

Bachelor Thesis

Wind-Aware Kinematic MPCC Formulation for Fixed-Wing Guidance

Spring Term 2025

Supervised by:

David Rohr
Dr. Andrea Carron

Author:

Mark Benazet Castells

Declaration of originality

The signed declaration of originality is a component of every written paper or thesis authored during the course of studies. In consultation with the supervisor, one of the following three options must be selected:

- I confirm that I authored the work in question independently and in my own words, i.e. that no one helped me to author it. Suggestions from the supervisor regarding language and content are accepted. I used no generative artificial intelligence technologies¹.
- I confirm that I authored the work in question independently and in my own words, i.e. that no one helped me to author it. Suggestions from the supervisor regarding language and content are accepted. I used and cited generative artificial intelligence technologies².
- I confirm that I authored the work in question independently and in my own words, i.e. that no one helped me to author it. Suggestions from the supervisor regarding language and content are accepted. I used generative artificial intelligence technologies³. In consultation with the supervisor, I did not cite them.

Title of paper or thesis:

Wind-Aware Kinematic MPCC Formulation for Fixed-Wing Guidance

Authored by:*If the work was compiled in a group, the names of all authors are required.***Last name(s):**

Benazet Castells

First name(s):

Mark

With my signature I confirm the following:

- I have adhered to the rules set out in the Citation Guide.
- I have documented all methods, data and processes truthfully and fully.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for originality.

Place, date

Zürich, 09.10.2024

Signature(s)

¹ E.g. ChatGPT, DALL E 2, Google Bard

² E.g. ChatGPT, DALL E 2, Google Bard

³ E.g. ChatGPT, DALL E 2, Google Bard

If the work was compiled in a group, the names of all authors are required. Through their signatures they vouch jointly for the entire content of the written work.

Contents

Preface	v
Abstract	vii
Symbols	ix
1 Introduction	1
1.1 Motivation	1
1.2 Current Limitations in UAV Path Following	1
1.3 Goal Definition	1
1.4 Structure of the Report	2
2 Literature Research	4
2.1 State-of-the-Art	4
2.1.1 The L_1 Guidance Algorithm	4
2.1.2 L_1 Guidance Extensions	5
2.2 Related Work	6
2.2.1 Model Predictive Contouring Control	6
3 Preliminaries	8
3.1 Predictive Controllers	8
3.2 Fixed-Wing Model	10
3.2.1 System Description	10
3.2.2 Lateral Kinematic Model	10
3.2.3 Complete Kinematic Model	11
4 Model Predictive Contouring Control	13
4.1 Model Predictive Control	13
4.2 Model Predictive Contouring Control	14
4.2.1 Path Parameterization	14
4.2.2 Cost Function Design	14
4.2.3 Constraint Formulation	17
4.3 Optimization Problem	17
4.4 Reference Tracking	18
4.4.1 Thrust Tracking	18
4.4.2 Roll Tracking	18
5 Results	21
5.1 Simulation Setup	21
5.1.1 ACADOS	24
5.2 Scenario Analysis	25
5.3 Big Radius Conditions	26

5.3.1	Scenario 1: Large Radius - No Wind	26
5.3.2	Scenario 2: Large Radius - High Wind	26
5.4	Small Radius Conditions	27
5.4.1	Scenario 3: Small Radius - High Wind	27
5.4.2	Scenario 4: Small Radius - Maximum Wind	27
5.4.3	Scenario 5: Small Radius - Excess Wind	28
5.5	Arbitrary conditions and paths	34
6	Conclusion and Outlook	37
6.1	Limitations of the Work	37
6.1.1	Future Work	37
6.2	Concluding Remarks	38
	Bibliography	40

Preface

This Bachelor Thesis, "Wind-Aware Kinematic MPCC Formulation for Fixed-Wing Guidance", represents the end of my undergraduate studies in mechanical Engineering at the *Eidgenössische Technische Hochschule* (ETH) Zürich. The research was conducted at the Autonomous Systems Lab (ASL) in collaboration with the Institute for Dynamical Systems and Control (IDSC).

I would like to express my gratitude to my supervisors, David Rohr and Dr. Andrea Carron, for their guidance and support throughout this project. Their feedback during our meetings was crucial. I am also thankful to Professor Dr. Roland Siegwart and Professor Dr. Melanie Zeilinger for providing me with the opportunity to work on this topic.

In the development of this thesis, generative artificial intelligence technologies were used to enhance both the textual content and associated coding elements of the project. Claude [1], an generative AI developed by Anthropic, refined the text, addressing grammar and syntax, while also improving L^AT_EX formatting. For programming tasks, Claude and GitHub Copilot provided code suggestions and debugging support, enhancing overall quality and efficiency.

Abstract

This thesis presents the development and evaluation of a Model Predictive Contouring Control (MPCC) framework for fixed-wing UAV guidance in high wind conditions. The work focuses on enhancing path-following precision while strictly adhering to kinematic limits (e.g excessive accelerations or excessive use of the thrust) and compensating for wind disturbances. While existing state-of-the-art methods often struggle to maintain path accuracy in extreme wind conditions without violating safety constraints, this approach addresses these limitations. A kinematic model of the UAV is developed, and the MPCC is formulated to balance path accuracy, progress, and control smoothness within defined operational limits. The controller is implemented using CasADi for problem formulation and ACADOS for rapid solving. Simulations are conducted across various scenarios, including different wind speeds and path radii. Results demonstrate the controller's effectiveness in maintaining path accuracy under challenging conditions while respecting safety limits, including its ability to adapt to extreme winds by allowing backward motion when necessary. The MPCC is designed to interface with the rest of the control system by generating optimal acceleration and heading rate commands, which are then tracked by a low-level controller to achieve the desired UAV behavior.

