the assumption  $\text{angle}(\mathbf{v}_M, \mathbf{u}_e) = 0$ , i.e. during the transient we might ask for residual accelerations that are higher than in steady-state. This doesn't ruin convergence, as the vehicle will keep turning until the  $\|\mathbf{v}_G\|$  eventually decreases and  $\theta_s$  can smoothly steer the trajectory curvature to the path curvature. In the end we apply:

$$\mathbf{u}_{\text{slow}} = \text{rot}(\mathbf{u}_e, \theta_s) \quad (3.29)$$

$\mathbf{u}_e$  defined as in (3.19),  $\theta_s$  as in (3.28), so that the goals in (3.8) are satisfied. We report in Fig. 3.4 a phase portrait showing global convergence in numerical simulations for a large variety of initial conditions and different wind speeds. That said, attractiveness to the equilibrium is not formally proved in this paper.

In Fig. 3.5, we can observe the performance of the algorithm for strong constant wind, still slower than the airspeed.

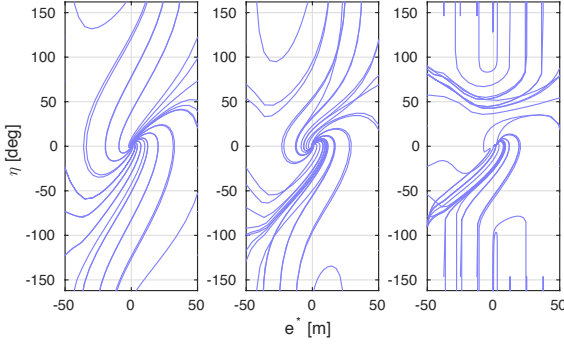
**Choice of  $k$ .** In order for the algorithm to keep null error in steady state, we have the lower bound:

$$k > \max_{|k_P|} \left( 1 + \frac{w^*}{v_M^*} \right)^2 |k_P| \quad (3.30)$$

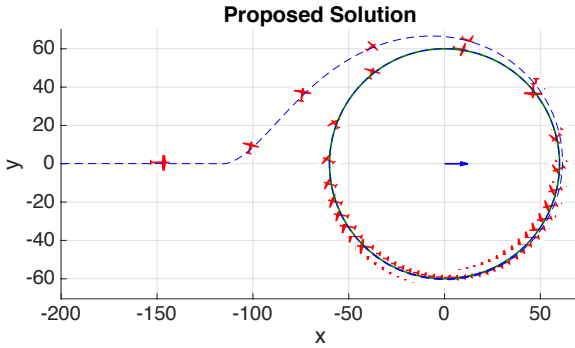
Similarly to [10], the derivation considers the highest acceleration we need in the worst case scenario ( $\hat{\mathbf{L}}_0 = \hat{\mathbf{w}}$ ,  $\mathbf{v}_M \parallel \mathbf{w}$ ).

## 4 The Higher Wind Case

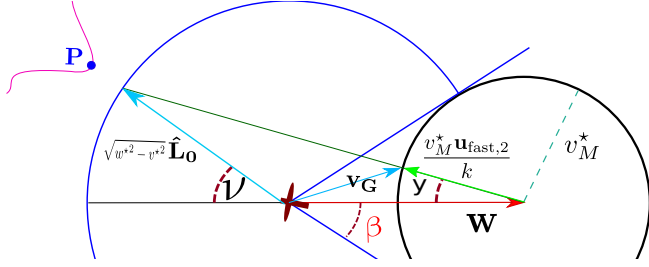
In this section we design  $\mathbf{u}_{\text{fast},1}$  and  $\mathbf{u}_{\text{fast},2}$  introduced in (3.12). Let us define the desired direction for the ground speed  $\hat{\mathbf{L}}_0$  as in equation (3.16), and the corresponding basic control input  $\mathbf{u}_e$  as in equation (3.19). It is convenient to reason considering the angles introduced in (3.18): refer to Fig. 3.6 for a better visualization.



**Figure 3.4:** Phase portraits of the proposed lower wind solution for  $w^*=0\text{ m s}^{-1}$  (left),  $7\text{ m s}^{-1}$  (middle), and  $13.5\text{ m s}^{-1}$  (right), respectively. The tracking angular error  $\eta = \text{atan2}(\hat{\mathbf{T}}_{P_y}, \hat{\mathbf{T}}_{P_x}) - \text{atan2}(\hat{\mathbf{T}}_{G_y}, \hat{\mathbf{T}}_{G_x}) \in [-\pi, \pi]$  is compared with the signed, one-dimensional cross-track error  $e^* = \mathbf{e} \cdot \frac{\mathbf{r}_M}{\|\mathbf{r}_M\|}$  to show algorithm convergence within the bounds of  $\delta_{BL} = 50\text{ m}$ , for  $k = 0.05$ ,  $R = 100\text{ m}$ , and  $v_M^* = 14\text{ m s}^{-1}$ .



**Figure 3.5:** Airspeed  $14\text{ m/s}$ . wind speed  $12\text{ m/s}$ . The proposed solution lets the vehicle achieve the goals in (3.8).



**Figure 3.6:**  $w^* > v_M^*$  case.  $\nu$  is the angle between  $-\mathbf{w}$  and the desired ground speed  $\hat{\mathbf{L}}_0$ ,  $y$  as in Figure 3,  $\mathbf{v}_G$  is the actual ground speed that we achieve,  $\mathbf{u}_{\text{fast},2}$  is the chosen control input.

#### 4.1 Solution for $\hat{\mathbf{L}}_0$ feasible, i.e. $\lambda \leq \beta$

As the desired ground speed direction  $\hat{\mathbf{L}}_0$  is feasible, we reason as in Section 3.2: choose the basic control input  $\mathbf{u}_e$  as in (3.19) and rotate it by a proper angle in order to achieve curvature convergence: this would mean  $\mathbf{u}_{\text{fast},1} = \mathbf{u}_{\text{slow}}$ , and doing so we would achieve curvature convergence as long as the  $\hat{\mathbf{L}}_0$  is still feasible. However, with usual shapes for the target curved path, at some point the desired direction *will* become infeasible: when this happens, we need the control input not to change abruptly, i.e. to be a continuous function of the desired  $\hat{\mathbf{L}}_0$ . Since we cannot in general achieve the goals in (3.8), we make a slightly different heuristic choice for  $\mathbf{u}_{\text{fast},1}$  that guarantees continuity of the commands (as better explained in [18]), while preserving curvature convergence to a good extent as long as the  $\hat{\mathbf{L}}_0$  is feasible:

$$\theta_{s2} = \frac{\sqrt{\left[1 - \left(\frac{w^* \sin(\lambda_e)}{v_M^*}\right)^2\right]}}{\cos(\lambda_e)} \theta_s \quad (3.31)$$

Notice indeed that at the infeasibility boundary  $\theta_{s2}|_{\lambda_e = \arcsin v_M^* w^* - 1} = 0$  and at the slower wind case boundary  $\theta_{s2}|_{w^* = v_M^*} = \theta_s$ ,  $\theta_s$  as in (3.28). So, in the end,

$$\mathbf{u}_{\text{fast},1} = \text{rot}(\mathbf{u}_e, \theta_{s2}) \quad (3.32)$$

#### 4.2 Strategy for $\hat{\mathbf{L}}_0$ infeasible, i.e. $\lambda > \beta$

We define an *infeasibility parameter*  $\alpha_{\text{out}}$  and a *safety function*  $\sigma_{\text{safe}}(\alpha_{\text{out}})$  as follows:

$$\alpha_{\text{out}} = \frac{\lambda - \beta}{\pi - \beta} \quad \sigma_{\text{safe}} = \frac{\frac{\pi}{2} - \beta - y(\alpha_{\text{out}})}{\frac{\pi}{2} - \beta} \quad (3.33)$$

where both indices have maximum value equal to 1. When  $\sigma_{\text{safe}} = 1$ , it means that we act conservatively and choose  $\frac{\mathbf{u}_{\text{fast},2}}{k} = -\hat{\mathbf{w}}$ : this has to happen only in the absolutely worst scenario of  $\hat{\mathbf{L}}_0 = -\hat{\mathbf{w}}$ , which corresponds to the maximum  $\alpha_{\text{out}} = 1$ . In all the intermediate cases, we want to guarantee a tradeoff between conservatism and tracking performance, i.e we want  $\sigma_{\text{safe}}(\alpha_{\text{out}})$  to be increasing with  $\alpha_{\text{out}}$ . This can be achieved by finding a proper mapping  $f$  from angle  $\nu = \pi - \lambda_e$  to angle  $y$  in the following form

$$f : \nu \in [0, \pi - \beta] \rightarrow y \in \left[0, \frac{\pi}{2} - \beta\right] \quad (3.34)$$

This mapping should satisfy, at least, these 3 properties:

$$\begin{aligned} f(0) &= 0 \\ f(\pi - \beta) &= \frac{\pi}{2} - \beta \\ f(a) &< f(b) \quad \forall a > b, \quad a, b \in \left[0, \frac{\pi}{2} - \beta\right] \end{aligned} \quad (3.35)$$

The first requirement is to guarantee that  $\sigma_{\text{safe}} = 1$  when  $\hat{\mathbf{L}}_0 = -\mathbf{w}$ . The second one is a boundary condition to guarantee that the input is continuous to the  $\hat{\mathbf{L}}_0$  switching from being feasible to infeasible (or vice versa). The third requirement is for finding a tradeoff between safety and performance: put in words, the more the  $\hat{\mathbf{L}}_0$  is infeasible for the ground speed, the more we want to turn against the wind and wait for it to stop.

By looking at Fig. 3.6, a natural choice that follows geometric intuition and is coherent with the requirements that we have just stated, is

$$\mathbf{u}_{\text{fast},2} = k \frac{\sqrt{w^*2 - v_M^*2} \hat{\mathbf{L}}_0 - \mathbf{w}}{\|\sqrt{w^*2 - v_M^*2} \hat{\mathbf{L}}_0 - \mathbf{w}\|} \quad (3.36)$$

In terms of the mapping that has been defined before, this choice corresponds to

$$f(\nu = \pi - \lambda) = y = \arcsin \frac{\sin \nu \cos \beta}{\sqrt{1 + \cos^2 \beta + 2 \cos \beta \cos \nu}} \quad (3.37)$$

this mapping satisfies the requirements (3.35), as can be easily verified by substitution and derivation with respect to  $\nu$ . For clarity, the function is plotted in Fig. 3.7 for different values of the wind-cone opening angle  $\beta$ . In Fig. 3.8 the performance of the algorithm is shown.

It is also worth highlighting the tradeoff introduced between performance and

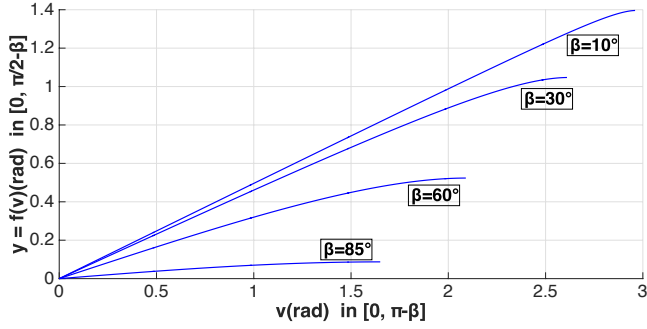


Figure 3.7: Proposed mapping  $y = f(v)$  for different values of  $\beta$

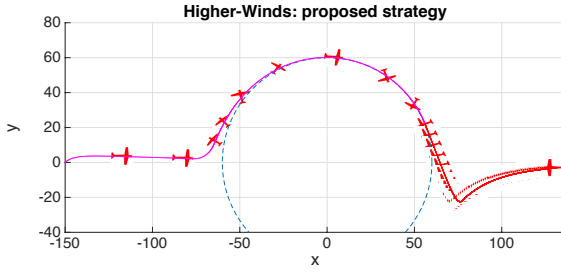


Figure 3.8: wind speed is 16  $m/s$ , airspeed is 14  $m/s$ . The proposed control input is used. Magenta line: feasible desired direction. Red line: infeasible desired direction.

safety (**incremental safety**) by computing the safety function  $\sigma_{\text{safe}}(\alpha_{\text{out}})$  :

$$\sigma_{\text{safe}} = \frac{\beta - \frac{\pi}{2} + \arcsin\left(\frac{\sin(\pi - \beta - (\pi - \beta)\alpha_{\text{out}}) \cos \beta}{\sqrt{1 + \cos^2 \beta + 2 \cos \beta \cos(\pi - \beta - (\pi - \beta)\alpha_{\text{out}})}}\right)}{\beta - \frac{\pi}{2}} \quad (3.38)$$

as is also shown in Fig. 3.9.



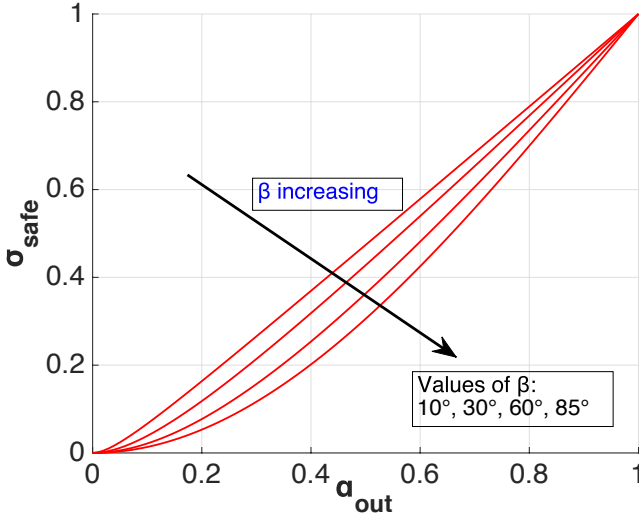


Figure 3.9: Infeasibility/safety relationship, for different  $\beta$ s

## 5 Continuity

In realistic scenarios, the wind is not going to be constant, but will likely switch between  $w^* < v_M^*$  and  $w^* > v_M^*$  several times. Not only that, the path is going to be curved, so the desired direction for the ground speed  $\hat{\mathbf{L}}_0$  is going to switch between being feasible and infeasible. All these switchings mean that it is very important for the command input  $\mathbf{u}$  to be continuous to changing winds and changing  $\hat{\mathbf{L}}_0$ . The control input was derived separately for the three subcases (slower winds, higher winds with feasible desired direction, higher winds with infeasible desired direction) in Sections 3, 4.1, 4.2.

We want to show here that the complete control input

$$\mathbf{u} = \begin{cases} \mathbf{u}_{\text{slow}}, & w^* \leq v_M^* \\ \mathbf{u}_{\text{fast},1}, & w^* > v_M^*, \lambda \leq \beta \\ \mathbf{u}_{\text{fast},2}, & w^* > v_M^*, \lambda > \beta \end{cases} \quad (3.39)$$

indeed guarantees continuity in this sense.

- **Switching between  $\mathbf{u}_{\text{slow}}$  and  $\mathbf{u}_{\text{fast},1}$ :** this happens as the wind passes from  $w^* < v_M^*$  to  $w^* > v_M^*$ . Let  $t^*$  be the boundary time instant in which  $w^*(t^*) = v_M^*(t^*)$ . Also, in this case,  $\lambda_e(t^*) \leq \frac{\pi}{2}$ . Looking at the formulation for  $\theta_s$  and  $\theta_{s2}$  in (3.31), we have:

$$\theta_{s2}|_{w^*=v_M^*} = \theta_s|_{w^*=v_M^*} \quad (3.40)$$

and so  $\mathbf{u}_{\text{slow}}(t^*) = \mathbf{u}_{\text{fast},1}(t^*)$  so the command  $\mathbf{u}(t)$  is continuous at this boundary condition

- **Switching between  $\mathbf{u}_{\text{slow}}$  and  $\mathbf{u}_{\text{fast},2}$ :** this happens as the wind passes from  $w^* < v_M^*$  to  $w^* > v_M^*$ . Let  $t^*$  be the boundary time instant in which  $w^*(t^*) = v_M^*(t^*)$ . Also, in this case,  $\lambda_e(t^*) > \frac{\pi}{2}$ . Solving the geometry in Fig. 3.3, we have that  $\mathbf{u}_e = -\mathbf{w}$ . Since  $\beta(t^*) = \frac{\pi}{2}$ , this implies that  $y(t^*) = 0$  as computed in (3.37): so  $\mathbf{u}_{\text{fast},2}(t^*) = -\mathbf{w}$  as well. Assuming  $\hat{\mathbf{T}}_M \approx \hat{\mathbf{L}}_{1e}$ , which is the case after some transient, we have that  $\|\mathbf{v}_G\| \approx 0$ . So by (??), we have  $\theta_s \approx 0$ , implying

$$\mathbf{u}_{\text{slow}}(t^*) \approx -\mathbf{w} \approx \mathbf{u}_{\text{fast},2}(t^*) \quad (3.41)$$

- **Switching between  $\mathbf{u}_{\text{fast},1}$  and  $\mathbf{u}_{\text{fast},2}$ :** this happens when  $w^* > v_M$  and  $\hat{\mathbf{L}}_0$  passes from being feasible to being infeasible. Let  $t^*$  be the boundary time instant. Then

$$w^*(t^*) \sin \lambda_e(t^*) = v_M^*(t^*) \quad (3.42)$$

so  $\theta_{s2}(t^*) = 0$ . This guarantees continuity, as no shifting angle is applied in the infeasible case.

In Fig. 3.10 we report a plot that highlights the continuity of the input as the wind increases and for a fixed  $\hat{\mathbf{L}}_0$ . We plot angle  $y$  associated with the direction of the control input  $\mathbf{u}_{\text{fast},2}$ , for different values of  $\nu = \text{angle}(-\mathbf{w}, \hat{\mathbf{L}}_0)$ .

## 5.1 Sinusoidal Winds

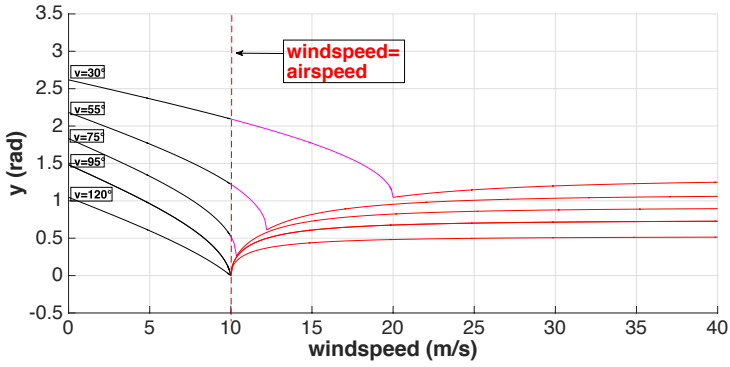
As an example of a more realistic varying wind profile, in order to show that the commands do not switch abruptly and are continuous, we consider the case of the wind having this sinusoidal profile

$$\mathbf{w}(t) = W \sin(\Omega t) [1 \quad 0 \quad 0]^T \quad (3.43)$$

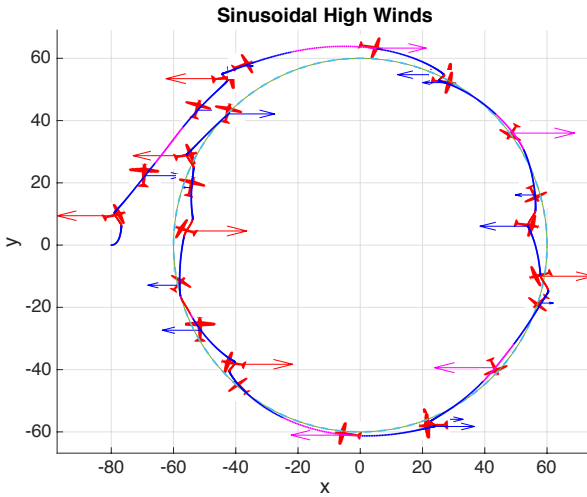
for some wind pulsation  $\Omega$  and amplitude  $W > v_M^*$ . The result is shown in Fig. 3.11, and the same is shown (more clearly) in an accompanying video<sup>1</sup>.

Switching between any couple of the three parts of the control input can happen in this case. The least smooth behaviour, as we have only approximate continuity, is when the switching is between  $\mathbf{u}_{\text{slow}}$  and  $\mathbf{u}_{\text{fast},2}$ .

<sup>1</sup>Sinusoidal wind simulation: <<https://www.youtube.com/watch?v=fpV5KkmrrUc>>



**Figure 3.10:** Continuity at  $w^* = v_M^*$ . Black line: slower wind. Magenta: inside the cone. Red: outside the cone.



**Figure 3.11:** Sinusoidal winds. Blue:  $\mathbf{u}_{\text{slow}}$  is applied. Magenta:  $\mathbf{u}_{\text{fast},1}$  is applied. Red:  $\mathbf{u}_{\text{fast},2}$  is applied

## 6 Flight Results

The proposed algorithm was implemented on a Pixhawk Autopilot in C++ and subsequently tested on a small fixed-wing UAV in high wind conditions. In Fig. 3.12, we show the results from the flight tests. The aircraft was commanded to follow a circular trajectory in counter-clockwise direction at a nominal airspeed of  $8 \text{ m s}^{-1}$ . The wind vector is represented in the figures using the following arrow, color scheme:  $w^* < v_M^*$  (black),  $w^* > v_M^* \cap (\hat{\mathbf{L}}_0 \text{ feasible})$  (magenta),  $w^* > v_M^* \cap (\hat{\mathbf{L}}_0 \text{ infeasible})$  (red). In Fig. 3.12a, the UAV can be seen to attempt curvature following despite the infeasible look-ahead direction until a point where the wind speed reduces and allows the start to convergence back to the path. Fig. 3.12b shows a wind-stabilized approach towards the trajectory until the point where simply pointing into the wind is the only option to reduce “runaway” from the track, recall the tracking direction is counter-clockwise.

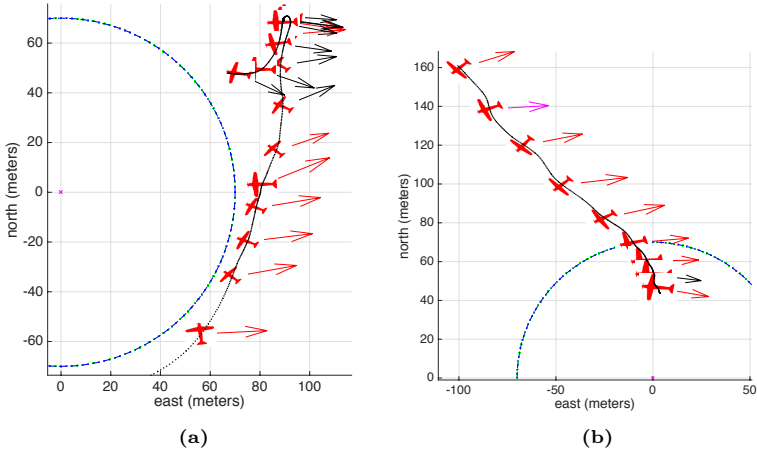


Figure 3.12: Windy flight experiments. Red arrow:  $w^* > v_M^*$

## 7 Conclusions

In this work, we extended a nonlinear guidance method based on a look-ahead vector, particularly suitable for fixed-wing UAVs, so as to actively take into account the measurements of external flowfields and drastically improve the tracking performance of the vehicle. Taking inspiration from issues that often arise when using small-sized UAVS, such as the maximum achievable airspeed being lower than the wind speed

and the commands to the aircraft being discontinuous when treating the higher-wind case as a corner faulty case, the proposed technique considers arbitrarily strong flowfields and is continuous to wind changes.

The slower wind case allows for exact convergence to the path, as all the directions for the ground speed are feasible. The higher-wind case was considered in two separate subcases, by defining the notion of *feasible* and *infeasible* desired ground speed directions for the aircraft. Exact tracking performance in the feasible case was shown preserved, while safety in the infeasible case was demonstrated and bounded to a minimum “run away” configuration; i.e. we define the concept of *asymptotic safety* for finite paths.

Future work will need to extend the proposed geometric approach to the more general case of 3D paths with 3D winds. A mathematical proof for the convergence with slower winds will also have to be provided.

## 8 Appendix: Stability Proof for High Wind Case

We are in the scenario of  $w^* > v_M^*$ . For finite-length paths, we want to show that we achieve the requirements in (3.9).

Here we will also consider briefly the case of infinite paths: the only realistic case in UAV application is that of infinite linear paths. In this case we want to show that:

$$\begin{cases} \lim_{t \rightarrow \infty} \mathbf{a}_N^M(t) = 0 \\ \hat{\mathbf{T}}_M = -\text{rot}(-\hat{\mathbf{w}}, f(\frac{\pi}{2} - \mu)) \end{cases} \quad (3.44)$$

where  $\mu = \arccos \hat{\mathbf{w}} \cdot \hat{\Lambda}$ ,  $\hat{\Lambda}$  is the direction of the target linear path, mapping  $f : \nu \rightarrow y$  has to be chosen. The second requirement in (3.44) asks for a trade-off between the linear-path direction and the anti-wind direction for the  $\hat{\mathbf{T}}_M$ , that results in an efficient direction for the actual  $\hat{\mathbf{T}}_G$ .

In both cases, the proof for the proposed algorithm will be structured as follows:

- First the so called *geometric case* will be tackled: the vehicle is considered to always be at the desired heading angle, i.e.  $\hat{\mathbf{T}}_M(t) = \frac{\mathbf{u}_{\text{fast},2}}{k}(t)$ ,  $\forall t$ .
- Then, the so called *dynamical case* (the vehicle is not always at the desired heading angle) will be considered and shown to fall into the geometrical case as time goes to infinity.

### 8.1 Geometric case: finite paths

#### Subcase 0. Single point path

Here we consider the path to be very far away and hence similar to a single point  $P$  for the aircraft to be reached. The radially shifted distance is indistinguishable from the error, so  $\theta_s(t) \approx 0 \forall t$ . Also, notice that with a point-path,  $\hat{\mathbf{e}} = \hat{\mathbf{L}}_0$ . By

defining

$$a = \sqrt{w^{*2} - v_M^{*2}}, \quad l = \|a\hat{\mathbf{L}}_0 - \mathbf{w}\| \quad (3.45)$$

we obtain

$$\begin{aligned} \mathbf{v}_G \times \hat{\mathbf{L}}_0 &= (w^* + v_M^* \hat{\mathbf{L}}_{1e}) \times \hat{\mathbf{L}}_0 = \\ &= (\mathbf{w} + \frac{v_M^* (\hat{\mathbf{L}}_0 a - \mathbf{w})}{l}) \times \hat{\mathbf{L}}_0 = \\ &= \underbrace{\left(1 - \frac{v_M^*}{l}\right)}_{>0} \mathbf{w} + \frac{av_M^*}{l} \hat{\mathbf{L}}_0 \times \hat{\mathbf{L}}_0 \\ &= \underbrace{\left(1 - \frac{v_M^*}{l}\right)}_{>0} \mathbf{w} \times \hat{\mathbf{L}}_0 + 0 \end{aligned} \quad (3.46)$$

Now let the line directed as  $\hat{\mathbf{L}}_0$  divide the plane into two half-planes: the previous considerations imply that  $\mathbf{v}_G$  and  $\mathbf{w}$  both lie in the same half-plane, so the  $\hat{\mathbf{L}}_0$  will rotate more towards the  $-\mathbf{w}$  direction in time until eventually  $\lim_{t \rightarrow \infty} \hat{\mathbf{L}}_0 = \lim_{t \rightarrow \infty} \hat{\mathbf{e}} = -\hat{\mathbf{w}}$ .

Another way to see this: the path-point  $P$  acts as a rotational joint for the error vector  $\mathbf{e}$ , which is fixed at one end in  $P$ : the  $\mathbf{v}_G$  is rotating the error vector in the same direction as the wind would rotate it, meaning that it will point instantaneously more in the anti-wind direction, i.e. even more outwardly with respect to the cone, until it reaches the antiwind direction (the ‘torque’ around point  $P$  is null at that point).

As in this case  $\hat{\mathbf{L}}_0 = \hat{\mathbf{e}}$ , the  $\hat{\mathbf{L}}_0$  rotation must stop here. Since  $f(0) = 0$ , then also  $\lim_{t \rightarrow \infty} \frac{\mathbf{u}_{\text{fast},2}}{k} = -\hat{\mathbf{w}}$  by construction. By hypothesis of geometrical case, this means  $\lim_{t \rightarrow \infty} \hat{\mathbf{T}}_M(t) = -\hat{\mathbf{w}}$ . Then, by definition of the normal acceleration command, also  $\lim_{t \rightarrow \infty} \mathbf{a}_N^M = 0$  so we reach **asymptotic safety** as defined in (3.9).

As an additional feature, note that

$$\begin{aligned} \text{sign}[(\mathbf{v}_G \times \mathbf{w}) \cdot \mathbf{k}](t_0) &= \\ \text{sign}[(\mathbf{v}_G \times \mathbf{w}) \cdot \mathbf{k}](t), \quad \forall t > t_0 \end{aligned} \quad (3.47)$$

so we reach the equilibrium without oscillations around that line such that  $\hat{\mathbf{e}} = -\hat{\mathbf{w}}$ .

### Subcase 1. Finite length paths

In this case, the  $\hat{\mathbf{L}}_0$  versor is a function of the particular path we are considering, so we can no longer assume it to coincide with  $\hat{\mathbf{e}}$  as in the single point path case. However, consider the following two facts:

- The path is finite

$$\bullet \forall t \in \mathbb{R}, \mathbf{r}_M(t) \cdot \hat{\mathbf{w}} \geq \mathbf{r}_M(0) \cdot \hat{\mathbf{w}} + \underbrace{(w^* - v_M^*)t}_{>0}$$

That is, as the wind is constantly stronger than the airspeed, the minimum growth of the projection of the error onto the wind direction has rate  $(w^* - v_M^*)t$ . This implies that

$$\lim_{t \rightarrow +\infty} \|\mathbf{r}_M(t)\| = +\infty \quad (3.48)$$

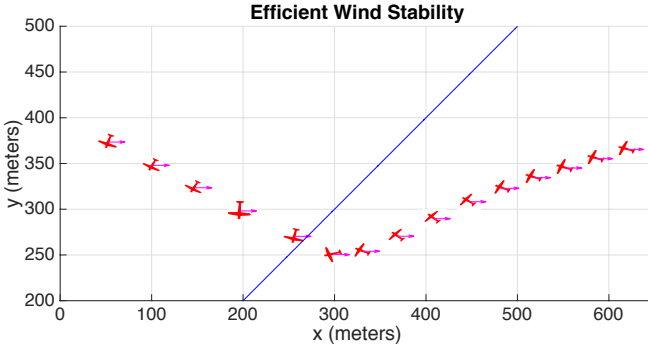
and since the path is finite

$$\lim_{t \rightarrow +\infty} \|\mathbf{d}(t)\| = \lim_{t \rightarrow +\infty} \|\mathbf{e}(t)\| = +\infty \quad (3.49)$$

As the distance grows to infinity, the path will look like a single point  $\mathbf{P}_\infty$ , that is the center of the smallest circle that contains the whole path. Then, we fall into the single point-path subcase.

## 8.2 Geometric case: infinite linear paths

Here the path is not finite. However, a common case in UAV applications is when the path is an infinite line. If this line is outside the wind-cone or the intersection with the cone is finite, it is not possible for the  $\mathbf{v}_G$  to align to it. In this case, the proposed algorithm achieves **efficient wind stability**, i.e. the objectives in (3.44). To show this, simply notice that if  $\mathbf{d} > \delta_{BL}$ , then whatever the vehicle position, we have  $\hat{\mathbf{L}}_0 \perp \hat{\mathbf{A}}$ , as the error direction will always be perpendicular to the line. A simulation for this situation is shown in Fig. 3.13.



**Figure 3.13:** Airspeed is 14 m/s. wind speed is 30 m/s, indicated by the magenta arrow on the aircraft.  $\hat{\mathbf{A}} = (\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2})$ . Line direction is hence infeasible.





**Subcase 1**

If

$$\beta < \nu < \pi - \beta \quad (3.50)$$

corresponding to  $\hat{\mathbf{L}}_0$  pointing outside of the “specular” cone, then it is easy to see that

$$\alpha_{Lg} < \pi, \forall \theta_g \quad (3.51)$$

meaning that

$$\dot{\nu} < 0 \quad (3.52)$$

independently from the actual aircraft orientation. This holds until we fall into subcase 2.

**Subcase 2**

If

$$0 < \nu < \beta \quad (3.53)$$

corresponding to  $\hat{\mathbf{L}}_0$  pointing inside of the “specular” cone, we need further considerations. It is not true anymore that  $\alpha_{Lg} < \pi, \forall \theta_g$ . Instead we have

$$\begin{cases} \alpha_{Lg} < \pi, & \text{if } \theta_g > -\nu \\ \alpha_{Lg} \geq \pi & \text{otherwise} \end{cases} \quad (3.54)$$

Then it’s possible that, depending on how the aircraft is oriented,  $\nu$  will increase while the angle between  $\hat{\mathbf{T}}_M$  and the commanded direction  $\frac{\mathbf{u}_{\text{fast},2}}{k}$  is smaller than  $\pi$ , which is undesirable as it would mean the  $\hat{\mathbf{L}}_0$  is “running away” from  $\hat{\mathbf{T}}_M$ . To show that eventually the aircraft can be considered to be aligned with its commanded control input versor  $\frac{\mathbf{u}_{\text{fast},2}}{k}$ , consider the following:

- We can increase parameter  $k$  in order to make the vehicle turn with faster dynamics.
- As time goes to infinity, eventually the “chasing” angle  $z$  will decrease to 0.

To show this last fact, first notice that for any given  $\hat{\mathbf{T}}_M$ , if  $\dot{\nu} > 0$  then  $\dot{\nu}$  is a decreasing function of  $|\mathbf{e} \cdot \mathbf{w}|$  that goes to 0 as  $\frac{1}{|\mathbf{e} \cdot \mathbf{w}|}$  or faster. Indeed, consider the case when  $\dot{\nu} > 0$  and has the maximum value, i.e.  $\nu = 0$  and  $\hat{\mathbf{T}}_M \perp \mathbf{w}$ . We have

$$\dot{\nu}_{\text{MAX}} = \frac{v_M^*}{|\mathbf{e} \cdot \mathbf{w}|} \quad (3.55)$$

which acts as an upperbound for all the other situations. Irrespectively from  $\hat{\mathbf{T}}_M$ , since  $w^* > v_M^*$ ,  $|\mathbf{e} \cdot \mathbf{w}|$  indeed increases, hence  $\dot{\nu}$  must decrease and tend to 0. Since  $y$  is a function of  $\nu$  such that  $\forall \nu, y(\nu) < \nu$ , than also  $\dot{y}$  decreases and tends to 0

as time goes to infinity. Now consider the time derivative of the “chasing” angle  $z$

$$\dot{z} = \dot{y} + \dot{\xi} \quad (3.56)$$

Since we showed  $\lim_{t \rightarrow +\infty} \dot{\nu}(t) = 0 = \lim_{t \rightarrow +\infty} \dot{y}(\nu(t))$ , irrespectively of what the orientation of the vehicle could be at any time, then, as  $\xi$  indicates the heading angle of the aircraft,

$$\lim_{t \rightarrow \infty} \dot{z}(t) = \dot{\xi}(t) \quad (3.57)$$

As the acceleration command is designed to steer the vehicle orientation onto the chosen look-ahead vector, which now is  $\frac{\mathbf{u}_{\text{fast},2}}{k}$ , as the look-ahead is bound to asymptotically stop changing as the vehicle gets further away from the path, then we actually have that  $\lim_{t \rightarrow \infty} \xi(t) = 0$ , with the vehicle aligned to the look-ahead vector. This, together with (3.57), translates in

$$\lim_{t \rightarrow \infty} \dot{z}(t) = 0 \quad (3.58)$$

Then we can say that we asymptotically fall into the geometrical case, and the proof holds.



# On Flying Backwards: Preventing Run-Away of Small, Low-Speed, Fixed-Wing UAVs in Strong Winds

Thomas Stastny and Roland Siegwart

## Abstract

Small, low-speed fixed-wing Unmanned Aerial Vehicles (UAVs) operating autonomously, beyond-visual-line-of-sight (BVLOS) will inevitably encounter winds rising to levels near or exceeding the vehicles' nominal airspeed. In this paper, we develop a nonlinear lateral-directional path following guidance law with explicit consideration of online wind estimates. Energy efficient airspeed reference compensation logic is developed for excess wind scenarios (i.e. when the wind speed rises above the airspeed), enabling either mitigation, prevention, or over-powering of excess wind induced run-away from a given path. The developed guidance law is demonstrated on a representative small, low-speed test UAV in two flight experiments conducted in mountainous regions of Switzerland with strong, turbulent wind conditions, gusts reaching up to 13 meters per second. We demonstrate track-keeping errors of less than 1 meter consistently maintained during a representative duration of gusting, excess winds and a mean ground speed undershoot of 0.5 meters per second from the commanded minimum forward ground speed demonstrated in over 5 minutes of the showcased flight results.

## 1 Introduction

In recent years, small, easily manageable, operated, and maintained fixed-wing Unmanned Aerial Vehicles (UAVs) are increasingly being applied to remote sensing ventures requiring long-range and/or long-endurance flight. For example, ETH Zürich's Autonomous Systems Lab (ASL) has developed Low-Altitude, Long-Endurance (LALE) solar-powered platforms capable of multi-day, payload-equipped flight [61], and further demonstrated the utility of such small platforms in beyond-visual-line-of-sight (BVLOS) science missions such as Arctic glacier monitoring (see project Sun2Ice<sup>1</sup>). The ability to reach far-away locations where humans either cannot or do not want to go is a great advantage, however it also comes with risks that operators, placated by seemingly fully automated aircraft, may not anticipate.

Endemic to small, low-speed fixed-wing platforms is a susceptibility to high winds. UAVs operating autonomously BVLOS, e.g. in mountainous areas or along coastlines, will doubtless encounter winds rising to levels near or exceeding the vehicles' nominal airspeed. Without a control law cognizant of the local wind field, or moreover, without logic to handle such cases, the aircraft risks mission delays (unable to make meaningful progress towards subsequent waypoints) or possibly loss of airframe (if operating near and blown into large structures, e.g. cliff walls or fjords).

As researchers and practitioners execute more flight hours and experience more weather conditions, wind hazards have started to be acknowledged within literature. Recent work has considered wind in various iterations of emergency landing planners using airmass relative Dubins aircraft curves (or trochoids in the inertial frame) [40, 85], however, these planners still require that the aircraft is able to move forward with respect to the ground and rely on some form of guidance law to follow the planned paths. Authors of [12] propose a vectorfield-based guidance law which considers the current wind estimate and may follow any smooth path curvature, however the algorithm uses level sets to describe the notion of distance from the trajectory which makes tuning specific to the curve for which the "distance" function is defined, as opposed to the more common direct relation to track error. A three-dimensional guidance approach with explicit consideration of wind is developed in [4], where the law further accounts for roll and flight path angle constraints using the theory of nested saturations.

More generic guidance logic for either waypoint tracking or path following typically takes inertial ground speed measurements as inputs, which contains the effect of wind. Perhaps the most widely used path following guidance running on small fixed-wing UAVs today, the nonlinear path following guidance developed in [75] (commonly known as  $L_1$  guidance), uses exactly this approach. This "look-ahead" method is simple (implemented easily on a microcontroller), intuitive to tune (following further extensions in [11]), and quite effective in practice; however, it has the detriments that 1) convergence to the path is only guaranteed for lines and circles, and 2) as shown in [19], the logic breaks down once winds approach

---

<sup>1</sup><http://www.sun2ice.ethz.ch>



**Figure 4.1:** Easyglider test platform flying in strong winds over the Jura, Switzerland.

and/or exceed the vehicle’s airspeed. The law was abstracted in [10] to a more general form capable of following any smooth, continuous 3D path, though without consideration of wind.

In our previous work, we took the method from [10] in a different direction and reformulated the 2D case for consideration of *excess wind* cases (i.e. wind speed  $>$  airspeed) [19]. To date, [19] is still the only guidance method in literature considering the particular problem of *excess wind*. While this logic will provide run-away *mitigation* (i.e. minimizing the rate at which the vehicle is blown away from the path), in the case the aircraft may have remaining energy available, the airspeed reference could be increased above the nominal value to further reduce, or even *prevent* run-away.

In this work, we propose utilizing the airspeed reference as control towards the development of an efficient airspeed reference compensation logic, running in parallel with an improved, wind-robust directional guidance. The resultant logic either regulates wind excess, stays on track, or maintains a minimum ground speed, depending on the operator’s chosen mode or the aircraft’s speed limits. We provide significant enhancements to the baseline algorithm with a heavy emphasis on practical implications of fielding the controller including, but not limited to, an improved notion of *bearing feasibility*, numerical stability considerations, reference command continuity, and condition independent tuning strategies. Finally, we demonstrate the effectiveness of the control strategy on a small fixed-wing test platform in thorough flight experiments in mountainous terrain with strong, turbulent winds. The

resulting guidance is, to the authors' best knowledge, the first example in literature of an algorithm considering both excess wind conditions on small fixed-wing UAVs as well as providing the means to fully prevent vehicle run-away and maintain track keeping.

The remainder of the paper is structured as follows: Section 2 presents the objective formulation and the concept of *bearing feasibility*, Section 3 outlines our previously developed directional guidance strategy [19] for both lower and *excess* wind conditions, detailing specific enhancements and modifications in the present work, Section 4 develops a new airspeed reference compensation logic, and Section 5 concludes with experimental results.

## 2 Bearing Feasibility

We consider a fixed-wing aircraft flying horizontally in two dimensions with ground velocity  $\mathbf{v}_G = \mathbf{v}_A + \mathbf{w}$  the sum of airspeed  $\mathbf{v}_A$  and wind  $\mathbf{w}$  vectors. In the case that the wind speed exceeds the UAV's airspeed, feasibility of flying a given bearing  $\chi_{\text{ref}}$  depends on the wind direction. A binary definition of the *bearing feasibility* can be formulated as:

$$\begin{aligned} \beta \sin |\lambda| \geq 1 \cup (|\lambda| \geq \frac{\pi}{2} \cap \beta > 1) & \quad (\text{infeasible}) \\ \text{else} & \quad (\text{feasible}) \end{aligned} \quad (4.1)$$

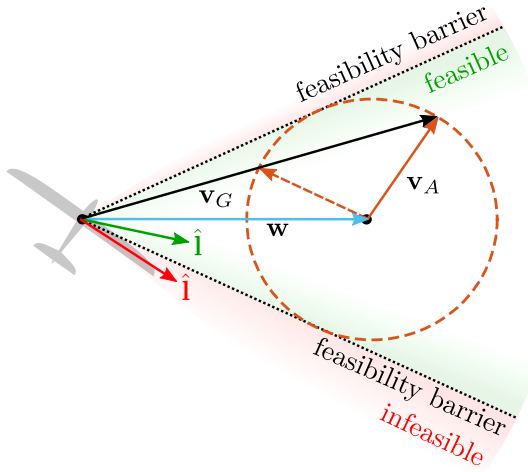
where the wind ratio  $\beta = w/v_A$  is the fraction of wind speed  $w = \|\mathbf{w}\|$  over airspeed  $v_A = \|\mathbf{v}_A\|$  and  $\lambda$  is the angle between the wind  $\mathbf{w}$  and (unit) bearing  $\hat{\mathbf{l}} = [\cos \chi_{\text{ref}}, \sin \chi_{\text{ref}}]^T$  vectors:

$$\lambda = \text{atan2} \left( \mathbf{w} \times \hat{\mathbf{l}}, \mathbf{w} \cdot \hat{\mathbf{l}} \right) \in [-\pi, \pi] \quad (4.2)$$

The relationship in (4.1) physically describes a "feasibility cone", fully open when  $\beta < 1$  and asymptotically decreasing to zero angular opening as  $\beta \rightarrow \infty$ , see Fig. 4.2. When the bearing  $\hat{\mathbf{l}}$  lies within this cone the bearing is *feasible* and contrarily, when outside, *infeasible*.