Symbols

Symbols

ϕ, θ, ψ	Roll, Pitch and Yaw angle
\mathcal{I}	Inertial Frame
\mathcal{B}	Body Frame
\mathbf{L}_1, L_1	Vector from vehicle to reference point and Magnitude
\mathbf{V}_g, v_g	Ground Speed Vector and Magnitude
a_c	Lateral Acceleration command
η	Angle between \mathbf{L}_1 and \mathbf{V}_g
\mathbf{V}_A, v_A	Airspeed Vector and Magnitude
\mathbf{W}, w	Wind Speed Vector and Magnitude
τ	Time constant
w_n	Natural frequency
ζ	Damping ratio
θ	Path parameter
\mathcal{P}	Path
$p(\theta)$	Point on the path
$\tau(\theta)$	Tangent vector to the path
e^c	Contouring error
e^l	Lag error
$\hat{\theta}$	Approximated path parameter
v_θ	Velocity along the path
ΔT	Time step
Q, R	Weighting matrices
q_c, q_l, q_v	Scalar weights
r_v	Velocity error weight
v_d	Desired velocity magnitude
$a_x^{\mathcal{B}}, a_y^{\mathcal{B}}$	Body-frame accelerations
e, n	East and North position components
$v_x^{\mathcal{B}}, v_y^{\mathcal{B}}$	Body-frame velocity components
u_x, u_y, u_ψ	Control inputs
$e_{v,k}$	Velocity error
N	Horizon Steps
$f(x_k, u_k)$	System dynamics function
$g_k(x_k, u_k)$	Inequality constraint function

$g(x_0)$	Equality constraint function
T_f	Horizon Length
T_{sim}	Simulation time
ϵ	Convergence threshold

Indices

x	x axis
y	y axis
k	Time step index
d	Desired value
ref	Referenced value
$(\hat{\cdot})$	Linearized value
g	Ground
a	Airspeed

Acronyms and Abbreviations

ETH	Eidgenössische Technische Hochschule
ASL	Autonomous Systems Lab
IDSC	Institute for Dynamic Systems and Control
UAV	Unmanned Aerial Vehicle
MPC	Model Predictive Control
NMPC	Nonlinear Model Predictive Control
NED	North East Down
FRD	Front Right Down
sUAS	small Unmanned Aircraft System
MPCC	Model Predictive Contouring Control
SQP	Sequential Quadratic Programming
RTI	Real-Time Iteration
HPIPM	High-Performance Interior Point Method
QP	Quadratic Programming
SiL	Software in the Loop
CasADi	Computer Algebra System for Automatic Differentiation

Chapter 1

Introduction

1.1 Motivation

Small fixed-wing UAVs have become increasingly popular due to their easy and cheap setup, as well as their efficient flight characteristics. These small planes can cover and map large areas in short time and with relatively low operational costs. These characteristics and their capability for modular payloads have made them be adopted across various sectors, including agriculture, environmental monitoring and disaster management.

Despite these advantages, small fixed-wing UAVs face challenges in certain operating conditions as they are heavily weather-dependent, like rain, snow or wind. Especially, these small aircraft are very susceptible to strong winds, as the wind speed can quickly exceed the airspeed and thus make them fly away from the path or even crash. The light weight of these UAVs makes them especially vulnerable to such disturbances.

1.2 Current Limitations in UAV Path Following

Current commercial UAV flight control software predominantly relies on reactive methods for lateral control, with the L_1 guidance algorithm being the popular choice. These methods are computationally efficient and easily implementable on various hardware platforms, making them widely adopted in commercial systems.

However, these reactive approaches have limitations. They lack the ability to anticipate future states or incorporate constraints into their decision-making process. These limitations highlight the potential for more sophisticated, predictive control strategies that can account for future states and adhere to strict safety constraints, especially in demanding wind conditions.

A detailed examination of current control methods and their limitations will be presented in the preliminaries chapter.

1.3 Goal Definition

The primary objective of this thesis is to develop and evaluate an outer-level controller for the path-following guidance problem in small UAVs operating under high

wind conditions. The work aims to formulate a Model Predictive Contouring Control (MPCC) framework that can effectively mitigate the impact of strong winds while strictly adhering to predefined safety constraints and maintaining a desired airspeed. This last point is particularly crucial for fixed-wing UAVs, which desire a fixed airspeed, and is not always explicitly considered in MPCC formulations.

This work specifically focuses on designing a controller that optimizes the precision of the path while minimizing the drift caused by high winds, adhering to safety constraints, and crucially, maintaining a specified airspeed.

It is important to note that this thesis deliberately excludes considerations of dynamics and system identification, instead concentrating solely on the kinematics of UAVs. This approach allows for a more focused examination of the path-following problem from a purely geometric perspective, while still addressing the critical aspect of velocity control. The MPCC framework developed in this thesis is based on kinematics, assuming the existence of a low-level controller capable of following the commands of the high-level controller, including maintaining the desired airspeed.

1.4 Structure of the Report

This report begins by presenting related work in Chapter 2, providing the state-of-the-art and identifying the research gap this thesis addresses. Chapter 3 then introduces the preliminaries, including the rationale for predictive controllers and the fixed-wing model used in the MPCC, which forms the foundation for subsequent computations.

The core of the work is detailed in Chapter 4, where the Model Predictive Contouring Controller (MPCC) is presented. This chapter covers the theoretical foundation of MPCC, including path parameterization, cost function design, constraint formulation, and the complete optimization problem. Additionally, it discusses the reference tracking models for the plant.

Chapter 5 presents the results of the study. It begins by describing the simulation setup, including the ACADOS solver configuration. The chapter then provides a comprehensive scenario analysis, examining the MPCC's performance under various conditions, from large radius paths with no wind to challenging scenarios with small radii and high winds. Each scenario is analyzed in terms of trajectory and control inputs.

Finally, Chapter 6 concludes the thesis with a discussion of the work's limitations, suggestions for future research, and closing remarks on the significance of the developed constraint-aware MPCC framework for UAV control in windy conditions.

Chapter 2

Literature Research

This section presents a comprehensive review of state-of-the-art approaches and relevant research in UAV control and guidance. Keywords include:

- Path following guidance
- Guidance strategies for high-wind scenarios
- Constraint-aware UAV guidance
- Predictive guidance methods

The primary interest of this work lies in approaches that enhance UAV performance in high-wind environments while adhering to kinematic constraints such as acceleration limits and bank angle restrictions.

Furthermore, this section examines previous implementations of Model Predictive Contouring Control (MPCC), analyzing its various applications.

2.1 State-of-the-Art

Commercially available flight control software for UAVs (e.g *PX4*¹ or *Ardupilot*²) predominantly utilize reactive methods for path-following control. Among these, the L_1 guidance algorithm has emerged as a prevalent choice due to its simplicity, computational efficiency, and effectiveness in path-following tasks[2].

2.1.1 The L_1 Guidance Algorithm

The L_1 algorithm is a nonlinear guidance method that operates on several key principles[3]. Firstly, it defines a reference point on the desired path at a fixed distance L_1 ahead of the vehicle's current position. Secondly, it considers two critical vectors: \mathbf{L}_1 , which extends from the vehicle to the reference point, and \mathbf{V}_g , representing the vehicle's ground-speed velocity vector.

Based on these vectors, the algorithm generates a lateral acceleration command (a_c) using the following equation:

$$a_c = 2 \frac{V_g^2}{L_1} \sin \eta, \quad (2.1)$$

¹<https://px4.io>

²<https://ardupilot.org>

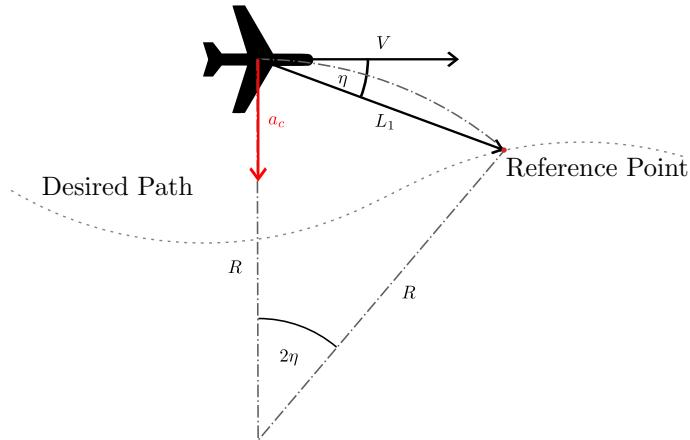


Figure 2.1: L_1 Guidance Algorithm per [3]

In this equation, V_g represents the ground speed of the vehicle, η is the angle between L_1 and V_g , and L_1 is the look-ahead distance. The primary objective of this approach is to minimize the angle η , thereby effectively guiding the vehicle onto the desired path. To maintain consistent performance across various operating conditions, the L_1 distance is often dynamically adapted based on the vehicle's speed[3].

The L_1 algorithm offers several advantages in UAV control. Its simplicity of implementation and low computational requirements make it an effective option for resource-constrained systems, as it can be embedded almost in every onboard platform, e.g. micro-controllers. Moreover, it demonstrates robust performance in typical flight conditions.

2.1.2 L_1 Guidance Extensions

The L_1 guidance algorithm, while effective in many scenarios, exhibits limitations in high wind conditions. Park et al. [2] established a critical convergence criterion for the algorithm: the look-ahead distance L_1 must not exceed the path's radius of curvature R , i.e., $L_1 \leq R$. This condition ensures the algorithm's stability and convergence towards the desired path. However, in high wind scenarios, this criterion is frequently violated. The main factor which contributes to this violation is:

1. Increased ground speed: Strong tailwinds can significantly elevate the vehicle's ground speed, necessitating a larger L_1 to maintain responsiveness.

These limitations have spurred the development of enhanced versions of the L_1 guidance algorithm. Notable extensions include the work of Fourier et al. [4] and Stastry et al. [5]. These algorithms not only address the convergence issues of the original L_1 guidance but also demonstrate robust performance in extreme wind conditions, including scenarios where wind speeds surpass the vehicle's airspeed.

The evolution of these guidance algorithms reflects a shift from trajectory tracking to path following. Trajectory tracking requires following a time-parameterized reference, aiming for specific locations at predetermined times. Path following, however, focuses on guiding the vehicle along a geometric path without strict timing constraints, offering more flexibility. This distinction is crucial for UAV guidance in windy conditions, where adhering to a time-based trajectory can be challenging

or impossible. This is particularly relevant in high wind situations, where following a time-parameterized reference might become unfeasible.

2.2 Related Work

Kang and Hedrick [6] applied nonlinear model predictive control (NMPC), demonstrating improved trajectory tracking compared to reactive methods. Osborne and Rysdyk [7] proposed a wind-aware waypoint guidance method, emphasizing the importance of integrating wind information directly into guidance algorithms.

Reinhardt and Johansen [8] developed an NMPC approach based on a wind frame formulation for attitude control, highlighting the benefits of wind-relative control design. Stastny et al. [9] focused on NMPC for lateral-directional control in windy conditions.

Chen and Tan [10] presented an NMPC-based approach for UAV path following, demonstrating superior performance compared to the L1 guidance law.

While NMPC approaches have been widely applied to UAV path following, it's worth noting another specialized predictive control technique: MPCC. Although not specifically designed for UAV applications, MPCC has found success in other domains where precise path following is crucial.

2.2.1 Model Predictive Contouring Control

The concept of Model Predictive Contouring Control (MPCC) evolved from earlier work on path following control. Faulwasser et al. [11] introduced model predictive path-following control, which separates the geometric path from its temporal parametrization. This allows for independent optimization of the path progress.

Building on this foundation, Lam et al. [12] developed MPCC for biaxial systems such as machine tools and laser cutting applications. Their formulation minimizes contouring error while maximizing path progress, subject to system constraints.

Liniger et al. [13] extended MPCC to the domain of autonomous racing, applying it to 1:43 scale RC cars. Recently, Romero et al. [14] further expanded MPCC to three-dimensional quadrotor control for drone racing. They showed that MPCC could follow simplified reference trajectories while achieving near time-optimal performance, outperforming standard trajectory tracking approaches in real-world experiments.

MPCC, originally designed for applications that demand precise timing, was developed to cater to the high-speed, competitive nature of these applications. Although these applications differ significantly from typical UAV missions, understanding the underlying principles and how they can be adapted for path following is crucial, as the work aims to convert the racing-oriented controller to fixed-wing cases.

It is crucial to note that previous MPCC implementations have primarily operated in controlled, relatively disturbance-free environments. In contrast, UAV flights, often encounter significant and unpredictable environmental factors. This disparity highlights the motivation of the work.

Chapter 3

Preliminaries

This chapter introduces the foundational concepts upon which the subsequent chapters are built. It begins by establishing the reasoning for employing a predictive controller in the context of fixed-wing UAV control. Following this, a kinematic model of the fixed-wing UAV is presented, which serves as the cornerstone for the MPCC framework developed in this thesis.