Two separate tracking objectives can then be intuited: 1) an *ideal* tracking objective, where we are able to track the prescribed bearing and 2) a *safety* objective, where we instead tend towards reducing run-away by turning against the wind and simultaneously leveling the aircraft as  $t \rightarrow \infty$ , where  $t$  is time. When the vehicle remains on or near the feasibility boundary (common when the wind speed is approaching the airspeed and small gusts or turbulence are present), it is desirable to transition continuously between these two states to avoid oscillating discretely between reference commands (see Section 3 for reference command generation). In [19], the following continuous feasibility function was proposed:

$$\text{feas}(\lambda, \beta) = \frac{\sqrt{1 - (\beta \sin \bar{\lambda})^2}}{\cos \bar{\lambda}} \quad (4.3)$$



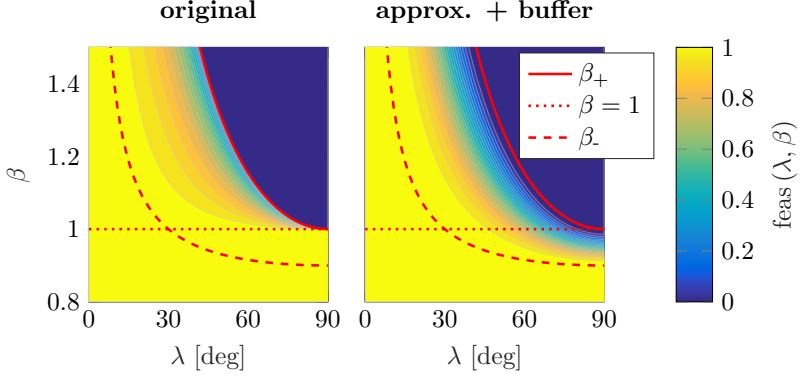
**Figure 4.2:** Feasibility “cone” (wind speed greater than airspeed). Note in the *excess wind* condition, two heading solutions exist for a given course.

where  $\text{feas}(\lambda, \beta) \in [0, 1]$  transitions from a value of 1 at “fully” feasible conditions ( $\beta < 1$ ) to 0 in infeasible conditions (definition in (4.1)), see Fig. 4.3 (left). Input  $\bar{\lambda} = \text{sat}(|\lambda|, 0, \frac{\pi}{2})$ , where operator  $\text{sat}(\cdot, \min, \max)$  saturates the input at the bounds min and max.

However, some practical issues exist with the function as defined in (4.3); namely:

- The function is continuous, but not smooth at the feasibility boundary, which can lead to fast changing and undesirably jagged reference commands.
- Numerical stability issues exist as  $\bar{\lambda} \rightarrow \frac{\pi}{2} \cap \beta \rightarrow 1$  due to the simultaneously decreasing magnitudes of the numerator and denominator (calculations with floating point precision on small microcontrollers then become an issue).
- A purely binary jump from feasible to infeasible conditions exists at  $|\lambda| \geq \frac{\pi}{2} \cap \beta = 1$ , which leads to jumping reference commands at a critical and common position in the state space: i.e. when the wind speed is very close the airspeed, the aircraft is facing against the wind  $\lambda = \pi$ , and small gusts perturb the system above and below the feasibility barrier.

To address these issues, a small buffer zone below the  $\beta = 1$  line is designed, considering some buffering wind ratio  $\beta_{\text{buf}} \in (0, 1)$ . The buffer’s magnitude may be set



**Figure 4.3:** Feasibility function: original formulation from [19] (left), new approximation with extended buffer zone (right).

e.g. corresponding to the airspeed reference tracking dynamics. An approximation of the feasibility function in (4.3) can be made incorporating the buffer zone, as well as maintaining both continuity and smoothness in the transition (see Fig. 4.3 (right)):

$$\text{feas}(\lambda, \beta) = \begin{cases} 0 & \beta > \beta_+ \\ \cos^2\left(\frac{\pi}{2} \text{sat}\left(\frac{\beta - \beta_-}{\beta_+ - \beta_-}, 0, 1\right)\right) & \beta > \beta_- \\ 1 & \text{else} \end{cases} \quad (4.4)$$

where the upper limit of the transitioning region  $\beta_+$  is approximated as a piecewise-continuous function with a linear finite cut-off to avoid singularities, the cut-off angle  $\lambda_{\text{co}}$  chosen small such that the regular operational envelope is not affected:

$$\beta_+ = \begin{cases} \beta_{+\text{co}} + m_{\text{co}}(\lambda_{\text{co}} - \bar{\lambda}) & \bar{\lambda} < \lambda_{\text{co}} \\ 1/\sin \bar{\lambda} & \text{else} \end{cases} \quad (4.5)$$

with  $\beta_{+\text{co}} = 1/\sin \lambda_{\text{co}}$  and  $m_{\text{co}} = \cos \lambda_{\text{co}}/\sin \lambda_{\text{co}}^2$ . The lower limit of the transitioning region  $\beta_-$  is similarly made piecewise-continuous to correspond with  $\beta_+$ :

$$\beta_- = \begin{cases} \beta_{-\text{co}} + m_{\text{co}}(\lambda_{\text{co}} - \bar{\lambda})\beta_{\text{buf}} & \bar{\lambda} < \lambda_{\text{co}} \\ (1/\sin \bar{\lambda} - 2)\beta_{\text{buf}} + 1 & \text{else} \end{cases} \quad (4.6)$$

where  $\beta_{-\text{co}} = (1/\sin \lambda_{\text{co}} - 2)\beta_{\text{buf}} + 1$ .



### 3 Directional Guidance

For purely directional guidance, traditional look-ahead approaches ([10, 75]) consider a constant speed unicycle model directionally driven via normal acceleration command

$$a_{G,\text{ref}}^N = kv_G^2 \sin \eta \quad (4.7)$$

typically defined about the ground speed vector, where  $k$  is a proportional gain and  $\eta = \chi_{\text{ref}} - \chi \in [-\pi, \pi]$  is the angular error in course  $\chi$  from the bearing  $\chi_{\text{ref}}$ , corresponding to look-ahead vector  $\hat{\mathbf{l}}$ . While a powerful control law, very high wind ratios degrade the performance and the ground speed based formulation does not handle *excess wind* conditions (i.e.  $\beta \geq 1$ ) [19].

Noting any normal acceleration command in reality is applied about the aircraft's *velocity-axis*, the reference acceleration may be reformulated about the airspeed vector

$$a_{A,\text{ref}}^N = kv_A^2 \sin \eta_A \quad (4.8)$$

where  $\eta_A = \xi_{\text{ref}} - \xi \in [-\pi, \pi]$  is the angular error in heading  $\xi$  from the heading reference  $\xi_{\text{ref}}$ , corresponding to air-mass relative look-ahead vector  $\hat{\mathbf{l}}_A$ .

In the following sections, the construction of the air-mass relative look-ahead vector  $\hat{\mathbf{l}}_A$  is built up step-by-step: first considering a baseline, purely track-error based, nonlinear path following law, followed by feed-forward rotations both converting the ground relative bearing  $\chi_{\text{ref}}$  to an air-mass relative heading reference  $\xi_{\text{ref}}$  and considering the path curvature, and finally partitioning the control law into *feasible* and *infeasible* bearing cases while maintaining reference command continuity. The approach follows closely to that in [19], with extensions/enhancements over the original formulation highlighted at each stage. All vectors and rotations from the following development are displayed in Fig. 4.4.

#### 3.1 Baseline path following

We consider a mathematical definition of the *ideal tracking objective* described in Sec. 2, i.e. the desired convergence behavior of the guidance law, assuming the bearing is feasible.

$$\text{ideal obj.} \quad \begin{cases} \lim_{t \rightarrow \infty} \mathbf{e}(t) = 0 \\ \lim_{t \rightarrow \infty} (\hat{\mathbf{t}}_P(t) - \hat{\mathbf{v}}_G(t)) = 0 \\ \lim_{t \rightarrow \infty} \left( \frac{d}{dt} \hat{\mathbf{t}}_P(t) - \frac{d}{dt} \hat{\mathbf{v}}_G(t) \right) = 0 \end{cases} \quad (4.9)$$

where track-error  $\mathbf{e} = \mathbf{p} - \mathbf{r}$ ,  $\mathbf{r}$  is the vehicle position,  $\hat{\mathbf{t}}_P$  is the path's unit tangent vector at the closest point  $\mathbf{p}$ , and  $\hat{\mathbf{v}}_G = \mathbf{v}_G / \|\mathbf{v}_G\|$ ,  $\|\mathbf{v}_G\| \neq 0$  is the unit ground speed vector. In words, the vehicle should converge to the path while heading in the correct direction.



which the aircraft begins turning into the path with respect to the approaching ground speed. A ground speed cut-off  $v_{G,co}$  is incorporated within a piecewise quadratic function to smoothly saturate  $e_b$  as  $v_G \rightarrow 0$ , avoiding singularities. It is from a distance of  $e_b$  the look-ahead vector will begin to transition from normal to tangent bearings, with respect to the path.

At this point, we need to augment  $\hat{\mathbf{I}}$  for both wind and path curvature. Look-ahead  $\hat{\mathbf{I}}_A$  requires separate control laws for *feasible* and *infeasible* bearings, corresponding to the ideal and safety objectives, respectively.

### 3.2 Feasible Bearing

The look-ahead vector  $\hat{\mathbf{I}}$  describes a bearing necessary to drive convergence to a path with no curvature. In the case that the bearing is *feasible*, wind vector information may be utilized to translate appropriate heading references  $\xi_{ref}$  necessary to achieve the ground relative motion defined by the bearing. Towards this end, we rotate the ground-based look-ahead vector  $\hat{\mathbf{I}}$  by angle  $x$ :

$$x = \sin^{-1}(\beta \sin \lambda) \quad (4.13)$$

the resulting (curvature independent, i.e.  $\kappa_P = 0$ ) heading vector reference then

$$\begin{aligned} \hat{\mathbf{I}}_A^{\kappa_P=0} &= \mathbf{H}(x) \hat{\mathbf{I}} \\ \text{where } \mathbf{H}(\cdot) &= \begin{pmatrix} \cos\{\cdot\} & -\sin\{\cdot\} \\ \sin\{\cdot\} & \cos\{\cdot\} \end{pmatrix} \end{aligned} \quad (4.14)$$

To further account for path curvature, we consider the “on-track” wind triangle, i.e. angles  $\lambda_0$ ,  $x_0$ , and  $y_0$  and ground velocity  $\mathbf{v}_{G_0} = v_{G_0} \hat{\mathbf{t}}_P$  at point  $\mathbf{p}$ , where

$$\lambda_0 = \text{atan2}(\mathbf{w} \times \hat{\mathbf{t}}_P, \mathbf{w} \cdot \hat{\mathbf{t}}_P) \quad (4.15)$$

$$x_0 = \sin^{-1}(\beta \sin \lambda_0) \quad (4.16)$$

$$y_0 = \pi - |x_0| - |\lambda_0| \quad (4.17)$$

$$v_{G_0} = \sqrt{v_A^2 + w^2 - 2v_A w \cos y_0} \quad (4.18)$$

In this condition, imagining the vehicle is already tracking the path with  $\|\mathbf{e}\| = 0$  and  $\mathbf{v}_A$  aligned with  $\hat{\mathbf{I}}_A^{\kappa_P=0}$  (if calculated as in (4.14) from the respective  $x_0$ ), an additional normal acceleration

$$a_{G_0}^N = v_{G_0}^2 \kappa_P \quad (4.19)$$

is required to follow the path’s curvature. With a quasi-steady assumption on wind

and differentiating (4.17),

$$\dot{\gamma}_0 = -\dot{\lambda}_0 - \frac{\beta \cos \lambda_0 \dot{\lambda}_0}{\sqrt{1 + (\beta \sin \lambda_0)^2}} \quad (4.20)$$

Noting the relationship between normal acceleration, linear speed, and angular speed  $a^N = v\omega$ , and following the guidance law in (4.8) it also holds:

$$\dot{\lambda}_0 = \frac{a_{G_0}^N}{v_{G_0}} \quad (4.21)$$

$$\dot{\gamma}_0 = kv_A \sin \eta_{c_0} \quad (4.22)$$

where  $\eta_{c_0}$  is the necessary additional rotation through which  $\hat{\mathbf{i}}_A^{\kappa_P=0}$  must be transformed to obtain  $\hat{\mathbf{I}}_A$ . Hence, plugging (4.19) into (4.21), (4.21) into (4.20), then equating (4.22) and (4.20), we may obtain on-track curvature rotation:

$$\eta_{c_0} = \sin^{-1} \left( \text{feas}(\lambda_0, \beta) \frac{v_{G_0} \kappa_P}{v_A k} \left( 1 + \frac{\beta \cos \lambda_0}{\sqrt{1 - (\beta \sin \lambda_0)^2}} \right) \right) \quad (4.23)$$

We highlight three enhancements present in the curvature rotation defined in (4.23), compared to that in [19]:

- Considering only the on-track wind triangle avoids the necessity to saturate input to the arcsine function, previously required due to the mismatch between on-track angles and the aircraft centric wind triangle.
- We embed the on-track bearing feasibility  $\text{feas}(\lambda_0, \beta)$  in order to zero-out the arcsine input argument as we approach the feasibility barrier, and thus that of an infeasible state where considering curvature no longer makes sense.
- To avoid observed occasionally unintuitive flight trajectories caused by considering the path curvature when far from the track, an additional smooth limiter  $\sigma_l = \sin^2 \theta_l$  is included to bring in curvature adjustments only as we converge to the path:

$$\eta_c = \text{feas}(\lambda, \beta) \sigma_l \eta_{c_0} \quad (4.24)$$

where  $\text{feas}(\lambda, \beta)$  again zeros-out the consideration of path curvature as we approach the feasibility barrier; in this case in the aircraft centric frame to maintain reference continuity.

With the final rotation necessary for tracking curvature, the airspeed relative



$$\text{safety obj.} \quad \begin{cases} \lim_{t \rightarrow \infty} a_{A,\text{ref}}^N(t) = 0 \\ \lim_{t \rightarrow \infty} \hat{\mathbf{v}}_A(t) = -\hat{\mathbf{w}}(t) \\ \lim_{t \rightarrow \infty} \hat{\mathbf{e}}(t) = -\hat{\mathbf{w}}(t) \end{cases} \quad (4.28)$$

The latter of the requirements in (4.28) correspond to the desire to minimize “run-away” from the track. Convergence analysis of the safety objectives (4.28) defined for look-ahead law (4.27) may be found in [19] which is similarly applicable to the present formulation. Note, in this work, the infeasible look-ahead reference  $\hat{\mathbf{I}}_{A,\text{infeas}}$  always uses the “faster” heading solution of the two seen for the excess wind case in Fig. 4.2.

### 3.4 Tuning

The  $k$  bounds for guaranteeing curvature convergence can be derived by considering the steady-state conditions  $\mathbf{e} = \mathbf{0}$  in the “worst-case” scenario  $\hat{\mathbf{t}}_P = \hat{\mathbf{w}}$  (i.e. maximum required normal acceleration to maintain curvature), where  $v_{G_0} = v_A + w$  and  $\lambda_0 = 0$  are substituted within (4.19), (4.21), and (4.20):

$$\dot{y}_{\text{ss}} = -(v_A + w) |\kappa_P| (\beta + 1) \quad (4.29)$$

Further considering the input argument of the arcsine function in (4.23), it may be noticed that  $k > \dot{y}_{\text{ss}}/v_A$  to ensure the equation is well defined, this resulting in the following  $k$  bounds:

$$k > (1 + \beta)^2 |\kappa_P| \quad (4.30)$$

While the above initial analysis was also present in [19], we handle a previously unconsidered practical implementation of this bound, that of potentially variable wind ratios  $\beta$  and path curvature  $\kappa_P$ , in an adaptive way:

$$k_{\text{adj}} = k_{\text{max}} + \sigma_l (k - k_{\text{max}}) \quad (4.31)$$

$$k_{\text{max}} = \begin{cases} \max(k, k_{\text{mult}} (1 + \beta)^2 |\kappa_P|) & \beta \geq 1 \\ \max(k, 4k_{\text{mult}} |\kappa_P|) & \text{else} \end{cases} \quad (4.32)$$

where  $k_{\text{max}}$  is the maximum of the operator defined proportional gain  $k$  and the minimum required gain from (4.30) (with some tolerance,  $k_{\text{mult}}$ ), and  $k_{\text{adj}}$  is the resulting adjusted gain used by the controller. Note  $(1 + (\beta = 1))^2 = 4$ , which is held as a constant multiplier in the  $\beta < 1$  case. This logic alleviates the need for condition specific tuning and ensures convergence is maintained while still allowing operator defined dynamics of the control response whenever the bounds are not exceeded.

### 3.5 Control allocation

The commanded lateral acceleration from the directional guidance (following the control law in (4.8)) is translated into a roll angle reference via the common coordinated turn assumption:  $\phi_{\text{ref}} = \tan^{-1} \left( a_{A,\text{ref}}^N / g \right)$ , where  $g$  is the acceleration of gravity. It is then the task of the lower-level control loops to track this reference, see Sec. 5 for control architecture details.

## 4 Airspeed Reference Compensation

In this section, we extend the high-level guidance logic by adding an additional control, the airspeed reference, developing an energy efficient *airspeed compensation* logic. With the assumption that extra commanded airspeed entails extra energy usage, we wish to only increase the reference as much as necessary to prevent run-away (stay on track) until winds have receded. I.e. when the feasibility barrier has been crossed, an ideal equilibrium point of  $v_G = \|\mathbf{e}\| = 0$  should be approached. The following subsections develop successive stages of compensation logic; namely, *wind excess regulation*, *track keeping*, and *minimum ground speed maintenance*.

### 4.1 Wind Excess Regulation

To achieve the first component of the desired equilibrium,  $v_G = 0$ , we define a positive speed increment  $\Delta v_A^w$  corresponding to the difference between the wind speed  $w$  and nominal reference airspeed  $v_{A,\text{nom}}$ ; i.e. the *excess wind speed*  $\Delta w = \text{sat}(w - v_{A,\text{nom}}, 0, \Delta v_{A,\text{max}})$ , where  $\Delta v_{A,\text{max}} = \max(v_{A,\text{max}} - v_{A,\text{nom}}, 0)$  is the maximum allowed airspeed reference increment, derived from the maximum available airspeed setting  $v_{A,\text{max}}$ :

$$\Delta v_A^w = \Delta w (1 - \text{feas}(\lambda, \beta)) \quad (4.33)$$

This wind excess -based speed increment may be added to the nominal reference  $v_{A,\text{nom}}$  towards regulating  $v_G \rightarrow 0$ , though small perturbations will induce small steady-state tracking errors which may only grow over time, unless wind speeds recede.

### 4.2 Track Keeping

To further stay on track in excess wind speeds, an additional speed increment  $\Delta v_A^e$  corresponding to the normalized track-error  $\bar{e}$  may be defined:

$$\Delta v_A^e = \Delta v_{A,\text{max}}^e k_{\bar{e}} k_w (1 - \text{feas}(\lambda, \beta)) \quad (4.34)$$

where  $\Delta v_{A,\text{max}}^e$  is the maximum allowed speed increment generated from track-error. The gains  $k_{\bar{e}}$  and  $k_w$  are used to tune track-error and wind speed excess derived

saturation ramps:

$$k_{\bar{e}} = \text{sat} \left( \frac{\bar{e}}{\bar{e}_{\text{buf}}}, 0, 1 \right) \quad (4.35)$$

$$k_w = \text{sat} \left( \frac{\Delta w}{\Delta w_{\text{buf}}}, 0, 1 \right) \quad (4.36)$$

$k_{\bar{e}}$  is scaled by a chosen fraction of the normalized track-error  $\bar{e}_{\text{buf}}$ , setting the proximity at which  $\Delta v_{A,\text{max}}^e$  is applied in full, while  $k_w$  is scaled by  $\Delta w_{\text{buf}}$  to ensure no airspeed increment is applied in the condition that the feasibility function lies within the extended buffer zone below  $\beta = 1$ . The track offset -based speed increment  $\Delta v_A^e$  assists  $\Delta v_A^w$  by increasing the airspeed enough to overpower the current wind speed, returning the aircraft to the path, at which point the term again zeros out. With both increments in play, the augmented airspeed reference combines them as follows:

$$v_{A,\text{ref}} = v_{A,\text{nom}} + \min(\Delta v_A^w + \Delta v_A^e, \Delta v_{A,\text{max}}) \quad (4.37)$$

### 4.3 Maintaining a Minimum Forward Ground Speed

Though the incremented airspeed reference in (4.37) will maintain zero track-error, it may further be desirable that forward progress is made on a given path – e.g. if the vehicle should attempt to return home, or at the least complete its current set of mission waypoints. Towards this end, the bearing feasibility function can be utilized for efficient tracking of an operator-set minimum *forward ground speed*  $v_{G,\text{min}}$ , with forward ground speed defined as the 2D (horizontal) projection onto the airspeed vector  $v_{G,\text{fwd}} = \mathbf{v}_G \cdot \frac{\mathbf{v}_A}{\|\mathbf{v}_A\|}$ .  $v_{G,\text{min}}$  may then be used to augment the wind ratio and, further, airspeed increments, effectively imitating a higher wind speed which the logic must compensate:

$$\beta_G = \frac{w + v_{G,\text{min}}}{v_A} \quad (4.38)$$

$$\Delta w = \text{sat}(w - v_{A,\text{nom}} + v_{G,\text{min}}, 0, \Delta v_{A,\text{max}}) \quad (4.39)$$

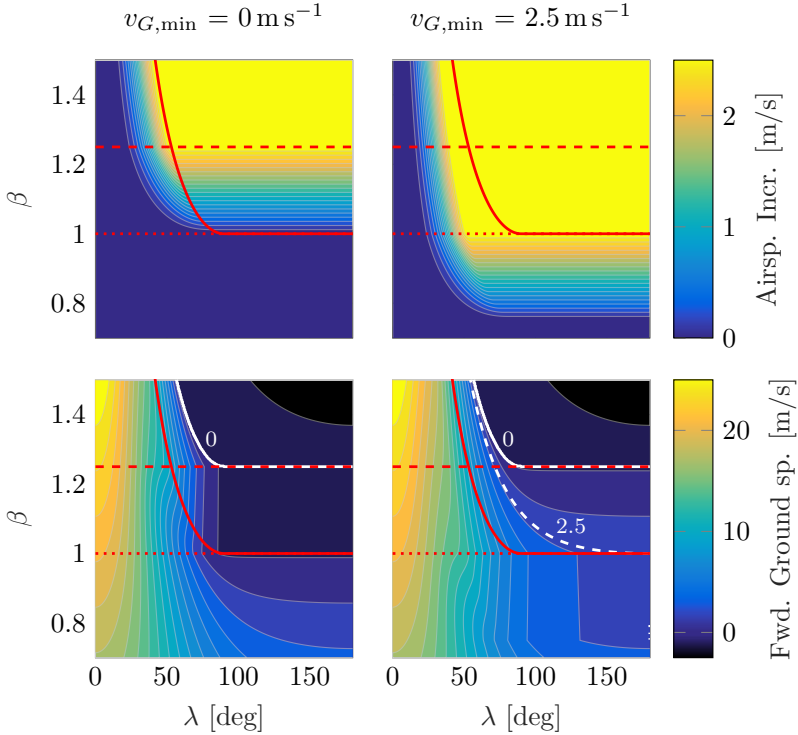
$$\Delta v_A^w = \Delta w(1 - \text{feas}(\lambda, \beta_G)) \quad (4.40)$$

Note that with a minimum ground speed defined, the track keeping logic defined in Sec. 4.2 should be disabled, i.e.  $\Delta v_A^e = 0$ . Fig. 4.6 illustrates the feed-forward airspeed reference mapping resulting from the minimum ground speed logic.

## 5 Flight Experiments

In this section, we present flight demonstrations on a small (1.8 m wingspan, 1.3 kg), low-speed test platform, Easyglider (see Fig. 4.1), from two mountainous regions in Switzerland, showcasing the performance of the developed guidance laws in strong





**Figure 4.6:** Airspeed reference compensation and resulting forward ground speed with (right) and without (left) a minimum ground speed applied (note no track-offset increment is added here). For illustrative purposes, the look-ahead logic defined in Sec. 3 is assumed perfectly tracked (or in steady-state condition).  $v_{A,\text{nom}} = 10 \text{ m s}^{-1}$  (dotted red line) and  $v_{A,\text{max}} = 12.5 \text{ m s}^{-1}$  (dashed red line). The solid red line indicates the feasibility boundary. Note airspeed is only incremented as necessary to achieve the desired forward ground speed until reaching the upper saturation bound of  $\Delta v_{A,\text{max}}$ .

winds. The guidance algorithm has been programmed in C/C++ on a *Pixhawk* autopilot (168 MHz Cortex-M4F microcontroller with 192 kB RAM) running *PX4*<sup>2</sup> firmware. PX4 implementations of a cascaded PID-based attitude/rate control (with feed-forward turn compensation), airspeed and altitude control via Total Energy Control System (TECS) [6], and an online Extended Kalman Filter (EKF) are utilized for tracking guidance commands and feeding back state estimates, respectively. We note that all underlying control and estimation structures are operational with a standard low-cost sensor suite for small fixed-wing UAVs (see exemplary sensor selection in [61]) and further require no model-based assumptions. Guidance parameters held constant for both flights may be found in Table 4.1. All displayed airspeeds are “true” airspeeds (TAS), i.e. relative to the airmass. To keep wind, ground speed, and TAS inputs to the guidance algorithm compatible with each other, a “filtered” TAS estimate is obtained by subtracting the wind estimate from the GNSS velocity. As the wind estimate is already filtered, this further smooths out the typically noisy airspeed measurements (from a pitot-static tube with differential and ambient pressure sensors) that would otherwise degrade the guidance commands.

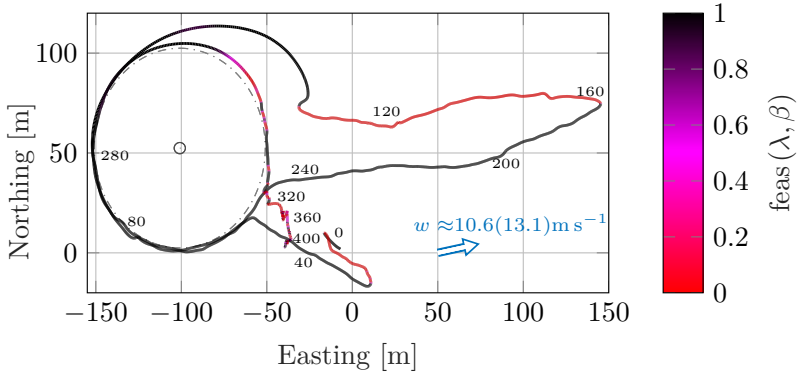
**Table 4.1:** Guidance parameters used in flight experiments.

| Param                 | Value | Unit | Param                       | Value | Unit              |
|-----------------------|-------|------|-----------------------------|-------|-------------------|
| $\beta_{\text{buf}}$  | 0.1   | - -  | $v_{A,\text{nom}}$          | 8.8   | $\text{m s}^{-1}$ |
| $\lambda_{\text{co}}$ | 1.0   | °    | $v_{A,\text{max}}$          | 15.0  | $\text{m s}^{-1}$ |
| $v_{G,\text{co}}$     | 1.0   | - -  | $\epsilon_{\text{buf}}$     | 0.5   | - -               |
| $T_b$                 | 7.0   | s    | $\Delta w_{\text{buf}}$     | 0.5   | $\text{m s}^{-1}$ |
| $k$                   | 0.11  | - -  | $\Delta v_{A,\text{max}}^e$ | 3.0   | $\text{m s}^{-1}$ |
| $k_{\text{mult}}$     | 1.1   | - -  |                             |       |                   |

## 5.1 Wind Excess Regulation and Minimum Ground Speed Maintenance

Figures 4.7 and 4.8 show the position and state trajectories of a fully automatic ca. 7 min flight experiment on a windy plateau in the Jura Mountain range in Switzerland. Stages (I)-(V) in Fig. 4.8 step through various compensated and uncompensated guidance modes, however no track keeping is enabled. A clear reduction in run-away can be seen in Fig. 4.7 when comparing the uncompensated case (III:  $t=101\text{-}142\text{s}$ ) to that of wind excess regulation (V:  $t=283\text{-}414\text{s}$ ). Further, when commanded, minimum forward ground speed is maintained with a mean undershoot of  $0.51 \text{ m s}^{-1}$  (with one standard deviation error  $1.07 \text{ m s}^{-1}$ ), increasing the airspeed reference when the bearing would otherwise become infeasible, and decreasing appropriately on the down wind legs, see (II), (IV), and (V); this, however, with the exception of the stall denoted by (VI).

<sup>2</sup><http://dev.px4.io>



**Figure 4.7:** Flight experiment: Lamboing, Switzerland (892 AMSL - Plateau de Diesse). Aircraft position colored with the current bearing feasibility. Mean wind speed was  $10.6 \text{ m s}^{-1}$ , gusting to  $13.1 \text{ m s}^{-1}$  during the flight period.

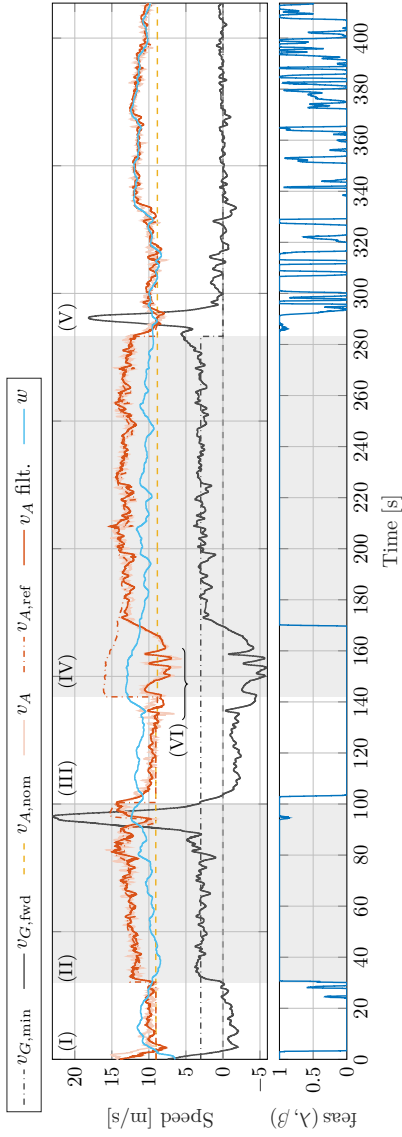
At  $t = 135 \text{ s}$ , while the Easyglider is running away at up to  $-5 \text{ m s}^{-1}$  forward ground speed (due to the disabled airspeed compensation logic), the aircraft encounters some turbulence (note the abrupt large airspeed fluctuations) followed by a gust increase of  $2.7 \text{ m s}^{-1}$ . These effects induce a stall from which the lower-level control loops spend the next several seconds regaining control. Once lower-level stabilization is regained, tracking of the guidance commands quickly resumes. This momentary lapse in low-level stabilization highlights the risk of flying small UAVs in such conditions and further motivates consideration of coupled longitudinal effects (e.g. angle of attack) within future iterations of wind-robust guidance and control.

## 5.2 Track Keeping

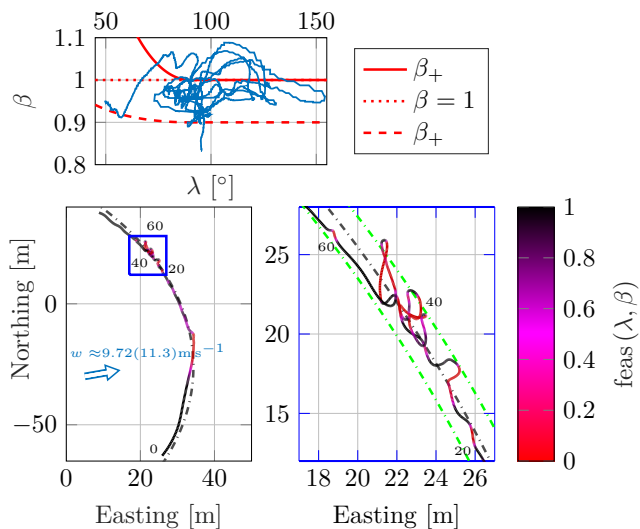
Figures 4.9 and 4.10 show position and state trajectories for a portion of a flight experiment conducted on the ridge of Uetliberg Mountain, Switzerland where the track keeping mode is enabled. Figure 4.9 details a full 40 s in which the guidance holds the aircraft at near zero ground speed with less than 1 m track error, despite facing into gusting winds nearly constantly above the nominal airspeed. In Fig. 4.9,  $\Delta v_A^w$  can be seen effectively tracking the wind excess, while  $\Delta v_A^e$  adjusts for deviations from the track caused by smaller turbulences and/or gusts. The importance of the bearing feasibility function's buffer zone is further highlighted here, as  $\beta$  stays near 1, and  $\lambda$  near  $90^\circ$ .

## **6 Discussion & Future Work**

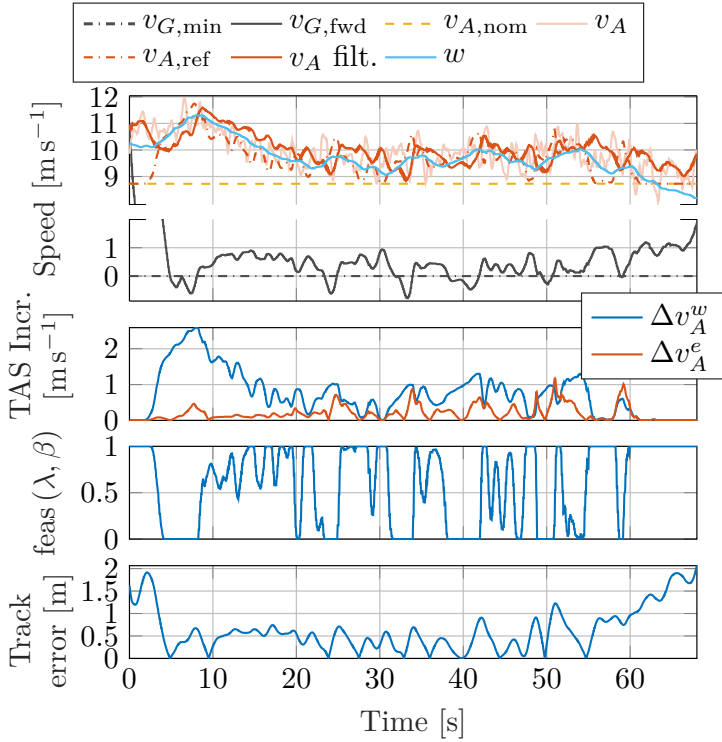
In this work, we developed and demonstrated a novel guidance law capable of preventing run-away of small, low-speed fixed-wing UAVs flying in strong winds. Though the controller was shown effective in flight experiments, one particularly noteworthy observation during testing was that care should be taken in tuning of noise values for the wind estimates within the EKF. Though highly dependent on the given estimation formulation, a general guidance may be taken that too slow or fast tuning of wind estimation responses causes, respectively, too slow or fast reactions the airspeed compensation logic, leading in the prior case to steady state tracking errors, and in the latter case to noisy, oscillatory guidance commands. A trade-off between the scale of gusts one wishes to capture vs the performance of the controller should be weighed. It is further apparent that flight within very turbulent conditions will require future work on coupled wind-robust algorithms considering longitudinal lower-level dynamics of the UAV. The authors lastly note that, though not shown here for brevity, the developed guidance law has further been applied/tested for line segment following.



**Figure 4.8:** Flight experiment: Lamboing, Switzerland (892 AMSL - Plateau de Diesse). Speeds and bearing feasibility. I. Automatic (backwards) take-off, airspeed compensation disabled, II. Approach and tracking of loiter circle with  $v_{G,\min} = 3 \text{ m s}^{-1}$ , III. Airspeed compensation disabled, leading to run-away bottoming out near  $v_{G,\text{fwd}} = -5 \text{ m s}^{-1}$ , IV.  $v_{G,\min} = 3 \text{ m s}^{-1}$  re-enabled, V. Wind excess regulation is demonstrated by setting  $v_{G,\min} = 0 \text{ m s}^{-1}$ , VI. Stall due to turbulence and gust increase while flying with negative ground speed.



**Figure 4.9:** Flight experiment: Uetliberg, Switzerland (943 AMSL - mountain top). Aircraft position colored with the current bearing feasibility. Green dashed-dot lines within the zoom-in plot (right) indicate 1 m bounds on the path. Mean wind speed was  $9.72 \text{ m s}^{-1}$ , gusting to  $11.3 \text{ m s}^{-1}$  during the flight period. Wind ratio vs. the angle between wind and bearing vectors over feasibility function boundaries (top).



**Figure 4.10:** Flight experiment: Uetliberg, Switzerland (943 AMSL - mountain top). Aircraft states for a track keeping experiment in strong ridge updrafts and horizontal winds.





Paper



# Wind Fighting Efficiency Revisited: Coupling Airspeed and Heading Guidance for Fixed-wing UAVs

Thomas Stastny

## Abstract

This technical brief investigates the energy efficiency of the lateral-directional guidance law developed in Papers I and II, exploring how taking a coupled approach to airspeed and heading commands can not only reduce energy demands, but may further improve tracking performance in highly variable winds. Simulation comparisons of the previous law and new coupled approach are presented which indicate preliminary support to this hypothesis.

## 1 Background & Scope

In Papers I and II, we developed lateral-directional path following guidance logic for small fixed-wing unmanned aerial vehicles (UAVs) flying in winds which may exceed the vehicle's nominal airspeed command. In particular, Paper II demonstrated how we may utilize the airspeed reference in an efficient way towards combating run-away of the aircraft. The airspeed reference compensation was run in a parallel feedback loop, decoupled from the directional commands.

While the controller was demonstrated effective in numerous flight experiments, the underlying approach, specifically the airspeed reference compensation, was largely heuristic and no deeper analysis was presented on the efficiency of the commands. Our aim in this brief is to investigate particularly this aspect of energy efficiency, exploring how taking a coupled approach to airspeed and heading commands can not only reduce energy demands, but further may potentially improve tracking performance in highly variable winds.

Section 2 will first provide a brief review of the previous guidance logic. Section 3 will develop a new coupled formulation, followed by a comparative simulation study in Section 4.

## 2 Windy Guidance Fundamentals

In this section, we provide a brief refresher of the guidance formulation and problem definition from Papers I and II. Note some details will be omitted, and it is intended that the reader refer to the previous two works for any definition not presented here in full.

### 2.1 Problem Definition

Consider a small fixed-wing UAV at two-dimensional (North, East) position  $\mathbf{p}$  with (air mass relative) airspeed vector  $\mathbf{v}_A$  composed of airspeed  $v_A$  and heading (clockwise from North)  $\xi$  flying in a static wind field with velocity  $\mathbf{v}_W$ . Air and wind velocities are summed resulting in ground velocity  $\mathbf{v}_G = \mathbf{v}_A + \mathbf{v}_W$ . The ratio of the wind speed to airspeed is notated as  $\beta = v_W/v_A$ . The aircraft is constrained to operate between a nominal (desired) and maximum airspeed,  $v_{A,\text{nom}}$  and  $v_{A,\text{max}}$ , respectively.

A desired ground-relative *bearing* is provided to the aircraft, here represented as unit vector  $\hat{\mathbf{I}}$ . Path following logic which generates this bearing may be found in Papers I and II. It is expected for the purposes of this brief that further, a closest point  $\mathbf{p}_P$  on the path to the aircraft as well as the path tangent  $\hat{\mathbf{t}}_P$  at that point are computable. This leads to a definition of the track error vector  $\mathbf{e} = \mathbf{p}_P - \mathbf{p}$ .

Translating bearings into heading references considers the geometry of the so-called "wind triangle", i.e. the triangle created by connecting the wind and airspeed vector sum to, ideally, a solution with the resulting ground speed vector lying in the direction of the current bearing. The rotation from ground velocity to air velocity

runs through angle  $x$ . The angle between wind and bearing vectors, a quantity used quite heavily in the preceding works, we notate as  $\lambda$ .

In Paper II, we formulated a “bearing feasibility” function  $\text{feas}(\lambda, \beta) \in [0, 1]$ , which we will further employ here. This function’s output describes the “feasibility” of following a given bearing at the given wind ratio, one and zero indicating fully feasible and fully infeasible, respectively.

*Ideal* and *safety* objectives were developed for the feasible and infeasible, respectively, tracking cases:

$$\begin{array}{ll}
 \text{Ideal Objective:} & \text{Safety Objective:} \\
 \left\{ \begin{array}{l} \lim_{t \rightarrow \infty} \|\mathbf{e}(t)\| = 0 \\ \lim_{t \rightarrow \infty} (\hat{\mathbf{t}}_P(t) - \hat{\mathbf{v}}_G(t)) = \mathbf{0} \\ \lim_{t \rightarrow \infty} \left( \frac{d}{dt} \hat{\mathbf{t}}_P(t) - \frac{d}{dt} \hat{\mathbf{v}}_G(t) \right) = \mathbf{0} \end{array} \right. & \left\{ \begin{array}{l} \lim_{t \rightarrow \infty} a_{A,\text{ref}}^N(t) = 0 \\ \lim_{t \rightarrow \infty} \hat{\mathbf{v}}_A(t) = -\hat{\mathbf{w}}(t) \\ \lim_{t \rightarrow \infty} \hat{\mathbf{e}}(t) = -\hat{\mathbf{w}}(t) \end{array} \right. \quad (5.1)
 \end{array}$$

where  $a_{A,\text{ref}}^N$  is a commanded acceleration normal to the airspeed vector (output from the directional guidance as a function of heading error).

## 2.2 Directional Guidance

We first define the bearing “feasibility” conditions, i.e. when the bearing can actually be tracked, given the current wind ratio  $\beta$  and wind-to-bearing angle  $\lambda$ :

$$\begin{cases} \text{infeasible} & |\mathbf{v}_W \times \hat{\mathbf{l}}| \geq v_A \cup (\mathbf{v}_W \bullet \hat{\mathbf{l}} \leq 0 \cap v_W \geq v_A) \\ \text{feasible} & \text{else} \end{cases} \quad (5.2)$$

where  $v_W \sin \lambda = \mathbf{v}_W \times \hat{\mathbf{l}}$  and  $v_W \cos \lambda = \mathbf{v}_W \bullet \hat{\mathbf{l}}$ ; the symbols  $\times$  and  $\bullet$  denoting the (scalar) vector cross product and dot product, respectively, of the fore and aft vectors.

If bearing is *feasible* and we take the “high speed” solution:

$$v_A \sin x = \mathbf{v}_W \times \hat{\mathbf{l}} \quad (5.3)$$

$$v_A \cos x = \text{proj}_{\hat{\mathbf{l}}} \mathbf{v}_A \quad (5.4)$$

$$v_{G,l} = \mathbf{v}_W \bullet \hat{\mathbf{l}} + \text{proj}_{\hat{\mathbf{l}}} \mathbf{v}_A \quad (5.5)$$

$$\sin \eta_c = \sigma_l \text{feas}(\lambda, \beta) \text{feas}(\lambda_0, \beta) \frac{v_G \kappa_P}{v_A k} \left( 1 + \frac{\mathbf{v}_W \bullet \hat{\mathbf{l}}}{\text{proj}_{\hat{\mathbf{l}}} \mathbf{v}_A} \right) \quad (5.6)$$

$$\cos \eta_c = \sqrt{1 - \sin^2 \eta_c} \quad (5.7)$$

$$\text{proj}_{\hat{\mathbf{l}}} \mathbf{v}_A = \sqrt{v_A^2 - (\mathbf{v}_W \times \hat{\mathbf{l}})^2} \quad (5.8)$$

where  $v_{G,l}$  is the ground speed along the bearing vector.  $\text{feas}(\lambda, \beta)$  and  $\text{feas}(\lambda_0, \beta)$  are instantaneous and on-track bearing feasibilities, "on-track" (denoted with subscript 0) indicating that we calculate the feasibility of the path tangent at the current closest point on the path from the aircraft at the current aircraft orientation and speed.  $\sigma_l$  is a smooth function of the track error bringing in the effect of this curvature dependent rotation only when we are near to the track. The multiplied feasibilities serve to zero-out the curvature adjustment when tracking the bearing is infeasible. Controller gain  $k$  is further elaborated in Paper II.

If the bearing is *feasible* and we take the "low speed" (or "backwards") solution, the following quantities are augmented as follows:

$$v_A \cos x = -\text{proj}_{\hat{\mathbf{l}}} \mathbf{v}_A \quad (5.9)$$

$$v_{G,l} = \mathbf{v}_W \bullet \hat{\mathbf{l}} - \text{proj}_{\hat{\mathbf{l}}} \mathbf{v}_A \quad (5.10)$$

For a given airspeed, equations (5.3) and (5.4) can be formed into a rotation matrix through angle  $x$ , and similarly equations (5.6) and (5.7) through  $\eta_c$ , transforming the bearing vector into a heading reference  $\hat{\mathbf{l}}_A$ .

If the bearing is *infeasible*, the safety objective is satisfied using the following control:

$$\hat{\mathbf{l}}_A = \frac{\sqrt{v_W^2 - v_A^2} \hat{\mathbf{l}} - \mathbf{v}_W}{\|\sqrt{v_W^2 - v_A^2} \hat{\mathbf{l}} - \mathbf{v}_W\|} \quad (5.11)$$

The relevant geometry from the above equations is visualized in Fig. 5.1.

**Practical Note.** *All calculations in this section and those that follow may be computed without use of any trigonometric functions which can save processing power on small, embedded autopilot platforms.*

### 2.3 Decoupled Airspeed Reference; Feedback Approach

In the previous works, the airspeed reference was utilized to combat wind excess (wind speeds exceeding the nominal airspeed) in a primarily feedback formulation, i.e. waiting for any feedback track errors or forward ground speed undershooting, then compensating by incrementing the airspeed reference above the nominal speed. Several modes were established:

- **Mitigation:** use only the directional guidance and reduce the *rate* of run-away from the track.
- **Regulation:** increment the airspeed reference to match the current wind speed, i.e. *regulating* wind excess.
- **Track Keeping:** inject additional energy (airspeed) to 1) overcome wind excess and 2) return to the track.
- **Minimum Ground Speed:** increase the airspeed reference to maintain a minimum forward ground speed.



where  $\Delta v_{A,\max} = v_{A,\max} - v_{A,\text{nom}}$ ,  $\Delta v_{A,e,\max}$  is the applied increment at the user defined track error threshold,  $k_e$  is a function of the track error,  $k_w$  prevents incrementing when  $\beta < 1$ , and  $\text{constrain}(\cdot, \min, \max)$  is a clamp function.

Minimum ground speeds were maintained by “faking” a higher wind speed in the wind ratio and airspeed increment feedback, i.e.:

$$\beta_G = \frac{v_W + v_{G,\min}}{v_A} \quad (5.15)$$

$$\Delta v_{A,w} = \text{constrain}(v_W - v_{A,\text{nom}} + v_{G,\min}, 0, \Delta v_{A,\max})(1 - \text{feas}(\lambda, \beta)) \quad (5.16)$$

### 3 Coupling Heading and Airspeed References

While the airspeed compensation approach in the prior work was shown effective in satisfying the objectives, the airspeed increments were heuristic by design and no proper investigation into their efficiency was shown. Further, we hypothesize that coupling the heading and airspeed references could additionally improve tracking performance, feeding forward the the correct directional command corresponding to the airspeed dependent wind triangle solution.

This section will develop a new coupled formulation following a principled approach to minimizing airspeed demand, the underlying assumption being that the nominal reference airspeed is optimized for efficiency by the operator and, within our flight envelope, any additional airspeed demand equates to greater power demands. Thus, our aim is to satisfy our objectives while minimizing the airspeed usage.

#### 3.1 Minimum Ground Speed Redefined

We define the “along-bearing” ground speed  $v_{G,l}$ , as our forward ground speed maintenance objective (corresponding to our desired minimum ground speed  $v_{G,\min}$ ), where forward ground speed is defined  $v_{G,\text{fwd}} = \mathbf{v}_G \bullet \hat{\mathbf{l}}$ , as oppose to  $\mathbf{v}_G \bullet \hat{\mathbf{v}}_A$  in Paper II. Our rationale being that forward progress along the bearing is the actual objective we wish to achieve, independent of our aircraft’s heading (so long as we satisfy command continuity). Further, as we’ll see in the development below, the wind triangle gives us an explicit solution for along-bearing ground speed, whereas maintaining the projection onto the heading vector would require a numerical solution.

#### 3.2 Reference Logic

If we are given a specific airspeed at which to operate, the previous works have already provided means by which to compute a wind triangle solution. E.g., if we want to operate at nominal airspeed  $v_{A,\text{nom}}$ , we can check the feasibility of the desired bearing  $\hat{\mathbf{l}}$  and then follow the logic provided in Section 2.2. When our bearing is *nominally infeasible*, i.e. infeasible for the nominal airspeed, but we have more allowable airspeed (up to the maximum) to use in combating the wind, the question

is then what reference to set our airspeed to, and what heading that reference corresponds to. We further want any air velocity command to continuously change with continuously variable bearings and wind ratios, and attempt to minimize airspeed usage while satisfying any given objectives.

To start, we'll consider simply attempting to stay on the desired the bearing at any forward speed. In Fig. 5.1 (left), we can see a nominally feasible bearing, solved on the nominal airspeed circle. If we imagine the bearing line rotating further counterclockwise until it is aligned with the nominal feasibility barrier (tangent to the nominal airspeed circle), we reach a heading to bearing angle of  $90^\circ$ . Rotating the bearing further outside the nominally feasible region, we can see that the minimum airspeed solution to the wind triangle is then to maintain this right angle, making our minimizing airspeed:

$$v_{A,\min} = |\mathbf{v}_W \times \hat{\mathbf{i}}| \quad (5.17)$$

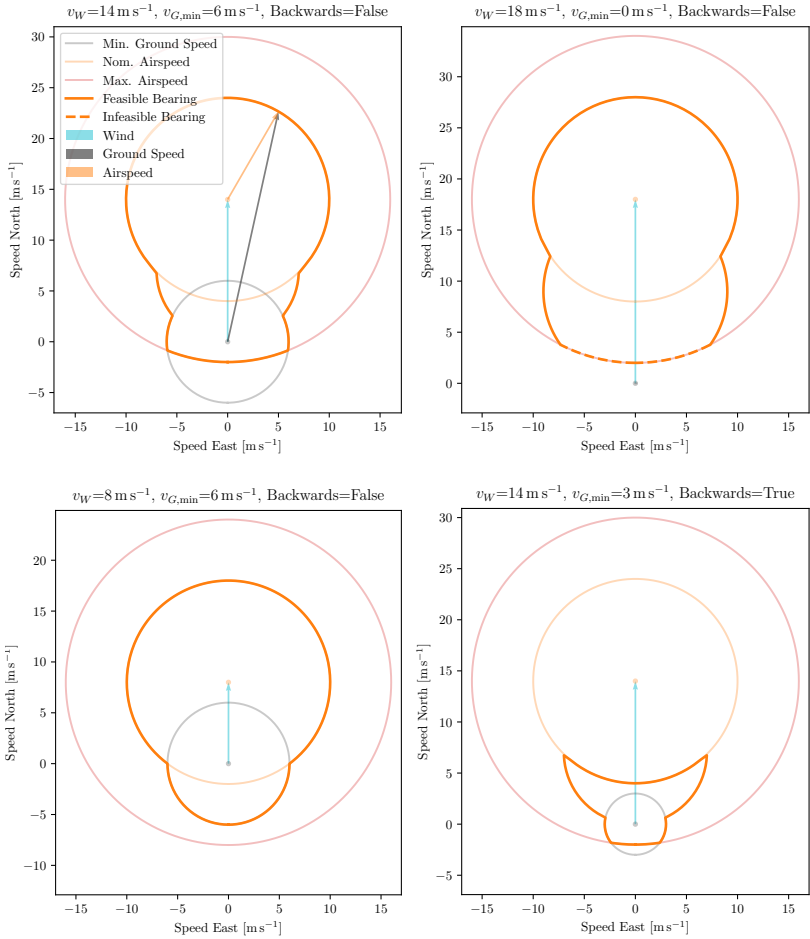
Rotating the bearing even further (beyond the maximally feasible region, i.e. all space contained within the  $v_{A,\max}$  radius), we then have no more option to increase the airspeed, and we can continuously default back to *mitigation* via (5.11).

The logic above effectively considers zero minimum ground speed in the maximally feasible region. If we want to make progress towards a goal, we may need to consider a non-zero minimum ground speed. Looking to Fig. 5.1 (right), we can see an example of the geometry of applying this minimum ground speed. If we start from the minimal airspeed usage condition shown in Fig. 5.1 (left), we can simply rotate the heading along the bearing line until intersect the minimum ground speed circle, and minimize our airspeed usage while still satisfying the desired along-bearing ground speed. Note this logic similarly applies for the nominally feasible condition; i.e. if the minimum ground speed radius obtrudes on the nominal airspeed radius, we can switch to following the prior as necessary. The minimizing airspeed for the violated minimum ground speed condition is then:

$$v_{A,\min} = \sqrt{(v_{G,\min} - \mathbf{v}_W \bullet \hat{\mathbf{i}})^2 + (\mathbf{v}_W \times \hat{\mathbf{i}})^2} \in v_{G,\min} > \mathbf{v}_W \bullet \hat{\mathbf{i}} \quad (5.18)$$

Figure 5.2 provides a more comprehensive visualization of how these airspeed reference paths look for different conditions in what we call “snowman” patterns. A pseudo algorithm is given in algorithm 1.

With an air velocity reference obtained from the new logic, we have both an airspeed and directional reference which can be applied to the vehicle. Note that path curvature may still be considered by rotating the air velocity reference through the path curvature adjustment angle. For the purposes of this brief, we will follow the same directional control allocation method as in Paper II.



**Figure 5.2:** “Snowman” patterns for various wind speeds and commanded minimum ground speeds. The highlighted curve shows, for each commanded bearing, the reference airspeed *and* heading necessary to achieve the desired minimum ground speed while minimizing airspeed usage. All diagrams assume the “fast” or “forward” nominally feasible solution except for the bottom right diagram, which demonstrates how the “backwards” nominally feasible solution behaves.  $v_{A,\text{nom}}$  and  $v_{A,\text{max}}$  are  $10 \text{ m s}^{-1}$  and  $16 \text{ m s}^{-1}$ , respectively, for all diagrams. Note the dashed “infeasible” reference refers to the inability to achieve the desired  $v_{G,\min}$ .



**Algorithm 1** Reference air velocity guidance.  $\text{solveFeasible}(\cdot)$  and  $\text{solveInfeasible}(\cdot)$  take an airspeed reference as input and solve for the feasible or infeasible wind triangle solution, respectively, via the geometry elaborated in Section 3.

---

```

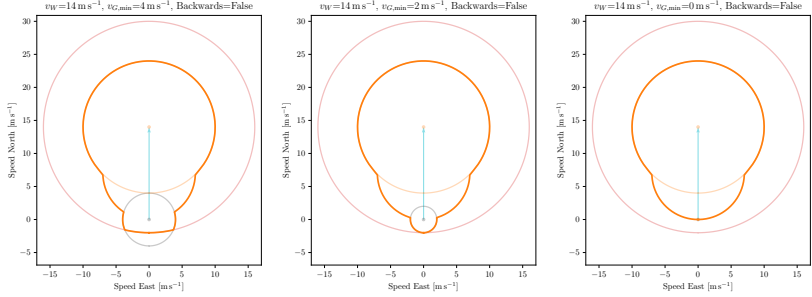
if  $v_{G,\min} > \mathbf{v}_W \bullet \hat{\mathbf{i}}$  then
    Minimum ground speed and/or track keeping:
    Compute  $v_{A,\min}$  with (5.18)
    if  $v_{A,\min} > v_{A,\max}$  then
        if bearing is maximally feasible then
             $\mathbf{v}_{A,\text{ref}} = \text{solveFeasible}(v_{A,\max})$ 
        else
             $\mathbf{v}_{A,\text{ref}} = \text{solveInfeasible}(v_{A,\max})$ 
        end if
    else if  $v_{A,\min} > v_{A,\text{nom}}$  then
         $\mathbf{v}_{A,\text{ref}} = \text{solveFeasible}(v_{A,\min})$ 
    else
         $\mathbf{v}_{A,\text{ref}} = \text{solveFeasible}(v_{A,\text{nom}})$ 
    end if
else
    Wind excess regulation and/or mitigation:
    if bearing is nominally feasible then
         $\mathbf{v}_{A,\text{ref}} = \text{solveFeasible}(v_{A,\text{nom}})$ 
    else if bearing is maximally feasible then
        if  $\mathbf{v}_W \bullet \hat{\mathbf{i}} \leq 0$  then
             $\mathbf{v}_{A,\text{ref}} = -\mathbf{v}_W$ 
        else
            Compute  $v_{A,\min}$  with (5.17)
             $\mathbf{v}_{A,\text{ref}} = \text{solveFeasible}(|\mathbf{v}_W \times \hat{\mathbf{i}}|)$ 
        end if
    else
         $\mathbf{v}_{A,\text{ref}} = \text{solveInfeasible}(v_{A,\max})$ 
    end if
end if
    
```

---

### 3.3 Track Keeping Objective as a Minimum Ground Speed

Differently from Paper II, we implement the track keeping objective as another superposed minimum ground speed using a simple proportional relationship to the normalized track error  $e' = \text{constrain}\left(\frac{\|\mathbf{e}\|}{e_b}, 0, 1\right)$ :

$$v_{G,\min_e} = v_{G,e} \text{ constrain}(k_e e', 0, 1) \quad (5.19)$$



**Figure 5.3:** Snowman patterns as the vehicle converges to the path (left to right) using a track keeping minimum ground speed.

where  $e_b$  is a speed dependent track error bound, see Paper II for definition,  $v_{G,e}$  is the minimum ground speed demand when  $k_e e' = 1$ , and  $k_e \geq 1$  (note the abuse of notation here) similar to Paper II scales the normalized track error to e.g. demand full minimum ground speed for more of the region between the track error bound and the path.

We can now use both the track keeping and minimum ground speed objectives simultaneously without any deleterious effects. An example of the progression of  $v_{G,\min_e}$  as the vehicle approaches the track can be seen in the track keeping snowman patterns in Fig. 5.3.

### 3.4 Applying the Backwards Solution

Primary reasons the “backwards” or “low-speed” heading solution to the feasible excess wind case has yet to be implemented in practice include 1) non-continuous heading references at unity wind ratio crossing and 2) non-trivial criteria for *when* either solution should optimally be utilized considering lags in aircraft dynamics and proximity to the track. For the purpose of this brief, we give an example of how the backwards heading solution could potentially be used with all other developed modes given that we are operating in *stable* excess wind, i.e. the wind speed is not expected to decrease below the nominal airspeed, where command continuity would then be violated. For the case that the nominal wind ratio is expected to vary above and below the unity line, either some mechanism of predicting the rate of this change or a hard bound on the excess wind level above which it is safe to apply the backwards solution would be required.

If our aim is to maintain the minimum energy solution in any airspeed compensation mode, we luckily do not need to change anything from the previous formulation. However, as the backwards solution is a “low-speed” one (ground relative), when we are far away from the track, it would benefit us to define some switch point before

which we can use the high speed solution and after provide enough time for a full turn around before converging to the path. Towards this end, a conservative “turn around” time  $t_{\text{turn}}$ , and corresponding distance  $d_{\text{turn}}$  can be defined as the time it takes to make a  $180^\circ$  turn in the current wind condition considering instantaneous acceleration at the maximum allowed roll angle and assuming we start from a direct approach to the path at nominal airspeed:

$$t_{\text{turn}} = \pi / \dot{\xi}_{\text{max}} \quad (5.20)$$

$$\dot{\xi}_{\text{max}} = \frac{g \tan \phi_{\text{max}}}{v_{A,\text{nom}}} \quad (5.21)$$

$$v_{G,l}(t) = v_W + v_{A,\text{nom}} \cos(\dot{\xi}_{\text{max}} t) \quad (5.22)$$

$$d_{\text{turn}} = \int_0^{t_{\text{turn}}} v_{G,l}(t) dt = v_W t + \dot{\xi}_{\text{max}}^{-1} v_{A,\text{nom}} \sin(\dot{\xi}_{\text{max}} t) \Big|_0^{t_{\text{turn}}} = \frac{v_W \pi}{\dot{\xi}_{\text{max}}} \quad (5.23)$$

The above relationship will be used in an exemplary simulation in Section 4.4.

## 4 Simulation

In this section, we present simulation comparisons of our new coupled guidance law with the controller developed in Paper II, evaluating airspeed reference efficiency and tracking performance. Last, we present an example simulation of the backwards wind triangle solution using the method described in Section 3.4.

### 4.1 Model and Environment

For a preliminary simulation study, we employ a simple two-dimensional unicycle model with first order time constants for airspeed and roll tracking, see (5.24)-(5.27).

$$\dot{\mathbf{p}} = v_A \begin{pmatrix} \cos \xi \\ \sin \xi \end{pmatrix} + \mathbf{v}_W \quad (5.24)$$

$$\dot{\xi} = \frac{g \tan \phi}{v_A} \quad (5.25)$$

$$\dot{\phi} = \frac{\phi_{\text{ref}} - \phi}{\tau_\phi} \quad (5.26)$$

$$\dot{v}_A = \frac{v_{A,\text{ref}} - v_A}{\tau_v} \quad (5.27)$$

All simulations are run in python using the *SciPy* odeint integrator for model propagation. The control loop is executed at 10 Hz. Model and controller parameters common for all simulations can be found in Table 5.1. Except where specified (or no longer existing), control parameters for both the decoupled and coupled approaches

**Table 5.1:** Common model and control parameters for simulations.

| Param              | Value | Unit              |
|--------------------|-------|-------------------|
| $\tau_\phi$        | 0.4   | s                 |
| $\tau_v$           | 1.0   | s                 |
| $v_{A,\text{nom}}$ | 10.0  | $\text{m s}^{-1}$ |
| $v_{A,\text{max}}$ | 16.0  | $\text{m s}^{-1}$ |

are identical to those used in Paper II. Where the airspeed reference logic for the decoupled controller will follow the approach described in Section 2.3, also from Paper II.

## 4.2 Airspeed Reference Efficiency Comparison

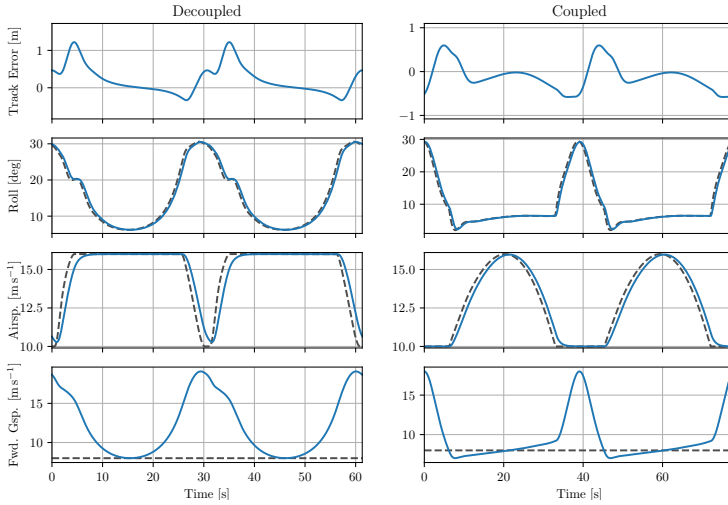
To compare the efficiency of the new reference commands to that of the decoupled approach, a loiter following simulation was run for either case. Simulation states are shown in Fig. 5.4 and the corresponding time averages of airspeed reference and track error are tabulated in Table 5.2.

**Table 5.2:** Airspeed reference efficiency comparison results for tracking of a 60 m radius loiter in  $8 \text{ m s}^{-1}$  East wind with a minimum ground speed setting of  $8 \text{ m s}^{-1}$ . See Fig. 5.4 for periodic states. Note the averages listed below are *time* averages.

|                            | Decoupled                | Coupled                  |
|----------------------------|--------------------------|--------------------------|
| Average Airspeed Reference | $15.04 \text{ m s}^{-1}$ | $12.73 \text{ m s}^{-1}$ |
| Average Track Error        | 0.21 m                   | -0.11 m                  |

Noteworthy is that the coupled approach indeed demands less airspeed (on average a 23 % reduction over the decoupled approach, normalized for the nominal airspeed) and further reduced track error, though both methods have track errors mostly below 1 m, so tuning could potentially close the small gap between them. Such a savings in airspeed demand are significant for prolonging aircraft flight endurance.

The coupled approach does, however, undershoot the prescribed minimum ground speed slightly more than the decoupled approach. This is due to the more exact reference (not over compensated) demand we are giving at all bearings, and the lag in airspeed response. The decoupled approach eliminates this undershoot by simply applying more than necessary airspeed commands throughout the loiter. Future work examining the change in airspeed reference with respect to the change in bearing could potentially provide a feed forward acceleration reference for the coupled approach which could mitigate these undershoots.



**Figure 5.4:** Airspeed reference efficiency comparison results for tracking of a 60 m radius loiter in  $8 \text{ m s}^{-1}$  East wind with a minimum ground speed setting of  $8 \text{ m s}^{-1}$ . Two converged loiter periods are shown for each simulation. References are shown in dashed grey lines, states in blue solid.

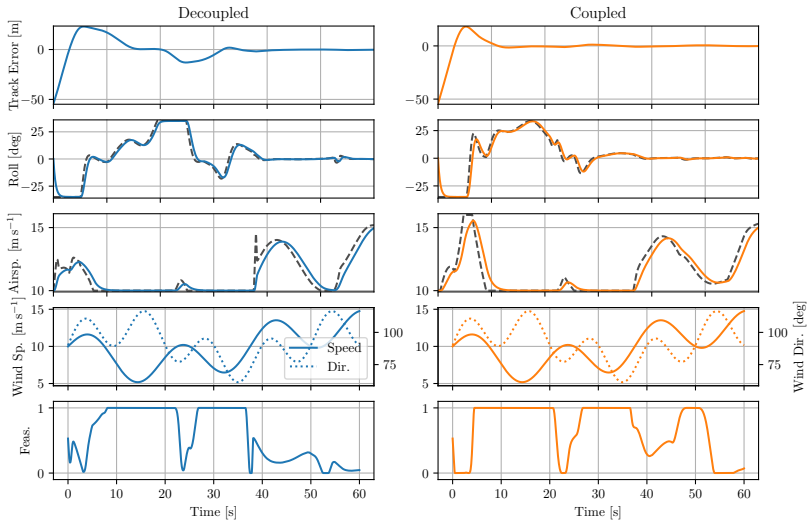
### 4.3 Track Keeping Comparison

As the simulation comparison in the last section was mostly a *static* evaluation of the reference commands, we wish to further explore the difference between coupled and decoupled approaches in a *dynamic* environment where winds, as in real life, may be quite variable, changing both direction and magnitude. We also wish to test the new minimum ground speed formulation of the track keeping objective.

An arbitrary combination of time varying sines and cosines are used to generate (quite aggressive) wind gusts acting on the aircraft. We note that the wind estimation is assumed perfect in this simulation, as we wish to evaluate the controllers in ideal conditions. Further, no force-dependent transients are modeled, meaning the aircraft is instantly in “steady state” with respect to the air mass. Investigation into force transients and wind estimation delays on the performance of these controllers should be conducted in future work.

The general minimum ground speed is set to zero, but track keeping is enabled with  $v_{G,e} = 4 \text{ m s}^{-1}$ .

Figures 5.5 and 5.6 show the results of the two control approaches. We can qualitatively see that the coupled approach maintains closer proximity to the path,



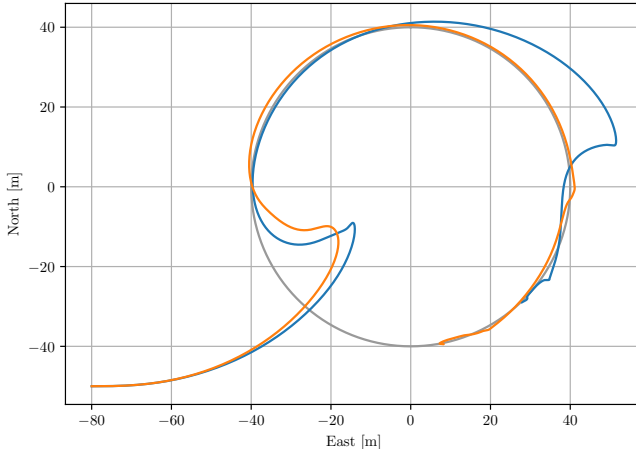
**Figure 5.5:** Track keeping in dynamic winds. Coupled and decoupled control performance.

actually commanding more airspeed than the decoupled approach within the first 10 s of the simulation. This is likely due to the feed forward behavior of the coupled controller, whereas the decoupled controller must wait until tracking errors are fed back.

#### 4.4 Backwards Solution

The simulation shown in Fig. 5.7 demonstrates the different tracking behaviors of the backwards and forward wind triangle solution. Note that both simulations shown are using the *coupled* guidance logic. As in the previous simulation, we set the general minimum desired ground speed to zero, but track keeping is enabled with  $v_{G,e} = 4 \text{ m s}^{-1}$ .

In the first 10 s of the backwards simulation we can see the aircraft adhering to the turn around condition until reaching c.a.  $-100 \text{ m East}$  (see the position plot). From here, the plane allows itself to be blown backwards into the path as we can see by the negative values in the  $\mathbf{v}_G \bullet \hat{\mathbf{v}}_A$  plot. Both forward and backwards solution are able to converge to the track, and, given we have static winds here, are able to converge to the limiting conditions outlined in the *track keeping objective*,



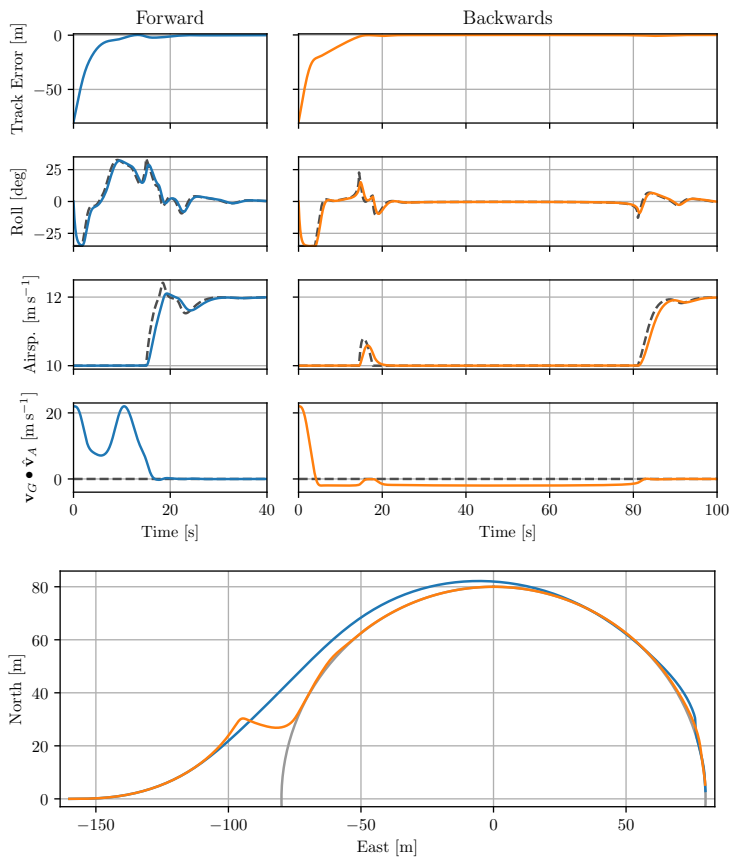
**Figure 5.6:** Position trajectories for track keeping in dynamic winds.

see 5.12, on the far (positive East) side of the loiter path. I.e., the ground speeds converge to zero, on track, with zero roll commanded, corresponding to zero lateral acceleration command.

We can see that the backwards solution, given the necessary utilization methodology, may be promising for further increasing controller efficiency, as the “slow” solution requires less instantaneous airspeed demand, or even increasing tracking performance on a given path segment since slower ground velocities require less roll control action to maintain curvature. However, quantifying the utility of this solution over the forward solution, and when it may be better or worse to employ, should be the subject of future work and surely will require some thought due to its discontinuous reference behavior at unity wind ratio (one could potentially imagine a branch of sampling based techniques for short flight segments).

## 5 Conclusions & Future Work

In this brief, we revisited the airspeed compensation logic developed in Paper II, developed a new coupled heading and airspeed reference guidance formulation, and compared the efficiency and tracking performance of the two approaches. Further, we explored a simple approach to implement the “backwards” wind triangle solution towards following a loiter path, discussing the difference in its behavior compared to the forward approach, and potential future utilization.



**Figure 5.7:** Backwards vs. forward flight simulation using the coupled control law. East wind at  $12 \text{ m s}^{-1}$ .



The coupled guidance law appears promising in preliminary simulation results, showing improved airspeed reference efficiency and tracking performance. However, more work is always necessary. Key points we wish to investigate in the future include:

- Formal stability analysis, evaluating the objectives in their limits via either a mathematical proof or appropriate Monte Carlo analysis.
- The influence of wind estimation errors on the performance of the algorithm. In particular, in dynamic winds (e.g. found in mountainous regions): What aircraft instrumentation is sufficient to quickly and reliably provide estimates? How much we should or should not filter the disturbances before providing these estimates to the controller? At what point (if any) do wind estimate errors actually degrade performance over a traditional ground speed -only feedback approach (with no formal wind estimation)?
- Extension to a coupled 3D guidance law. Consideration of the quite different longitudinal vs. lateral dynamics and constraints of aircraft should be considered in any extension, including prioritization of vertical vs. horizontal tracking preferences, when high wind components in one or the other axis may not be simultaneously mitigated.
- Of course, we also must deploy the newly developed algorithms on a real platform and go into the wild for testing!



## **Part B**

# MODEL PREDICTIVE CONTROL & LOCAL RE-PLANNING



# Nonlinear MPC for Fixed-Wing UAV Trajectory Tracking: Implementation and Flight Experiments

Thomas Stastny, Adyasha Dash, and Roland Siegwart

## Abstract

In this work we design a high-level Nonlinear Model Predictive Controller for lateral-directional fixed-wing UAV trajectory tracking in wind. Model identification of closed loop low-level roll channel dynamics is conducted towards representing a low-order equivalent of the low-level autopilot response to high-level commands. We show trajectory tracking with various horizon lengths in high winds in simulation and demonstrate track convergence to sequential Dubins Car segments in flight experiments with a small autonomous unmanned aircraft using the designed algorithm. Discussion on appropriate objective formulation and weighting is given.

## 1 Introduction

Unmanned aerial robots are becoming ubiquitous in today's society. Whether from impactful public exposure such as demonstrations of large scale three-dimensional aerial modeling projects [67], propositions of forward-looking postal delivery [84], or increasing interest in airborne support for applications including disaster relief [13, 78], crop monitoring [33], and infrastructure inspection [50], the word "drone" (for better or worse) is now within the common vernacular. Particularly interesting platforms, relative to rotor-craft, for their longer endurance and speed in mapping and sensing applications, are fixed-wing unmanned aerial vehicles (UAVs). Even more advantageous, within the fixed-wing vehicle class, are small scale and hand-launchable platforms for their ease of deployment and minimal system complexity. While a wealth of work on advanced control algorithm design, specifically towards such systems, is present in recent literature, much still remains within the confines of simulation, and needs experimental validation.

One particular avenue within this realm is work on optimization-based approaches towards trajectory tracking. In particular, Nonlinear Model Predictive Control (NMPC) algorithms offer a broad range of possible formulations and applications. For instance, one of the earliest uses of NMPC for trajectory tracking control of unmanned fixed-wing aircraft can be found in the work of Kang et al. [38] Their cost function was designed to minimize the normal distance from a UAV to a desired line segment thereby turning the tracking problem into a regulation problem with an objective to drive the error to zero. The single line tracking cost function is then extended to allow the tracking of multiple line segments with obstacle avoidance. Kang et al. follow this up with stability analysis, and verification using "hardware in the loop simulation." [39] Both works involve a kinematic model of an aircraft assuming planar motion and the existence of a low level controller to track the high level NMPC commands. Yang et al. further extend the previous two works formulating an adaptive NMPC for fixed-wing navigation through a cluttered environment, which varies the control horizon according to the path curvature profile for tight tracking. [86] By adding actuator slew limit to the optimization termination requirement in addition to the cost monotonicity, they show that the proposed optimization algorithm removes control input oscillations and tracks the trajectory more accurately than the conventional fixed horizon NMPC.

The path-following problem for fixed-wing UAV in presence of wind disturbances is addressed by Rucco et al. [73], where the objective of trajectory tracking with minimum control effort is fulfilled for a planar aircraft model using a sample-data Model Predictive Control (MPC) approach. Extension to three dimensions is seen in [22], where Gavilan et al. describe a high-level guidance algorithm based on MPC using a nonlinear 3DoF aircraft model for state prediction. The nonlinear optimization problem is then solved using an iterative scheme which uses a modified robust missile guidance law as hotstart to guarantee feasibility.

As outlined from the literature, many possibilities exist for the use of NMPC in high-level fixed-wing UAV control. However, real-time implementation has been challenging up until recently, owing to the large computational complexity and time

taken by the non-convex optimization problem posed. Efforts by researchers such as Quirynen et al. have focused on exploiting the structure of nonlinear models with linear subsystems using Implicit Runge Kutta methods, resulting in improved speeds of the solver.[71] The tools have since then been made available as a part of the ACADO toolkit [30], allowing user-friendly implementation of system dynamics and objectives in a MATLAB or C interface, and generating real-time capable code. With improved computational runtimes, an opportunity now exists to more broadly field NMPC algorithms on small fixed-wing UAV platforms.

While many of the NMPC approaches cited tend only to consider vehicle kinematics, assuming lower-level loops will sufficiently track the higher-level commands, the concept of identifying lower level dynamics has promise in enhancing the performance of the controller. However, system identification of fixed-wing UAVs in the classical sense, i.e. open-loop analysis, is often off-putting to researchers as it can be time consuming, tedious, and moreover challenging when e.g. wind tunnel facilities are not available and flight test based identification must be applied. As opposed to open-loop analysis where the system response is obtained with no feedback control, e.g. a direct aileron→roll system model and/or an aileron→roll rate system model, the objective of closed-loop system analysis is to obtain the system response to the low-level controller's command inputs (e.g. stick deflections or roll reference commands from the high-level controller) to actual roll or roll rate output.

Early examples of system identification for control-augmented fixed-wing aircraft can be found in the works of Murphy [54], and Mitchell et al. [51]. Their work concerns gaining an insight into pilot comfort levels with the augmented systems of highly maneuverable fighter aircrafts by reducing the complex high order longitudinal and lateral directional dynamics to Low Order Equivalent Systems (LOES), thereby obtaining a quantitative measurement of pilot ratings, as well as possible explanations of unexpected aircraft behavior. As an example, Murphy uses rudder pedal force and input stick deflection as inputs, with sideslip angle, stability axis, roll rate, stability axis yaw rate, bank angle, and lateral acceleration as system states to identify the closed-loop lateral dynamics. The system is treated as a Multiple Input Multiple Output (MIMO) system and parameter estimation methods in Frequency domain are used. LOES models are typically fixed a priori whose structure can be found in this detailed report[52]. For all these systems, the input delays play a significant role in flight quality performance and pilot experience.

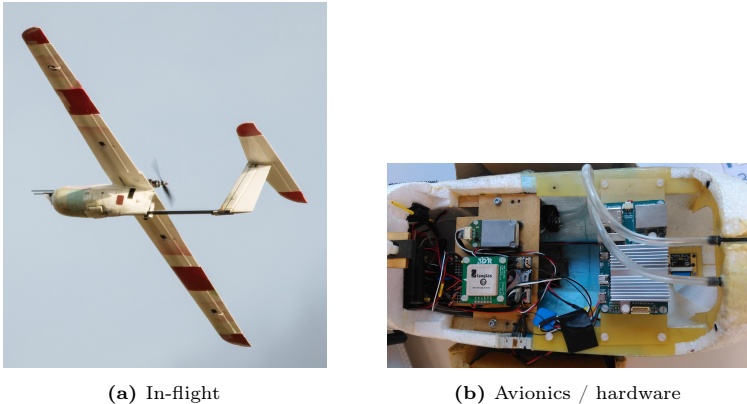
Morelli took a similar approach to identify the closed loop system for both longitudinal and lateral directions using LOES models[53]. However, unlike the previous example, SISO approach is favored here. Different types of maneuvers are executed, and it is found that certain multi-step maneuvers in combination with appropriate identification methods could be used instead of frequency sweeps, which often take longer times and are difficult to execute at high pitch angles.

In recent times, Luo et al.[47] identified the inner closed-loop system of the roll-channel of a small fixed-wing UAV as a first-order plus time delay model in order to design a fractional order  $(PI)^\lambda$  controller. A fifth order ARX model is calculated first using least squares algorithms which is then approximated to a continuous time first order plus time-delay model.

Capitalizing on new methods for auto-generation of fast NMPC code and the experience of previous authors in LOES modeling, we explore in this work the design of a high-level NMPC for real-time implementation on a small UAV, as well as simplified closed-loop identification procedures for modeling low order system equivalents of the low-level autopilot system. The identified model is incorporated into a modular high-level NMPC for general trajectory tracking in wind and verified through simulation and flight experiments.

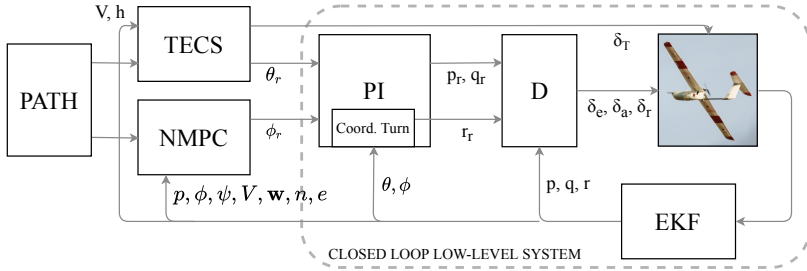
## 2 System Overview

System identification, controller design, and flight experiments within this work are performed on a small, 2.6 m wingspan, light-weight 2.65 kg, low-altitude, and hand-launchable fixed-wing UAV, Techpod, see Fig. 6.1a. The platform is a standard T-tail configuration, fixed-pitch, pusher propeller. Onboard avionics including a 10-axis ADIS16448 Inertial Measurement Unit (IMU), u-Blox LEA-6H GPS receiver, and Sensirion SDP600 flow-based differential pressure sensor feed measurements to a Pixhawk Autopilot, an open source/open hardware project started at ETH Zurich [68]. Pixhawk features a 168 MHz Cortex-M4F microcontroller with 192 kB RAM for online state estimation and low-level control. A light-weight, robust Extended Kalman Filter runs onboard fusing sensor measurements to provide state estimates including a local three-dimensional wind vector, modeled statically with slow dynamics. [42] As Techpod flies at a nominal airspeed of  $14 \text{ m s}^{-1}$ , the aircraft is easily susceptible to high winds present in low flight altitudes (below 500 m AGL), motivating explicit consideration of wind estimates within high-level position control.



**Figure 6.1:** Techpod, test platform.





**Figure 6.2:** Control architecture

As processing power on the Pixhawk microcontroller is somewhat limited, an additional onboard ODROID-U3 computer with 1.7 GHz Quad-Core processor and 2 GB RAM, running Robotic Operating System (ROS) [72] is integrated into the platform for experimentation with more computationally taxing algorithms. High-level controllers can be run within ROS node wrappers which communicate with the Pixhawk via UART serial communication; average communication latency was observed  $<3 \mu\text{s}$ , see Fig. 6.1b.

The control architecture implemented on Techpod in the current work can be seen in Fig. 6.2. The low-level autopilot, all processed onboard the Pixhawk, contains a standard cascaded PID structure with additional compensation for coordinated turns, i.e. a yaw damper signal,  $r_r = \frac{g \sin \phi}{V}$ . Attitude errors are fed to a PI block followed by rate errors running through a D block (proportional gain on rates), finally generating appropriate actuator commands  $\delta_e$ ,  $\delta_a$ ,  $\delta_r$  (elevator, aileron, and rudder deflections, respectively). The EKF feeds back appropriate signals to each respective controller resulting in a stabilized closed loop low-level system. For the purposes of this work, we consider only the typical flight regimes employed in the vast majority of UAV missions, where flight is mostly planar, and maneuvers are mostly docile. Within this regime, the described PID control architecture is reasonably fit to track attitude commands for higher level trajectory following objectives.

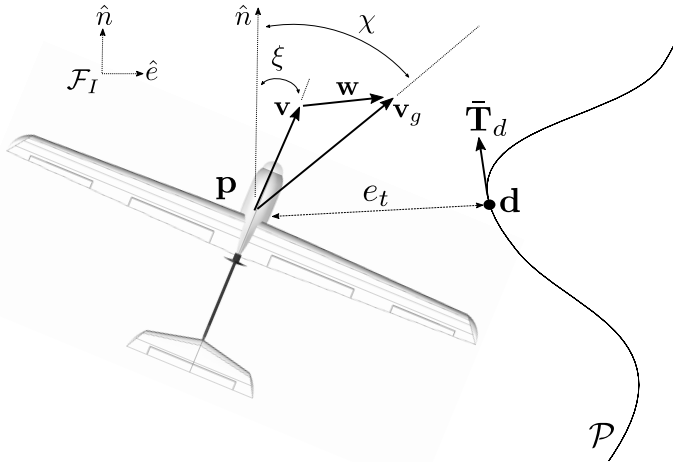
High-level altitude and airspeed control is achieved with an implementation of the Total Energy Control System (TECS), also onboard the Pixhawk Flight Management Unit (FMU). Airspeed references are mostly tracked utilizing pitch commands, and altitude holds are maintained using throttle inputs  $\delta_T$ . We assume altitude and airspeed are reliably tracked, and our focus lies with lateral-directional trajectory tracking. Detail into the last remaining blocks in the control architecture will be elaborated in Section 4.

### 3 Flight Dynamics & Identification

In this section, we consider the lateral-directional kinematics of fixed-wing aircraft, expressed under a coordinated turn assumption, as follows in the inertial frame,  $\mathcal{F}_I$ :

$$\begin{aligned}\dot{n} &= V \cos \xi + w_n \\ \dot{e} &= V \sin \xi + w_e \\ \dot{\xi} &= \frac{g \tan \mu}{V}\end{aligned}\quad (6.1)$$

where  $n$  and  $e$  are the Northing and Easting positions, respectively,  $\xi$  is the heading,  $V$  is the air-mass-relative airspeed,  $\mu$  is the bank angle,  $g$  is the acceleration of gravity, and  $w_n$  and  $w_e$  are the Northing and Easting components of the wind vector, respectively. Note that bank,  $\mu$  and heading,  $\xi$  angles are defined about the air-mass-relative airspeed vector. This distinction is important when considering flight dynamics in wind, where the ground-relative flight path of the vehicle is defined as the course angle,  $\chi$  from North,  $\hat{n}$  to the ground speed vector,  $\mathbf{v}_g$ , see Fig. 6.3.



**Figure 6.3:** Inertial Frame Definitions

Making the assumption that the low-level controller is able to adequately regulate sideslip and altitude, i.e. the airspeed vector lies on the body- $x$  axis, we may reasonably make the simplification of equating roll,  $\phi$  and bank,  $\mu$  angles, which are typically very close in value. This assumption is useful when prescribing attitude references to the low-level controller, which considers estimated body-axis,  $\mathcal{F}_B$  defined Euler angles as feedback.

The kinematics described in equation (6.1) can be further enhanced with knowledge of the underlying low-level closed-loop plant dynamics. I.e. identifying a representable transfer function,  $\frac{\phi}{\phi_r}(s)$  from the roll input,  $\phi_r$  received by the low-level autopilot, to the resulting roll angle,  $\phi$  measured in stabilized flight.

The objective of closed-loop system identification is to capture the dynamics of the aircraft with the low-level controller in the loop, which guarantees the flight stability of the UAV under various maneuvers. The low-level PID controller should be appropriately tuned before the system identification experiments, though to what degree the low-level loops perform is not necessarily important (outside of instability), as the identification should capture whatever dynamic is present for use in the high-level controller.