3.1 Predictive Controllers

An alternative to reactive approaches for handling high winds is predictive control, as discussed in [9], [10], [6], and [8]. By incorporating an approximation of the wind and the UAV's own model into the controller, the UAV's behavior can be anticipated, enabling proactive measures to mitigate these disturbances. This predictive strategy enables the controller to foresee potential problems and avoid them, addressing a key limitation of reactive controllers.

Safety constraints are integral to this approach, keeping the UAV within a safe flight regime by avoiding extreme accelerations and bank angles. These constraints ensure stable flight operations, especially in challenging wind conditions.

Before delving into specific predictive control use cases, it's crucial to understand the relationship between airspeed, wind speed, and ground speed.

$$\mathbf{V}_g = \mathbf{V}_a + \mathbf{W}, \quad (3.1)$$

This relationship is often visualized using the wind triangle, as shown in Figure 3.1.

In this figure, we can see that for a given ground speed vector and wind vector, there are typically two possible airspeed solutions. These solutions represent different flight strategies:

- Solution 1: Flying more with the wind, resulting in a larger ground speed magnitude.
- Solution 2: Flying more against the wind, resulting in a smaller ground speed magnitude.

The choice between these solutions impacts the UAV's safety. A predictive controller can leverage this information to optimize the flight path based on mission requirements and environmental conditions.

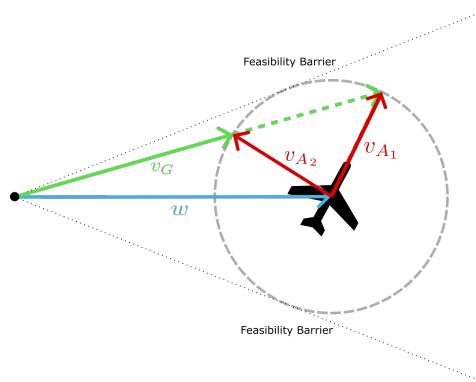


Figure 3.1: Wind Triangle with Two Airspeed Solutions

Consider Figure 3.2, which illustrates two UAVs navigating a circular path under wind conditions, demonstrating different flight strategies and their implications for ground speed and bank angle.

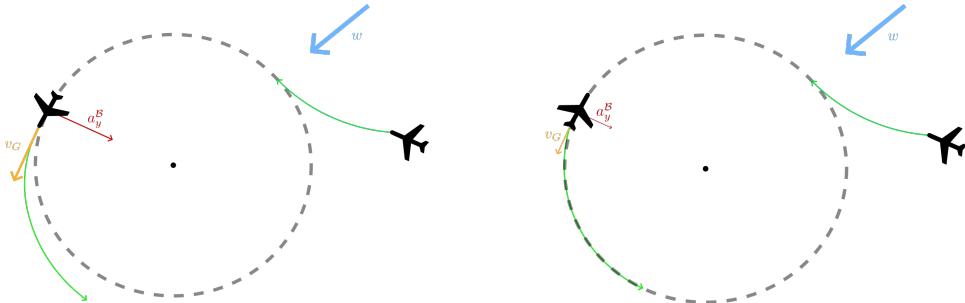


Figure 3.2: Predictive Approach: Flying With and Against Wind

1. Flying with the wind: The UAV moves in the direction of the wind, resulting in higher ground speed. This necessitates a steeper bank angle to generate the required centripetal force. However, if the required bank angle exceeds the UAV's safe operational limits, the aircraft may be unable to maintain the intended path, leading to deviations.
2. Flying against the wind: The UAV moves against the wind direction, reducing its ground speed. This allows for a smaller bank angle to maintain the turn radius, generally providing a larger safety margin.

A predictive controller can anticipate these conditions and optimize the UAV's trajectory, considering both wind impact and safety constraints. This aligns with the thesis objective stated in Section 1.3: developing an MPCC formulation that mitigates strong winds while maintaining safety constraints.

3.2 Fixed-Wing Model

This section explores the lateral kinematics and underlying assumptions for a rigid, fixed-wing UAV. The kinematic model presented here forms the foundation for the subsequent development of the MPCC.

3.2.1 System Description

The following fundamental assumptions define the system and its environment:

1. The operational area is sufficiently small to consider the Earth's surface flat, neglecting curvature effects.
2. Earth-induced effects, such as the Coriolis effect, are disregarded in the aircraft modeling.
3. The UAV performs coordinated turns¹.

The focus is on a planar scenario, effectively reducing the problem to two dimensions. Two coordinate systems are used to represent the problem.

1. Only horizontal wind is considered, with wind speed not exceeding airspeed unless otherwise stated.
2. Earth-fixed local North East Down (NE) Cartesian system: x-axis (East), y-axis (North).
3. Plane-fixed Front Right Down (FR) Cartesian system: x-axis (Front), y-axis (Right).
4. The model assumes parameters for a small fixed-wing UAV, with an existing flight controller preventing stall.

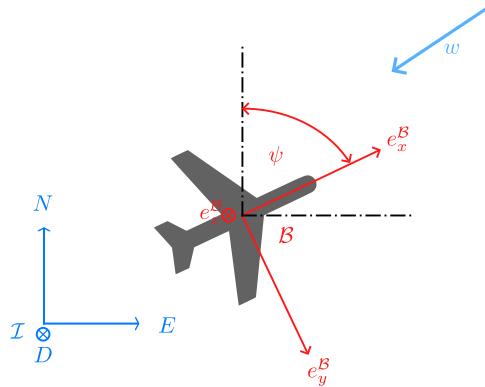


Figure 3.3: Frame convention used in this work: \mathcal{I} denotes the North East (NE) frame, while \mathcal{B} denotes the body-fixed Front Right (FR) frame.