For identification of the roll channel, a series of multi-step inputs called 2-1-1 maneuvers were chosen. A 2-1-1 maneuver is a modified doublet input, which ideally consists of alternating pulses with pulse widths in the ratio 2-1-1. As demonstrated by Morelli[53], flight time required for the 2-1-1 maneuver is approximately one-sixth of the time required for the standard frequency sweep maneuver, thus enabling one to gather more data in the same flight time, which is often limited by battery capacity. At the same time, concatenated 2-1-1 maneuvers make for suitable identification inputs for both frequency and time domain system identification approaches, at par with frequency sweeps. For our purposes, a unit pulse width of 1s was chosen. Amplitude of these maneuvers were varied in the range allowed by the low-level controller, i.e.,  $|\phi| \in [0^\circ, 30^\circ]$ .

Ordinarily, once in stabilized mode, the low-level controller receives its pitch and roll command references from a high-level controller, such as  $\mathcal{L}_1$ -Navigation (also known as Nonlinear Guidance Law, NLGL) [75] for lateral-directional path following and TECS (Total Energy Control System) for airspeed and altitude control. For system identification experiments, the autopilot source code was modified such that, once the system identification switch on the RC transmitter was on, pitch and roll commands from high level controller are over-written by commands for system identification maneuvers. For roll channel identification, pitch reference command to the low-level controller was kept at zero, while the roll reference command was varied as per the 2-1-1 maneuvers. An example can be seen in Fig. 6.4. With appropriate settling time added before and after the maneuvers, a set of two 2-1-1 maneuvers could be performed at a stretch, which facilitates concatenation of data later for system identification. Several different data sets are collected in this manner, including some with different roll amplitudes.

For system identification analysis, simple ARX models up to fifth order were chosen. Models beyond that order were likely to overfit the data, and additionally, would add too many additional states to the high-level controller, increasing computational complexity and size. For every data-set, 20 ARX models with different numbers of pole-zero combination were created. Additionally, each of the 20 models was also evaluated with and without a set of delays, resulting in 10 variations of every ARX model. See time responses of the various models in Fig. ??.

For system identification, time domain based Instrument Variables (IV) method

is used considering the ease of use facilitated by ready to use computation tools, and guaranteed good results[47] [53].

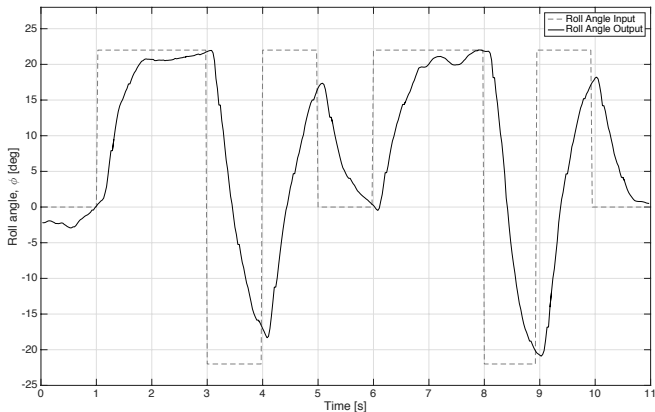


Figure 6.4: Sample system ID maneuver with respective roll angle output.

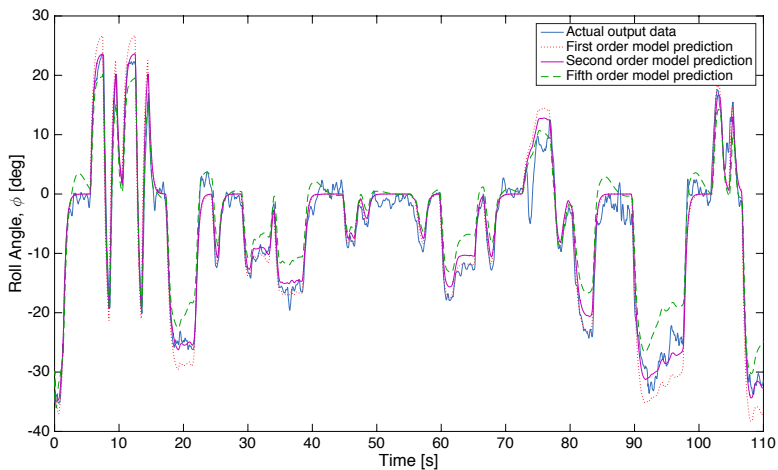
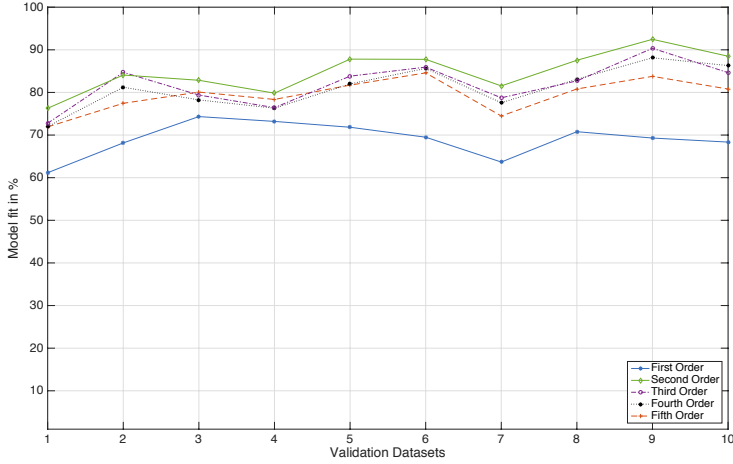


Figure 6.5: Actual roll angle output vs. time response of various models.



**Figure 6.6:** Validation fits on different data sets with different order models.

Once the individual ARX models with and without delays for a particular order model were evaluated, the best among them was chosen. This was then compared with similarly chosen ARX models of different combinations of zeros and poles. With every data-set, fifth order models typically gave best individual fits to the data sets, however second and third order models tended to generalize more and give better validation fits. Further, the addition of delay did not have any significant effect on improving the model fits to validation data. Since our NMPC is formulated in continuous time (see more information in Section 4), and modeling of a delay there is more difficult, inclusion of delay for the sake of an insignificant increase in model fit is not justified. The models of a particular order with the best fits among the different models of the same order are shown in Fig. 6.6

Considering all the factors, namely ease of augmentation to the system dynamics and the size of computational complexity thereby, and good model fits to validation data-sets, a second order model of the following form was chosen.

$$\frac{b_0}{s^2 + a_1 s + a_0} \quad (6.2)$$

From this model, the kinematics in equation (6.1) are augmented in the time domain with two differential equations describing the identified second order roll channel dynamics. Note, again, that we assume  $\mu \approx \phi$ .

$$\begin{pmatrix} \ddot{\mu} \\ \dot{\mu} \end{pmatrix} = \begin{pmatrix} \dot{\mu} \\ b_0 \mu_r - a_1 \dot{\mu} - a_0 \mu \end{pmatrix} \quad (6.3)$$

## 4 Nonlinear Model Predictive Control

In this section, we formulate a high-level lateral-directional trajectory tracking controller in a Nonlinear Model Predictive Control (NMPC) scheme. A general objective is constructed for minimizing the position error to a given track,

$$e_t = (\mathbf{d} - \mathbf{p}) \times \bar{\mathbf{T}}_d \quad (6.4)$$

where  $\bar{\mathbf{T}}_d$  is the unit tangent vector at the closest point  $\mathbf{d}$  from the UAV position  $\mathbf{p}$  to the current path, while also aligning the vehicle course with the desired trajectory direction, i.e. minimize

$$e_\chi = \chi_d - \chi \quad (6.5)$$

where  $\chi_d = \text{atan2}(\bar{T}_{d_e}, \bar{T}_{d_n}) \in [-\pi, \pi]$ . Here, we use the  $\text{atan2}$  function from the standard C math library. See also Fig. 6.3. Use of this general objective formulation allows inputting any path shape, so long as the nearest point from the UAV position can be calculated and a direction of motion along the path (i.e. the path tangent) is given for minimization throughout the horizon. In this paper, we limit our discussion *circle* and *line* following, geometry for which finding the closest point in the two-dimensional plane has a simple analytic form, the calculation of which will be omitted for brevity.

We define the state vector  $\mathbf{x} = [n, e, \mu, \xi, \dot{\mu}, x_{sw}]^T$ , and control input  $\mathbf{u} = \mu_r$ , where the augmented state  $x_{sw}$  is a switch state used within the horizon in the case that desired trajectories are piece-wise continuously, or generally discretely, defined. The switch variable has no dynamic until a switch condition is detected within the horizon, at which point an arbitrary differential is applied for the remainder of the horizon calculations. I.e.

$$\dot{x}_{sw} = \begin{cases} \alpha & \text{switch condition met} \\ 0 & \text{else} \end{cases} \quad \parallel \quad x_{sw} > \text{threshold} \quad (6.6)$$

The switch state, then, either has a value of zero (i.e. the aircraft has not met the switching condition), or some value greater than zero, at which point the internal model within the controller will switch to tracking the next path throughout the remainder of the control horizon. When a track switch has been fully achieved (i.e. with respect to the actual current aircraft position/velocity), the value of the switch state is reset to zero throughout the horizon.

A relevant example of such a case for fixed-wing vehicles is that of Dubins Car or Dubins Aircraft, in the three-dimensional case, path following, see [3]. Dubins paths can be used to describe the majority of desired flight maneuvers in a typical fixed-wing UAV mission. Further, using continuous curves such as arcs and lines allow time-invariant trajectory tracking, as oppose to desired positions in time, a useful quality when only spatial proximity to the track is desired and timing is less important; for instance, if energy conservation is required and a single low airspeed reference is given to be tracked. For the remainder of the paper, we will consider Dubins segments as path inputs to the high-level controller, though it should be

noted that the objective formulation is not limited to these.

We use the ACADO Toolkit [30] for automatic generation of fast, embedded C-code implementations of nonlinear solvers and integration methods. Though formulated in continuous time, a direct multiple shooting technique is used to solve the optimal control problem (OCP), where dynamics, control action, and inequality constraints are discretized over a time grid of a given horizon length  $N$ . A boundary value problem is solved within each interval (or shooting node) and additional continuity constraints are imposed. Sequential Quadratic Programming (SQP) is used to solve the individual Quadratic Programs using the active set method implemented in the qpOASES solver [17]. A Gauss-Newton based real-time iteration scheme is used which iteratively improves the current online solution during each step of the process during runtime [31]. Note that the ACADO framework does not explicitly support non-smooth functions. Here, we exploit an option to use externally defined C-based model and objective functions. Numerical jacobians are implemented using a finite difference. Both control and numerical stability properties for this implementation are not guaranteed; however, extensive simulation studies and flight experimentation have to-date shown no ill effects, provided control input constraints are set. A more robust investigation into possible instabilities is to be conducted in future work.

The OCP takes the following continuous form:

$$\begin{aligned}
 \min_{\mathbf{U}, \mathbf{X}} \int_{t=0}^T & \left( (\mathbf{y}(t) - \mathbf{y}_{ref}(t))^T \mathbf{Q} (\mathbf{y}(t) - \mathbf{y}_{ref}(t)) \right. \\
 & + (\mathbf{u}(t) - \mathbf{u}_{ref}(t))^T \mathbf{R} (\mathbf{u}(t) - \mathbf{u}_{ref}(t)) \left. \right) dt \\
 & + (\mathbf{y}(T) - \mathbf{y}_{ref}(T))^T \mathbf{P} (\mathbf{y}(T) - \mathbf{y}_{ref}(T)) \tag{6.7} \\
 \text{subject to } & \dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) \quad (\text{equations (6.1) \& (6.3) \& (6.6)}) \\
 & \mathbf{y} = h(\mathbf{x}, \mathbf{u}) \\
 & \mathbf{u}(t) \in \mathbb{U} \\
 & \mathbf{x}(0) = \mathbf{x}(t_0).
 \end{aligned}$$

where  $\mathbf{y} = [e_t, e_\chi, \mu, \dot{\mu}, \mu_r]^T$  and  $\mathbb{U}: \mu_{r_{min}} \leq \mu_r \leq \mu_{r_{max}}$ . Here,  $\mu_r$  is included in the objective function doubly; once within the standard control penalty (i.e.  $(\mathbf{u}(t) - \mathbf{u}_{ref}(t))^T \mathbf{R} (\mathbf{u}(t) - \mathbf{u}_{ref}(t))$ ), and again within  $\mathbf{y}$ , allowing the formulation of a slew rate  $\Delta\mu_r(t) = \mu_r(t) - \mu_{r_{k-1}}(t)$ , i.e. the deviation from the previous horizon control solution, which may be penalized by the weight component  $Q_{\Delta\mu_r}$ . The previous control horizon  $\mu_{r_{k-1}}$  is stored after the last NMPC iteration step and input as a reference value during the next. Note, this is not identical to the typical slew rate penalty often utilized in discrete MPC formulations, but actually a comparison at each shooting node to the previous control solution at that same node. The difference between each subsequent time step within the horizon in a given optimization step is not considered. This penalty is implemented to penalize

bang-bang control action caused by unregulated action within the first shooting node between each NMPC iteration. In particular, we are interested in relatively long horizons for considering optimal path convergence in various windy scenarios where the ground speed may grow faster than the chosen horizon allows adequate reaction time. Longer horizons can be achieved without overly increasing the horizon length, and thus the dimensionality of the problem, by the use of larger discretization steps, i.e. we use  $T_{step}=0.1$  s. This, however, in turn exacerbates the mentioned issue of bang-bang control action, as the next NMPC iteration step measurement can possibly deviate enough to induce a large control step. An example of the issue can be seen in Section 5. As the next action in the control horizon is applied to the vehicle at each iteration, the deviation in the first few shooting nodes should be penalized, but not the latter steps, as this would result in an overall sluggish control performance. To remedy this discrepancy, a decreasing quadratic function is defined in the control deviation weight horizon, so that latter nodes are not penalized, and earlier nodes are.

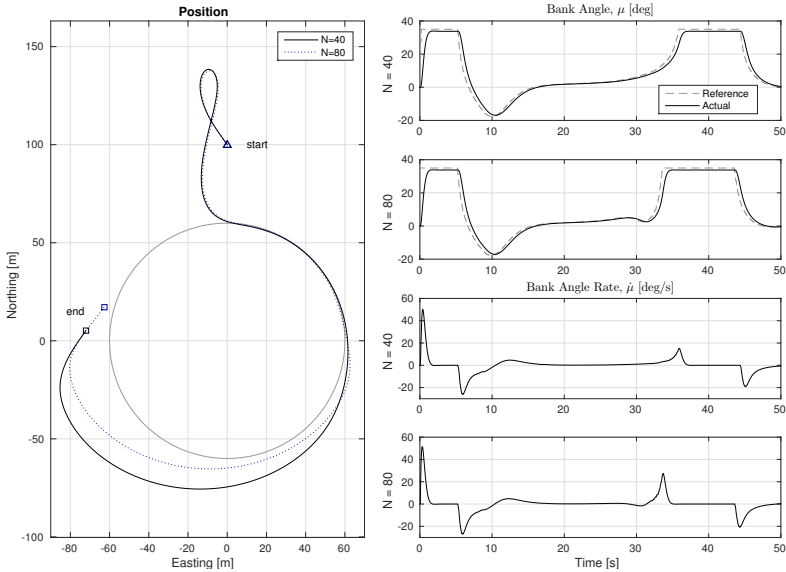
Other online parameters augmented to the model and held constant through the horizon are the current airspeed  $V$ , current wind vector  $\mathbf{w}$ , and the current and next sets of Dubins path parameters  $\mathcal{P}_{cur}$ ,  $\mathcal{P}_{next}$ , where line parameters include  $\mathcal{P} = \{type = 0, \mathbf{a}, \mathbf{b}\}$ ,  $\mathbf{a}$  and  $\mathbf{b}$  are two waypoints defining a straight segment, and arc parameters include  $\mathcal{P} = \{type = 1, \mathbf{c}, R, dir, \xi_0, \Delta\xi\}$ ,  $\mathbf{c}$  is the center point of the arc,  $R$  is the radius,  $dir$  is the loiter direction, and  $\xi_0$  is the heading pointing towards the entrance point on the arc, and  $\Delta\xi$  is the arclength traveled. The path segments are managed and rotated based on an acceptance radius and heading direction criteria.

## 5 Simulations & Flight Experiments

In this section, we present simulation results as well as real world flight experiments with the Techpod test platform using the designed NMPC with the identified low-level model.

Before testing on the platform, extensive two-dimensional simulation of the kinematics with augmented second order bank dynamics was carried out to obtain a rough tuning of objective weights as well as experiment with various magnitudes of wind. A sample of the simulation findings is shown. Displayed simulation results were obtained using objective weights  $\mathbf{Q}_{diag} = [0.01, 1, 0.1, 0.01, 100]$ ,  $\mathbf{R}_{diag} = 10$  and end term objective weights  $\mathbf{P}_{diag} = [0.1, 10, 0, 0.01, 0]$ . The discretization time step within the horizon is  $T_{step}=0.1$  s, and the NMPC is iterated every 0.05 s. In Fig. 6.7, two simulations, one with a control horizon length of  $N = 40$ , set as a minimum to ensure capturing an entire  $90^\circ$  turn at maximum bank, and the other with  $N = 80$  are initialized at the same position and orientation and commanded to track a circle in high wind. The vehicles in the simulation were set to an airspeed of  $14 \text{ m s}^{-1}$ . Further, the desired loiter radius is smaller than the minimum trackable radius at the given flight speed plus added wind component. Thus, as shown, the high winds cause a large deviation from the track when the UAVs fly down wind.





**Figure 6.7:**  $N = 40$  and  $N = 80$  length horizons for circle tracking in  $w_e = -10 \text{ m s}^{-1}$ .

The UAV with the longer horizon is able to foresee the future deviation, and plans an adverse control action earlier in the loiter to enable less deviation over the remainder of the horizon. Despite the windy conditions, both horizons, however, are able to converge initially to the circle when feasible. Further, bank angle rates are within acceptable limits, and control input constraints are respected.

Fig. 6.8 demonstrates the possible bang-bang effect when the deviation of the control horizon from the previous solution is not penalized. The displayed control solutions are taken from the same simulation shown in Fig. 6.7.

After testing the controller in simulation. Two flight experiments were conducted to demonstrate various trajectory following performance. A horizon length of  $N = 40$  was used with objective weights and end term objective weights set to  $\mathbf{Q}_{diag} = \mathbf{P}_{diag} = [0.01, 10, 0.1, 0.01, 100]$ ,  $\mathbf{R} = 10$ . The discretization time step within the horizon is  $T_{step} = 0.1 \text{ s}$ , and the NMPC is iterated every  $0.05 \text{ s}$ . Solve times for the NMPC running on the ODR0ID-U3 vary, depending on the type of path  $\mathcal{P}_{cur}$  the aircraft is following, as well as the next path  $\mathcal{P}_{next}$  in the trajectory, if there are switches within the given horizon length. Table 6.1 shows the mean,

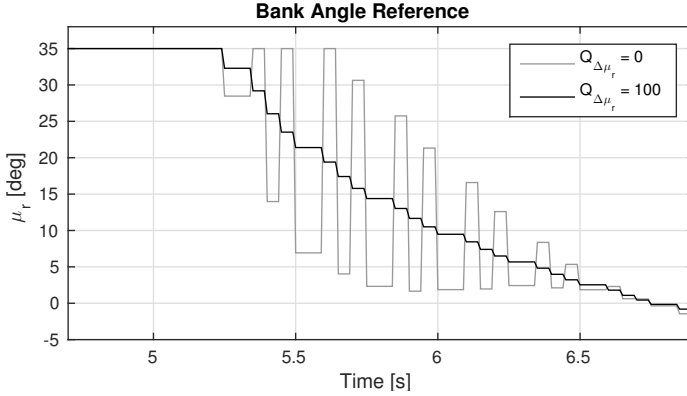


Figure 6.8: Effect of horizon slew weight on control action.

Table 6.1: NMPC computational load on ODROID-U3.

|                  | Mean [ms] | Standard Dev. [ms] | Maximum [ms] |
|------------------|-----------|--------------------|--------------|
| Line following   | 9.96      | 0.250              | 10.5         |
| Circle following | 13.5      | 0.439              | 15.1         |

standard deviation, and maximum computation times for the controller running during the Dubins path following experiment (see Figure 6.11). Note these times include the solver as well as required message handling, path management, and data conversion executed within each iteration of the ROS node. Computation times are taken from portions of flight data wherein the aircraft is only tracking a line or circle, respectively, throughout the control horizon. Switching within the horizon results in increases or decreases in computations depending on which path is being tracked and for what portion of the horizon (e.g. circle tracking times will decrease as a switch to a line path is observed within the control horizon). Both experiments took place during very calm conditions, and the wind speed was negligible.

In Fig. 6.9, Techpod is commanded towards a box pattern until returning to a loiter circle. Minimal overshoot is observed, considering the set acceptance radius of 35 m, and convergence within less than 1 m of position error is observed for each line segment and the final loiter circle. Figure 6.10 shows the commanded and actual roll angles as well as the roll rate, which are both kept within acceptable bounds.

In Fig. 6.11, an arbitrary sequence of Dubins segments were given to the high-level NMPC. Again, good convergence to the path is seen, with acceptable state responses shown in Fig. 6.12. Steady-state position error remained within 1 m

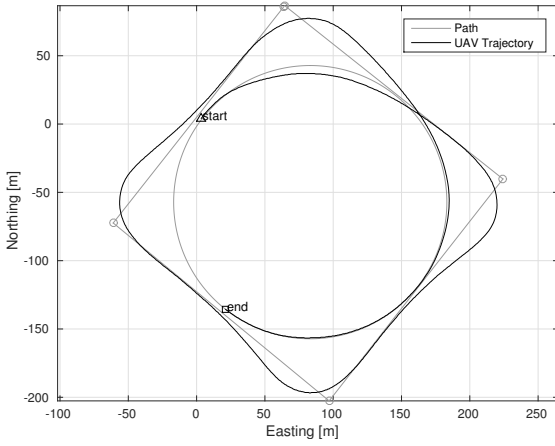


Figure 6.9: Flight experiment: box tracking.

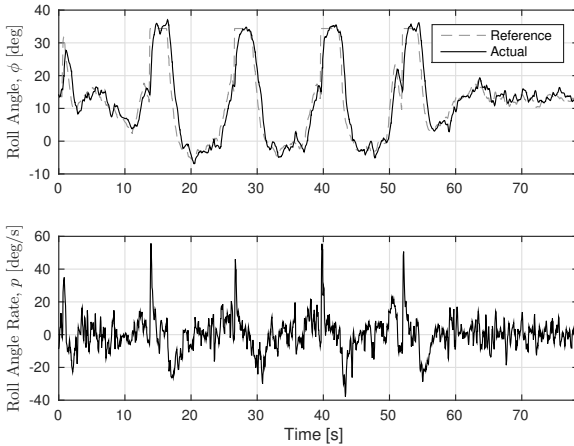
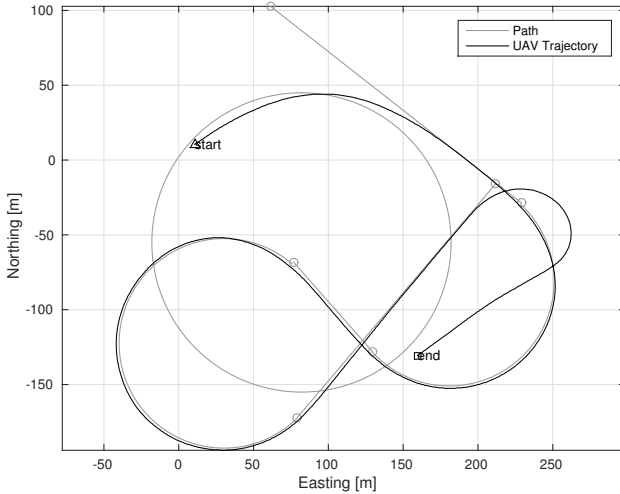


Figure 6.10: Flight experiment: box tracking attitude and rates.

after convergence to the path. Note this is without the inclusion of integral action, and either model uncertainties, or variable wind conditions perhaps not properly estimated, or gusts could cause larger track errors. The end of the shown flight path is stopped just before converging to the final loiter due to rain fall starting during the flight experiment and manual take-over of the aircraft for landing.

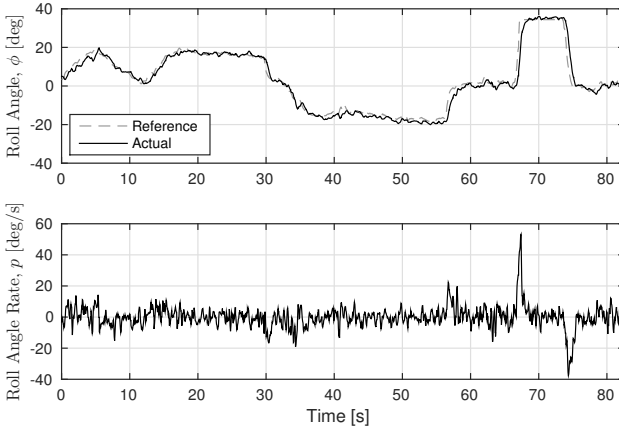


**Figure 6.11:** Flight experiment: Dubins tracking.

It should be noted that each flight experiment shown here was also flown separately with  $\mathcal{L}_1$  guidance for roll command generation, achieving similar performance in these non-windy conditions. Our focus in this work, however, is not on comparing methods, but verifying the feasibility of the given NMPC formulation in real flights. In higher wind scenarios such as those shown in simulation, similar control performance with  $\mathcal{L}_1$  loops would require some form of wind-dependent gain scheduling. Similar scheduling would also likely be required for vector-field based approaches, notorious for being somewhat difficult to tune [3, 80, 81].

## 6 Conclusions & Future Work

In this paper, we outlined an approach for low order equivalent system modeling and identification of control-augmented low-level roll channel dynamics for a small fixed-



**Figure 6.12:** Flight experiment: Dubins tracking attitude and rates.

wing UAV and, further, demonstrated the importance of their inclusion within the model of a high-level Nonlinear Model Predictive Controller. The control-augmented model identification process was observed to significantly decrease identifying flight time, as well as simplify the resulting model structure, when compared with open-loop, low-level aerodynamic identification. Open-loop simulation of the identified dynamics also demonstrated predictable behavior, even within long horizons, due to the stabilized dynamics; a useful trait for high-level controllers.

An NMPC was designed for Dubins car path following in the two-dimensional lateral-directional plane, and was shown capable of good tracking performance, even in high wind conditions (through simulation experiments), and arbitrary path combinations, shown in flight experiments. Design of the objective function was elaborated towards avoiding bang-bang control action, and including track switching behavior within the control horizon. Computation times onboard the ODROID-U3 were observed to be well within feasible limits for online solutions of adequate high-level command generation (or guidance). Horizon lengths up to eight seconds were investigated in simulation, and four seconds within flight experiments, showing utility in the determination of optimal flight path convergence to a given track in strong wind.

Though promising results, alternative objective functions could easily be designed and tested as off-shoots from the given basic formulation, e.g. inclusion of obstacle avoidance, or more complex paths. The true benefit of such an online optimization-based approach to control is the modularity of the cost function, as well as the inherent online adaptability to changing environmental conditions. It is this point

where more simple analytic approaches, e.g.  $\mathcal{L}_1$  navigation or vector field based approaches, often require gain scheduling per condition, a process which takes time to properly tune.

Further future work will focus on extending the approach to three dimensions, with longitudinal dynamics also included into the formulation. Implications of a three-dimensional controller on low-level model identification would likely necessitate MIMO Linear Time Invariant (LTI) plants, or possibly nonlinear model structures, as longitudinal aircraft dynamics, even when stabilized, typically vary with airspeed and angle of attack. Last, future work should include stability analysis, both in the control algorithm, as well as the numerical methods used to solve the sometimes non-smooth optimization problem posed.

# Nonlinear Model Predictive Guidance for Fixed-Wing UAVs Using Identified Control Augmented Dynamics

Thomas Stastny and Roland Siegwart

## Abstract

As off-the-shelf (OTS) autopilots become more widely available and user-friendly and the drone market expands, safer, more efficient, and more complex motion planning and control will become necessary for fixed-wing aerial robotic platforms. Considering typical low-level attitude stabilization available on OTS flight controllers, this paper first develops an approach for modeling and identification of the control augmented dynamics for a small fixed-wing Unmanned Aerial Vehicle (UAV). A high-level Nonlinear Model Predictive Controller (NMPC) is subsequently formulated for simultaneous airspeed stabilization, path following, and soft constraint handling, using the identified model for horizon propagation. The approach is explored in several exemplary flight experiments including path following of helix and connected Dubins Aircraft segments in high winds as well as a motor failure scenario. The cost function, insights on its weighting, and additional soft constraints used throughout the experimentation are discussed.

## 1 Introduction

As off-the-shelf (OTS) autopilots become more widely available and user-friendly and the drone market expands, safer, more efficient, and more complex motion planning and control will become necessary for aerial robotic platforms. Tools for auto-code-generation of fast, efficient embedded nonlinear solvers, e.g. ACADO Toolkit [30] or FORCES<sup>1</sup>, are becoming popular for the high-level control design of such systems. Exemplary applications of these tools, using Nonlinear Model Predictive Control (NMPC), have been experimentally shown on multi-copters for various high-level tasks such as trajectory tracking [37], inter-vehicle collision avoidance [36], and aerial manipulation [46]. The NMPC formulation conveniently offers the capability to solve receding horizon optimal control problems with consideration of nonlinear dynamics and handling of state/input constraints, a valuable set of functionalities for flying platforms aiming to satisfy the ever-increasing complexity of desired autonomous behaviors.

For large-scale sensing and mapping applications, small fixed-wing unmanned aerial vehicles (UAVs) provide advantages of longer range and higher speeds than rotorcraft. However, unlike their multi-copter counterparts, experimental implementation and validation of NMPC approaches on fixed-wing platforms is almost non-existent. To examine the state-of-the-art in fixed-wing specific NMPC formulation, one must consider simulation studies within the literature. High-level guidance formulations, using two-degrees-of-freedom (2DOF) kinematic models for horizon propagation, have been shown for the 2D path following case [38, 39, 86], and with 3DOF kinematic models for 3D soaring [45] or automatic landing [34]. Other works have considered lower-level formulations, either incorporating all objectives from obstacle avoidance to actuator penalty directly [24], focusing on low-level states only, e.g. for deep-stall landing [48], or augmenting the internal low-level model with guidance logic [80, 81].

Higher-level formulations typically utilize simple parameterless kinematic models, assuming that lower-level controllers adequately track high-level commands. These approaches rarely consider details of integration with increasingly ubiquitous OTS autopilots and their low-level control structures. On the other hand, lower-level formulations, if implemented on real aircraft, require extensive wind tunnel testing and/or flight experimentation for actuator-level aerodynamic system identification (ID), a time consuming and potentially safety-critical process. In our previous work [82], we first explored the concept of encapsulating the closed-loop autopilot roll channel response dynamic into the internal model of a high-level lateral-directional NMPC, taking a ‘middle road’ between full classical ID and model-free formulations. Broader application of this approach to 3D problems requires extending the control augmented modeling to a full, coupled lateral-directional and longitudinal structure; an extension we provide in the present work.

In this paper, we will first develop an approach to modeling and identification of *control augmented dynamics* for a conventional fixed-wing platform with a widely

---

<sup>1</sup><https://www.embotech.com/FORCES-Pro>





**Figure 7.1:** Techpod, fixed-wing unmanned aerial test platform.

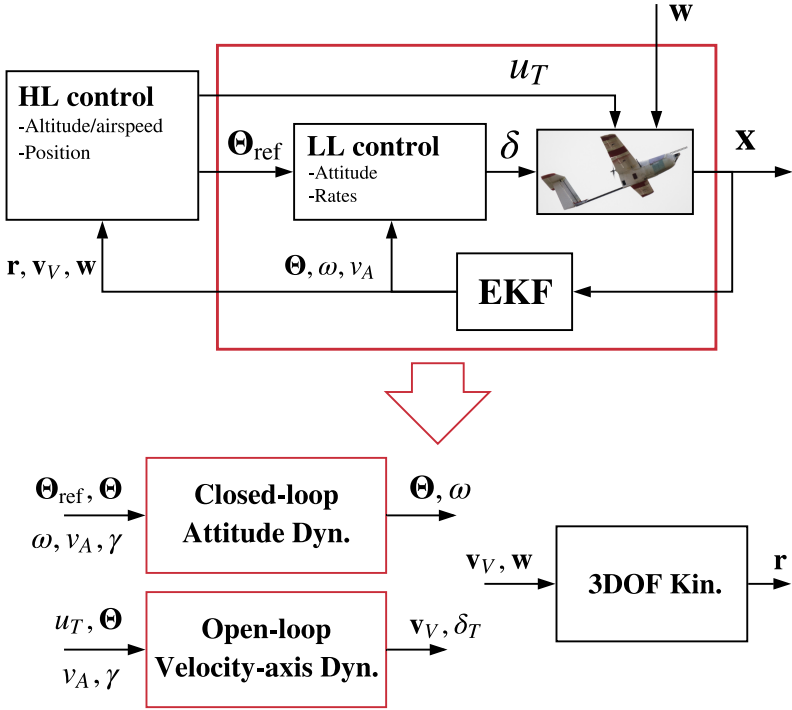
available OTS autopilot in the loop, utilizing a standard sensor suite. We will secondly detail a high-level NMPC cost function design for simultaneous airspeed stabilization, path following, and soft constraint handling, utilizing the identified model internally. We take special consideration of practical implementation insights throughout this work, such as explicit consideration of high winds as well as on-board computational constraints, and conclude with a set of representative flight experiments for validation of the approach.

## 2 Control Augmented Modeling

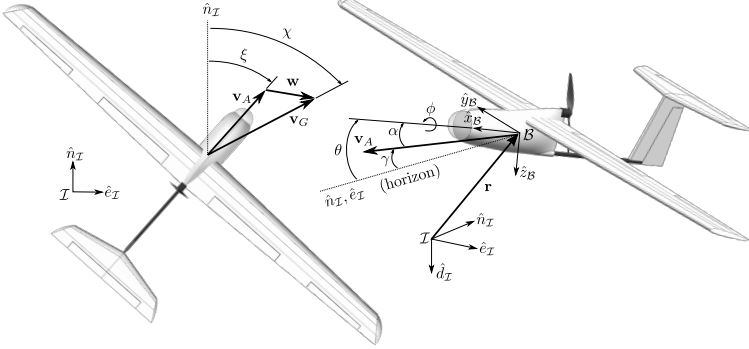
A typical fixed-wing system/control architecture is shown in Fig. 7.2, e.g. similarly implemented in open-source autopilot firmwares *PX4*<sup>2</sup> and *ArduPilot*<sup>3</sup>. A low-level (LL) control structure runs on the pixhawk microcontroller consisting of a cascaded PD attitude control/rate damping approach with coordinated turn feed-forward terms, using both rudder and elevator compensation, as well as dynamic pressure scaling on control actuation, see [60] for more details. High-level (HL) lateral guidance, e.g. within *PX4* firmware  $\mathcal{L}_1$ -guidance [75], steers the aircraft position and velocity toward waypoints or paths by commanding roll angle references  $\phi_{\text{ref}}$ , and airspeed/altitude are controlled, e.g. via Total Energy Control System (TECS) [6], by commanding pitch angle references  $\theta_{\text{ref}}$  and throttle input  $u_T$ . To replace this high-level module with a unified model predictive controller, one must characterize the underlying dynamics.

<sup>2</sup><http://pixhawk.org>

<sup>3</sup>[ardupilot.org](http://ardupilot.org)



**Figure 7.2:** Model abstraction of the closed-loop attitude dynamics, open-loop velocity-axis dynamics, and 3DOF kinematics.



**Figure 7.3:** Inertial  $\mathcal{I}$  and body  $\mathcal{B}$  axes and state definitions.

Here, we propose a “cascaded” modeling approach, defining two low-level model structures as grey-box models: 1) the stabilized, closed-loop attitude dynamics, (7.1), and 2) the open-loop velocity-axis dynamics, (7.2). Their outputs are fed to the standard (parameterless) 3DOF kinematic equations (7.4). State/axes definitions may be seen in Fig. 7.3.

## 2.1 Closed-loop attitude dynamics

We model the input-output relationship of the closed-loop LL attitude controlled system in (7.1); specifically, how the attitude and body rates respond to attitude references. The structure contains coupled lateral-directional and longitudinal states and parameters as well as nonlinearities, particularly owing to the longitudinal effects. LL controllers are often tuned for one, or very few, trim conditions around the standard flight operation point, allowing the control augmented behavior throughout the flight envelope to vary, further motivating included nonlinear airspeed dependence.

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} = \begin{pmatrix} p \\ q \cos \phi - r \sin \phi \\ l_p p + l_r r + l_{e_\phi} (\phi_{\text{ref}} - \phi) \\ v_A^2 (m_0 + m_\alpha \alpha + m_q q + m_{e_\theta} (\theta_{\text{ref}} - \theta)) \\ n_r r + n_\phi \phi + n_{\phi_{\text{ref}}} \phi_{\text{ref}} \end{pmatrix} \quad (7.1)$$

where  $\Theta = [\phi, \theta]$  is the aircraft attitude (roll and pitch, respectively),  $\omega = [p, q, r]^T$  are the body roll, pitch, and yaw rates, respectively,  $v_A$  is the airspeed (air-mass

relative), and  $\varphi_{\text{CL}} = [l_p, l_r, l_{e_\phi}, m_0, m_\alpha, m_q, m_{e_\theta}, n_r, n_\phi, n_{\phi_{\text{ref}}}]^T$  is the set of parameters to identify.

## 2.2 Open-loop dynamics

As airspeed is controlled on a high-level basis within the given autopilot structure, there exists a non-stabilized (open-loop) dynamic from throttle input to UAV outputs we must model, see (7.2).

$$\begin{pmatrix} \dot{v}_A \\ \dot{\gamma} \\ \dot{\xi} \\ \dot{\delta}_T \end{pmatrix} = \begin{pmatrix} \frac{1}{m} (T \cos \alpha - D) - g \sin \gamma \\ \frac{1}{mv_A} [(T \sin \alpha + L) \cos \phi - mg \cos \gamma] \\ \frac{\sin \phi}{mv_A \cos \gamma} (T \sin \alpha + L) \\ (u_T - \delta_T) / \tau_T \end{pmatrix} \quad (7.2)$$

where  $m$  is the mass,  $g$  is the acceleration of gravity,  $\gamma$  is the air-mass relative flight path angle,  $\alpha$  is the aircraft angle of attack (AoA),  $\delta_T$  is a virtual throttle *state* lagged from input  $u_T \in [0, 1]$  by time constant  $\tau_T$ , and  $\xi$  is the aircraft heading, defined from North to the airspeed vector.

This 3DOF model is often used as a simplified dynamic formulation in aerospace controls literature, containing only forces, the assumption being that moments are controlled on a lower-level and the overall behavior of the high-level states may be described in a quasi-steady manner. Further, the model only considers forces in the longitudinal axis, making the assumption that no aerodynamic or thrusting side force is generated, in part due to an assumption that the low-level controller appropriately regulates sideslip. By neglecting sideslip, which is challenging to observe without a vector airdata probe or alpha-beta vane, we are also able to make the approximate relationship  $\alpha \approx \theta - \gamma$  (similarly necessary as  $\alpha$  is not directly measured) and that heading angle  $\xi$  is assumed equivalent to aircraft yaw angle. We, however, retain the velocity-axis convention,  $\mathbf{v}_V = [v_A, \gamma, \xi]^T$ , for aircraft heading  $\xi$  as a means to distinguish between lower- and higher-level modeling descriptions. The force equations are shown in (7.3).

$$\begin{aligned} T &= (c_{T_1} \delta_T + c_{T_2} \delta_T^2 + c_{T_3} \delta_T^3) / v_{\infty \text{prop}} \\ D &= \bar{q} S (c_{D_0} + c_{D_\alpha} \alpha + c_{D_{\alpha^2}} \alpha^2) \\ L &= \bar{q} S (c_{L_0} + c_{L_\alpha} \alpha + c_{L_{\alpha^2}} \alpha^2) \end{aligned} \quad (7.3)$$

where motor thrust  $T$  is modeled as power  $\mathcal{P}$ , a function of throttle input, over the effective propeller free stream, approximated as  $v_{\infty \text{prop}} \approx v_A \cos \alpha$ . Lift  $L$  and drag  $D$  forces are scaled with dynamic pressure  $\bar{q}$  and wing surface area  $S$ . The elaborated model structure contains grey parameters  $\varphi_{\text{OL}} = [c_{T_1}, c_{T_2}, c_{T_3}, \tau_T, c_{D_0}, c_{D_\alpha}, c_{D_{\alpha^2}}, c_{L_0}, c_{L_\alpha}, c_{L_{\alpha^2}}]^T$  to be identified.

### 2.3 3DOF kinematics

Finally, parameterless 3DOF kinematics propagate the position through time in wind:

$$\begin{pmatrix} \dot{n} \\ \dot{e} \\ \dot{d} \end{pmatrix} = \begin{pmatrix} v_A \cos \gamma \cos \xi + w_n \\ v_A \cos \gamma \sin \xi + w_e \\ -v_A \sin \gamma + w_d \end{pmatrix} \quad (7.4)$$

where  $\mathbf{r} = [n, e, d]^T$  are the inertial frame Northing, Easting, and Down position components, respectively, and  $\mathbf{w} = [w_n, w_e, w_d]^T$  are the inertial frame wind components, modeled as static disturbances.

## 3 System Identification

### 3.1 System overview

All development and experimentation within this work is conducted on the 2.6 m wingspan, 2.65 kg, hand-launchable fixed-wing UAV – Techpod, see Fig. 7.1. The platform is a standard T-tail configuration, fixed-pitch, pusher propeller integrated with a 10-axis ADIS16448 Inertial Measurement Unit (IMU), u-Blox LEA-6H GPS receiver, and Sensirion SDP600 flow-based differential pressure sensor coupled with a one-dimensional pitot-static tube configuration. Sensor measurements are fused in a light-weight, robust Extended Kalman Filter (EKF) [42] running on board a *Pixhawk* Autopilot (168 MHz Cortex-M4F microcontroller with 192 kB RAM) generating state estimates including a local three-dimensional wind vector, modeled statically with slow dynamics.

### 3.2 Data collection and organization

Data from five approx. 40 min flight tests was collected containing 72 experiment sets (with 1 or 2 identification maneuvers each) spanning a range of 28 *static*, 35 *dynamic*, and 9 *free-form* preprogrammed maneuvers, covering the operational flight envelope (i.e.  $v_A \in [11, 18] \text{m s}^{-1}$ ,  $\phi \in [-30, 30]^\circ$ , and  $\theta \in [-15, 15]^\circ$ ), all with active attitude stabilization. A 70-30 percent ratio was used for training and validation groups on the static and dynamic sets (together, 87.5% of the total number of sets), while the free-form sets were all held back for a “testing” group (the remaining 12.5%).

- *Static* experiment sets refer to fixed airspeed  $v_A$ , throttle input  $u_T$ , and flight path angle  $\gamma$  with no dynamic maneuvering (i.e. constant  $\phi_{\text{ref}}$  and  $\theta_{\text{ref}}$ ).
- *Dynamic* experiment sets were conducted at various flight speeds and flight path angles utilizing 2-1-1 step inputs (see [53]) for all  $u_T$ ,  $\phi_{\text{ref}}$ , and  $\theta_{\text{ref}}$  to excite the low-level autopilot response dynamics.

- *Free-form* experiment sets refer to manually commanded attitude references and throttle inputs to the stabilized system in an arbitrary fashion.

Note that inputs were applied in both independent and coupled combinations (see Figures 7.5 and 7.6 for an example of a coupled identification maneuver). All static and dynamic maneuvers were initialized at trim reference commands for a settling period before enacting the steps. For repeatable experiments, commands were generated in a mostly automated fashion on-board the pixhawk. A data logging rate of 40 Hz was found sufficient to observe the stabilized dynamic responses within the maneuvers. Care was taken to fly on windless days, and on-board estimates from the EKF are used within the parameter estimation process without any post processing.

### 3.3 Time-domain nonlinear grey-box identification

The MATLAB System Identification Toolbox (*ver. R2016b*) was used for nonlinear grey-box estimation. The closed-loop  $\varphi_{\text{CL}}$  and open-loop  $\varphi_{\text{OL}}$  model parameters were identified in a decoupled manner, focusing the parameters to their respective dynamics and outputs in an attempt to avoid any erroneous cost minimization in the optimization across model structures. Further, decoupling the identifications allows any future change in low-level attitude control parameters only to require adapting the closed-loop attitude response model, while the quasi-steady open-loop model should not change with respect to slightly varying attitude stabilization.

The grey-box structure for the closed-loop attitude dynamics contains states  $\mathbf{x}_{\text{CL}} = [\Theta^T, \omega^T]^T$ , inputs  $\mathbf{u}_{\text{CL}} = [\Theta_{\text{ref}}^T, v_A, \gamma]^T$ , and outputs for error minimization  $\mathbf{y}_{\text{CL}} = [\Theta^T, \omega^T]^T$ , and dynamic equations (7.1). Note the airspeed and flight path angles are input from the logged data, and not propagated within the model structure.

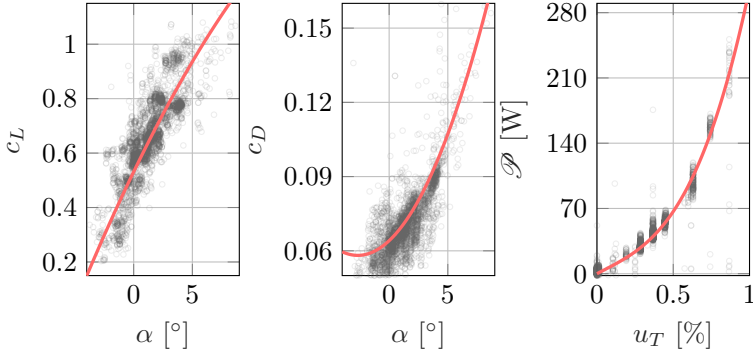
The grey-box structure for the open-loop dynamics contains states  $\mathbf{x}_{\text{OL}} = [v_A, \gamma, \delta_T]^T$ , dynamic equations (7.2), inputs  $\mathbf{u}_{\text{OL}} = [\Theta^T, u_T]^T$ , and outputs  $\mathbf{y}_{\text{OL}} = [v_A, \gamma, a_x, a_z]^T$ , where  $a_x$  and  $a_z$  are the  $x$ -body and  $z$ -body axis accelerations, related to the internal model states as:

$$\begin{pmatrix} a_x \\ a_z \end{pmatrix} = \begin{pmatrix} \cos \alpha & \sin \alpha \\ \sin \alpha & -\cos \alpha \end{pmatrix} \begin{pmatrix} (T \cos \alpha - D) / m \\ (T \sin \alpha + L) / m \end{pmatrix} \quad (7.5)$$

The minimization of body acceleration errors during parameter estimation proved especially useful. Prior to the optimization process itself, the same acceleration measurements could be used to fit an initial guess of the lift and drag curves. Such a plot can be seen in Fig. 7.4.

### 3.4 Model validation

After optimizing the model parameter estimates of the two model structures, the models were validated on data not used within the training. Figures 7.5 and 7.6



**Figure 7.4:** Static aerodynamic and power curves: lift coefficient (left), drag coefficient (center), power (right). Acceleration data with corresponding body rates below  $1^\circ \text{s}^{-1}$  are displayed. Note this “sanity check” is important during the identification and model selection process, as the output-error method can easily misrepresent the underlying physics, despite obtaining a low-cost fit.

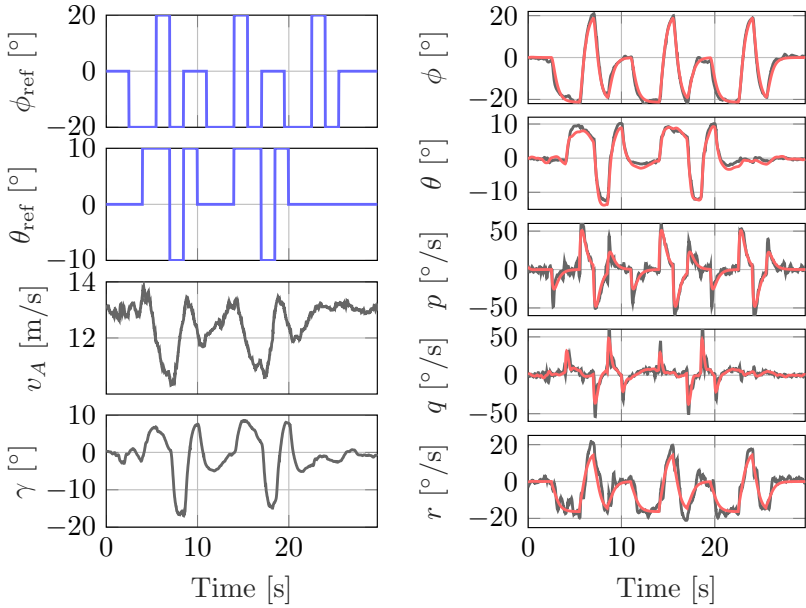
show a representative validation of a coupled pitching and rolling maneuver for the closed- and open-loop dynamics, respectively. All outputs are well matched to the flight data. Table 7.1 displays the average Root Mean Squared Error (RMSE) for each output signal over all validation sets.

**Table 7.1:** Average Root Mean Squared Error (RMSE) over all validation sets.

| YCL      |       |                        | YOL      |       |                   |
|----------|-------|------------------------|----------|-------|-------------------|
| Signal   | RMSE  | Unit                   | Signal   | RMSE  | Unit              |
| $\phi$   | 1.610 | $^\circ$               | $v_A$    | 0.424 | $\text{m s}^{-1}$ |
| $\theta$ | 0.921 | $^\circ$               | $\gamma$ | 1.680 | $^\circ$          |
| $p$      | 5.140 | $^\circ \text{s}^{-1}$ | $a_x$    | 0.217 | $\text{m s}^{-2}$ |
| $q$      | 3.390 | $^\circ \text{s}^{-1}$ | $a_z$    | 0.660 | $\text{m s}^{-2}$ |
| $r$      | 2.650 | $^\circ \text{s}^{-1}$ |          |       |                   |

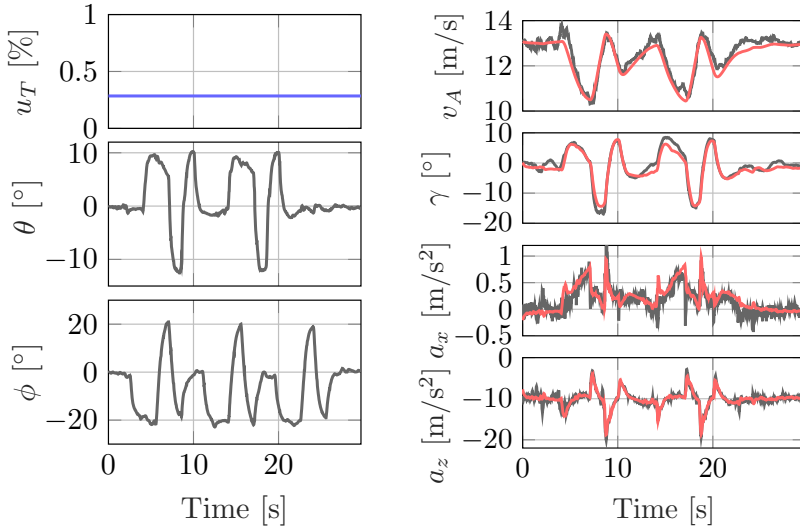
As both identified models are to be propagated simultaneously within the horizon of the MPC, a subset of *free-form* flight data was used to *test* the fully integrated model in open-loop simulation. Figures 7.7 and 7.8 show a comparison of one such simulation against over 1 min of flight data. Despite not being trained or validated with the combined model, the results show good tracking - validating the decoupled (open-loop vs. closed-loop) modeling assumptions made within the identification. Notably, the largest errors within the experiment were seen during

extended maximum roll angle commands while simultaneously flying at airspeeds exceeding the identified state-space – suggesting, in particular, that the model structure for  $q$  and  $r$  dynamics may begin to break down near the boundary of the identified flight envelope. For more aggressive flight with higher roll angles or airspeeds, these unmodeled effects would need further consideration.

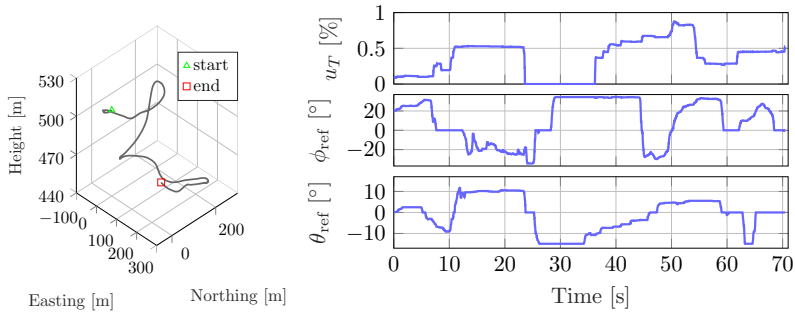


**Figure 7.5:** Inputs (left) and outputs (right) for a coupled validation experiment on the closed-loop attitude dynamics. (blue is the input signal, red is the simulated output)

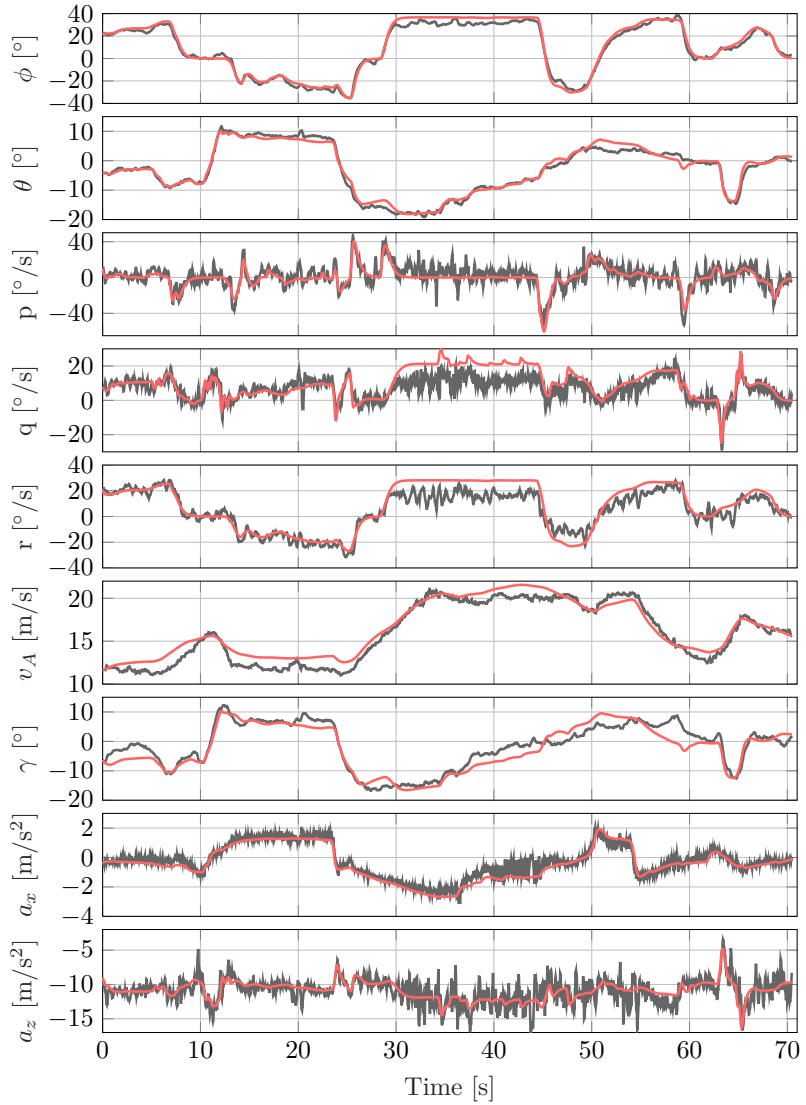




**Figure 7.6:** Inputs (left) and outputs (right) for a coupled validation experiment on the open-loop dynamics. (blue is the input signal, red is the simulated output)



**Figure 7.7:** Flight path of the Techpod UAV (left) and corresponding full-model inputs (right) during a *free-form* validation experiment.



**Figure 7.8:** Free-form open-loop simulation of the combined, identified model (red) compared to flight data.

## 4 Control Formulation

In this section we formulate a high-level Nonlinear Model Predictive Controller (NMPC) for multi-objective guidance of the UAV, embedding the path following problem for optimization within the horizon, stabilizing airspeed, and considering soft constraints on the angle of attack.

### 4.1 Path following

Dubins Aircraft segments (lines and arcs in 3D) [3] can be used to describe the majority of desired flight maneuvers in a typical fixed-wing UAV mission. Further, using simple paths such as arcs and lines allows spatially defined path following, independent of time (or consequently speed), a useful quality when only proximity to the track is desired and winds can significantly change the ground velocity. For the remainder of the section, we will consider Dubins segments as path inputs to the high-level controller, though it should be noted that the path objective formulation is not limited to these; e.g. within this work we have also incorporated a special case of the Dubins arc, the common unlimited loiter circle.

#### Path geometry

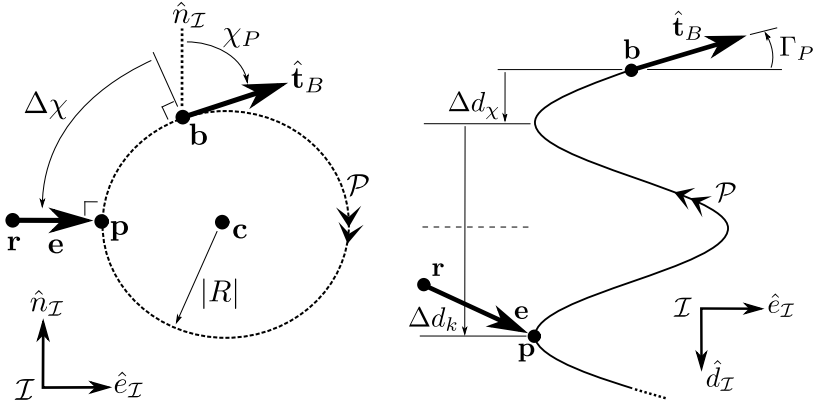
A minimum set of path parameters are required to define each time independent segment type as follows:

- Dubins line:  $\mathcal{P} \in \text{line} = [\mathbf{b}, \chi_P, \Gamma_P]$
- Dubins arc:  $\mathcal{P} \in \text{arc} = [\mathbf{c}, \pm R, \chi_P, \Gamma_P]$
- Loiter unlim.:  $\mathcal{P} \in \text{loit} = [\mathbf{c}, \pm R]$

where  $\mathbf{b}$  is the terminal point on a Dubins line,  $\mathbf{c}$  is the center point of a Dubins arc (or loiter circle) at the terminal altitude,  $\pm R$  is the arc (or loiter) radius with sign indicating clockwise (positive) or counter-clockwise (negative) direction,  $\chi_P$  is the exit course for a Dubins arc or line, and  $\Gamma_P$  is the inertial-frame elevation angle of a Dubins arc or line. Figure 7.9 describes the geometry.

To date, fixed-wing guidance logic for tracking helix-type paths has largely been limited to ‘pose-in-time’-based definitions (e.g. [3, 21]), i.e. desired positions and orientations are prescribed in time from initially starting to follow a given path segment. As the aircraft ground speed may change significantly over time, e.g. due to wind, we desire a time-independent formulation, defined only by spatial proximity to the path. A unique spatially defined solution for the closest point on a *line* in three-dimensions can be analytically calculated. However, to avoid multiple solutions or the necessity of numerical methods when finding the closest point on the 3D *arc* paths, we define an approximate of the closest point by decoupling the problem into lateral-directional and longitudinal planes.

We first consider the closest point in the lateral-directional plane to a circle with radius  $|R|$  (with unique spatially defined solution, except on the center point),



**Figure 7.9:** Lateral-directional (left) and longitudinal (right) *arc* path geometry.

and subsequently choosing the nearest arc ‘leg’ (assuming an infinite helix in the direction opposite the sign of the elevation angle). In (7.6), the  $d$  component of the closest point on the path  $p_d$  is calculated via summation of the terminal altitude  $b_d$ , the altitude deviation  $\Delta d_\chi$  due to the angular distance  $\Delta\chi$  from the exit point, and the altitude deviation  $\Delta d_k$  corresponding rounded  $k$  number of arc ‘legs’ away from the terminal point; see Fig. 7.9.

$$\begin{aligned}\Delta d_\chi &= \Delta\chi |R| \tan \Gamma_P \\ \Delta d_k &= \text{round} \left( \frac{d - (b_d + \Delta d_\chi)}{2\pi |R| \tan \Gamma_P} \right) 2\pi |R| \tan \Gamma_P \\ p_d &= b_d + \Delta d_\chi + \Delta d_k\end{aligned}\quad (7.6)$$

### Lateral-directional guidance

As fixed-wing aircraft behave dissimilarity in lateral-directional and longitudinal states, we also decouple the guidance objectives. The lateral track-error is defined:

$$e_{\text{lat}} = \bar{t}_{P_n} (p_e - r_e) - \bar{t}_{P_e} (p_n - r_n) \quad (7.7)$$

where  $\bar{t}_{P_n} = \frac{t_{P_n}}{\| [t_{P_n}, t_{P_e}] \|}$  and  $\bar{t}_{P_e} = \frac{t_{P_e}}{\| [t_{P_n}, t_{P_e}] \|}$ , for  $\| [t_{P_n}, t_{P_e}] \| \neq 0$ . The unit path tangent  $\hat{\mathbf{t}}_P$  is defined from the current path parameters  $\mathcal{P}_{\text{cur}}$ , specifically  $\chi_P$  and  $\Gamma_P$ .

One approach to the path following problem is to minimize the track-error itself along with the error between the aircraft course angle, inertial flight path angle, and

the path tangent, e.g.  $\chi - \chi_P$ , as designed in the 2D case presented in our previous work [82], as well as other works [38, 39, 86]. However, the NMPC's internal model also includes airspeed in an open-loop formulation, requiring simultaneous reference tracking. This dichotomy was found to present a challenge in properly defining a time-independent and velocity independent path following objective, as large track-errors would induce increased airspeed commands in an attempt to quickly reduce the larger cost in the shortest time within the horizon. Rather than attempt to define a complicated prioritized approach to weighting these two competing objectives, we instead embed unified, speed independent guidance logic, incorporating both directional and position errors into one lateral-directional and one longitudinal error term.

Augmentation of the NMPC internal model to include the guidance formulation has also previously been explored [21, 80, 81]; however, in these approaches, the analytic guidance law was used to generate attitude references within the control horizon, commands we wish our high-level NMPC to allocate itself. We therefore propose in this work to leave the control allocation open-ended for the nonlinear optimization to solve in real-time, while providing the NMPC with an error angle in the objective which, when minimized, results in convergence to the path.

The lateral-directional guidance error is formulated as the error angle  $\eta_{\text{lat}}$  from a *look-ahead* (or line-of-sight) guidance approach, commonly used in high-level lateral-directional position control for fixed-wing UAVs, see (7.8) and Fig. 7.10. Specifically we formulate our look-ahead vector  $\hat{\mathbf{I}}$  in a similar manner to the formulation found in [10], though it should be noted that several similar formulations exist, e.g. [11, 75].

$$\hat{\mathbf{I}} = \begin{pmatrix} l_n \\ l_e \end{pmatrix} = \begin{pmatrix} (1 - \theta_{l_{\text{lat}}}) \bar{t}_{P_n} + \theta_{l_{\text{lat}}} \bar{e}_n \\ (1 - \theta_{l_{\text{lat}}}) \bar{t}_{P_e} + \theta_{l_{\text{lat}}} \bar{e}_e \end{pmatrix} \quad (7.8)$$

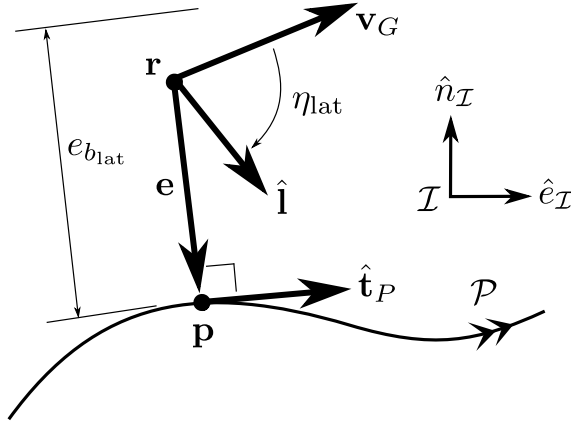
where  $\bar{e}_n = \frac{e_n}{\| [e_n, e_e] \|}$  and  $\bar{e}_e = \frac{e_e}{\| [e_n, e_e] \|}$ , for  $\|e_n, e_e\| \neq 0$ , and  $\mathbf{e} = \mathbf{p} - \mathbf{r}$ .  $\theta_{l_{\text{lat}}}$  is a mapping function for the lateral-directional track-error, equal to 1 at the track-error boundary  $e_{b_{\text{lat}}}$  and 0 when  $e_{\text{lat}} = 0$ . We choose a quadratic shape for  $\theta_{l_{\text{lat}}}$  such that beyond the track-error boundary a perpendicular approach to the path is demanded, and at the track-error boundary, the commanded direction begins to transition smoothly towards the unit tangent vector on the path:

$$\theta_{l_{\text{lat}}} = -e'_{\text{lat}} (e'_{\text{lat}} - 2) \quad (7.9)$$

where the normalized and saturated lateral-directional track-error is defined:

$$e'_{\text{lat}} = \text{sat} (|e_{\text{lat}}|/e_{b_{\text{lat}}}, 0, 1) \quad (7.10)$$

and  $\text{sat}(\cdot, \min, \max)$  is a saturation function. Borrowing a similar effect from the developments in [11], the track-error boundary is defined in an adaptive way with respect to the current ground speed  $e_{b_{\text{lat}}} = \|\mathbf{v}_{G_{\text{lat}}}\| T_{b_{\text{lat}}}$ , for  $\|\mathbf{v}_{G_{\text{lat}}}\| \neq 0$ , where  $T_{b_{\text{lat}}}$  is a tuning constant, varying the steepness of the look-ahead vector mapping on approach to the path, and  $\mathbf{v}_{G_{\text{lat}}} = [v_{G_n}, v_{G_e}]^T$ . In this work, we limit



**Figure 7.10:** Lateral-directional guidance logic.

the minimum track error bound  $e_{b_{\text{lat}}}$  with a smooth piecewise relationship to a minimum ground speed, arbitrarily set to  $1 \text{ m s}^{-1}$ , see (7.11).

$$e_{b_{\text{lat}}} = \begin{cases} \|\mathbf{v}_{G_{\text{lat}}}\| T_{b_{\text{lat}}} & \|\mathbf{v}_{G_{\text{lat}}}\| > 1 \\ \frac{1}{2} T_{b_{\text{lat}}} (1 + \|\mathbf{v}_{G_{\text{lat}}}\|^2) & \text{else} \end{cases} \quad (7.11)$$

The final guidance objective is defined as the error angle between the aircraft ground speed vector and the look-ahead vector.

$$\eta_{\text{lat}} = \text{atan2}(l_e, l_n) - \text{atan2}(\dot{e}, \dot{n}) \quad (7.12)$$

where  $\text{atan2}$  is the four quadrant arctangent operator, and  $\eta_{\text{lat}}$  should be wrapped to remain within  $\pm\pi$ . Note this discrete switch at  $180^\circ$  entails an instability point within the guidance formulation when propagated within the horizon; e.g., in the case that the aircraft is traveling on the track in the opposite of the desired direction. Careful initialization of the NMPC horizon should be considered to ensure operation outside of some range of this condition. A hard end term constraint bounding the aircraft to find trajectories leading away from this zone would also be advised, however, at this point, the ACADO framework [30] (used for auto-generation of optimized C code for real time control in this paper) does not support externally defined constraints. In lieu of this, some handling of this case outside of the NMPC (e.g. shifting the track away when close to this configuration) is possible.

One further consideration is the limitation of maintaining an airspeed greater than the wind speed. While the present guidance formulation will not fall into a

singularity, the resulting guidance commands may be erroneous (e.g. when the atan2 function on ground speed is very near zero on both input arguments), and this work does not further consider their influence on the NMPC's corresponding objective cost, apart from the insight that some objective weight retuning was found to be required. However, one may look to appropriate guidance enhancements for these particular conditions in, e.g., the formulation presented in [19].

### Longitudinal guidance

We approach longitudinal guidance in a slightly different manner than that of the lateral-directional, as longitudinal fixed-wing states do not have the full range of their counterparts in the 2D, horizontal plane, and are non-symmetric in climbing and sinking flight performance. We define the desired (on-track) vertical velocity  $\dot{d}_P = \|\mathbf{v}_G\|t_{P_d} \in (\dot{d}_{\text{climb}}, \dot{d}_{\text{sink}})$  corresponding to the path elevation and current ground speed, further bounded by the maximum climb rate  $\dot{d}_{\text{climb}}$  and sink rate  $\dot{d}_{\text{sink}}$ . Depending on the sign and magnitude of the longitudinal track-error  $e_{\text{lon}} = p_d - r_d$ , a vertical velocity setpoint  $\dot{d}_{\text{sp}}$  is modulated in an asymmetric manner between the bounds of maximum sinking and climbing using a quadratic look-ahead mapping function similarly defined to that within the lateral guidance, Sec. 4.1.

$$\dot{d}_{\text{sp}} = \Delta \dot{d} \theta_{l_{\text{lon}}} + \dot{d}_P \quad (7.13)$$

with look-ahead mapping  $\theta_{l_{\text{lon}}} = -e'_{\text{lon}} (e'_{\text{lon}} - 2)$  and normalized track-error  $e'_{\text{lon}} = \text{sat}(|e_{\text{lon}}/e_{b_{\text{lon}}}|, 0, 1)$ , and track-error boundary defined similar to the lateral-directional:

$$e_{b_{\text{lon}}} = \begin{cases} T_{b_{\text{lon}}} |\Delta \dot{d}| & |\Delta \dot{d}| > 1 \\ \frac{1}{2} T_{b_{\text{lon}}} (1 + \Delta \dot{d}^2) & \text{else} \end{cases} \quad (7.14)$$

with  $\Delta \dot{d}$  defined for climbing or sinking:

$$\Delta \dot{d} = \begin{cases} \Delta \dot{d}_{\text{climb}} & e_{\text{lon}} < 0 \\ \Delta \dot{d}_{\text{sink}} & \text{else} \end{cases} \quad (7.15)$$

where  $\Delta \dot{d}_{\text{climb}} = -\dot{d}_{\text{climb}} - \dot{d}_P$  and  $\Delta \dot{d}_{\text{sink}} = \dot{d}_{\text{sink}} - \dot{d}_P$ .

The resultant guidance error term is formulated as the vertical velocity offset, normalized by the range of climbing and sinking rates.

$$\eta_{\text{lon}} = \frac{\dot{d}_{\text{sp}} - \dot{d}}{\dot{d}_{\text{climb}} + \dot{d}_{\text{sink}}} \quad (7.16)$$

where  $\eta_{\text{lon}}$ , though not a true angular error as in the lateral-directional case, is then mostly defined between  $[-1, 1]$  (except with large vertical velocity deviations). Another purpose of the present formulation is to allow for high horizontal wind scenarios where we may still be able to climb or sink to a desired altitude, despite

flying at close to zero horizontal ground speed, something an angle based guidance objective would not readily handle.

### Switching conditions

Terminal conditions for Dubins arc segments require the aircraft to be within some acceptance radius  $R_{\text{acpt}}$  of the segment terminal point  $\mathbf{b}$  (proximity), traveling within an acceptance angle  $\eta_{\text{acpt}}$  of the exit course  $\chi_P$  (bearing), and beyond the terminal point  $\mathbf{b}$  in the path axis (travel). Only the travel condition is set for line segments to avoid runaway behavior when the other conditions are missed due to e.g. the path being commanded while the aircraft is not already close to the track and correct orientation. No terminal condition is set for unlimited loiter circles. Switching conditions are summarized in (7.17), and shown graphically in Fig. 7.9.

$$\begin{aligned} \|\mathbf{r} - \mathbf{b}\| &< R_{\text{acpt}} && \text{(proximity)} \\ \mathbf{v}_G \cdot \hat{\mathbf{t}}_B &> \cos \eta_{\text{acpt}} && \text{(bearing)} \\ (\mathbf{r} - \mathbf{b}) \cdot \hat{\mathbf{t}}_B &> 0 && \text{(travel)} \end{aligned} \quad (7.17)$$

where  $\hat{\mathbf{t}}_B$  is the unit tangent at the terminal point  $\mathbf{b}$  of the current path  $\mathcal{P}_{\text{cur}}$ . Note for Dubins arcs,  $\mathbf{b}$  must be calculated from the arc center  $\mathbf{c}$  and exit course  $\chi_P$ .

As in [37, 82], a switching state  $x_{sw}$  is defined and augmented to the model, with dynamic shown in (7.18). Then,  $x_{sw}$  defines the desired path in the queue to follow internally within the horizon.

$$\dot{x}_{sw} = \begin{cases} \rho & \text{terminal conditions met} \\ 0 & \text{else} \end{cases} \quad \parallel \quad x_{sw} > \text{threshold} \quad (7.18)$$

where  $\rho$  is an arbitrary constant.

## 4.2 Optimal control problem

We use the ACADO Toolkit [30] for the generation of a fast C code based nonlinear solver and implicit Runge-Kutta integration scheme. A direct multiple shooting technique is used to solve the optimal control problem (OCP), where dynamics, control action, and inequality constraints are discretized over a time grid of a given horizon length  $N$ . A boundary value problem is solved within each interval and additional continuity constraints are imposed. Sequential Quadratic Programming (SQP) is used to solve the individual QPs, using the active set method implemented



in the *qpOASES*<sup>4</sup> solver. The OCP takes the continuous time form:

$$\begin{aligned}
 \min_{\mathbf{x}, \mathbf{u}} \quad & \int_{t=0}^T \left( (\mathbf{y}(t) - \mathbf{y}_{\text{ref}}(t))^T \mathbf{Q}_y (\mathbf{y}(t) - \mathbf{y}_{\text{ref}}(t)) \right. \\
 & \left. + (\mathbf{z}(t) - \mathbf{z}_{\text{ref}}(t))^T \mathbf{R}_z (\mathbf{z}(t) - \mathbf{z}_{\text{ref}}(t)) \right) dt \\
 & + (\mathbf{y}(T) - \mathbf{y}_{\text{ref}}(T))^T \mathbf{P} (\mathbf{y}(T) - \mathbf{y}_{\text{ref}}(T)) \quad (7.19) \\
 \text{subject to} \quad & \dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}), \\
 & \mathbf{u}(t) \in \mathbb{U}, \\
 & \mathbf{x}(0) = \mathbf{x}(t_0)
 \end{aligned}$$

where control vector  $\mathbf{u} = [u_T, \phi_{\text{ref}}, \theta_{\text{ref}}]^T$  and state vector  $\mathbf{x} = [\mathbf{r}^T, \mathbf{v}_V^T, \Theta^T, \boldsymbol{\omega}^T, \delta_T, x_{sw}]^T$ .  $\mathbf{Q}_y$ ,  $\mathbf{R}_z$ , and  $\mathbf{P}$  are state, control, and end-term non-negative diagonal weighting matrices. State and control-dependent output vectors are compiled from path following objectives, airspeed stabilization, rate damping, and soft constraints:

$$\begin{aligned}
 \mathbf{y} &= [\boldsymbol{\eta}^T, v_A, \boldsymbol{\omega}^T, \alpha_{\text{soft}}]^T \\
 \mathbf{z} &= [\dot{\delta}_T, \mathbf{u}^T]^T \quad (7.20)
 \end{aligned}$$

where  $\boldsymbol{\eta} = [\eta_{\text{lat}}, \eta_{\text{lon}}]^T$ , and  $\alpha_{\text{soft}}$  is a soft constraint on the angle of attack (to be defined in Sec. 4.2).

### Feed-forward terms

Note that penalization of the attitude references and throttle input is necessary to avoid bang-bang control behavior. The selection of trim values, however, is important to avoid lowered performance in other objectives. Constant trim values (for level-cruise flight) are used as control output references  $\mathbf{z}_{\text{ref}} = [0, u_{T_{\text{trim}}}, 0, \theta_{\text{trim}}]$ . However, it was found that a feed-forward calculation for an approximate roll angle reference  $\phi_{\text{ref}}$  improves path following objective performance, as roll angle trims for turning flight have large offsets from level flight. The feed-forward term is calculated and subtracted from the roll angle reference  $\phi_{\text{ref}}$  throughout the horizon for the final output  $z_{\phi_{\text{ref}}} = \phi_{\text{ref}} - \phi_{\text{ff}}$ :

$$\phi_{\text{ff}} = \begin{cases} \tan^{-1} \left( \frac{\|\mathbf{v}_{G_{\text{lat}}}\|^2}{gR} \right) \frac{1 + \cos(\pi e'_{\text{lat}})}{2} & \mathcal{P}_{\text{cur}} \in \text{arc, loit} \\ 0 & \mathcal{P}_{\text{cur}} \in \text{line} \end{cases} \quad (7.21)$$

Note  $\phi_{\text{ff}}$  is not a commanded value to be explicitly tracked, but only gives guidance when near the track (via the multiplied smooth trig function as a function of the normalized lateral track error  $e'_{\text{lat}}$ ), keeping the weighted roll trim closer to

<sup>4</sup><http://www.qpOASES.org/>

the region of the optimal solution.

### Soft constraints

As the NMPC is also required to stabilize the open-loop dynamics of the vehicle, inappropriate commands could lead to a stall of the aircraft. To mitigate the potential for stall, we include a soft constraint on the angle of attack  $\alpha$ , keeping zero cost within the “safe” range, and quadratically increasing cost outside of these minimum  $\alpha_-$  and maximum  $\alpha_+$  bounds. A transition zone is defined by  $\Delta\alpha$  to allow tuning of the constraint’s steepness.

$$\alpha_{\text{soft}} = \begin{cases} \left( \frac{\alpha - (\alpha_+ - \Delta\alpha)}{\Delta\alpha} \right)^2 & \alpha > \alpha_+ \\ 0 & \alpha_+ \geq \alpha \geq \alpha_- \\ \left( \frac{\alpha - (\alpha_- + \Delta\alpha)}{\Delta\alpha} \right)^2 & \text{else} \end{cases} \quad (7.22)$$

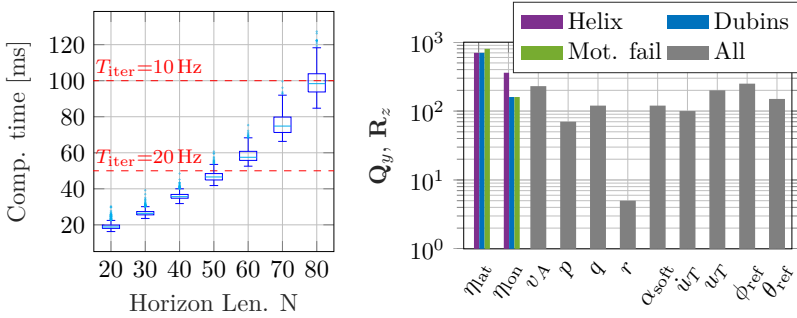
We have chosen soft constraints over hard constraints for several reasons. Namely, as the NMPC is operating on the high-level state-space, angle of attack rates and other fast modes are not modeled and likely not to be regulated well using only the attitude commands at its disposal (stall prevention is traditionally a low-level control problem). We thus only consider future prevention through foresight into the horizon, and allow momentary violations of the soft bounds during abrupt events like an actuator failure, strong gust, or poor initialization of the controller, instances where a hard constraint could result in either no control solution or more iteration steps leaving the low-level controller without commands for some time. Note that other works have incorporated similar soft constraints within the MPC framework at the same time combining them with hard constraints [36], in the cited example for the purpose of collision avoidance. A similar option could be explored for stall prevention in future work.

## 5 Flight Experiments

As the primary focus of this paper was the experimental implementation and validation of the proposed guidance methods, we present several indicative flight experiments, providing insights gained from the field experience, and omit simulation studies for brevity.

### 5.1 Hardware setup

Low-level attitude stabilization was run at 50 Hz on the Pixhawk, state estimates from the EKF were transferred via UART connection at 40 Hz over MAVLink/MAVROS to an on-board computer running Robotic Operating System (ROS), where a wrapper node iterated the NMPC solver at a specified fixed time interval  $T_{\text{iter}}$ . As future applications of NMPC based guidance approaches may include objectives



**Figure 7.11:** ROS node bench-tested on an Intel<sup>®</sup> UP board (Quad Core, 1.92 GHz CPU, 4 GB RAM) in HIL configuration with horizon step size  $T_{step}=0.1$  s. State measurement updates were randomized and a 3D Dubins arc segment was input to maximize computational load. Approx. 1 min of data collected for each configuration. (left) Objective weights for the normalized error outputs during each flight experiment. (right)

such as high speed obstacle avoidance, and noting that fixed-wing platforms require some time/space to maneuver, it is important to examine achievable real-time horizon lengths for experimentation. Hardware-in-the-loop (HIL) bench tests were performed for this purpose, see box plot results in Fig. 7.11. Here, *computation time* is defined as the sum of all operations conducted within the ROS node (solve time, array allocation, waypoint management, etc.) for one iteration.

## 5.2 Experimental results

Objective weighting was kept mostly constant throughout all experiments, save for some minor tuning adjustments, see Fig. 7.11 for a comparison. Note that the output error signals  $\mathbf{y}_{ref} - \mathbf{y}$  and  $\mathbf{z}_{ref} - \mathbf{z}$  were normalized by the expected error ranges for nominal flight in an attempt to improve intuition on relative weighting between signals. Guidance parameters were fixed throughout all experiments, see Table 7.2. Further insight into the specific weighting is elaborated within each experiments subsection.

### Helix following

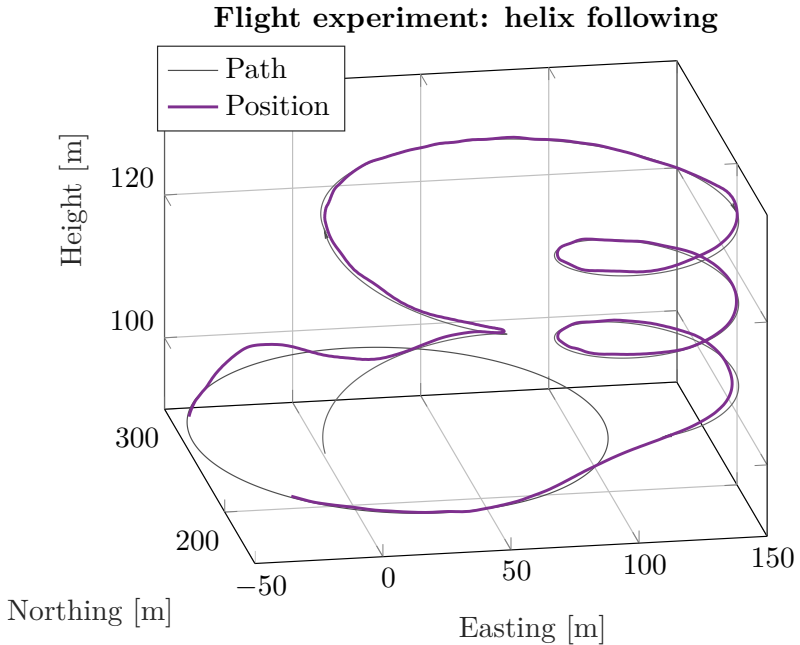
To push the boundaries of the control horizon in a real world setting, a horizon length of  $N = 70$  was used with  $T_{step}=0.1$  s, corresponding to a 7 s horizon. Control solutions were iterated at  $T_{iter}=0.1$  s, or 10 Hz.

Figures 7.12 and 7.13 show path following and airspeed stabilization on steep ascending and descending Dubins arcs. The arc radii were chosen to be just above

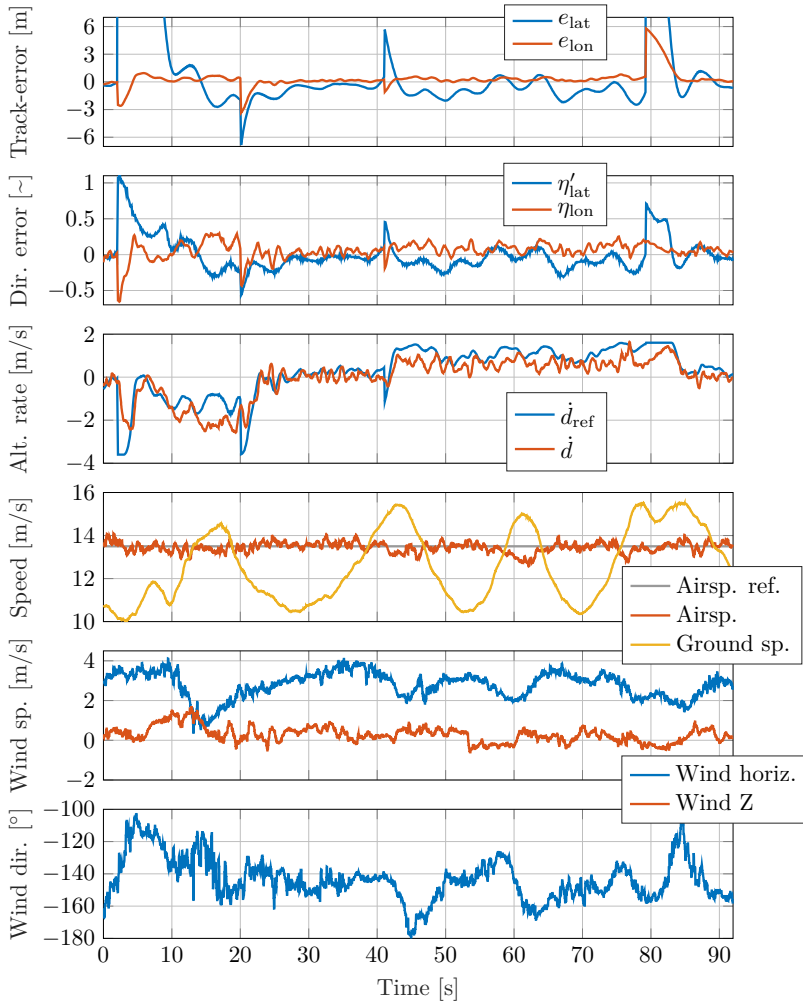
**Table 7.2:** Guidance parameters used for flight experiments.

| Param          | Value | Unit     | Param             | Value | Unit              |
|----------------|-------|----------|-------------------|-------|-------------------|
| $\Delta\alpha$ | 2     | $^\circ$ | $T_{b_{lat}}$     | 1     | s                 |
| $\alpha_-$     | -3    | $^\circ$ | $T_{b_{lon}}$     | 1     | s                 |
| $\alpha_+$     | 8     | $^\circ$ | $\dot{d}_{climb}$ | 3.5   | $\text{m s}^{-1}$ |
| $R_{acpt}$     | 30    | m        | $d_{sink}$        | 1.5   | $\text{m s}^{-1}$ |
| $\eta_{acpt}$  | 15    | $^\circ$ |                   |       |                   |

the physical limit for the given roll angle constraints ( $\pm 30^\circ$ ) and commanded flight speed ( $13.5 \text{ m s}^{-1}$ ). Final horizontal track-errors are kept within  $\pm 2 \text{ m}$  once settled to the path, and the vertical track-errors mostly regulated below  $\pm 0.5 \text{ m}$ .



**Figure 7.12:** Techpod tracks a steep ascending ( $8^\circ$  incline) Dubins arc with a 35 m radius, then summits on a large constant altitude arc before descending on another 35 m radius arc ( $3^\circ$  glide).