3.2.2 Lateral Kinematic Model

The lateral motion of the fixed-wing UAV is described by the following state and input variables:

¹A coordinated turn combines yaw rate ($\dot{\psi}$) and roll (ϕ) such that there is no sideslip. ($v_y^{\mathcal{B}} = 0$)

State Variables

- e, n : Position in the inertial frame \mathcal{I} [m]
- $v_x^{\mathcal{B}}, v_y^{\mathcal{B}}$: Velocity components in the body frame \mathcal{B} [m/s]
- ψ : Yaw angle [rad]
- $a_x^{\mathcal{B}}, a_y^{\mathcal{B}}$: Acceleration components in the body frame \mathcal{B} [m/s²]

Input Variables

- u_x, u_y : Acceleration commands [m/s²]
- $u_{\dot{\psi}}$: Yaw rate command [rad/s]

The kinematics are derived as follows:

The UAV's velocity in the inertial frame \mathcal{I} is given by:

$$\begin{bmatrix} \dot{e} \\ \dot{n} \end{bmatrix} = \begin{bmatrix} \cos(\psi) & \sin(\psi) \\ \sin(\psi) & -\cos(\psi) \end{bmatrix} \begin{bmatrix} v_x^{\mathcal{B}} \\ v_y^{\mathcal{B}} \end{bmatrix} + \begin{bmatrix} w_e^{\mathcal{I}} \\ w_n^{\mathcal{I}} \end{bmatrix}, \quad (3.2)$$

where $w_e^{\mathcal{I}}$ and $w_n^{\mathcal{I}}$ are wind components in the inertial frame.

The UAV's acceleration in the body frame \mathcal{B} is expressed as:

$$\begin{bmatrix} \dot{v}_x^{\mathcal{B}} \\ \dot{v}_y^{\mathcal{B}} \end{bmatrix} = \begin{bmatrix} 0 & -\dot{\psi} \\ \dot{\psi} & 0 \end{bmatrix} \begin{bmatrix} v_x^{\mathcal{B}} \\ v_y^{\mathcal{B}} \end{bmatrix} + \begin{bmatrix} a_x^{\mathcal{B}} \\ a_y^{\mathcal{B}} \end{bmatrix}, \quad (3.3)$$

The use of first-order dynamics for longitudinal acceleration and second-order dynamics for lateral acceleration is simplifying assumptions about the UAV's behavior.

The evolution of the control inputs is modeled as:

$$\dot{\psi} = u_{\dot{\psi}}, \quad (3.4)$$

$$\dot{a}_x^{\mathcal{B}} = \frac{u_x - a_x^{\mathcal{B}}}{\tau}, \quad (3.5)$$

$$\ddot{a}_y^{\mathcal{B}} = \omega_n^2(u_y - a_y^{\mathcal{B}}) - 2\omega_n\zeta\dot{a}_y^{\mathcal{B}} \quad (3.6)$$

where τ is the time constant for longitudinal acceleration, and ω_n and ζ are the natural frequency and damping ratio for lateral acceleration dynamics, respectively.

A more detailed discussion of this modeling choice and its implications will be presented in Chapter 4.4.1 and 4.4.2.

3.2.3 Complete Kinematic Model

Combining all the derived equations, the complete lateral kinematic model is obtained:

$$\left\{ \begin{array}{l} \dot{e} = v_x^{\mathcal{B}} \cos(\psi) + v_y^{\mathcal{B}} \sin(\psi) + w_e^{\mathcal{I}}, \\ \dot{n} = v_x^{\mathcal{B}} \sin(\psi) - v_y^{\mathcal{B}} \cos(\psi) + w_n^{\mathcal{I}}, \\ \dot{v}_x^{\mathcal{B}} = a_x^{\mathcal{B}} - v_y^{\mathcal{B}} \dot{\psi}, \\ \dot{v}_y^{\mathcal{B}} = a_y^{\mathcal{B}} + v_x^{\mathcal{B}} \dot{\psi}, \\ \dot{\psi} = u_{\dot{\psi}}, \\ \dot{a}_x^{\mathcal{B}} = \frac{u_x - a_x^{\mathcal{B}}}{\tau}, \\ \ddot{a}_y^{\mathcal{B}} = \omega_n^2(u_y - a_y^{\mathcal{B}}) - 2\omega_n\zeta\dot{a}_y^{\mathcal{B}} \end{array} \right. \quad (3.7)$$

Chapter 4

Model Predictive Contouring Control

In this chapter, the foundation of Model Predictive Contouring Control (MPCC) as well as the low-level reference tracking models are presented.

4.1 Model Predictive Control

Model Predictive Control (MPC) is an advanced control technique that optimizes system behavior over a future time horizon. The core objective is to minimize the quadratic error between predicted states and inputs and their dynamically feasible references, while adhering to system constraints, initial conditions, terminal conditions and dynamics. The standard MPC optimization problem can be formulated as:

$$\underset{\substack{x_0, u_0, x_1, \dots, \\ u_{N-1}, x_N}}{\text{minimize}} \quad \sum_{k=0}^{N-1} \Delta \mathbf{x}_k^\top Q \Delta \mathbf{x}_k + \Delta \mathbf{u}_k^\top R \Delta \mathbf{u}_k + \Delta \mathbf{x}_N^\top P \Delta \mathbf{x}_N \quad (4.1a)$$

$$\text{subject to} \quad \mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k), \quad k = 0, \dots, N-1, \quad (4.1b)$$

$$g_k(\mathbf{x}_k, \mathbf{u}_k) \leq \mathbf{0}, \quad k = 0, \dots, N-1, \quad (4.1c)$$

$$g(\mathbf{x}_0, \mathbf{x}_N) = \mathbf{0} \quad (4.1d)$$

In this formulation, $\Delta \mathbf{x}_k = \mathbf{x}_k - \mathbf{x}_{k,\text{ref}}$ represents the state error, $\Delta \mathbf{u}_k = \mathbf{u}_k - \mathbf{u}_{k-1}$ the control input change, and $\Delta \mathbf{x}_N = \mathbf{x}_N - \mathbf{x}_{N,\text{ref}}$ the terminal state error. The weighting matrices $Q \succeq 0$, $R \succ 0$, and $P \succ 0$ penalize state deviations, control effort, and terminal state error, respectively.

It's worth noting that this formulation penalizes the change in control input ($\Delta \mathbf{u}_k$) rather than the input itself. An alternative approach would be to directly penalize the control input \mathbf{u}_k , replacing the term $\Delta \mathbf{u}_k^\top R \Delta \mathbf{u}_k$ with $\mathbf{u}_k^\top R \mathbf{u}_k$ in the cost function. The choice between these two forms depends on the specific control objectives and system characteristics.

The function $f(\mathbf{x}_k, \mathbf{u}_k)$ encapsulates the system dynamics in discrete time. The inequality constraint $g_k(\mathbf{x}_k, \mathbf{u}_k) \leq \mathbf{0}$ represents a general form of state and input constraints, which may include bounds on states and inputs. The equality constraint $g(\mathbf{x}_0, \mathbf{x}_N) = \mathbf{0}$ imposes conditions on the initial and terminal states.

Upon solving this optimization problem, the controller applies only the first computed input \mathbf{u}_0 to the system. The process then repeats at the next time step, with the optimization problem solved again using updated state information.

A significant challenge in applying MPC to guidance problems lies in ensuring the reference trajectory is dynamically feasible within the prediction horizon N . Infeasibility may arise if the reference is unattainable given the system constraints and dynamics. To address this, specialized MPC formulations for path following problems have been developed, such as Model Predictive Contouring Control (MPCC).

4.2 Model Predictive Contouring Control

To address the challenges of path following and mitigate the issues discussed in Section 4.1, MPCC is designed to handle path following tasks more effectively by separating the geometric aspects of the path from its temporal progression.

The path following problem for a fixed-wing UAV in wind conditions aims to balance three key objectives:

1. Minimizing deviation from a predefined spatial path
2. Following a desired constant airspeed
3. Compensating for wind disturbances

Unlike trajectory tracking, which defines references in both time and space, path following allows for temporal flexibility. This distinction is important when dealing with wind conditions, as it allows the controller to adapt itself without being constrained by strict time requirements.

4.2.1 Path Parameterization

The reference path is represented as a continuous curve in space, parameterized by a scalar variable θ .

The path \mathcal{P} is defined as:

$$\mathcal{P} = \{p(\theta) \in \mathbb{R}^2 : \theta \in [0, \theta_{max}]\}, \quad (4.2)$$

where $p(\theta)$ is a vector-valued function that maps the path parameter θ to a point in the 2-dimensional space, and θ_{max} represents the total path length or the final value of the path parameter.

For a two-dimensional path, $p(\theta)$ can be expressed as:

$$p(\theta) = [x(\theta), y(\theta)]^\top, \quad (4.3)$$

The tangent vector to the path at any point is given by:

$$\tau(\theta) = \frac{\partial p(\theta)}{\partial \theta} = [x'(\theta), y'(\theta)]^\top, \quad (4.4)$$

4.2.2 Cost Function Design

The MPCC arises from minimizing the true contouring error, which is the shortest distance from the current position to the path. Mathematically, this can be expressed as:

$$e_k^c = \min_{\theta} \|p_k - p(\theta)\|_2, \quad (4.5)$$

where $p_k = [x_k, y_k]^\top$ is the current position of the UAV, and $p(\theta)$ represents a point on the reference path parameterized by θ . The goal is to find θ_d , which gives the closest point $p(\theta_d)$ on the path to p_k .

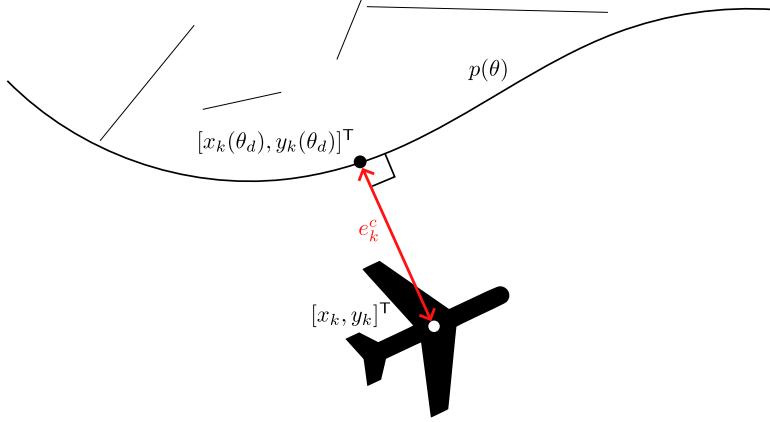


Figure 4.1: True Contouring Error

However, computing the true contouring error involves solving an optimization problem at each time step, which is computationally inefficient. Therefore, θ_d is approximated by $\hat{\theta}$, whose evolution is described by:

$$\hat{\theta}_{k+1} = \hat{\theta}_k + v_\theta \Delta T,$$

where v_θ represents the speed along the path and becomes part of the control input vector, and ΔT is the time step between consecutive time instants in the discretized system.

To address the challenge of not knowing the true contouring point θ_d during computation, the error is linearized around the current position. This linearization allows the formulation to be separated into two components: the linearized contouring error \hat{e}_k^c , which quantifies precision error, and the linearized lag error \hat{e}_k^l , which measures error along the path reference. These components are illustrated in Figure 4.2.

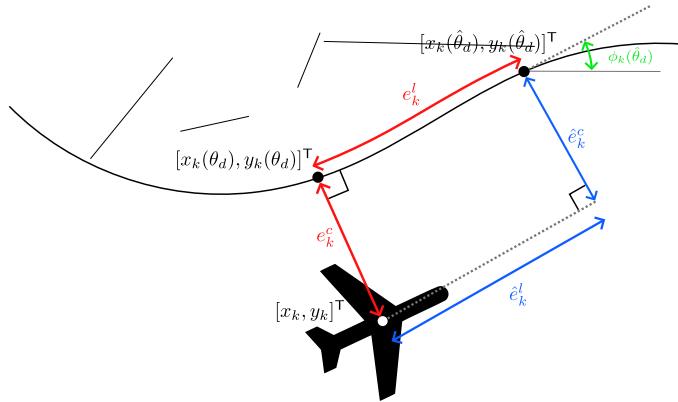


Figure 4.2: Linearized Lag and Contouring Error

The linearized contouring error is defined as the projected perpendicular distance from the aircraft to the reference point on the path. Conversely, the linearized

lag error is the parallel distance from the aircraft to this point. This linearization enables a more computationally efficient formulation of the path following problem.