**Figure 7.13:** Guidance/track errors and air-, wind, and ground speeds during helix testing.

More extensive flight testing on these tight and steep helix paths, especially

when flying in wind, showed that appropriate relative tuning of the weights for the lateral-directional and longitudinal guidance errors is important. Less relative weighting on the longitudinal path objective, at times, allowed significant altitude deviations on down-wind portions of the helix; though, once turning back into the wind the ground speed was lowered and the altitude error was again regulated. The altitude deviations on the down-wind leg could become significant enough that the midpoint between helix legs was passed, causing a discrete switch to the lower leg within the horizon (recall the position-based helix logic in Sec. 4.1). While the latter point was later solved with a simple "arc length traveled" logic to refuse *previous* legs in each horizon, the prior required some investigation and subsequent adjustment to the objective weighting. Convergence to this local minimum was in part due to the greater roll angle requirements for tracking the arc down-wind (faster ground speed), which reduced the vertical component of lift available for the steep climb and thus increased the required thrust and/or angle of attack (controlled with pitch), which would, in turn, cause greater deviations of these values from their constant objective references. This, coupled with the higher weight on lateral-directional track error, induced lessened prioritization of the altitude error within the optimization. Higher weighting on the longitudinal path objective (see 7.11) resulted in the improved performance seen in Fig. 7.12.

### Connected Dubins segments

Figures 7.14 and 7.15 show path following of connected Dubins lines and arcs. As in the previous experiment, tight radii were commanded on each arc segment, and newly, the incorporation of line segments with  $90^\circ$  corners. A 7 s horizon (length  $N = 70$ ,  $T_{\text{step}} = 0.1$  s) with iteration rate 10 Hz was again used.

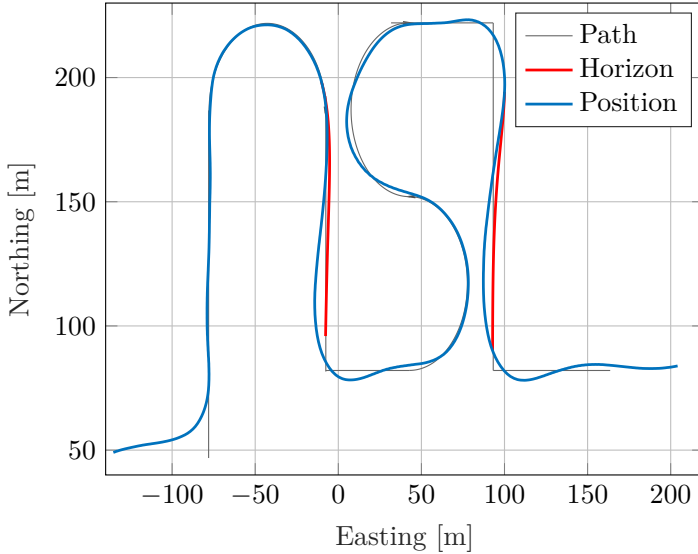
Note despite the  $\sim 5 \text{ m s}^{-1}$  wind and tight arcs, the horizontal track-errors are kept within  $\pm 1$  m once settled to a given set of smoothly connected path segments (see the left leg and top curvature of the  $A$  in *ASL*, Fig. 7.14) and the vertical track-errors mostly regulated below  $\pm 0.5$  m. Once the terminal conditions are met, the planned trajectory considers not only the current path, but also the next, allowing reduced tracking performance in the down-wind leg (also considering body-rate penalties) in order to reduce overshoot after the  $90^\circ$  turn onto the next.

One may notice the slightly noisy attitude and throttle reference signals. Though in the present cases not having significant detrimental effect on the aircraft performance, it is worth noting the origin is primarily from the unfiltered (aside from a subtracted estimated bias) angular rate feedback to the NMPC. Future consideration of some feedback or control output filtering may be warranted for particular signals, though care should be taken not to add undesired delays on the controller response.

### Motor failure

To explore potential fault tolerance of the designed guidance algorithm, we simulate a motor outage during a flight experiment. This tests the NMPC's capability of

## Flight experiment: Dubins segments in wind



**Figure 7.14:** Techpod tracks connected Dubins arcs and lines in ca.  $5 \text{ m s}^{-1}$  winds. The red NMPC horizons showcase the planned trajectory converging to the straight segments before the horizon reaches the terminal conditions for switching to the next segment.

reconfiguring the control allocation for the multi-objective problem in real-time. We assume detection of a motor failure, generally (though perhaps not the exact type of failure), can be accomplished by monitoring the expected current draw (with respect to the throttle input) and comparing to a threshold value, a feature presently integrated in the custom flight software running on the Techpod UAV.

In this experiment, we change the horizon length to  $N = 40$ , again with  $T_{\text{step}}=0.1 \text{ s}$ , a 4 s horizon. However, with the reduced computational load (see Fig. 7.11) we are able to increase the iteration rate to 20 Hz (or  $T_{\text{iter}}=0.05 \text{ s}$ ). A higher iteration rate allows faster feedback for the quicker dynamics expected to need mitigation in a motor failure situation.

Figures 7.16 and 7.17 show the Techpod UAV ascending to a loiter circle, when the motor is cut at  $t=15.5 \text{ s}$ . We simultaneously apply an arbitrarily large weight on the throttle input (e.g.  $1e6$ ). This causes the NMPC to reallocate the remaining

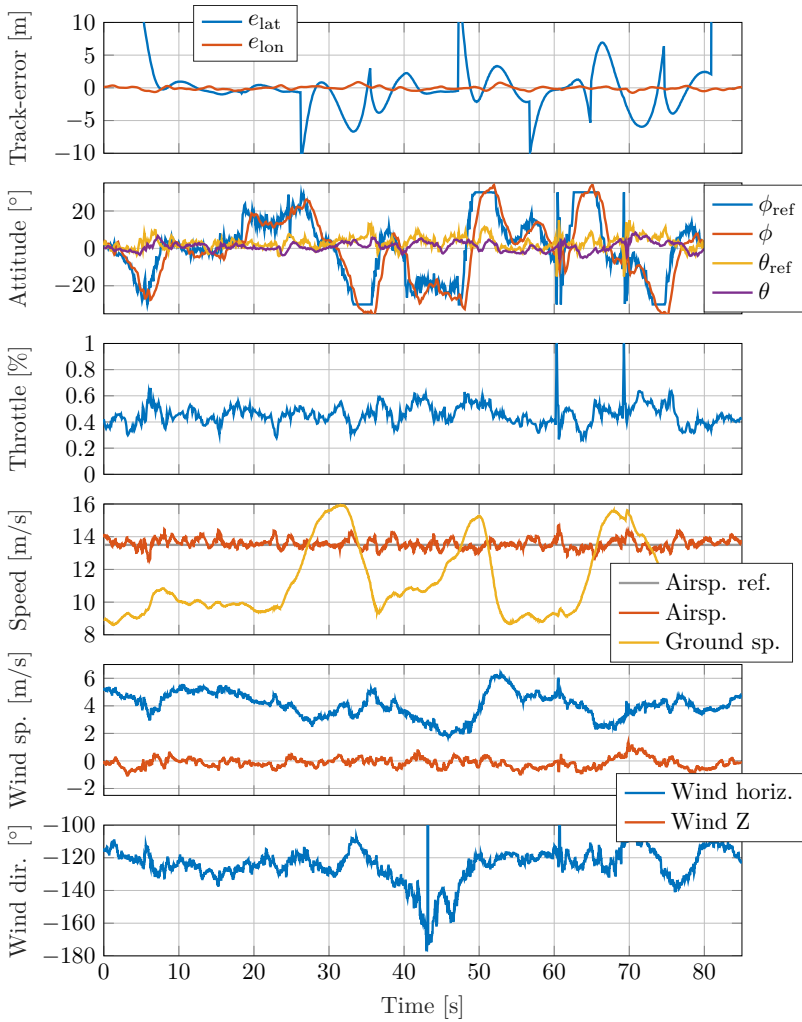
control signals, in this case, immediately pitching down and stabilizing a glide at the commanded airspeed. Notice that the lateral-directional track-error still remains below  $\pm 1$  m, as the optimization is able to maintain this particular tracking objective. A brief, negative spike in the angle of attack  $\alpha$  is seen at the time of the failure, but this is quickly returned to nominal values, and well within the bounds of the soft constraints (stopping just before the buffer zone). At  $t=34$ s, the motor is reactivated, and the NMPC is similarly able to quickly reconfigure the control allocation and resume ascending to the loiter circle. Despite the almost halving of the NMPC horizon, and the doubling of the iteration rate, no major retuning of the objective weights and parameters was necessary to maintain good performance.

## 6 Conclusions & Future Work

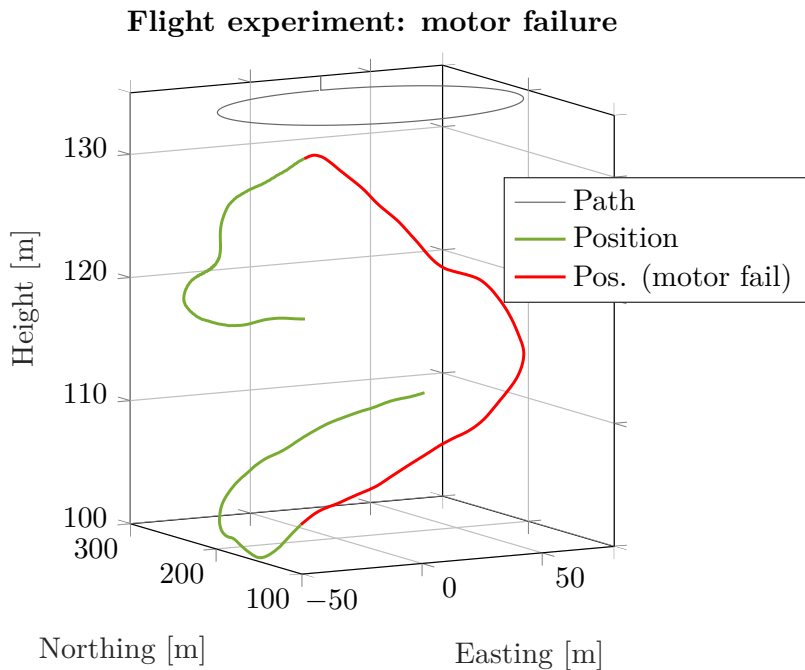
In this work, we presented an approach for modeling and identification of the control augmented dynamics of a small fixed-wing UAV with a typical OTS autopilot in the loop, and further, utilized these dynamics within the internal model in the design of a high-level NMPC for simultaneous airspeed stabilization, 3D path following, and handling of soft angle of attack constraints. The identified model structure demonstrated good predictive qualities, as shown with comparisons of open-loop simulation times on the order of tens of seconds, with respect to flight data. The designed high-level NMPC showed good performance for the multi-objective problem in a variety of experimental scenarios; in particular showcasing the benefit of explicit consideration of wind within the formulation and sufficiently long horizon times.

In future work, considering both airspeed and attitude within the low-level control loop would be advantageous, allowing a simplification of the control augmented model structure and, further, the possibility of increased horizon lengths and/or additional objectives such as obstacle avoidance and/or terrain constraints on autonomous landing.

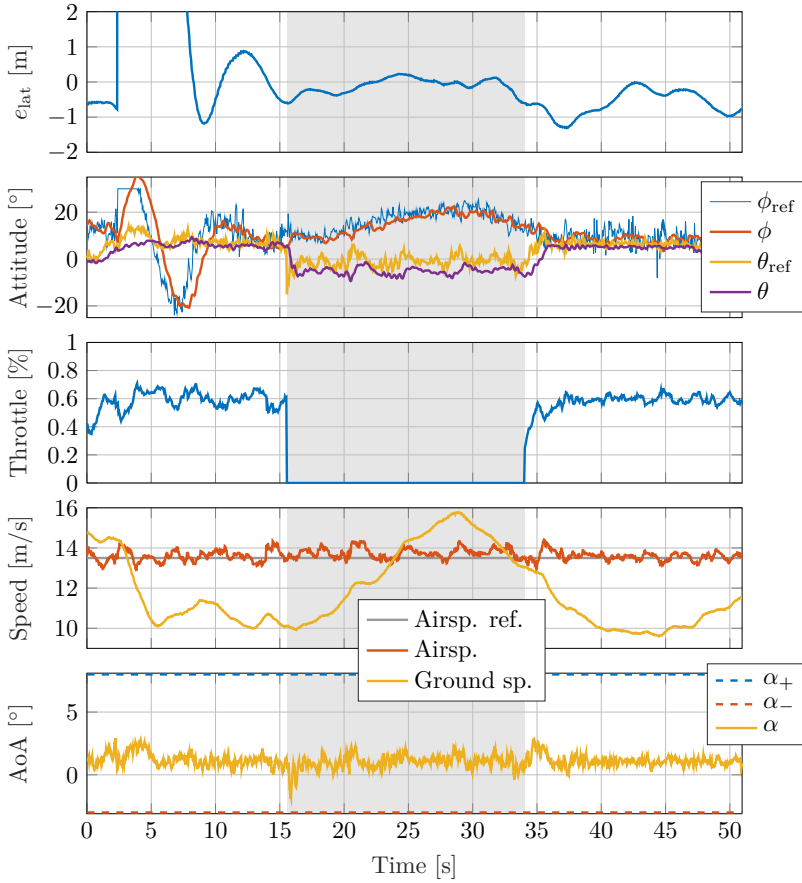




**Figure 7.15:** Track-errors and control inputs and air-, wind, and ground speeds during Dubins segment tracking in wind.



**Figure 7.16:** Techpod experiences a mock- motor failure during ascent to a loiter path.



**Figure 7.17:** Lateral-directional track-error and control inputs (left) and airspeed, ground speed, and angle of attack (right) during a mock- motor failure.



# An Outlook on Environment-Aware Local Re-Planning for Safe Near-Terrain Operation of Fixed-wing UAVs

Thomas Stastny, Timo Hinzmann, and David Rohr

## Abstract

Future fixed-wing and/or hybrid platforms, operating autonomously beyond the horizon, near terrain and disturbed by variable, turbulent wind fields, will need more than global, open loop motion plans to safely execute their missions. Handling all real-time variables not foreseen by the original planner will require direct and generalized feedback of real-time, on-board environmental awareness within the control loop. In this brief, we showcase some ongoing work in its preliminary stages of development towards utilizing the developed control structures in Papers IV and V, reconceptualizing the work as an environment-aware local re-planner. We first present some design trade-offs particular to long horizon Nonlinear Model Predictive Controllers (NMPCs), leading to some enhancements to our previous control structure. The brief then turns to environmental perception, where vision-based elevation mapping is utilized to provide a generalized 2.5D world representation to the aircraft. The map is bilinearly interpolated for height feedback, and we design an efficient ray casting approach for detection of forward (line of flight) and lateral occlusions. The occlusions are used to construct novel “relative” Euclidean Signed Distance Fields (RESDFs), which are a function of the relative velocity between the vehicle and obstacle. The RESDFs are transformed into optimizable soft constraints for the objective function of the NMPC. A preliminary example of the full system acting to avoid an obstructing hillside is demonstrated in hardware-in-the-loop (HITL) simulation.

## 1 Introduction

In Papers IV and V, fixed-wing implementations of high-level model predictive guidance were developed and demonstrated in flight experiments for three-dimensional path segment following. Paper V in particular showed how simultaneous airspeed control, angle of attack constraints, and even a limited example of motor failure could be encapsulated within the same controller. A control augmented model formulation was used which we were able to demonstrate as a sufficiently predictive model for longer control horizons, and easily identifiable from flight data.

The results from Papers IV and V also indicate what a predictive optimization based control approach can potentially provide over more deterministic controllers; e.g. in Paper V, how the NMPC utilized it's long predictive horizon to "locally" adapt it's trajectory around a 90° path change in strong winds, increasing the instantaneous track error while finding a state trajectory which more globally reduced track error along the full transfer. These types of insights, while on their own may not merit full time employment of such control approaches for simple tasks, forecast the potential of handling the more advanced control objectives that will inevitably be required of autonomous UAVs in the coming future.

In this brief, we aim to showcase where this research is headed, outlining some of our ongoing work in its preliminary stages of development which illuminates our original motivations in developing the ability to deploy long-horizon model predictive controllers on fixed-wing UAVs: Specifically, our aim is to utilize the developed control framework and reconceptualize it as an **environment-aware local re-planner**.

Our primary hypothesis in this outlook is that future fixed-wing and/or hybrid platforms, operating autonomously beyond the horizon, near terrain and disturbed by variable, turbulent wind fields, whether for close surveying, payload drop and recovery, or even physical interaction with the environment, will need more than global, open loop motion plans to safely execute their missions. Handling all real-time variables not foreseen by the original planner will require direct and generalized feedback of real-time, on-board environmental awareness *within the control loop*. We note that this functionality should be seen as a *complement*, not necessarily a replacement, for existing motion planners.

### 1.1 Related Work

While we discussed estimation and guidance level control solutions with wind awareness throughout Part A, here our primary environmental concern will be *terrain*.

Tackling terrain avoidance (when it is indeed our aim *not* to interact with the environment) gives three main challenges:

1. The ability to *sense* the terrain with enough distance that our (relatively) high speed vehicles have time to react and avoid.
2. A method of compressing map information efficiently for controller compre-

hension.

3. Fast motion planning which respects vehicle dynamics and constraints.

The following subsections detail some of the most recent literature related to the challenges listed above, with a particular focus on small fixed-wing UAVs.

### Sensing and Mapping

A large amount of work has been conducted on real-time mapping in the context of Micro Aerial Vehicles (MAVs), particularly multi-copter platforms. Widely used mapping frameworks populate 3D occupancy grids such as an octomap [29] which may be used by planning algorithms for collision checking. More recently the *voxblox* [63] and *FIESTA* [25] frameworks were released, proposing efficient methods for incrementally building Euclidean Signed Distance Fields (ESDFs) which can more readily provide collision cost and gradient information needed for e.g. trajectory optimization based motion planners. For instance, the *voxblox* pipeline was used on-board multi-copters for continuous time reactive re-planning in [62] and safe outdoor exploration in a forest in [64].

Off the shelf rigid stereo rigs or depth cameras have shown to be sufficient for populating occupancy grids and achieving autonomous operation of many ground robots and MAVs. However, as detailed in [27] and [28], similar off the shelf solutions applied to fast moving (relative to multi-copters) fixed-wing vehicles fail for several reasons. On the vision side, the primary deficit of small baselines and/or current depth cameras is low range and depth uncertainty. E.g. a 10 m range on an Intel RealSense<sup>1</sup> is nowhere near functional for an aircraft flying above  $15 \text{ m s}^{-1}$ .

Other sensing possibilities include scanning lasers (popular on large autonomous helicopters, see e.g. [76], and ground vehicles) or radar. The former being heavy and expensive, or when in smaller scale (for example as outfitted on an indoor fixed-wing in [7]), again impeded by range, and the latter not yet miniaturized for small UAV use.

Longer range (on the order of 60 m) single point lidars are often used by fixed-wing vehicles for height above ground (HAG) measurements once entering a landing approach. An example of using several point lidars pointed in various directions for terrain avoidance was shown on a small fixed-wing UAV flying through a canyon in [23]. However, this approach has limited applicability in more complex environments where much broader knowledge of the terrain is required, and even this 60 m may not be enough for a fast flying plane to avoid an obstacle.

To our best knowledge, the only work to date solving the problem of depth and range for fast fixed-wing vision solutions was tackled in [27] and [28], where wide baselines (up to 3 m from wing tip to wing tip) increased depth perception up to the order of 100 m-200 m, and a “flexible” baseline estimation framework mitigates errors from the inherent non-rigidity of the aircraft’s wings in flight. Within [28], the approach was tested in flight experiments on our own Techpod UAV.

<sup>1</sup><https://click.intel.com/intelr-realsensetm-depth-camera-d435.html>

## Obstacle Avoidance

Motion planning, and more specifically obstacle avoidance, is an enormously studied topic in MAV research. An entire book could be written on all the competing approaches for various platform types, sizes, and applications (and several have been). However, we will limit our focus here to specifically recent works pertaining to fixed-wing avoidance, attempting mostly to single out those works with flight experimentation to aid our understanding of this real-world challenge.

Here we need to consider the scale of the avoidance problem we want to solve. If the desired stand off distance from any given terrain is large enough that even vision solutions with very large depth uncertainty at far ranges would detect a large previously unknown structure some 500 m-1 km away, the a simple solution would be to move position waypoints out of the way, e.g. set a stop and loiter point, and then wait for a global path planner to take some seconds and solve for a new plan with the updated information. This condition is quite common for high-flying fixed-wing missions, and there are a variety of such global planners existing. For example, see [59], where wind is also predicted online and considered in the 3D path planner. [70] provides another example where a digital elevation model (DEM) is provided a priori; loiter paths are used to climb to safer altitudes when the employed search algorithm (independently surveying) runs up against terrain in the map.

However, as introduced above, our aim here is *near terrain* operation, i.e. we actually want to get as close as possible without crashing. Looking to fixed-wing literature for similar work, the lion's share of implementations appear to favor fast sampling and/or single selection of maneuvers from trajectory libraries, checking each potential trajectory for collisions with whatever map is preloaded or estimated on-board.

In [2], a “pushbroom stereo” vision setup approached a reduction in block matching processing requirements by choosing a single distance at which to search for matches, storing obstacles as the vehicle progresses forward. A small library of manually demonstrated maneuvers, e.g. veer left, climb, fly straight, were programmed into a selection module and a 12-state time varying LQR controller was used to track the pre-determined state references. Their experimental results were impressive, however the obstacle detection range was still limited to ca. 10 m and only the agile nature of their vehicle allowed avoidance. Further the reactive nature of the obstacle detection algorithm with only the limited “swept up” set of points ahead to plan through caused further issues in complex obstacle scenarios.

Other notable works include [9] and [43], where a trajectory library is built up of optimized state trajectories, in the latter work even for partially stalled aerobatic maneuvers, and feedback control was executed via a simple nonlinear attitude and thrust response law along the planned trajectories. In [9], a simulated Intel RealSense was used to generate point clouds at a maximum of 10 m in front of the aircraft, while in [43], real flight experiments were conducted, no on-board mapping was performed, and “faked” obstacles were preloaded into the flight computer. Both works utilized a very agile aerobatic, lightweight airframe with 0.86 m wingspan.



Another notable mention, though not utilizing trajectory libraries, [7] mounted a Hokuyo scanning laser to a 2 m wingspan plane and indoors through a parking garage using a pre-optimized polynomial trajectory from hand picked waypoints. The feedback control law exploited differential flatness, mapping the Dubins-polynomial trajectories back to actuator commands.

One commonality among the works listed above is their reliance on their lower-level controllers to perfectly track the generated trajectories, an assumption that is clearly not achieved in many of the experimental results shown in the papers. This can be quite dangerous if no other feedback is provided while executing the trajectories. The underlying issue here is almost certainly the separation between the fast kinematics based trajectory planning and underlying unmodeled low-level response dynamics acting on open loop trajectories. Further, these works use highly agile and lightweight airframes which are not representative of the majority of small fixed-wing UAVs in operation. Consideration of these lower-level response dynamics and tighter flight regime constraints must be taken for a broader class of less maneuverable vehicles.

Obstacle avoidance with consideration of underlying dynamics, in particular model predictive control formulations where obstacle information is embedded in the objective function, has also been studied. In [24], full body control of a fixed-wing UAV down to actuator level was implemented as a NMPC with trajectory tracking and avoidance of sphere obstacles (modeled as constraints). The results were, however, provided in simulation with the same dynamics modeled within the controller and perfect state information. [36] presents a multi-agent obstacle avoidance scheme for multi-copters, where spheres around Vicon provided vehicle positions were similarly the obstacle representation of choice. However here, state estimation uncertainty on the vehicle position is taken into account to provide a widening (with horizon time) uncertainty used for scaling up objective costs in the more uncertain regions of the trajectory. Instead of only hard constraints, the work further employs potential fields in the objective function applied within some range of the obstacle to aid the optimizer in finding avoidance trajectories. The demonstrated flight experiments were quite convincing, and motivate the concept of embedding of obstacle and lower-level dynamics into a single cost function.

## 1.2 Approach

In this brief, we will approach the problem of fixed-wing terrain avoidance by re-evaluating and expanding our previously developed long-horizon NMPC from Paper V, incorporating *generalized terrain feedback* to the objective function.

The wide-baseline flexible stereo vision solution from [28] will be used to provide real-time sensory input of the environment. However, instead of either populating a 3D occupancy grid and/or utilizing *voxblox* for ESDF generation, we propose to fuse our vision measurements to the cells of a “Grid Map” [14], a 2.5D elevation map representation of the environment. Our rationale being that full 3D maps are more computationally expensive (and time consuming) to compute, may not be as scalable for large environments, and fixed-wing vehicles typically do not operate in

truly 3D environments (e.g. with occlusions overhead), even when operating near terrain.

With an elevation map being populated on the fly, we propose an efficient means of interpreting the grid through height look-ups and radial ray casting, constructing local “relative” ESDFs (RESDFs) which consider not only the distance to terrain in the map, but also the relative approach velocity of the aircraft, only building distance fields where the aircraft is currently planning to fly. These new RESDFs are designed for direct incorporation in the NMPC objective function, providing a *generalized* representation of obstacles which may adapt over time and space, without the need for any further shape abstraction (e.g. planes or spheres) or tedious collision checking through point clouds along trajectories. To preliminarily evaluate our approach in a worst case scenario, no global or local planner external to the NMPC will be used (aside from arbitrarily “poorly” placed path setpoints), forcing the controller to react and *re-plan* on its own.

The remainder of this brief is organized as follows: Section 2 details some afterthoughts and remedies to left-over issues (particular to long-horizon MPC) from Papers IV and V. Section 3 describes our proposed environmental mapping method as well as how we can interpret this information within the MPC feedback loop. Section 4 formally defines the updated controller, followed finally by Hardware-in-the-Loop (HITL) simulation results in Section 5 and overall conclusions in Section 6.

## 2 “Outsmarting” the Optimizer: Safe Objectives for Long Horizons

First, we need to consider some of the lingering issues found in the previous works and attempt to resolve them before including more objectives. Specific reformulations are outlined in the following subsections.

### 2.1 Challenges for Long Horizon Optimizations

In Paper IV, path following was considered in the objective by finding the closest point on the path to each node in the horizon, applying the corresponding position error and directional error (from the path tangent at the closest point on the path) as quantities to be regulated. While this worked for the 2D case in Paper IV, we found in Paper V that a similar approach fails once airspeed is included as a state, as the optimizer now has knowledge that increasing airspeed reduced that track error faster than maintaining the desired flight speed.

Paper V remedied this issue by encapsulating path following logic into the objective model, reducing the “path errors” only to directional ones, independent of the airspeed. This allowed the optimizer to mostly decouple airspeed control and path following objectives, treating them as parallel requirements.

While the Paper V approach showed excellent tracking performance, we later observed some issues with the region of attraction of the path following objective.

An instability point existed when the horizon was initialized opposite the normal direction to the current track, which caused the aircraft to oscillate and actually fly away in the opposite direction! This was due to the discrete switch in the path following objective Jacobians at that point in the state space. Such a point exists in nearly every path following controller, though the issue normally resolves itself with any numerical error or in real flights any disturbance that jostles the aircraft far enough in one direction or the other that the gradient then becomes well defined. However, as we use long horizons, each horizon node independently may be settling on one side or the other of this discrete reference switch, at times causing the resulting summed full horizon path gradient to vary around that mean (opposite direction), hence our fly away situation.

Another related issue was found with horizon initializations in the opposite rotational direction from a loiter path, but within close proximity to the track (within the reference vector field bend). Here the summed costs over the long horizon entered a local minimum which kept the aircraft flying in the opposite direction at a stand off distance from the path.

Note these issues similarly may arise with large changes in path reference, effectively acting as a new horizon initialization.

One potential fix would be removing the objective Jacobian feedback from the directional guidance commands, effectively reducing the node-wise path objective to a single direction at each point in time, i.e. we run the guidance logic on each node external to the objective cost, and simply command tracking of those individual reference directions. As far as the optimizer would then be concerned, there is only one direction (that will not change with any other state) in which to point the aircraft. However, even this approach, given our longer horizons, runs into problems. For one, horizon initialization again is an issue, and we can very well run into e.g. the same counter rotating stand off local minimum. More critically, when converging to the path with this formulation, a “slapping” effect, depending on how aggressively the path following cost is weighted, can develop in the end of the horizon caused by the optimizer only considering directional errors and taking too large of steps due to the lack of Jacobian information on how track relative position influences the guidance commands. De-tuning the path following objective can reduce this effect, but then of course greatly reduces the path following performance.

Three primary takeaways arise from these observations, specific to long horizon MPC formulations:

1. Attractive regions for path following logic are not universal when applied in a node-wise fashion over the horizon.
2. Proper initialization of longer horizons can be challenging, but is critical to performance and stability.
3. Position dependence in the path following reference Jacobian is necessary for fast track convergence without negative externalities (e.g. “slapping”).

## 2.2 Wind-Aware Lateral-Directional Reference Trajectory Generation

In this section, we reformulate the path following objectives to a “mobile” trajectory tracking approach, using a wind-aware trajectory generation module. While the previous two papers spent quite some efforts to formulate *path following* objectives into a time-horizon MPC, we pose here that *trajectory tracking* is 1) a more natural approach for the given control structure, and 2) will resolve many of the issues seen in the previous section. The *mobile* designation here refers to our aircraft relative regeneration of the reference trajectories. As aircraft are in constant nonholonomic motion, we cannot fix a given trajectory from one origin and feed the time dependent points to the optimizer states, as is commonly done for multi-copters, as this could lead to run away from the reference points when our vehicle inevitably is disturbed. Here, we reset the generated trajectory from the current aircraft state at every optimization iteration step.

While ideally more optimized trajectory libraries or path planners such as those elaborated in Section 1.1 could be utilized and sampled for our horizon, here we instead simply propagate our wind-aware guidance logic from Paper III in time from the aircraft current position and heading, assuming perfect tracking of lateral acceleration and airspeed reference commands. This provides a horizon of horizontal (2D) position  $\mathbf{p}_{\text{lat}}$ , airspeed  $v_A$ , and heading  $\xi$  references.

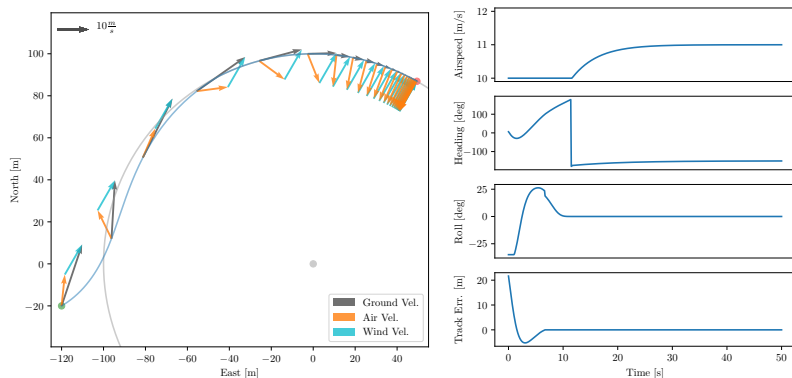
The air mass relative commands from our wind-aware guidance moreover add the ability to consider excess wind conditions such as those discussed in Part A. By applying the simple roll reference  $\phi_{\text{ref}}$  allocation used for all controllers in Part A, we can further provide a feed-forward term to the NMPC.

To avoid purely relying on the tuning of the guidance logic used for propagating the trajectories, we set path convergence criteria which “snaps” the samples to the path once below a given cross track error and directional threshold. Euler propagation is used with a sample time equal to the respective NMPC time step discretization. An example reference trajectory is shown in Fig. 8.1.

The following outcomes are realized by using this trajectory generation approach:

- Though Jacobian information from the guidance setpoints is no longer embedded in the NMPC objective function, defining the additional position objective for the NMPC to track effectively dampens any possibility for “slapping”. Note, as we will see in Section 5.2, only a light weight on position error is necessary.
- We achieve a more ideal approach to the path. Before, the node-wise approach was in reality treating each horizon node as an individual aircraft, resulting in the horizon as a whole “drifting” towards the path, rather than forming the state trajectory we would expect.
- Including both direction and position references generated in sequence makes us much less prone to the instability condition when the horizon is initialized in the opposite direction of the track. This will further become important

when we introduce obstacles and the aircraft will be consistently deviating away from the path.



**Figure 8.1:** Example reference trajectory to a loiter path in excess wind. Notice the trajectory “snaps” to the path at ca. 7s, seen by the small kink in the roll reference, beyond which point all references are sampled directly on the track. The trajectory enters a terminal condition by ca. 30s where the wind-aware guidance logic compensates the excess wind and simultaneously keeps the aircraft on the path. 10 Hz time discretization is used for a horizon length of 500. In practice note our horizon lengths are an order of magnitude smaller, only elongated here for visualization purposes.

### 2.3 Longitudinal Guidance, as Feedback

While in the last section we externalized the lateral-directional guidance from the NMPC internal cost model and implemented a reference trajectory generation approach, this was not necessary for the longitudinal axis, as the range of flight path angles in which the aircraft operates is vastly smaller (i.e. no possible “turn arounds”, we are not performing aerobatic maneuvers). With this in mind, each horizon node is penalized at its previously optimized state against a longitudinal guidance reference, as done in Paper V.

We use a simple quadratically shaped vector field which steers the vehicle from a maximum approach vector at a fixed vertical position error boundary until reaching the path vertical tangent  $t_{P,d}$  at track error  $e_{10n} = 0$ . However, different from the

last paper, we incorporate the vertical wind estimate and formulate an air mass relative flight path angle setpoint.

The guidance law acts on the fed back longitudinal track error:

$$e_{\text{lon}} = p_{P,d} - p_d \quad (8.1)$$

where  $p_{P,d}$  is the closest vertical position on the commanded path. To consider asymmetric climbing and sinking flight performance, we define climbing and sinking maximum approach flight path angles  $\gamma_{\text{climb}}$  and  $\gamma_{\text{sink}}$ , respectively. To maintain a continuous *and* smooth guidance command, some modulation of the track error boundary is necessary.

$$e'_{b,\text{lon}} = \frac{e_{b,\text{lon}}}{2} \left( 1 + \left| \frac{\gamma_{\text{sink}}}{\gamma_{\text{climb}}} \right| \right) \quad (8.2)$$

where

$$\bar{\gamma} = \frac{\gamma_{\text{sink}} + \gamma_{\text{climb}}}{2} \quad (8.3)$$

is the midpoint of the flight path setpoint axis, and

$$\gamma_{b/2} = \frac{|\gamma_{\text{sink}}| + |\gamma_{\text{climb}}|}{2} \quad (8.4)$$

is the half span. The former two quantities are used for scaling/shifting the control mapping, as will be shown in the following.

An “on-track” flight path angle  $\gamma_P$  is determined considering the current ground speed, airspeed, and vertical wind component:

$$\gamma_P = \text{constrain} \left( -\frac{v_G \hat{t}_{P,d} - v_{W,d}}{v_A}, \gamma_{\text{sink}}, \gamma_{\text{climb}} \right) \quad (8.5)$$

From here, we can compute a unit track error increment necessary for shifting the setpoint bounds

$$\Delta e'_{\text{lon}} = \begin{cases} \sqrt{1 + \Delta\gamma'} - 1 & \Delta\gamma' < 0 \\ 1 - \sqrt{1 - \Delta\gamma'} & \text{else} \end{cases} \quad (8.6)$$

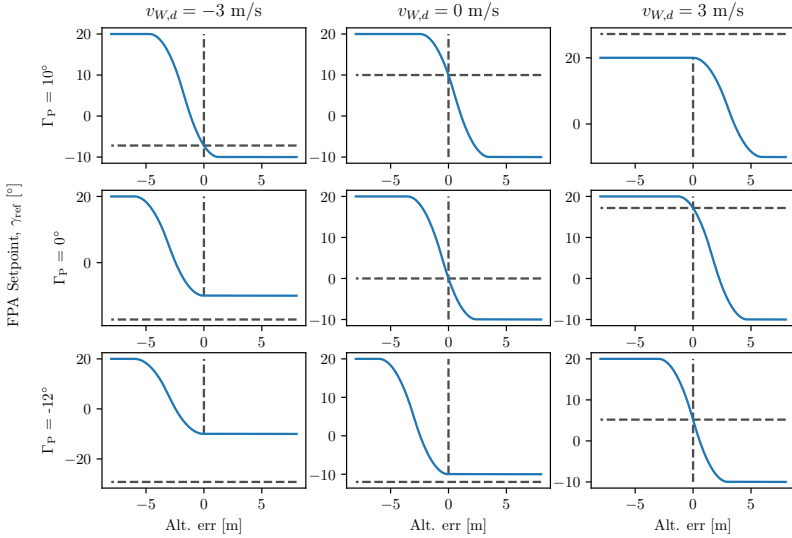
where  $\Delta\gamma' = \frac{\bar{\gamma} - \gamma_P}{\gamma_{b/2}}$ . A normalized (unit) longitudinal track error is now calculated with vertical wind, non-zero path inclination, and approach angle asymmetries considered with the unit track error increment.

$$e'_{\text{lon}} = \text{constrain} \left( \frac{e_{\text{lon}}}{e'_{b,\text{lon}}} + \Delta e'_{\text{lon}}, -1, 1 \right) \quad (8.7)$$

Finally, then leading to the flight path angle reference to command the vehicle:

$$\gamma_{\text{ref}} = \begin{cases} -e'_{\text{lon}} (e'_{\text{lon}} + 2) \gamma_{b/2} + \bar{\gamma} & e'_{\text{lon}} < 0 \\ e'_{\text{lon}} (e'_{\text{lon}} - 2) \gamma_{b/2} + \bar{\gamma} & \text{else} \end{cases} \quad (8.8)$$

Several setpoint patterns are displayed against longitudinal track error in Fig. 8.2 for varying winds and path inclinations.



**Figure 8.2:** Flight path angle setpoints vs. longitudinal track error for varying vertical wind speeds and path inclinations  $\Gamma_P$  (defined positive up).

Note that we augment the longitudinal guidance to the NMPC internal model such that the dynamics of the flight path angle setpoint are known (via objective Jacobians with respect to vehicle states) and we avoid the previously described “slapping” issues caused by external reference commands.

### Implementation Note

For the case of longer horizons (our case) with a more aggressive guidance tuning (i.e. a small  $e_{b,\text{lon}}$ ), it can happen that the optimizer gets stuck in a local minimum

near the track where position and flight path errors on the latter end of the horizon are converged, but the start of the horizon (near the current aircraft position) settles just beyond the boundary. At and beyond the track error bounds, the piecewise quadratic guidance law has zero gradient with respect to vertical position, which causes the boundary value problems for those few multiple shooting nodes to consider flight path errors with no Jacobian feedback on how a change in vertical position may improve the cost. The result is then that the optimizer chooses the greater reduction in total cost settling the latter majority of nodes to the track, but leaving the beginning nodes with a, relatively, smaller flight path error cost.

In place of the quadratic formulation, asymptotic function such as the *arctangent* could potentially be used. However, in the interests of reducing trigonometric function usage wherever possible (attempting to make any CPU savings we can), a “quick and dirty” remedy while still using the quadratic formulation is to threshold the unit track error in the feedback Jacobian, such that there is always a small position dependence the optimizer can exploit.

## 2.4 Soft Exponential Huber Constraint Formulation

In Paper V, we formulated soft constraints as heavily weighted objectives. In particular, a piecewise quadratic formulation was used where, up to some threshold, no cost was applied to the state (thus no objective penalty), and beyond the threshold, the soft objective then increased quadratically (quartically in the least squares cost function). The concept being that soft penalization allows some foresight for the optimizer to gradually adjust for the bound, rather than running up against a constraint wall directly and executing aggressive (often oscillatory) commands to return to permitted state space. Particularly states like the angle of attack and airspeed for fixed-wing UAVs are good candidates for soft constraints, as:

1. We often want the optimizer to freely choose at what angle of attack or airspeed to operate, e.g. given range/endurance or climbing and sinking performance objectives which have dissimilar and often unknown ideal operating states.
2. These states are very susceptible to violating the constraint (even if briefly), which can lead to the general hard constraint issues listed above; or, if the optimizer does not find a feasible solution in the current state, we could be left for some iteration steps without new controls, possibly leading to a crash.

While the quadratic formulation was shown to work as an angle of attack constraint in flight tests from Paper V, we observed some small oscillatory behavior near the threshold at times (though much less than that of hard constraints). This is due to the discrete step in the objective Jacobian at that boundary.

To address this, we propose an exponential formulation for soft bounds which effectively goes to zero cost beyond some threshold, but still always maintains some gradient. However, a purely exponential cost can easily “blow up” the solver if operating near the constraint. We can take some inspiration from the machine



learning community to remedy this by formulating a one-sided, exponential version of the *Huber loss* [32], which appends a linear cost increase beyond the constraint in a piecewise fashion. Allowing the further reaches of our long horizon to occasionally violate some constraints momentarily with more relaxed gradients pushing us back.

We’ll call this soft bound an “exponential Huber constraint”:

$$\sigma = \begin{cases} 1 - k_\sigma y' & y' < 0 \quad (\text{linear}) \\ \exp(-y' k_\sigma) & \text{else} \quad (\text{exponential}) \end{cases} \quad (8.9)$$

where

$$k_\sigma = \log \sqrt{\frac{w_\sigma}{\sigma_1}} \quad (8.10)$$

and  $\sigma_1$  is the desired value of  $\sigma$  at  $y' = 1$ , assuming a weight of  $w_\sigma = 1$ . Note that the square root seen in (8.10) is present to account for the least squares formulation this output is processed through with weight  $w_\sigma$  applied post-squaring. Input to the soft constraint is normalized with respect to the constraint wall  $y_0$  and a buffer region  $\Delta y$ , wherein the exponential ramps up. The normalized input is then

$$y' = \frac{y - y_0}{\Delta y} \quad (8.11)$$

where  $y$  is the objective on which we are applying the soft constraint.

A graphical representation of the exponential Huber constraint can be seen in Fig. 8.3.

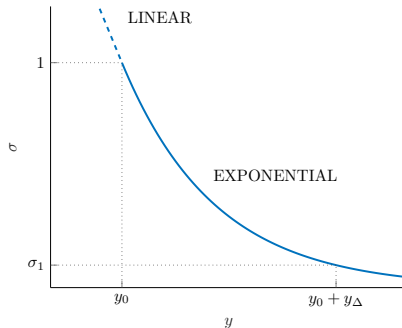


Figure 8.3: Exponential Huber constraint

### Implementation Note

The formulation in (8.9)-(8.10) achieves the desired cost at the constraint and end of the buffer region, however, is in actuality shaped quadratically in the linear portion once run through the least squares formulation. Simply setting the true huber loss as an objective formulation is disallowed in the ACADO framework, requiring the following formulation, if one desired a true linear progression beyond the constraint.

$$\sigma = \begin{cases} \sqrt{1 - k_\sigma y'} & y' < 0 \quad (\text{linear}) \\ \sqrt{\exp(-y'k_\sigma)} & \text{else} \quad (\text{exponential}) \end{cases} \quad (8.12)$$

where

$$k_\sigma = \log\left(\frac{w_\sigma}{\sigma_1}\right) \quad (8.13)$$

This formulation is not used here, as it adds unnecessary computations to both the cost and especially the Jacobians due to the square roots.

### Jacobian

To use the Huber constraint as a soft constraint on a variety of objectives, each with possibly different numbers of other state/control dependencies, we can construct a general Jacobian structure which can later be chained to any other objective specific Jacobian. In the following, we will assume the objective  $y(x)$ , the constraint  $y_0(x)$ , and the buffer region  $\Delta y(x)$  may all be functions of the aircraft state such that

$$\sigma(x) = f(y'(y(x), y_0(x), \Delta y(x))) \quad (8.14)$$

$$\frac{\partial \sigma}{\partial x} = \frac{\partial \sigma}{\partial y'} \left( \frac{\partial y'}{\partial y} \frac{\partial y}{\partial x} + \frac{\partial y'}{\partial y_0} \frac{\partial y_0}{\partial x} + \frac{\partial y'}{\partial \Delta y} \frac{\partial \Delta y}{\partial x} \right) \quad (8.15)$$

$$(\sigma)_x = (\sigma)_{y'} \left( (y')_y (y)_x + (y')_{y_0} (y_0)_x + (y')_{\Delta y} (\Delta y)_x \right) \quad (8.16)$$

where e.g. shorthand  $(y)_x = \frac{\partial y}{\partial x}$ .

From (8.9) it follows

$$(\sigma)_{y'} = \begin{cases} -k_\sigma & y' < 0 \quad (\text{linear}) \\ -k_\sigma \exp(-y'k_\sigma) & \text{else} \quad (\text{exponential}) \end{cases} \quad (8.17)$$

and from (8.11)

$$(y')_y = \Delta y^{-1} \quad (8.18)$$

$$(y')_{y_0} = -\Delta y^{-1} \quad (8.19)$$

$$(y')_{\Delta y} = -\frac{y - y_0}{\Delta y^2} \quad (8.20)$$

$$(8.21)$$

Returning to (8.16)

$$(\sigma)_x = (y\sigma)_{y'} \frac{(y)_x - (y_0)_x - y'(\Delta y)_x}{\Delta y} \quad (8.22)$$

where  $(y)_x$ ,  $(y_0)_x$ , and  $(\Delta y)_x$  must be provided for the given objective / state dependencies.

### 3 Terrain Interpretation from Vision-based Elevation Maps

In this section we describe the vision-based mapping approach taken in this work and how that map is interpreted as objective feedback for the NMPC optimization.

#### 3.1 Vision-based Mapping from Fixed-Wing Aircraft

Vision based elevation mapping, using the flexible stereo vision pipeline from [28], is employed to give our vehicle environmental awareness, on the fly. A densification module takes as input the undistorted images of the left, right, and center camera as well as the estimated center camera pose in world coordinates and the relative camera pose of the right and left camera. For each stereo pair, a depth map can be computed by rectifying the image and applying an efficient stereo block-matching (BM) algorithm. The depth maps are then integrated into the world coordinates by projecting every pixel of the depth map onto a 2.5D elevation map, following the approaches outlined in [15] and [16]. For every cell in the grid map, a height and covariance value is stored. With new measurements arriving, a Kalman Filter update scheme is applied as follows:

$$\hat{h}_k = \frac{\hat{h}_{k-1}\Sigma_{m,k} + h_{m,k}\Sigma_{k-1}}{\Sigma_{m,k} + \Sigma_{k-1}} \quad (8.23)$$

$$w_h^{-1} = \frac{\Sigma_{m,k} + \Sigma_{k-1}}{\Sigma_{m,k}\Sigma_{k-1}} \quad (8.24)$$

where  $\hat{h}_k$  and  $w_h^{-1}$  are the updated elevation estimate and its inverse uncertainty, respectively, with elevation and variance priors  $\hat{h}_{k-1}$ ,  $\Sigma_{k-1}$  and measurements  $h_{m,k}$ ,  $\Sigma_{m,k}$ .

While in this work, we will simulate the camera inputs in the Gazebo [41] and *RotorS* [20] simulation environment, an image of the practical implementation of this setup on the Techpod UAV is shown in Fig. 8.4.

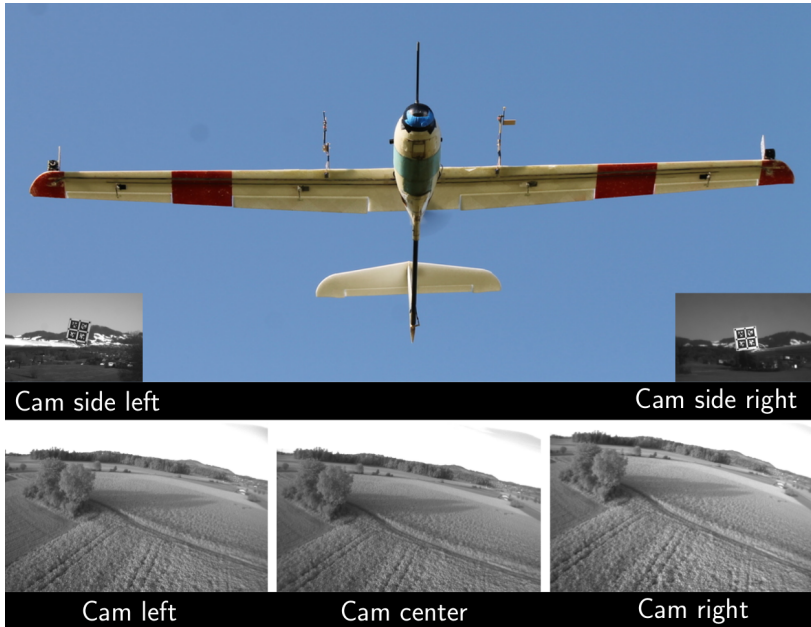


Figure 8.4: Flexible trinocular vision setup on the Techpod UAV. [28]

### 3.2 Occlusion Detection

From the elevation map, we can directly look-up terrain height at a given 2D position. Bilinear interpolation is used within the given cell to ensure continuous height measurements and further their Jacobian information with respect to position, which is fed back to the NMPC.

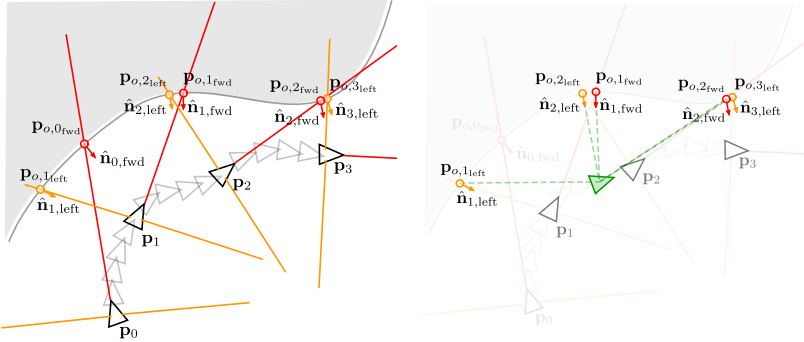
However, early on in our development we realized height feedback alone was not enough to provide safe avoidance of sharp increases in terrain (e.g. a wall, cliff, or steep hillside), as our aircraft is moving fast enough that even the long horizons will not pick up the elevation change in time to allow an avoidance maneuver. Increasing the horizon length could potentially remedy this issue, but we are already pushing the computation limits with the horizon size currently in use.

If we were to instead employ a 3D voxel map, e.g. using *voxblox* [63], we could also retrieve Euclidean Signed Distance Fields (ESDFs) which can be directly formulated into optimizable avoidance objectives. However, the processing time necessary for ESDF construction, as the wave propagation acts over the entire 3D map, may not be fast enough to provide the immediate distance feedback we require for fast

moving fixed-wing UAVs. Other works have taken points directly from the point cloud to check their trajectories against, though even when down sampling the relevant points, the set grows quite large and further poses the issue *how* to down sample without missing obstacles, as described in [2].

Here, we use efficient 2D ray casting, common in computer graphics, through our existing elevation map from various origins close to the aircraft's current position to detect upcoming occlusions by checking for triangle-ray intersections in any grid cells that rise above the ray's height. The idea being that the point cloud has already been considered in the generation of the elevation map, and we can assume (at some larger discretization) that all obstacles are contained within this representation.

To focus our search space (and computational efforts) on the most relevant regions for the vehicle, rays are cast from NMPC horizon nodes in the direction of each node's current ground velocity, as well as perpendicular (left and right) to that vector. The length of each ray corresponds to the inertial velocity of the node; a function of ground speed for the "forward" ray, and a fixed distance for the lateral rays. Using this scheme, our aim is to provide the NMPC with potential upcoming terrain collisions in the direction that the optimizer is current planning to fly. To save on computational requirements, we cast rays at a defined interval of the horizon nodes (the ray casting interval). The subset of horizon nodes used for ray casting are called *ray casting nodes*. Detected occlusions are stored in a short time buffer (with corresponding occlusion buffer length), such that the information is not immediately lost, but will timeout when we presumably move on from that area or they are replaced by new detections. Figure 8.5 shows an illustration of the ray casting and occlusion detection concept.



**Figure 8.5:** Ray casting and occlusion summing scheme. (Left) Forward, left, and right rays cast from the vehicle at each ray casting node ( $p_0$ - $p_3$ ). (Right) For a specific horizon node, detected occlusions from the nearest ray casting nodes up to the occlusion window length are considered for RESDF construction.

The next sections will detail how we translate these occlusions into an optimizable objective for the NMPC.

Before moving on, we note this general concept of limited ray casting is not particularly new. In [23], two point lidars are oriented out the side of a small delta wing vehicle, measuring the distance from canyon walls to the aircraft and adapting their waypoints accordingly to keep a stand off distance. However, one of the largest differences between our approach and other vision (or lidar) -based reactive avoidance approaches is that we have much more information at our disposal from the stored vision populated elevation map. We are able to not only *react* to the current depth or laser measurement, but actively *plan* due to our horizon’s prediction of where the vehicle will go in the future, even as far as full turn arounds, assuming we have previously mapped some area behind us. Additionally each detection in our elevation map provides 3D slope information we can use as gradient feedback for the avoidance objective.

### 3.3 Local “Relative” Euclidean Signed Distance Fields

The occlusion detection in the last section provides us with a small “point cloud” of the most immediately relevant upcoming terrain. The triangle intersections further provided a terrain normal for each point.

However, as elaborated in the previous sections, our optimizer needs a continuous representation of the environment. While ESDFs would provide an ideal avoidance objective format, we have found that distance metrics alone are not sufficient to drive our high speed vehicle away from obstacles. Here, we develop a new concept of “Relative” ESDFs (RESDFs), which are modulated by the aircraft speed and direction. An illustration of an ESDF and corresponding RESDF for a 2D grid is shown in Fig. 8.6.

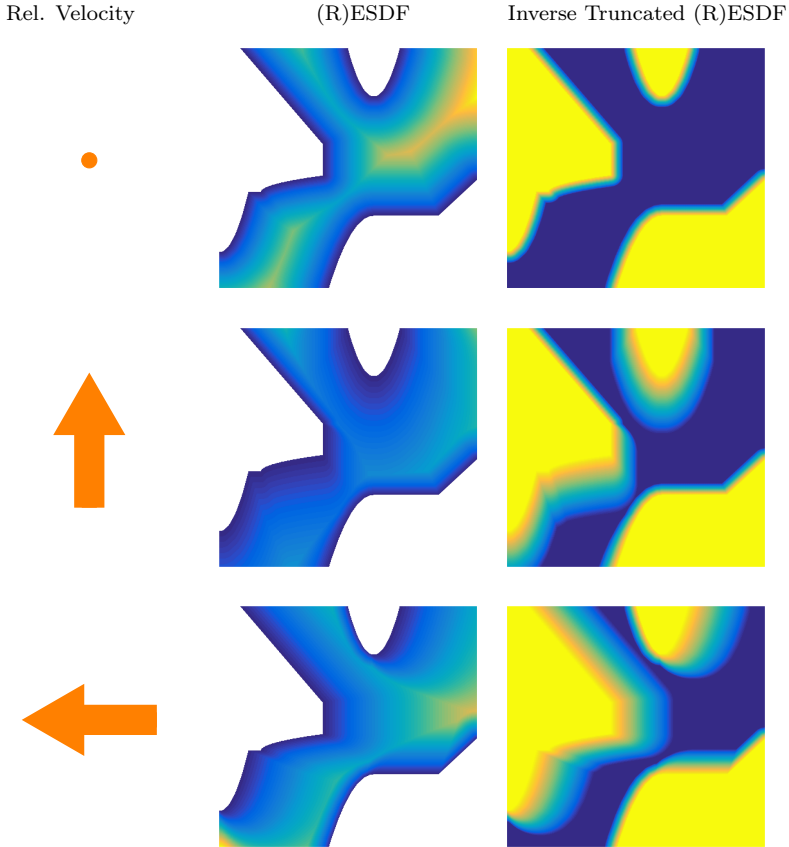
#### Constructing Local RESDFs

Occlusions can be represented either purely as points  $\mathbf{p}_o$  or, to utilize the terrain gradient information included with each occlusion, as *surfels*, i.e. points with a small surface of radius  $R_s$  about their normal vector  $\hat{\mathbf{n}}_o$ . We define the radial terrain distance (RTD)  $d$  and “relative” radial terrain distance  $d'$ .

$$d'(\mathbf{p}, \mathbf{v}) = \frac{d(\mathbf{p}) - d_0(\mathbf{p}, \mathbf{v})}{\Delta d(\mathbf{p}, \mathbf{v})} \quad (8.25)$$

where distance constraint  $d_0$  and buffer region  $\Delta d$  are functions of both position and velocity. Then,

$$d(\mathbf{p}) = \begin{cases} \|\mathbf{r} - R_s \frac{\mathbf{r}_n}{\|\mathbf{r}_n\|}\| & \|\mathbf{r}_n\| > R_s \quad (\text{point}) \\ \|\text{proj}_{\hat{\mathbf{n}}_o} \mathbf{r}\| & \text{else} \quad (\text{surface}) \end{cases} \quad (8.26)$$



**Figure 8.6:** (R)ESDFs for an arbitrary 2D occupancy grid. White space in the ESDFs correspond to occupied space and distance from the nearest wall is presented from darker to lighter colors. The inverse truncated fields flip the color spectrum and illustrate how an optimizer may view such a field, proximity to the walls incurring higher costs. RESDFs show how the field is modulated for various relative velocities, showcasing “relative” distances, as opposed to actual distances.

where  $\mathbf{r} = \mathbf{p}_o - \mathbf{p}$ , and the projection of  $\mathbf{r}$  onto the surfel plane is

$$\mathbf{r}_{||} = \mathbf{r} - \text{proj}_{\hat{\mathbf{n}}_o} \mathbf{r} \quad (8.27)$$

where

$$\text{proj}_{\hat{\mathbf{n}}_o} \mathbf{r} = (\mathbf{r} \bullet \hat{\mathbf{n}}_o) \hat{\mathbf{n}}_o \quad (8.28)$$

Note in the point description of occlusions, we still distinguish whether the vehicle is “in front” or “behind” the point, corresponding to the surface normal, to neglect consideration of points if the NMPC horizon is potentially momentarily inside the terrain. Defining a surface within a limited range of the occlusion point also helps to smooth out the resultant RESDF from our sparse detections by using the terrain normal information. Sizing should be consistent with the elevation map discretization.

The dynamic constraint  $d_0(\mathbf{p}, \mathbf{v})$  and delta  $\Delta d(\mathbf{p}, \mathbf{v})$  are designed as

$$d_0(\mathbf{p}, \mathbf{v}) = d_{0,v=0} + k_{d,0} \frac{v_{\text{rel}}(\mathbf{p}, \mathbf{v})^2}{g \tan \phi_{\text{max}}} \quad (8.29)$$

$$\Delta d(\mathbf{p}, \mathbf{v}) = \Delta d_{v=0} + k_{d,\Delta} \frac{v_{\text{rel}}(\mathbf{p}, \mathbf{v})^2}{g \tan \phi_{\text{max}}} \quad (8.30)$$

where  $d_{0,v=0}$  and  $\Delta d_{0,v=0}$  are the desired nominal terrain offset and delta, respectively, at zero speed.  $k_{d,0}$  and  $k_{d,\Delta}$  are multiples of the minimum turn radius  $R_{\text{min}}$  w.r.t. the occlusion at the current relative speed  $v_{\text{rel}}$ .

$$R_{\text{min}} = \frac{v_{\text{rel}}(\mathbf{p}, \mathbf{v})^2}{g \tan \phi_{\text{max}}} \quad (8.31)$$

$$v_{\text{rel}} = \begin{cases} \mathbf{v}_G \bullet \hat{\mathbf{d}} & v_{\text{rel}} \geq 0 \\ 0 & \text{else} \end{cases} \quad (8.32)$$

A sample based approach is used to construct the local RESDF at a given aircraft position from the set of occlusions, where the minimum relative terrain distance out of the samples is then applied.

### Translating RESDFs to Optimization Feedback

At this point, one may notice a similarity between the relative terrain distance  $d'$  formulation and the normalized Huber constraint input  $y'$  from Section 2.4. This relation can be exploited to directly translate the relative terrain distance into an optimizable cost for the controller.

The full cost (using the Huber formulation) is calculated at each sample, still taking the minimum relative distance (maximum cost) of all the samples, but evaluating a weighted (by the sample costs) sum of all sample Jacobians. Note that if we took only the Jacobians corresponding to the minimum relative distance at each time step, the feedback could become very noisy. The resulting average Jacobian is then representative of the sampled cost field, and somewhat “filtered” over time by the averaging operation.

Computational cost can be further reduced by using a less expensive operation for



the exponential component of the Huber constraint, in fact reverting to the original quadratic Huber formulation. A ray must be cast to obtain each sample and for computational efficiency, is only extended slightly more than the corresponding speed dependent constraint and delta terms. This effectively provides us with a truncated RESDF. As the truncated field anyway loses gradient information beyond the delta boundary, in this case, the quadratic Huber formulation is no worse than the exponential.

The sample quadratic Huber constraint cost for RTD takes the form:

$$\sigma_d = \begin{cases} 1 - 2d' & d' < 0 \quad (\text{linear}) \\ (1 - d')^2 & d' < 1 \quad (\text{quadratic}) \\ 0 & \text{else} \end{cases} \quad (8.33)$$

and corresponding sample Jacobian:

$$(\sigma_d)_{d'} = \begin{cases} -2 & d' < 0 \quad (\text{linear}) \\ -2(1 - d') & d' < 1 \quad (\text{quadratic}) \\ 0 & \text{else} \end{cases} \quad (8.34)$$

Finally, the sample Jacobians are combined in a weighted sum:

$$(\bar{\sigma}_d)_x = \frac{\sum_i^{n_{\text{samples}}} (\sigma_d)_{x,i} \sigma_{d,i}}{\sum_i^{n_{\text{samples}}} \sigma_{d,i}} \quad (8.35)$$

where  $\sum_i^{n_{\text{samples}}} \sigma_{d,i} \neq 0$  and  $(\bar{\sigma}_d)_x$  is the (weighted) average Jacobian of the Huber constraint cost with respect to an arbitrary state  $x$ .

A pseudo algorithm for the objective formulation is given in Algorithm 2. Descriptions of several settings related to the occlusion detector and RESDF construction algorithm are summarized in Table 8.1.

**Remark.** *Why use a soft constraint on radial terrain distance? Primarily, we run into a similar issue as noted in Section 2.4 for airspeed or angle of attack, that we fully expect occlusions to be popping up where they previously weren't within the latter nodes of the horizon which may at first violate our constraints. Using a soft constraint heavily penalizes this occurrence, but allows the optimizer a more relaxed path back to safe spaces.*

In the end, we end up with an output somewhat similar to what voxblox would have given, but something of a “poor man’s” implementation, where only the local distance field is derived within a “slice” on the plane created by our current horizon and radial ray projections. Further, our RTD cost is terrain relative velocity dependent, which is not feasible to construct within an incremental ESDF builder.

**Algorithm 2** Radial terrain distance objective cost construction. This process is run at each NMPC horizon node.

---

```

for all occlusion detections  $\mathbf{p}_o$  in occlusion window and buffer do
   $\mathbf{r} = \mathbf{p}_o - \mathbf{p}$ 
  if  $(\mathbf{r} \bullet \hat{\mathbf{n}}_o) < 0$  then
     $\mathbf{d} = \text{proj}_{\hat{\mathbf{n}}_o} \mathbf{r}$ 
     $\mathbf{r}_{||} = \mathbf{r} - \mathbf{d}$ 
    if  $\|\mathbf{r}_{||}\| > R_s$  then
      recalculate:  $\mathbf{d} = \mathbf{r} - R_s \frac{\mathbf{r}_{||}}{\|\mathbf{r}_{||}\|}$ 
      Calculate distance norm and Jacobian (without surface normal dependency)
    else
      Calculate distance norm and Jacobian (with surface normal dependency)
    end if
  end if
end for
Take the maximum objective cost from all samples.
Calculate weighted average of all sample cost Jacobians.

```

---

**Table 8.1:** Occlusion detection parameters.

| Setting                 | Description   |
|-------------------------|---|
| Ray casting interval    | The interval between horizon nodes from which rays are cast (defining the <i>ray casting nodes</i> ).   |
| Occlusion window length | The number of ray casting nodes included in the RESDF construction for a given horizon node. The window is centered at the horizon node in consideration. The nearest ray nodes, adding to the occlusion window length, are included. |
| Occlusion buffer length | The number of past occlusion detections per ray casting node which are stored. The buffer is FIFO, the oldest values getting replaced either with new information or when timed out.  |

We note the velocity relative augmentation of the ESDF presented here is similar to the “morphing potential field” concept proposed in [80] and extended to 3D in [81]. However, our approach applies to generalized terrain of any shape, as oppose to simple shapes like spheres or cylinders.

### 3.4 Filtering Terrain Feedback

As the elevation map will be constantly changing with newer vision updates arriving every instant and terrain may be inherently discrete, the incoming data needs to be processed before reaching the optimizer. The short buffer of previous occlusion detections used within the cost and gradient sums for the RESDFs helps in filtering out some of the terrain noise. However, an additional first order time delay  $\tau_{\text{terr}}$  is included on the incoming cost and Jacobian calculations for both the HAG and RTD.

## 4 NMPC Problem Definition

This section formalizes the model, objectives, and constraints of the NMPC.

### 4.1 Model

The model used for control in this brief mostly follows that of the one presented in Paper V, though with some adjustments, in particular towards reducing the order of the model, which we will describe here.

States  $\mathbf{x} = [p_n, p_e, p_d, v_A, \gamma, \xi, \phi, \theta, n_p]$  are propagated within the MPC, driven by controls  $\mathbf{u} = [u_T, \phi_{\text{ref}}, \theta_{\text{ref}}]$  using the following differential equations:

$$\dot{p}_n = v_A \cos \gamma \cos \xi + v_{W,n} \quad (8.36)$$

$$\dot{p}_e = v_A \cos \gamma \sin \xi + v_{W,e} \quad (8.37)$$

$$\dot{p}_d = -v_A \sin \gamma + v_{W,d} \quad (8.38)$$

$$\dot{v}_A = \frac{1}{m} (T \cos \alpha - D) - g \sin \gamma \quad (8.39)$$

$$\dot{\gamma} = \frac{1}{m v_A} [(T \sin \alpha + L) \cos \phi - mg \cos \gamma] \quad (8.40)$$

$$\dot{\xi} = \sin \phi \frac{T \sin \alpha + L}{m v_A \cos \gamma} \quad (8.41)$$

$$\dot{\phi} = \frac{k_\phi \phi_{\text{ref}} - \phi}{\tau_\phi} \quad (8.42)$$

$$\dot{\theta} = \frac{k_\theta \theta_{\text{ref}} - \theta}{\tau_\theta} \quad (8.43)$$

$$\dot{n}_p = \frac{u_n - n_p}{\tau_n} \quad (8.44)$$

where the state vector contains 3D inertial position  $\mathbf{p}$ , air mass relative airspeed  $v_A$ , flight path angle  $\gamma$ , and heading  $\xi$ , roll and pitch angles  $\phi$  and  $\theta$ , and propeller speed  $n_p$ .

Aerodynamic and thrusting forces are provided in equations (8.45)- (8.47):

$$T = \rho n_p^2 D_p^4 \left( c_{T_0} + c_{T_1} \frac{v_p}{n_p D_p} \right) \quad (8.45)$$

$$D = \bar{q} S \left( c_{D_0} + c_{D_\alpha} \alpha + c_{D_{\alpha^2}} \alpha^2 \right) \quad (8.46)$$

$$L = \bar{q} S \left( c_{L_0} + c_{L_\alpha} \alpha \right) \quad (8.47)$$

where  $\rho$  is the density of dry air,  $\bar{q} = 0.5\rho v_A^2$  is the dynamic pressure,  $S$  is the wing planform area,  $D_p$  is the diameter of the propeller, and  $[c_{T_0}, c_{T_1}]$ ,  $[c_{D_0}, c_{D_\alpha}, c_{D_{\alpha^2}}]$ , and  $[c_{L_0}, c_{L_\alpha}]$  are identifiable coefficients for thrust, drag, and lift. As in Paper V, aerodynamic and thrusting moments are neglected, maintaining the quasi-steady assumption from the control augmented model formulation.

## Thrust

A change in our model, from Paper V, can be seen in (8.45) and (8.44). The propeller thrust model in (8.45) approximates the axial force as a function of the advance ratio (c.f. [77]). Normal propeller force is neglected.

In (8.44),  $n_p$  refers to the propeller speed (here Techpod is a single pusher propeller) in revolutions per second. We use a first order spool-up lag from the commanded propeller speed input  $u_n$ . As the PX4 operates on normalized throttle settings  $u_T \in [0, 1]$ , a mapping can be generated between  $u_T$  and  $u_n$ :

$$u_n = \left( n_{0,\min} + u_T (n_{\max} - n_{0,\min}) \right) (1 - \sigma_{v_p}) + \left( n_{0,\max} + u_T (n_{\max} - n_{0,\max}) \right) \sigma_{v_p} \quad (8.48)$$

where  $v_p$  is the axial component of the incoming airflow, simplified here to be  $v_p = v_A \cos(\alpha - \epsilon_T)$ ,  $\epsilon_T$  is the propeller axis offset from the body x-axis. Propeller speeds  $n_{0,\min}$  and  $n_{0,\max}$  are the “zero-thrust” values for a user defined minimum and maximum incoming flow speeds  $v_{p,\min}$  and  $v_{p,\max}$ , respectively. These values are backed out of (8.45), post identification of the thrust coefficients.

$$n_{0,\min} = \frac{-c_{T_1} v_{p,\min}/D_{\text{prop}} + \sqrt{(c_{T_1} v_{p,\min}/D_{\text{prop}})^2 - 4c_{T_0}}}{2c_{T_0}} \quad (8.49)$$

$$n_{0,\max} = \frac{-c_{T_1} v_{p,\max}/D_{\text{prop}} + \sqrt{(c_{T_1} v_{p,\max}/D_{\text{prop}})^2 - 4c_{T_0}}}{2c_{T_0}} \quad (8.50)$$

$\sigma_{v_p}$  transitions the mapping scale from the propeller speed range at minimum and maximum flow speeds;  $n_{\max}$  being the maximum propeller speed the motor can

command (measured statically at zero flow on the ground).

$$\sigma_{v_p} = \frac{v_p - v_{p,\min}}{v_{p,\max} - v_{p,\min}} \quad (8.51)$$

The advantage of such a mapping becomes clear considering the control constraint formulation, where the NMPC commands  $u_T$  (as to directly feed it to the PX4), and may be limited by simple static bounds of 0 and 1. The mapping is then implemented within the model such that controller is still aware of the speed dependence on thrust force, but additional evaluation of dynamics on the constraints are then not necessary.

### First Order Attitude Response Dynamics

An additional change is the removal of body angular rates, previously present in the model from Paper V. In their place, we use simple first order control augmented models for the attitude response dynamics (8.42)- (8.43). As we wish to add several new objectives to the controller in this brief, any computational cost savings we can make will help.

Though having angular rates available for damping feedback was beneficial in Paper V, we found that the additional prediction errors introduced by using first order approximations were small enough to neglect these states, so long as sufficient trim elevator scheduling per airspeed (and if possible roll angle) is implemented within the low-level attitude controller. Examples of the first order control augmented attitude response formulation were shown laterally in Paper IV as well as in a newer work we conducted on NMPC for fixed-wing airborne wind energy [83]. The identification process remains nearly identical to that proposed in Paper V, here with a reduced parameter set  $[k_\phi, \tau_\phi, k_\theta, \tau_\theta]$ .

## 4.2 Objectives

The objective vector consists of several state reference tracking terms, control penalization, and soft constraints:

$$\mathbf{y} = [p_n, p_e, v_A, e_\gamma, \xi, \sigma_{v_A}, \sigma_\alpha, \sigma_h, \sigma_d, u_T, \phi_{\text{ref}}, \theta_{\text{ref}}]^T \quad (8.52)$$

with objective reference vector

$$\mathbf{y}_{\text{ref}} = [p_{n,\text{ref}}, p_{e,\text{ref}}, v_{A,\text{ref}}, 0, \xi_{\text{ref}}, 0, 0, 0, 0, u_{T,\text{trim}}, \phi_{\text{ff}}, \theta_{\text{trim}}]^T \quad (8.53)$$

where end term objective  $\mathbf{y}_N = \mathbf{y}_{1-9}$  and end term objective reference  $\mathbf{y}_{N,\text{ref}} = \mathbf{y}_{\text{ref}_{1-9}}$ .

### Reference Trajectories

Lateral-directional trajectory generation (see Section 2.2) provides  $p_{n,\text{ref}}$ ,  $p_{e,\text{ref}}$ ,  $v_{A,\text{ref}}$ , and  $\phi_{\text{ff}}$ , with  $v_{A,\text{ref}}$  constrained between  $v_{A,\text{nom}}$  and  $v_{A,\text{max}}$ . Longitudinal guidance (Section 2.3) is embedded into the objective model, producing flight path error  $e_\gamma = \gamma_{\text{ref}} - \gamma$  internally.

### Soft Constraints

$\sigma_{v_A}$  is a lower bounded exponential Huber constraint on airspeed  $v_A$  with lower bound  $v_{A,0}$ .  $\sigma_\alpha$  is an upper and lower bounded exponential Huber constraint on the angle of attack  $\alpha$  with upper constraint  $\alpha_+$ , lower constraint  $\alpha_-$ .  $\sigma_h$  is a lower bounded exponential Huber constraint on the HAG, and  $\sigma_d$  is a lower bounded quadratic Huber constraint on RTD.

### Control Penalization

Control signals  $u_T$  and  $\theta_{\text{ref}}$  are regulated towards fixed trims  $u_{T,\text{trim}}$  and  $\theta_{\text{trim}}$ , respectively. Additional *hard* upper and lower constraints are applied to the NMPC controls  $u_T$ ,  $\phi_{\text{ref}}$ ,  $\theta_{\text{ref}}$ .

## 4.3 Objective Weight Prioritization

To organize the focus of the NMPC on the many competing objectives we task it to optimize, we prioritize several objectives by diminishing their weights relative to others when close to terrain. Towards this end, the positive definite weighting matrix  $\mathbf{Q}(t)$  is variable over the horizon length, varied from nominal weighting  $\mathbf{Q}_0$ .

Inverse priorities for HAG and RTD are defined as functions of their respective normalized objectives (pre-weighting):

$$\sigma_{h,\text{prio}} = \text{constrain}(h', 0, 1) \quad (8.54)$$

$$\sigma_{d,\text{prio}} = \text{constrain}(d', 0, 1) \quad (8.55)$$

where the term ‘‘inverse’’ implies that the objective has higher priority when the value approaches zero. The inverse priority product at each horizon node is used for de-weighting

$$\sigma_{\text{prio}}(t) = \sigma_{h,\text{prio}}(t)\sigma_{d,\text{prio}}(t) \quad (8.56)$$

where individual weights

$$q_{p_n}(t) = \sigma_{\text{prio}}(t)q_{0,p_n} \quad (8.57)$$

$$q_{p_e}(t) = \sigma_{\text{prio}}(t)q_{0,p_e} \quad (8.58)$$

$$q_\gamma(t) = \sigma_{\text{prio}}(t)q_{0,\gamma} \quad (8.59)$$

$$q_\xi(t) = \sigma_{\text{prio}}(t)q_{0,\xi} \quad (8.60)$$

and objective reference

$$y_{\text{ref},v_A}(t) = v_{A,\text{ref}}(t)\sigma_{\text{prio}}(t) + (1 - \sigma_{\text{prio}}(t))v_{A,\text{ref}}(t) \quad (8.61)$$

$$(8.62)$$

Weight prioritization occurs during the *preparation step* of the NMPC iteration loop. First, all objectives and Jacobians are calculated on each horizon node given the current measured state and previous state horizon. Then, we scale the above weights as necessary before proceeding to the *feedback step*, where the multiple shooting nodes are linearized and forwarded to the solver (see more details on the optimizer in the next section).

## 4.4 Optimization

As in Papers IV and V, we use the ACADO Toolkit [30] for the generation of a fast C code based nonlinear solver and implicit fourth order Runge-Kutta integration scheme. A direct multiple shooting technique is used to solve the optimal control problem (OCP), where dynamics, control action, and inequality constraints are discretized by  $T_{\text{nmmpc}}$  over a time grid of a given horizon length  $N_{\text{nmmpc}}$ . A boundary value problem is solved within each interval and additional continuity constraints are imposed. Sequential Quadratic Programming (SQP) is used to solve the individual QPs, using the active set method implemented in the *qpOASES*<sup>2</sup> solver.

The OCP takes the continuous time form:

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{u}} \int_{t=0}^T & \left( (\mathbf{y}(t) - \mathbf{y}_{\text{ref}}(t))^T \mathbf{Q}(t) (\mathbf{y}(t) - \mathbf{y}_{\text{ref}}(t)) \right) dt \\ & + (\mathbf{y}_N(T) - \mathbf{y}_{N,\text{ref}}(T))^T \mathbf{Q}_N(T) (\mathbf{y}_N(T) - \mathbf{y}_{N,\text{ref}}(T)) \end{aligned} \quad (8.63)$$

$$\begin{aligned} \text{subject to } \dot{\mathbf{x}} &= f(\mathbf{x}, \mathbf{u}), \\ \mathbf{u}(t) &\in \mathbb{U}, \\ \mathbf{x}(0) &= \mathbf{x}(t_0) \end{aligned}$$

## 5 Hardware-in-the-Loop Simulation

### 5.1 Simulation Environment

Critical for initial testing of dangerous flight operations (such as the proposed near-terrain operation) is a high fidelity simulation environment capable of mimicking real-world sensors and vehicle models. The employed flight simulation framework is based on the Gazebo multi-robot simulator [41] and *RotorS* [20].

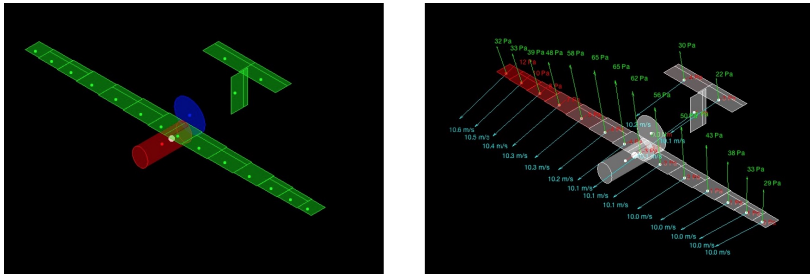
*RotorS* implements the basic functionality required to simulate multi-rotor and fixed-wing Micro Aerial Vehicles (MAV)s within the Gazebo simulator. It provides propulsion-, lift-/drag- and sensor models that can be added to the simulated vehicle

<sup>2</sup><http://www.qpOASES.org/>

in order to generate aerodynamic forces/torques and stream sensor signals based on the simulated state, respectively. *RotorS'* aerodynamics have lately been extended to increase fidelity, to allow for a modular approach to simulating aerodynamic wrenches and to model the interaction between propulsive units and airfoils.

### Fixed-Wing Aerodynamics/Propulsion

In the selected simulation approach, the aircraft is composed of multiple elements which are categorized as either 'airfoil', 'body' or 'propeller'. Aerodynamic forces and -torques are modeled independently for every element and applied to the aircraft at the element's center of pressure. The only exception to the independence of elements is given by the modeled interaction between the slipstream generated by propulsive elements and elements affected thereof. Fig.8.7 illustrates the partitioning of the Techpod UAV into multiple elements as well as exemplary aerodynamic forces generated by each of them.



**Figure 8.7:** Element-based composition of the aerodynamic model for the Techpod UAV. Left: Airfoil segments are shown in green, the single body element in red and the propeller in blue. The points indicate each element's center of pressure at which the aerodynamic forces and torques are resolved based on the local inflow condition, c.f. right fig. The red colored airfoil segments indicate stalled airflow, here due to excessive rolling.

Overall, this approach allows for a flexible and modular 'bottom-up' composition of an aerodynamic model which is, furthermore, computationally lightweight and therefore suited for real-time simulation in, e.g., a hardware-in-the-loop (HITL) setup.

It is important to note that the presented approach neglects the mutual influence between airfoils and/or bodies. Hence, effects such as e.g. induced drag or wing-fuselage interference do not implicitly emerge. Instead, they need to be included via corrections in the model's underlying aerodynamic parameters. More sophisticated lifting-line or vortex-lattice/panel methods are able to model these effects to some extent but are less modular and, more importantly, can get computationally

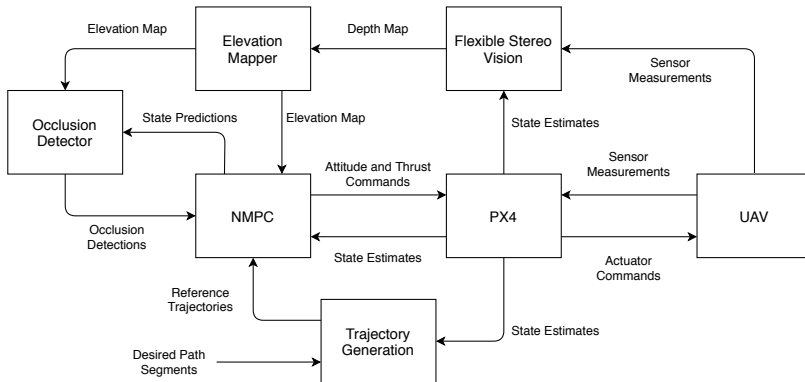


prohibitive when considering a potential real-time simulation requirement.

An actuator-level, nonlinear time-domain system identification from flight data was conducted in order to compare and “tailor” the simulated Techpod model towards a closer match to the dynamics we see in flight. Though it should be noted that the “lumped sum” identification parameters typically observable in common flight path reconstruction techniques are not trivially comparable to the more detailed *RotorS* model used here. Consequently, the simulation model can only be said “qualitatively” comparable to the real-world Techpod UAV, but is sufficiently high fidelity for our purposes.

## System Overview

In the results that follow, low-level control (attitude/rates) and state estimation are performed on a PX4 Autopilot, fed by (artificially noisy) simulations of sensor feedback from the Gazebo/*RotorS* simulation environment. Images of the Gazebo world from the wing tips are fed to the vision pipeline at ca. 10 Hz. The elevation grid map is discretized to 5 m cells. Occlusion detection, trajectory generation, and the NMPC iteration step are performed single thread, sequential process run at 10 Hz. A general overview of the full system which feeds (and is fed by) the NMPC is given in Fig. 8.8.



**Figure 8.8:** Full system diagram containing vision-based sensor input, elevation mapping, occlusion detection, trajectory generation, nonlinear model predictive control, and low-level attitude control and state estimation (running on the PX4 autopilot).

As this research is in its preliminary stages, all high-level control (NMPC) and vision algorithms are running in ROS (in tandem with the real-time Gazebo simulation environment) on a Lenovo Thinkpad P-50 laptop (2.8-GHz Intel Xeon E3-

1505M v5 CPU, 16GB RAM). Performance checks on smaller computer platforms such as the Intel Up Squared (2.4GHz CPU, 8GB RAM) for operation on-board the Techpod UAV are ongoing, e.g. determining maximum bounds on the MPC horizon length and occlusion detector parameters as well as optimization of the vision pipeline.

## 5.2 Results & Discussion

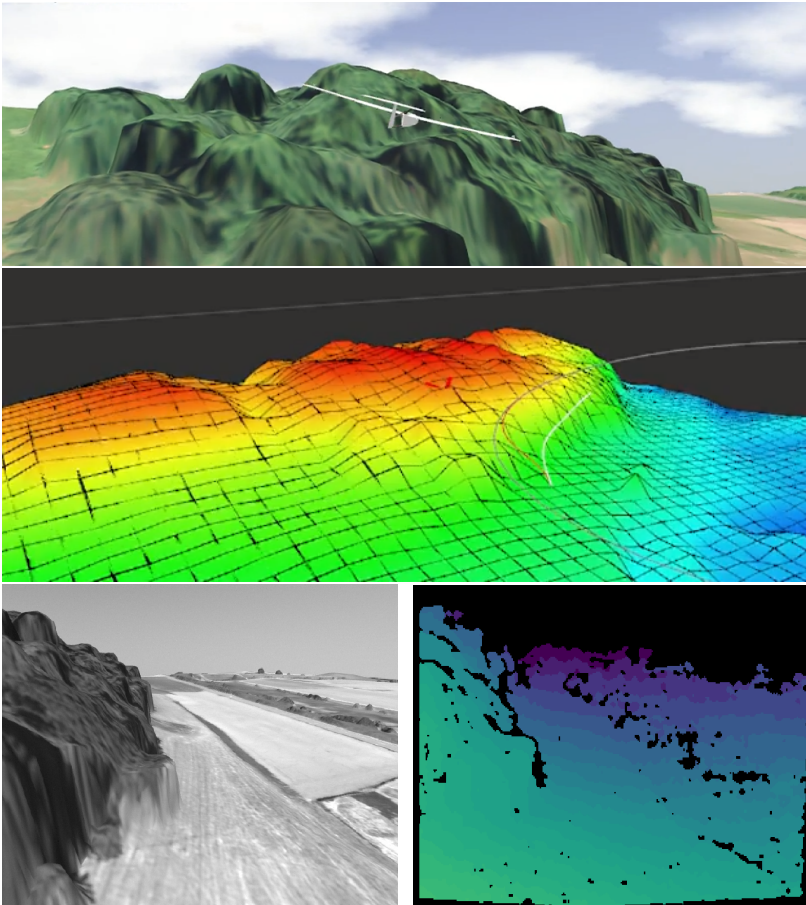
To evaluate the functionality of the terrain feedback objective developed in this brief, a scenario was set up in a Gazebo world map produced from a hill side in Hinwil, Switzerland where the Techpod is commanded to follow a “poorly planned” loiter path which nearly intersects with the elevated terrain. Screenshots from the simulated hillside, resulting vision-based elevation map, a snapshot of the left wing camera and corresponding disparity map (generated comparing the left and right wing tip cameras) are shown in Fig. 8.9. Parameters and weights used for the NMPC and guidance (trajectory generation) are tabulated in Table 8.2. Settings for the occlusion detector listed in Table 8.3.