These errors can be computed using the angle between the tangent of the linearized path at $\hat{\theta}_d$ and the horizontal axis. Given the tangent vector $\tau(\hat{\theta}_d) = [x'(\hat{\theta}_d), y'(\hat{\theta}_d)]^\top$, the angle is computed as:

$$\phi_k(\hat{\theta}_d) = \arctan \frac{y'(\hat{\theta}_d)}{x'(\hat{\theta}_d)}, \quad (4.6)$$

This angle then allows us to compute the linearized contouring and lag errors:

$$\hat{e}_k^c = \sin(\phi_k(\hat{\theta}_d)) \cdot (x_k - x_k(\hat{\theta}_d)) - \cos(\phi_k(\hat{\theta}_d)) \cdot (y_k - y_k(\hat{\theta}_d)), \quad (4.7)$$

$$\hat{e}_k^l = -\cos(\phi_k(\hat{\theta}_d)) \cdot (x_k - x_k(\hat{\theta}_d)) - \sin(\phi_k(\hat{\theta}_d)) \cdot (y_k - y_k(\hat{\theta}_d)), \quad (4.8)$$

where $[x_k, y_k]^\top$ is the current position of the aircraft and $[x_k(\hat{\theta}_d), y_k(\hat{\theta}_d)]^\top$ is the reference point on the path.

As previously mentioned, v_θ becomes an additional control input, and $\hat{\theta}$ is incorporated into the state vector. Consequently, the augmented state and control input vectors are defined as:

$$\mathbf{x} = [e, n, v_x^B, v_y^B, \psi, a_x^B, a_y^B, \hat{\theta}]^\top, \quad (4.9)$$

$$\mathbf{u} = [u_x, u_y, u_\psi, v_\theta]^\top, \quad (4.10)$$

In this MPCC formulation, $\hat{\theta}$ becomes an additional optimization state. This augmentation enables the controller to simultaneously pursue two objectives:

1. Minimize path-following errors using the control inputs $[u_x, u_y, u_\psi]^\top$
2. Maximize progress along the path by optimizing v_θ

The cost function of the MPCC incorporates several weighted components to balance path accuracy, progress, and control smoothness. Path-following errors, are weighted to penalize deviations from the path:

$$\begin{bmatrix} \hat{e}_k^c \\ \hat{e}_k^l \end{bmatrix}^\top \begin{bmatrix} q_c & 0 \\ 0 & q_l \end{bmatrix} \begin{bmatrix} \hat{e}_k^c \\ \hat{e}_k^l \end{bmatrix}, \quad (4.11)$$

where $q_c > 0$ and $q_l > 0$ are positive weighting scalars. To encourage forward motion along the path, a progress term is introduced:

$$-q_v v_\theta, \quad (4.12)$$

with $q_v > 0$ as a scalar weight. The negative sign indicates that increasing v_θ reduces the cost, thus promoting forward motion. Smooth control actions and prevention of excessive control effort are ensured by penalizing changes in control inputs:

$$\Delta \mathbf{u}_k^\top R \Delta \mathbf{u}_k, \quad (4.13)$$

where $\Delta \mathbf{u}_k = \mathbf{u}_k - \mathbf{u}_{k-1}$ represents the change in control inputs between consecutive time steps, and $R \in \mathbb{R}^{n_{\text{inputs}} \times n_{\text{inputs}}}$ is a diagonal positive definite matrix.

To mitigate the UAV's tendency to rush along the path, a penalty for deviating from a desired airspeed is introduced. This is incorporated into the cost function as:

$$e_{v,k}^\top r_v e_{v,k}, \quad (4.14)$$

where $e_{v,k} = v_d - |\mathbf{v}_k^B|_2$ represents the scalar airspeed error. Here, v_d is the desired airspeed magnitude, $|\mathbf{v}_k^B|_2$ is the magnitude of the current airspeed vector in the body frame, and $r_v > 0$ is a scalar weighting factor. This term encourages the UAV to maintain a specified airspeed magnitude while following the path.

Joining together all error components yields:

$$\mathbf{e}_k^T Q \mathbf{e}_k = \begin{bmatrix} \hat{e}_k^c \\ \hat{e}_k^l \\ e_{v,k} \end{bmatrix}^T \begin{bmatrix} q_c & 0 & 0 \\ 0 & q_l & 0 \\ 0 & 0 & r_v \end{bmatrix} \begin{bmatrix} \hat{e}_k^c \\ \hat{e}_k^l \\ e_{v,k} \end{bmatrix}$$

The cost function for the MPCC, combining all components, is given by:

$$J = \sum_{k=0}^{N-1} (\mathbf{e}_k^T Q \mathbf{e}_k - q_v v_{\theta,k} + \Delta \mathbf{u}_k^T R \Delta \mathbf{u}_k)$$

This formulation encapsulates the path-following objectives, progress along the path and control smoothness.

4.2.3 Constraint Formulation

The MPCC optimization problem incorporates constraints to ensure safe and feasible UAV operation. These include state limitations ($\underline{\mathbf{x}} \leq \mathbf{x}_k \leq \bar{\mathbf{x}}$), input boundaries ($\underline{\mathbf{u}} \leq \mathbf{u}_k \leq \bar{\mathbf{u}}$), initial conditions ($\mathbf{x}_0 = \mathbf{x}(t)$), and kinematic ($\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k)$). The kinematic equations, presented in Section 3.2.2, ensure the optimized trajectory adheres to the UAV's physical behavior.

These state and input constraints, as well as the initial condition constraint, can be expressed more compactly using vector-valued functions $g_k(\mathbf{x}_k, \mathbf{u}_k) \leq \mathbf{0}$ and $g(\mathbf{x}_0) = \mathbf{0}$, which encapsulate inequality and equality constraints respectively. This formulation provides a concise representation of all constraints within the optimization problem.

By enforcing these constraints, the MPCC generates trajectories that are not only optimal in terms of the cost function but also physically realizable and safe for the UAV to execute.

4.3 Optimization Problem

Combining everything together from 4.2.1, 4.2.2 and 4.2.3 the final optimal problem defining the MPCC is:

$$\begin{aligned} & \underset{\substack{x_0, u_0, x_1, \dots, \\ u_{N-1}, x_N}}{\text{minimize}} \quad J = \sum_{k=0}^{N-1} (\mathbf{e}_k^T Q \mathbf{e}_k - q_v v_{\theta,k} + \Delta \mathbf{u}_k^T R \Delta \mathbf{u}_k) \\ & \text{subject to} \quad \mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k), \quad k = 0, \dots, N-1, \\ & \quad g_k(\mathbf{x}_k, \mathbf{u}_k) \leq \mathbf{0}, \quad k = 0, \dots, N-1, \\ & \quad g(\mathbf{x}_0) = \mathbf{0} \end{aligned}$$

To summarize, the MPCC algorithm seeks to determine the optimal control horizon that balances multiple objectives, as illustrated in Figure 4.3. It aims to minimize the sum of the errors between the predicted UAV positions and their corresponding reference points on the path. Simultaneously, it strives to maximize progress along the path by pushing the reference points as far ahead as possible, while also ensuring smooth control actions by minimizing the differences between consecutive inputs.