**Table 8.2:** Control and guidance parameters used for HITL simulation. Nominal end term weights are identical to their corresponding nominal horizon weights,  $\mathbf{Q}_{0,N} = \mathbf{Q}_{0,1-9}$ .  $\sigma_1 = 0.001$  for all soft constraints. All lateral-directional guidance parameters not listed are identical to those in Paper III.

| NMPC Weights          |           | Soft Constraints     |                        | Misc.                             |                       |
|-----------------------|-----------|----------------------|------------------------|-----------------------------------|-----------------------|
| Objective             | $q_{0,y}$ | Param                | Value                  | Param                             | Value                 |
| $p_n$                 | 20        | $v_{A,0}$            | $12 \text{ m s}^{-1}$  | $v_{A,\text{nom}}$                | $14 \text{ m s}^{-1}$ |
| $p_e$                 | 20        | $\Delta v_A$         | $1.5 \text{ m s}^{-1}$ | $v_{A,\text{max}}$                | $18 \text{ m s}^{-1}$ |
| $v_A$                 | 300       | $\alpha_+$           | $7^\circ$              | $u_{T,\text{trim}}$               | 0.4                   |
| $\gamma$              | 900       | $\alpha_-$           | $-6^\circ$             | $\theta_{\text{ref},\text{trim}}$ | $1^\circ$             |
| $\xi$                 | 300       | $\Delta\alpha$       | $3^\circ$              | $u_{T,\text{constr}}$             | $[0, 1]$              |
| $\sigma_{v_A}$        | $10^8$    | $h_0$                | 10 m                   | $\phi_{\text{constr}}$            | $[-35, 35]^\circ$     |
| $\sigma_\alpha$       | $10^8$    | $\Delta h$           | 20 m                   | $\theta_{\text{constr}}$          | $[-15, 25]^\circ$     |
| $\sigma_h$            | $10^7$    | $d_{0,v=0}$          | 10 m                   | $\gamma_{\text{climb}}$           | $15^\circ$            |
| $\sigma_d$            | $10^7$    | $\Delta d_{v=0}$     | 20 m                   | $\gamma_{\text{sink}}$            | $-11^\circ$           |
| $u_T$                 | 70        | $k_{d,0}$            | 0.5                    | $e_{b,\text{lon}}$                | 1 m                   |
| $\phi_{\text{ref}}$   | 100       | $k_{d,\Delta}$       | 1                      | $N_{\text{nmpc}}$                 | 50                    |
| $\theta_{\text{ref}}$ | 400       | $\tau_{\text{terr}}$ | 0.5 s                  | $T_{\text{nmpc}}$                 | 0.1 s                 |

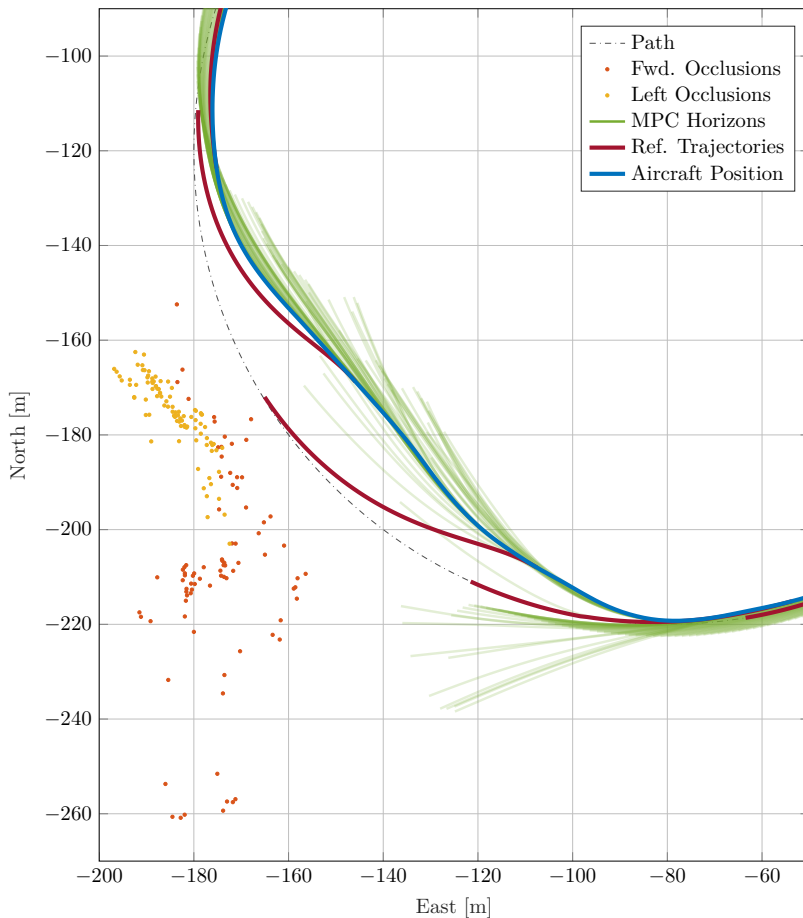
**Table 8.3:** Occlusion detection settings used in HITL simulation.

| Setting                 | Value |
|-------------------------|-------|
| ray casting interval    | 5     |
| occlusion window length | 10    |
| occlusion buffer length | 5     |
| surfel radius           | 5 m   |



**Figure 8.9:** Screenshots during an avoidance maneuver from the Gazebo simulated hillside (top), resulting vision-based elevation map, displayed in rviz (second from top), a snapshot of the left wing camera (bottom left), and corresponding disparity map (generated comparing the left and right wing tip cameras) (bottom right). The white and red trajectories shown in the rviz screenshot are the MPC predicted state and reference state trajectories, respectively. Note the predicted horizon states veer from the path (gray circle) as the controller is actively avoiding the terrain.

In Fig. 8.10 the aircraft's position trajectory as well as MPC state horizons, position reference trajectories, and the detected occlusions on the hillside throughout the maneuver are shown. As the aircraft approaches the hill, the first occlusions

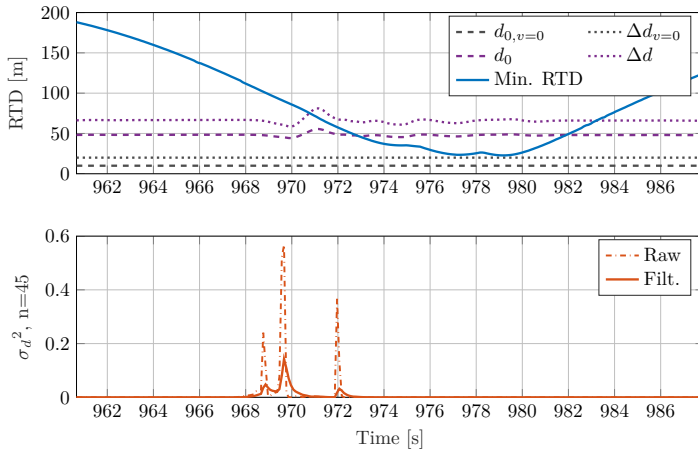


**Figure 8.10:** Avoidance maneuver; aircraft position, reference trajectories, predicted horizons, and all occlusion detections made throughout the maneuver by either the forward or left ray.

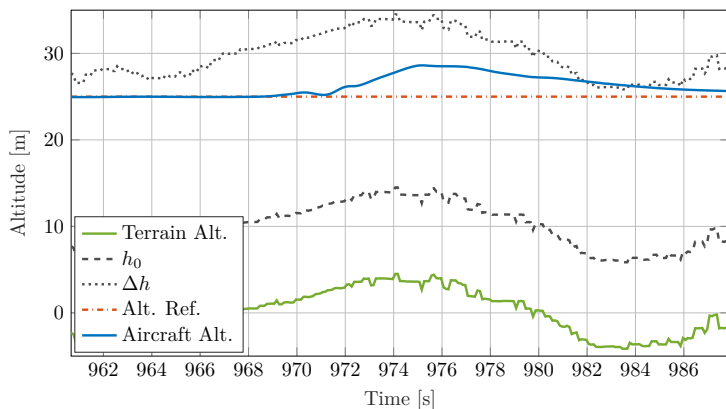
it detects are on the South facing slope, leading to the left veering motion plans at first, until the reference trajectory error (trajectories traveling along the loiter to the North) eventually provides enough incentive for the optimizer to attempt turning right, allowing more occlusion detections on the South-Eastern hill face which further encourage the rightward avoidance maneuver. Once the aircraft clears the hillside obstacle and the path is again unobstructed, the vehicle returns to the path.

Particularly important from this simulation is the illumination of some prospective robustness to the “split decision” problem gradient descent approaches to avoidance scenarios often face. I.e., the aircraft may face close to equally viable avoidance directions, with respect to the cost function, and each new measurement may tip the scale one way or the other. Sampling based motion planners would typically be a prudent way to handle such a scenario, but we can see here that including the “moving” reference trajectory generation in the formulation aided in forcing the optimizer to make a decision without any other external aid. Of course, this is only one scenario, and we should not extrapolate that this behavior will always result in any given obstacle laden predicament.

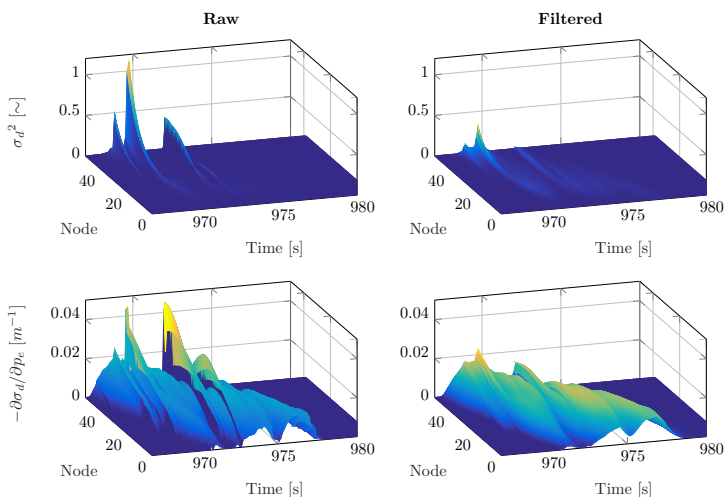
Looking deeper into the objective performance throughout the maneuver, we can see in Figures 8.11 and 8.12 that the nominal RTD and HAG constraints were respected. Additionally, Figures 8.11 and 8.13 showcase the effect of filtering the incoming terrain distance costs and cost Jacobians.



**Figure 8.11:** Avoidance maneuver; (top) the minimum terrain distance at each instance taken from all detected occlusions through the maneuver. (bottom) Squared RTD cost at horizon node 45, both filtered and raw.



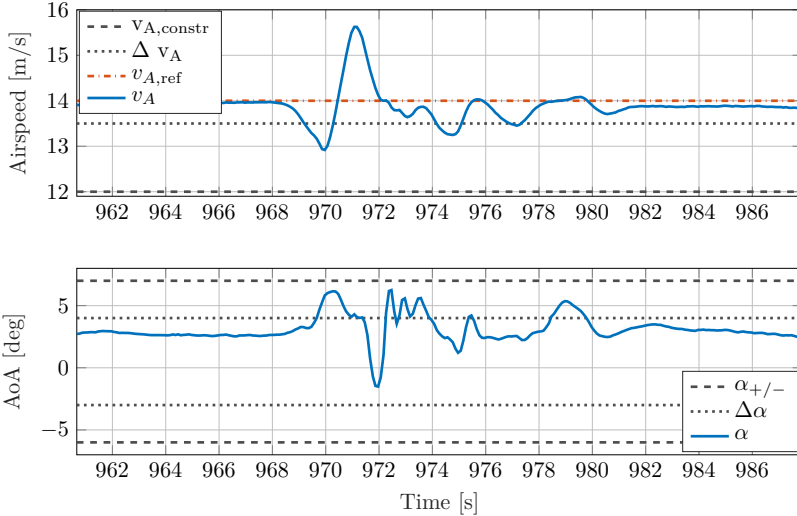
**Figure 8.12:** Avoidance maneuver; aircraft, reference, terrain, and constraint altitudes.



**Figure 8.13:** Avoidance maneuver; raw vs. filtered Radial Terrain Distance (squared) cost and (negative) Jacobian with respect to east position along horizon.

Despite the mostly smooth control response during the avoidance maneuver, the raw costs and Jacobians show much noisier/jumpy behavior which would have resulted in more aggressive control action, possibly destabilizing the aircraft.

Beyond the terrain objectives, Fig. 8.14 shows both airspeed and angle of attack constraints respected. Control outputs and attitude responses are shown in Fig. 8.15.



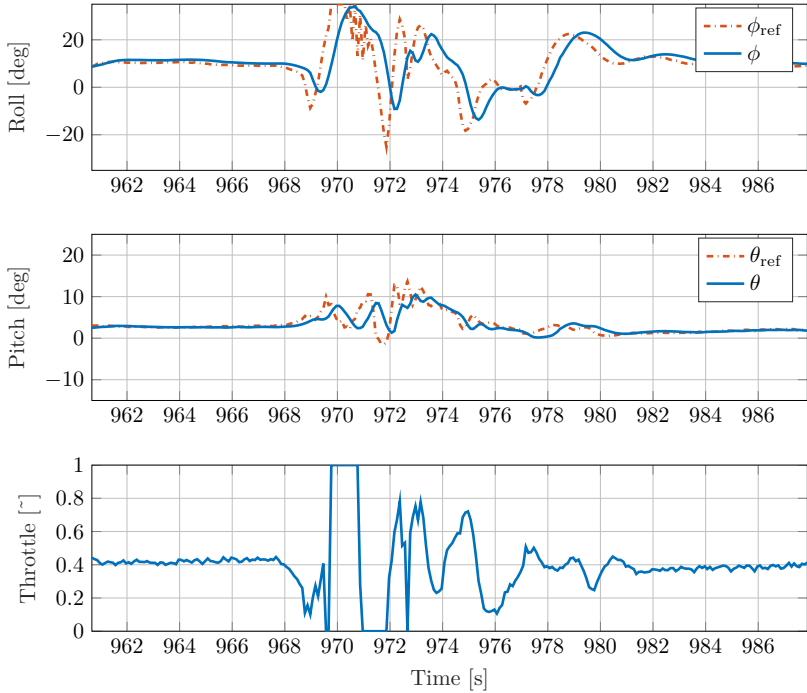
**Figure 8.14:** Avoidance maneuver; airspeed and AoA are kept within their soft constrained bounds during the maneuver.

## 6 Conclusions

Likely evident, due to the length and widely varied topics of this “brief”, is that combining real-time sensing and mapping of large scale, generalized environments with fast local re-planning algorithms for efficient and safe near terrain fixed-wing flight is decidedly non-trivial.

### 6.1 Summary

Section 2 showcased the some of the struggles of implementing long horizon optimizations for fixed-wing vehicles. We detailed several significant issues and some contained corner cases from our previous formulations and proposed remedies. In particular, a wind-aware lateral-directional trajectory generation algorithm was



**Figure 8.15:** Avoidance maneuver; optimized control references and the PX4 attitude response – note the short oscillations in roll when the trajectory reference errors are overtaking the preliminary RTD costs (see Fig. 8.10).

constructed from our wind-robust guidance logic (Part A), providing a reference set of positions and air mass relative speeds and headings for the NMPC to track. Vertical wind estimates were additionally incorporated into our longitudinal guidance law, providing an air mass relative flight path setpoint.

We improved our soft constraint formulation from Paper V, converting the piecewise quadratic cost into a one-sided exponential Huber loss. The new soft constraint formulation provides gradient information at every point in the soft constrained state space, reducing oscillatory control feedback when the state approaches the bound.

Section 3 presents a preliminary new approach to *generalized* terrain interpretation from vision-based elevation maps, tailored for implementation with fixed-wing



UAVs utilizing optimization based controllers. HAG look-ups with bilinear interpolation were used to provide a vertical terrain avoidance constraint on the vehicle. Radial terrain interpretation was accomplished using an occlusion detector designed to efficiently provide a small list of terrain point and surface normals in the most relevant areas of the elevation map which we could then use to construct novel, relative velocity dependent, “relative” ESDFs. The local RESDFs were molded into soft Huber constraints, and input to the NMPC optimization as radial terrain distance penalties, importantly providing terrain feedback weighted on those occlusions which the aircraft is most *quickly* approaching.

Our control augmented model formulation was reduced, allowing the elimination of three states from the optimization and easing identification complexity with reduced parameterization. We further introduced an enhanced thrust model.

Combining all objectives together in the optimizer was accomplished using a weight prioritization scheme which devalues certain objective references whenever relative proximity to the terrain increases.

Finally, our complete terrain avoidance framework was evaluated in HITL simulation with a high-fidelity model of the Techpod UAV. The occlusion detector was able to identify relevant surfels from which the optimizer then steered the aircraft away from elevated terrain obtruding on its desired path, all while satisfying all soft objective bounds. Our filtering approach for the terrain costs and Jacobians was further demonstrated to be an important feature, providing less noisy feedback which could have otherwise destabilized the optimizer.

## 6.2 Future Work

While our initial look into the terrain avoidance problem for fixed-wing UAVs reaped some insightful results, the preliminary concepts for environment-aware control presented herein are exactly that, preliminary, and many more avenues exist to explore within this research. Immediate technical aspects that must be born out include computational complexity analysis of the myriad of parameter settings/combinations for the occlusion detector and how these interact with varying horizon lengths of the NMPC, initialization and/or representation of unseen space in the horizon by either map priors or free vs. occupied assumptions, and field validation of all algorithms running onboard the vehicle. A conceptual outlook on what we find to be the most important and/or interesting directions for extensions and future work, including some ongoing work we are indeed already pursuing, may be found in Chapter 9.



## Conclusion & Outlook

---

In this thesis, we developed practical control and local re-planning strategies for small, low-flying fixed-wing UAVs towards enabling safe operation in strong winds and near uncertain terrain. In this chapter, we summarize the core contributions from the cumulative developments and sketch some potential future research directions.

### 9.1 Contributions

#### Handling High Winds

We developed a simple, safe, and computationally efficient guidance strategy for navigation of small, fixed-wing UAVs in *arbitrarily* strong wind fields. To the best of our knowledge, it was the first with specific consideration of *excess winds*, i.e. wind speeds that exceed the vehicle's nominal airspeed. The nonlinear controller guarantees convergence to a safe and stable vehicle configuration with respect to the wind field while preserving some tracking performance with respect to the path target, and command continuity is maintained throughout all state transitions, avoiding discrete switching which could degrade vehicle flight behavior. We expanded the control law by developing an energy efficient airspeed reference compensation logic which not only mitigates, but also actively works against excess winds which would otherwise cause the aircraft to “run-away”. Extensive field testing was conducted, demonstrating track keeping errors of less than 1 m consistently maintained during gusting excess winds over various mountainous regions in Switzerland. A coupled approach to airspeed and heading reference commands was further developed with a more principled consideration of airspeed reference minimization. With this new approach we additionally provided a simulation comparison to the previous decoupled approach in static and dynamic winds showcasing, qualitatively, improved airspeed reference efficiency and track keeping performance, potentially enabling

battery/fuel savings on long endurance flights.

### Model Predictive Control & Local Re-Planning

We proposed a control augmented modeling strategy as a means of capturing low-level autopilot response dynamics and quasi-steady aerodynamic and thrusting behavior of fixed-wing UAVs using simple and safe system identification procedures. The models were shown predictive on the order of tens of seconds, motivating their utility for use in long-horizon model predictive controllers. NMPC objective formulations particular to fixed-wing aircraft were designed achieving simultaneous three-dimensional Dubins Aircraft path segment following, airspeed stabilization, and stall prevention as well as the ability to adapt to control failures and strong wind disturbances. We conceptually evaluated the potential of incorporating obstacles into the NMPC formulation in the form of *generalized terrain feedback*. Real-time vision-based elevation maps were used to provide a 2.5D world representation to the aircraft, providing a “simulated” environment for the local re-planner to optimize within. The map was bilinearly interpolated for height feedback, and we designed an efficient ray casting approach for detection of forward (line of flight) and lateral occlusions. We proposed a novel “relative” Euclidean Signed Distance Field (RESDF) formulation as a function of the relative velocity between the vehicle and occlusion and developed a method to translate these RESDFs into optimizable soft constraints appended to the NMPC objective function. Finally, a preliminary example of the full local re-planning system acting to avoid an obstructing hillside was demonstrated in hardware-in-the-loop (HITL) simulation, all while respecting soft angle of attack and airspeed constraints.

## 9.2 Research Outlook

Our stated vision at the outset of this thesis was a progression towards fully autonomous small, fixed-wing UAVs able to reliably enter the wild, safely navigating in strong winds and complex terrain to perform their missions. With the wind-robust control and multi-objective local re-planning strategies we’ve developed throughout this work, we believe we’ve made a significant step towards these goals. However, as always, there is much we can improve, and many other avenues to explore. Our recommendations for the next steps forward specific to each part of the thesis are listed below, followed by some potential future remote fixed-wing applications to which we hope to expand our work.

### 9.2.1 Handling High Winds

**Robustification to Vertical Wind** While we were able to present an overall effective solution to the problem of two-dimensional wind robust guidance in Part A of the thesis, one aspect that remains is the vertical dimension. Up and down drafts in e.g. thermals or more critically rotors near hillsides were shown even in Paper II to cause problems for the aircraft, leading in the worst case to

momentary stall. 3D wind estimation techniques and low-level controllers capable of handling angle of attack constraints should be employed.

**Wind Prediction** Most of our efforts in this thesis, even the title of a chapter, was on “fighting” excess winds. However, if we look to nature, birds often take advantage of thermals or ridge updrafts to save energy. Some work has already been done on reactively tracking thermal updrafts [61], and more recently on *predicting* wind [1]. Considering wind as ally, as oppose to a foe, should be investigated, potentially incorporating not only feedback, but feed-forward wind predictions to either the the local re-planner or a separate trajectory generation unit for more optimized near terrain maneuvering.

## 9.2.2 Model Predictive Control and Local Re-planning

**Multiple Obstacle Abstractions** We only focused in this work on terrain avoidance. However, small fixed-wing UAVs may indeed encounter obstacles that will be missed by the elevation map, e.g. power lines, telephone poles. A parallel running algorithm similar to the “pushbroom stereo” in [2] could potentially be used to identify “likely misses” while populating the elevation map, and provide these as individual point obstacles we could incorporate into the re-planner separately.

**Trim Maps, or Separate Loops** Tuning the multi-objective NMPC, along with all peripheral algorithms is challenging. Especially in the longitudinal axis. Either supplying trim maps relating airspeed, angle of attack, pitch and throttle to the guidance logic, or down-grading longitudinal states to a lower-level controller could help to focus the optimization. However, in the latter case, a new control augmented model for the airspeed inclusive low-level response would need to be developed.

**Trajectory Libraries** Higher-level trajectory libraries or planners, operating at a reduced rate, acting in tandem to the NMPC, would attack the avoidance problem from two angles, a reasonable guess at a dynamic maneuver we could attempt and model-aware NMPC with terrain feedback to “dynamically smooth” the kinematic plan.

**Exploiting Map Layers** The elevation mapping approach used in this work also provides uncertainty information on the grid cells it populates; incorporating this uncertainty in the local re-planner could enhance robustness of avoidance maneuvers. Additional map layers could be populated with other high-level information derived from vision-based classifiers running on-board (see [26]); e.g. surface roughness, proximity to edges, or generally “safeness” for landing, and used to re-optimize landing targets online.

### 9.2.3 Future Applications

**Fixed-Wing Payload Drop & Recovery** Fig. 9.1 shows a photo from a recent ASL campaign to Gorner Glacier, Switzerland, where a multi-copter is dropping a GNSS unit onto the crevassed glacier surface. While in situ sensor placement is a natural task for a hover-capable vehicle, for large sensor networks or very remote drop locations, higher-endurance, longer-range fixed-wing UAVs would be required. However, precise dropping and recovery requires the plane to fly in close proximity to the possibly obstacle laden, variable terrain (see glacier surroundings), executing highly dynamic maneuvers, and simultaneously mitigating the abrupt mass changes without letting the aircraft stall. We see this task as an excellent challenge for our terrain-aware local re-planning framework, where payload related objectives could additionally be included in the optimization. Fig. 9.1 additionally shows a rendering of how a fixed-wing payload recovery may look.

**Deep Stall** In this thesis, we placed a significant priority on constraining the angle of attack to a safe, non-stalled region. In contrast, Fig. 9.1, shows the Easy Glider test UAV executing a deep stall descent, near  $60^\circ$  angle of attack!. Deep stall is a stable region far beyond the stall point where rapid air mass relative descents can be achieved using an all moving tail. The transition into or out of deep stall is a highly nonlinear process, including state and time hysteresis. However, with adequate modeling of these effects, NMPC could potentially be utilized to optimize these state transitions for e.g. landing in areas obstructed by tall trees or structures, potentially expanding the operational bounds of fixed-wing vehicles where otherwise only hover-capable platforms could go. We believe our work on wind relative guidance in this thesis could also be utilized for deep stall guidance, as the horizontal component of the airspeed is vastly reduced, and mimics our excess wind conditions. Efficient or optimal planning in strong winds, obstacle avoidance, and with additional stalled states provides many research avenues to explore.



**Figure 9.1:** Future fixed-wing UAV applications: (Left) Dropping a GNSS station on the Gorner Glacier, Switzerland, from a MAV. (Center) Rendering of fixed-wing payload recovery. (Right) The Easy Glider UAV in stabilized deep stall.

## Bibliography

---

- [1] F. Achermann, N. R. J. Lawrance, R. Ranftl, A. Dosovitskiy, J. J. Chung, and R. Siegwart. Learning to predict the wind for safe aerial vehicle planning. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 2311–2317, 2019.
- [2] A. J. Barry, P. R. Florence, and R. Tedrake. High-speed autonomous obstacle avoidance with pushbroom stereo. *Journal of Field Robotics*, 35(1):52–68, 2018. doi: 10.1002/rob.21741. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21741>.
- [3] R. Beard and T. McLain. Implementing dubins airplane paths on fixed-wing uavs. *Contributed Chapter to the Springer Handbook for Unmanned Aerial Vehicles*, 2013.
- [4] R. W. Beard, J. Ferrin, and J. Humpherys. Fixed wing uav path following in wind with input constraints. *IEEE Transactions on Control Systems Technology*, 22(6):2103–2117, Nov 2014. ISSN 1063-6536. doi: 10.1109/TCST.2014.2303787.
- [5] A. Brezoescu, T. Espinoza, P. Castillo, and R. Lozano. Adaptive trajectory following for a fixed-wing uav in presence of crosswind. *Journal of Intelligent and Robotic Systems*, 2012.
- [6] K. R. Bruce, J. R. Kelly, and L. H. Person. NASA B737 flight test results of the total energy control system. *AIAA Guidance Navigation and Control (GNC) Conference*, 1987. URL <http://hdl.handle.net/2060/19870017485>.
- [7] A. Bry, C. Richter, A. Bachrach, and N. Roy. Aggressive flight of fixed-wing and quadrotor aircraft in dense indoor environments. *The International Journal of Robotics Research*, 34(7):969–1002, 2015. doi: 10.1177/0278364914558129. URL <https://doi.org/10.1177/0278364914558129>.
- [8] Y. Bühler, M. S. Adams, R. Bösch, and A. Stoffel. Mapping snow depth in alpine terrain with unmanned aerial systems (uass): potential and limitations. *Cryosphere*, 10(3):1075–1088, 2016.
- [9] E. Bulka and M. Nahon. High-speed obstacle-avoidance with agile fixed-wing aircraft. In *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 971–980, 2019.

- [10] N. Cho, Y. Kim, and S. Park. Three-dimensional nonlinear differential geometric path-following guidance law. *Journal of Guidance, Control, and Dynamics*, 2015.
- [11] R. Curry, M. Lizarraga, B. Mairs, and G. H. Elkaim. L2+, an improved line of sight guidance law for UAVs. In *2013 American Control Conference*, pages 1–6, June 2013. doi: 10.1109/ACC.2013.6579804.
- [12] H. G. de Marina, Y. A. Kapitanyuk, M. Bronz, G. Hattenberger, and M. Cao. Guidance algorithm for smooth trajectory tracking of a fixed wing uav flying in wind flows. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5740–5745, May 2017. doi: 10.1109/ICRA.2017.7989674.
- [13] Falcon UAV, 2015. <http://www.falconunmanned.com/>.
- [14] P. Fankhauser and M. Hutter. A Universal Grid Map Library: Implementation and Use Case for Rough Terrain Navigation. In A. Koubaa, editor, *Robot Operating System (ROS) – The Complete Reference (Volume 1)*, chapter 5. Springer, 2016. ISBN 978-3-319-26052-5. doi: 10.1007/978-3-319-26054-9\\_5. URL <http://www.springer.com/de/book/9783319260525>.
- [15] P. Fankhauser, M. Bloesch, C. Gehring, M. Hutter, and R. Siegwart. *Robot-Centric Elevation Mapping with Uncertainty Estimates*, pages 433–440. doi: 10.1142/9789814623353\_0051. URL [https://www.worldscientific.com/doi/abs/10.1142/9789814623353\\_0051](https://www.worldscientific.com/doi/abs/10.1142/9789814623353_0051).
- [16] P. Fankhauser, M. Bloesch, and M. Hutter. Probabilistic terrain mapping for mobile robots with uncertain localization. *IEEE Robotics and Automation Letters*, 3(4):3019–3026, 2018.
- [17] H. Ferreau, A. Potschka, and C. Kirches. qpOASES webpage. <http://www.qpOASES.org/>, 2007-2015.
- [18] L. Furieri, T. Stastny, L. Marconi, R. Siegwart, and I. Gilitschenski. Gone with the wind: Nonlinear guidance for small fixed-wing aircrafts in arbitrarily strong windfields. *arXiv:1609.07577 [cs.SY]*, 2016.
- [19] L. Furieri, T. Stastny, L. Marconi, R. Siegwart, and I. Gilitschenski. Gone with the wind: Nonlinear guidance for small fixed-wing aircraft in arbitrarily strong windfields. In *2017 American Control Conference (ACC)*, pages 4254–4261, May 2017. doi: 10.23919/ACC.2017.7963609.
- [20] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart. *Robot Operating System (ROS): The Complete Reference (Volume 1)*, chapter RotorS—A Modular Gazebo MAV Simulator Framework, pages 595–625.



- 
- Springer International Publishing, Cham, 2016. ISBN 978-3-319-26054-9. doi: 10.1007/978-3-319-26054-9\_23. URL [http://dx.doi.org/10.1007/978-3-319-26054-9\\_23](http://dx.doi.org/10.1007/978-3-319-26054-9_23).
- [21] G. Garcia and S. Keshmiri. Nonlinear Model Predictive Controller for Navigation, Guidance and Control of a Fixed-Wing UAV. *AIAA Guidance, Navigation, and Control Conference (GNC)*, 2011.
- [22] F. Gavilan, R. Vazquez, and E. F. Camacho. A high-level model predictive control guidance law for unmanned aerial vehicles. In *Control Conference (ECC), 2015 European*, pages 1362–1369. IEEE, 2015.
- [23] S. Griffiths, J. Saunders, A. Curtis, B. Barber, T. McLain, and R. Beard. Obstacle and terrain avoidance for miniature aerial vehicles. *Advances in Unmanned Aerial Vehicles. Intelligent Systems, Control and Automation: Science and Engineering*, 33, 2007.
- [24] S. Gros, R. Quirynen, and M. Diehl. Aircraft control based on fast non-linear MPC & multiple-shooting. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 1142–1147, Dec 2012. doi: 10.1109/CDC.2012.6426439.
- [25] L. Han, F. Gao, B. Zhou, and S. Shen. Fiesta: Fast incremental euclidean distance fields for online motion planning of aerial robots. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4423–4430, 2019.
- [26] T. Hinzmann, T. Stastny, C. Cadena, R. Siegwart, and I. Gilitschenski. Free lsd: Prior-free visual landing site detection for autonomous planes. *IEEE Robotics and Automation Letters*, 3(3):2545–2552, 2018.
- [27] T. Hinzmann, T. Taubner, and R. Siegwart. Flexible stereo: Constrained, non-rigid, wide-baseline stereo vision for fixed-wing aerial platforms. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2550–2557, 2018.
- [28] T. Hinzmann, C. Cadena, J. Nieto, and R. Siegwart. Flexible trinocular: Non-rigid multi-camera-imu dense reconstruction for uav navigation and mapping. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1137–1142, 2019.
- [29] A. Hornung, K. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard. Octomap: an efficient probabilistic 3d mapping framework based on octrees. *Autonomous Robots*, 34(4):189–206, April 2013. ISSN 1573-7527. doi: 10.1007/s10514-012-9321-0. URL <https://www.sciencedirect.com/science/article/pii/S0266892097000362>.

- [30] B. Houska, H. Ferreau, and M. Diehl. ACADO Toolkit – An Open Source Framework for Automatic Control and Dynamic Optimization. *Optimal Control Applications and Methods*, 32(3):298–312, 2011.
- [31] B. Houska, H. J. Ferreau, and M. Diehl. An auto-generated real-time iteration algorithm for nonlinear {MPC} in the microsecond range. *Automatica*, 47(10):2279 – 2285, 2011. ISSN 0005-1098. doi: <http://dx.doi.org/10.1016/j.automatica.2011.08.020>. URL <http://www.sciencedirect.com/science/article/pii/S0005109811003918>.
- [32] P. J. Huber. Robust estimation of a location parameter. *Ann. Math. Statist.*, 35(1):73–101, 03 1964. doi: 10.1214/aoms/1177703732. URL <https://doi.org/10.1214/aoms/1177703732>.
- [33] E. R. Hunt, W. D. Hively, S. J. Fujikawa, D. S. Linden, C. S. T. Daughtry, and G. W. McCarty. Acquisition of nir-green-blue digital photographs from unmanned aircraft for crop monitoring. *Remote Sensing*, 2(1):290–305, 2010.
- [34] A. Joos, M. Müller, D. Baumgärtner, W. Fichter, and F. Allgöwer. Nonlinear predictive control based on time-domain simulation for automatic landing. In *AIAA Guidance, Navigation, and Control Conference (GNC)*, 2011.
- [35] M. Kamel, K. Alexis, M. Achtelik, and R. Siegwart. Fast nonlinear model predictive control for multicopter attitude tracking on  $so(3)$ . In *2015 IEEE Conference on Control Applications (CCA)*, pages 1160–1166, Sept 2015. doi: 10.1109/CCA.2015.7320769.
- [36] M. Kamel, J. Alonso-Mora, R. Siegwart, and J. Nieto. Robust collision avoidance for multiple micro aerial vehicles using nonlinear model predictive control. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 236–243, Sept 2017. doi: 10.1109/IROS.2017.8202163.
- [37] M. Kamel, T. Stastny, K. Alexis, and R. Siegwart. *Model Predictive Control for Trajectory Tracking of Unmanned Aerial Vehicles Using Robot Operating System*, pages 3–39. Springer International Publishing, Cham, 2017. ISBN 978-3-319-54927-9. doi: 10.1007/978-3-319-54927-9\_1. URL [https://doi.org/10.1007/978-3-319-54927-9\\_1](https://doi.org/10.1007/978-3-319-54927-9_1).
- [38] 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 Conference (GNC)*, 2006.
- [39] Y. Kang and J. K. Hedrick. Linear tracking for a fixed-wing uav using nonlinear model predictive control. *Control Systems Technology, IEEE Transactions on*, 17(5):1202–1210, 2009.

- 
- [40] M. Klein, A. Klos, J. Lenhardt, and W. Schiffmann. Wind-aware emergency landing assistant based on dubins curves. In *2017 Fifth International Symposium on Computing and Networking (CANDAR)*, pages 546–550, Nov 2017. doi: 10.1109/CANDAR.2017.99.
- [41] N. Koenig and A. Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, volume 3, pages 2149–2154. IEEE.
- [42] S. Leutenegger, A. Melzer, K. Alexis, and R. Siegwart. Robust state estimation for small unmanned airplanes. In *2014 IEEE Conference on Control Applications (CCA)*, pages 1003–1010, Oct 2014. doi: 10.1109/CCA.2014.6981466.
- [43] J. M. Levin, M. Nahon, and A. A. Paranjape. Real-time motion planning with a fixed-wing uav using an agile maneuver space. *Autonomous Robots*, 43(8): 2111–2130, 2019. doi: 10.1007/s10514-019-09863-2. URL <https://doi.org/10.1007/s10514-019-09863-2>.
- [44] C. Liu, O. McAree, and W.-H. Chen. Path-following control for small fixed-wing unmanned aerial vehicles under wind disturbances. *International Journal of Robust and Nonlinear Control*, 2012.
- [45] Y. Liu, S. Longo, and E. C. Kerrigan. Nonlinear predictive control of autonomous soaring UAVs using 3DOF models. In *2013 European Control Conference (ECC)*, pages 1365–1370, July 2013.
- [46] D. Lunni, A. Santamaria-Navarro, R. Rossi, P. Rocco, L. Bascetta, and J. Andrade-Cetto. Nonlinear model predictive control for aerial manipulation. In *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 87–93, June 2017. doi: 10.1109/ICUAS.2017.7991347.
- [47] Y. Luo, H. Chao, L. Di, and Y. Chen. Lateral directional fractional order ( $\pi$ )  $\pi$  control of a small fixed-wing unmanned aerial vehicles: controller designs and flight tests. *Control Theory & Applications, IET*, 5(18):2156–2167, 2011.
- [48] S. Mathisen, K. Gryte, T. Johansen, and T. Fossen. Non-linear model predictive control for longitudinal and lateral guidance of a small fixed-wing UAV in precision deep stall landing. In *AIAA Infotech @ Aerospace, AIAA SciTech Forum*, 2016.
- [49] T. McGee and K. Hedrick. Path planning and control for multiple point surveillance by an unmanned aircraft in wind. *2006 American Control Conference*, 2006.

- [50] N. Metni and T. Hamel. A UAV for bridge inspection: Visual servoing control law with orientation limits. *Automation in Construction*, 17(1):3–10, November 2007.
- [51] D. G. Mitchell and R. H. Hoh. Low-order approaches to high-order systems: Problems and promises. *Journal of Guidance, Control, and Dynamics*, 5(5): 482–489, 1982.
- [52] E. A. Morelli. Piloted parameter identification flight test maneuvers for closed loop modeling of the f-18 high alpha research vehicle (harv). 1996.
- [53] E. A. Morelli. Low-order equivalent system identification for the tu-144ll supersonic transport aircraft. *Journal of guidance, control, and dynamics*, 26(2):354–362, 2003.
- [54] P. C. Murphy. Closed-loop system identification experience for flight control law and flying qualities evaluation of a high performance fighter aircraft. 1996.
- [55] D. Nelson, B. Barber, T. McLain, and R. Beard. Vector field path following for miniature air vehicles. *Robotics, IEEE Transactions on*, 2007.
- [56] M. Neunert, C. de Crousaz, F. Furrer, M. Kamel, F. Farshidian, R. Siegwart, and J. Buchli. Fast nonlinear model predictive control for unified trajectory optimization and tracking. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1398–1404, 2016.
- [57] P. Oettershagen, A. Melzer, T. Mantel, K. Rudin, T. Stastny, B. Wawrzacz, T. Hinzmänn, S. Leutenegger, K. Alexis, and R. Siegwart. Design of small hand-launched solar-powered UAVs: From concept study to a multi-day world endurance record flight. *Journal of Field Robotics*, 2016.
- [58] P. Oettershagen, T. Stastny, T. Mantel, A. Melzer, K. Rudin, P. Gohl, G. Agamennoni, K. Alexis, and R. Siegwart. *Long-Endurance Sensing and Mapping Using a Hand-Launchable Solar-Powered UAV*, pages 441–454. Springer International Publishing, Cham, 2016. ISBN 978-3-319-27702-8. doi: 10.1007/978-3-319-27702-8\_29.
- [59] P. Oettershagen, F. Achermann, B. Müller, D. Schneider, and R. Siegwart. Towards fully environment-aware uavs: Real-time path planning with online 3d wind field prediction in complex terrain. *arXiv:1712.03608 [cs.RO]*, 2017.
- [60] P. Oettershagen, A. Melzer, T. Mantel, K. Rudin, T. Stastny, B. Wawrzacz, T. Hinzmänn, S. Leutenegger, K. Alexis, and R. Siegwart. Design of small hand-launched solar-powered UAVs: From concept study to a multi-day world endurance record flight. *Journal of Field Robotics*, 34(7):1352–1377, 2017. ISSN 1556-4967. doi: 10.1002/rob.21717. URL <http://dx.doi.org/10.1002/rob.21717>.

- 
- [61] P. Oettershagen, T. Stastny, T. Hinzmann, K. Rudin, T. Mantel, A. Melzer, B. Wawrzacz, G. Hitz, and R. Siegwart. Robotic technologies for solar-powered uavs: Fully autonomous updraft-aware aerial sensing for multiday search-and-rescue missions. *Journal of Field Robotics*, 35(4):612–640, 2018. doi: 10.1002/rob.21765. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21765>.
- [62] H. Oleynikova, M. Burri, Z. Taylor, J. Nieto, R. Siegwart, and E. Galceran. Continuous-time trajectory optimization for online uav replanning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.
- [63] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto. Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1366–1373, 2017.
- [64] H. Oleynikova, Z. Taylor, R. Siegwart, and J. Nieto. Safe local exploration for replanning in cluttered unknown environments for micro-aerial vehicles. *IEEE Robotics and Automation Letters*, 2018.
- [65] J. Osborne and R. Rysdyk. Waypoint guidance for small uavs in wind. *American Institute of Aeronautics and Astronautics*, 2005.
- [66] S. Park, J. Deyst, and J. How. A new nonlinear guidance logic for trajectory tracking. *AIAA Guidance, Navigation, and Control Conference and Exhibit, Guidance, Navigation, and Control and Co-located Conferences*, 2004.
- [67] Pix4D, 2015. <http://pix4d.com/>.
- [68] Pixhawk Autopilot, 2015. <http://pixhawk.org/>.
- [69] M. Popović, G. Hitz, J. Nieto, I. Sa, R. Siegwart, and E. Galceran. Online informative path planning for active classification using uavs. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5753–5758, 2017.
- [70] M. Quigley, B. Barber, S. Griffiths, and M. A. Goodrich. Towards real-world searching with fixed-wing mini-uavs. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3028–3033, 2005.
- [71] R. Quirynen, S. Gros, and M. Diehl. Efficient nmpc for nonlinear models with linear subsystems. In *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, pages 5101–5106. IEEE, 2013.
- [72] Robotic Operating System, 2016. <http://ros.org/>.

- [73] A. Rucco, A. P. Aguiar, F. L. Pereira, and J. B. de Sousa. A predictive path-following approach for fixed-wing unmanned aerial vehicles in presence of wind disturbances. In *Robot 2015: Second Iberian Robotics Conference*, pages 623–634. Springer, 2016.
- [74] R. Rysdyk. Unmanned aerial vehicle path following for target observation in wind. *Journal of Guidance, Control, and Dynamics*, 2006.
- [75] S. Park, J. Deyst and J. P. How. Performance and lyapunov stability of a nonlinear path following guidance method. *Journal of Guidance, Control, and Dynamics*, 30(6):1718–1728, 2007.
- [76] S. Scherer, S. Singh, L. Chamberlain, and M. Elgersma. Flying fast and low among obstacles: Methodology and experiments. *The International Journal of Robotics Research*, 27(5):549–574, 2008. doi: 10.1177/0278364908090949.
- [77] M. Selig. *Modeling Propeller Aerodynamics and Slipstream Effects on Small UAVs in Realtime*. doi: 10.2514/6.2010-7938. URL <https://arc.aiaa.org/doi/abs/10.2514/6.2010-7938>.
- [78] SHERPA Project, 2015. <http://www.sherpa-project.eu/>.
- [79] T. Stastny and R. Siegart. Nonlinear model predictive guidance for fixed-wing uavs using identified control augmented dynamics. In *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 432–442, June 2018. doi: 10.1109/ICUAS.2018.8453377.
- [80] T. Stastny, G. Garcia, and S. Keshmiri. Collision and obstacle avoidance in unmanned aerial systems using morphing potential field navigation and nonlinear model predictive control. *Journal of Dynamic Systems, Measurement, and Control*, 137(1), 2014. doi: 10.1115/1.4028034.
- [81] T. Stastny, G. Garcia, and S. Keshmiri. Robust three-dimensional collision avoidance for fixed-wing unmanned aerial systems. In *AIAA Guidance, Navigation, and Control Conference (GNC), AIAA SciTech Forum*, 2015.
- [82] T. Stastny, A. Dash, and R. Siegart. Nonlinear MPC for fixed-wing UAV trajectory tracking: Implementation and flight experiments. In *AIAA Guidance, Navigation, and Control Conference (GNC), AIAA SciTech Forum*, 2017.
- [83] T. Stastny, E. Ahbe, M. Dangel, and R. Siegart. Locally power-optimal nonlinear model predictive control for fixed-wing airborne wind energy. In *2019 American Control Conference (ACC)*, pages 2191–2196, 2019.
- [84] The Local CH, 2015. <http://www.thelocal.ch/20150419/swisspost-launches-tests-for-drone-deliveries>.

- [85] M. Warren, L. Mejias, J. Kok, X. Yang, F. Gonzalez, and B. Upcroft. An automated emergency landing system for fixed-wing aircraft: Planning and control. *Journal of Field Robotics*, 32(8):1114–1140, 2015. doi: 10.1002/rob.21641. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21641>.
- [86] K. Yang, S. Sukkarieh, and Y. Kang. Adaptive nonlinear model predictive path tracking control for a fixed-wing unmanned aerial vehicle. In *AIAA Guidance, Navigation, and Control Conference*, 2009.
- [87] J. Zhang, L. Cheng, and B. Liang. Path-following control for fixed-wing unmanned aerial vehicles based on a virtual target. *Journal of Aerospace Engineering*, 2012.





# Curriculum Vitae

---

## **Thomas Stastny**

born May 22, 1990

citizen of the United States of America

- 2014–2020 *ETH Zurich, Switzerland*  
Doctoral studies at the Autonomous Systems Lab; Supervised  
by Prof. Roland Siegwart
- 2012–2014 *University of Kansas, USA*  
Master of Science in Aerospace Engineering (with Honors)
- 2012 *TU Delft, Netherlands*  
Masters coursework in Systems & Control and Aerospace En-  
gineering Programs
- 2008–2012 *University of Kansas, USA*  
Bachelor of Science in Aerospace Engineering