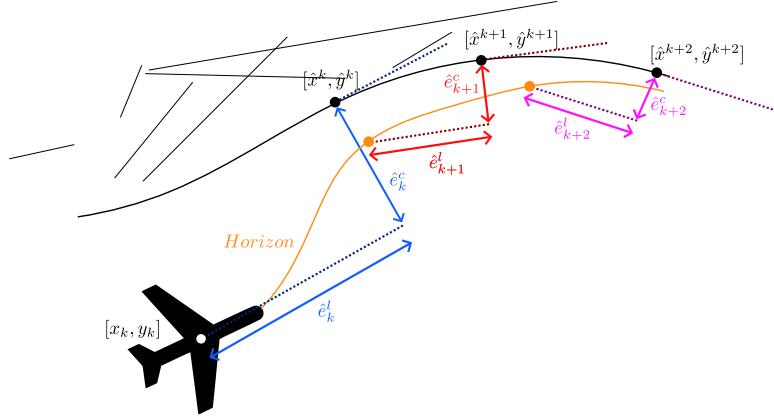


Figure 4.3: Optimal Horizon in MPCC

4.4 Reference Tracking

The execution of commands generated by the MPCC relies on tracking of acceleration references. As introduced in Section 3.2.2, models simulating the dynamic response of the UAV to control inputs have been implemented. These models account for the inherent delays and characteristics of the system, providing a more accurate representation of the UAV's response. It is important to note that these models and their parameters need to be identified and tuned for each specific UAV platform through flight testing, as the dynamic characteristics can vary significantly between different aircraft designs. This section elaborates on these tracking models, detailing their structure and significance.

4.4.1 Thrust Tracking

For longitudinal acceleration (x-axis), a first-order lag model is employed, as shown in Figure 4.4. This approach simulates a system response that approaches the commanded value over time, rather than instantaneously. It represents the gradual change in the UAV's acceleration due to the dynamics of the thrust system. The model can be described by the following differential equation:

$$\dot{a}_x^B = \frac{u_x - a_x^B}{\tau}, \quad (4.15)$$

where a_x^B is the actual longitudinal acceleration, u_x is the commanded acceleration from the MPCC, and τ is the time constant that characterizes the system's response time.

4.4.2 Roll Tracking

For lateral acceleration (y-axis), which involves rolling the aircraft, a second-order model is used, illustrated in Figure 4.5. This more complex model captures the slower dynamics involved in commanding a lateral acceleration: the aircraft must first deflect the ailerons and start the roll until a certain angle is archived before reaching the desired lateral acceleration. The model is described by:

$$\ddot{a}_y^B = \omega_n^2(u_y - a_y^B) - 2\omega_n\zeta\dot{a}_y^B, \quad (4.16)$$

where a_y^B is the actual lateral acceleration, u_y is the commanded acceleration from the MPCC, ζ is the damping ratio, and ω_n is the natural frequency of the system.

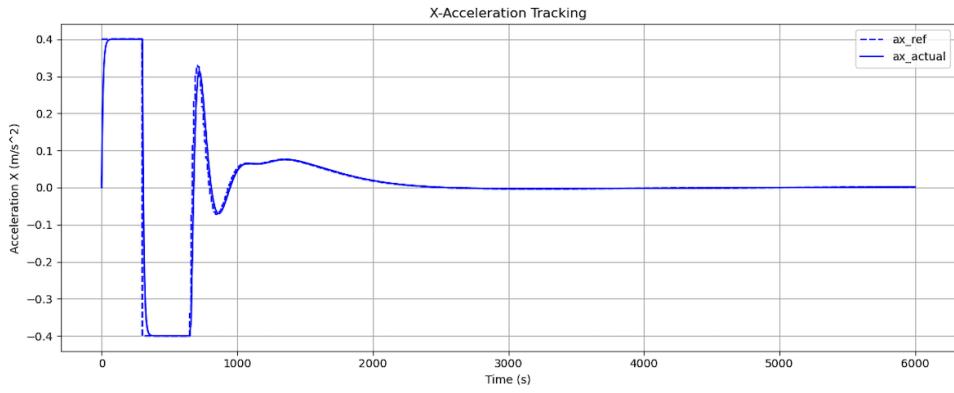


Figure 4.4: Thrust tracking

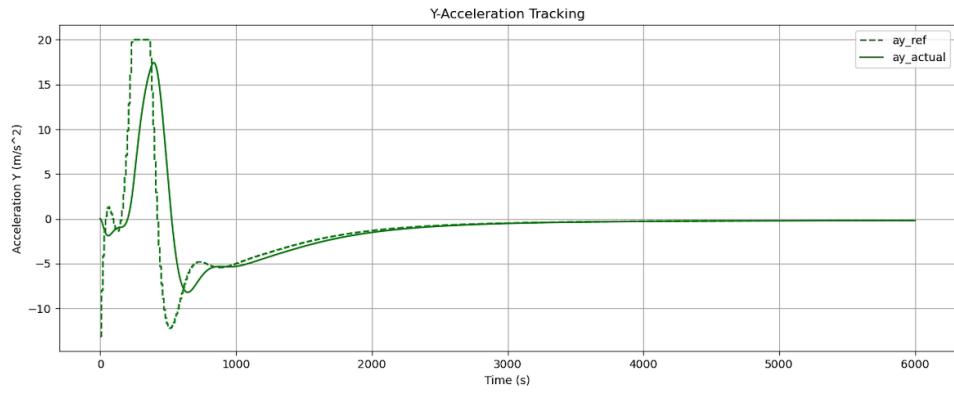


Figure 4.5: Roll tracking

As seen in Figure 4.4 and 4.5, the roll tracking model exhibits a more pronounced delay and overshoot compared to the thrust tracking model, reflecting the more complex dynamics involved in lateral maneuvers.

Chapter 5

Results

This chapter presents the evaluation results of the controller. The structure is as follows: First, the simulation setup and solver are introduced. Then, the UAV parameters used throughout the evaluation are specified. Finally, a scenario analysis is conducted, presenting and discussing the results of various test conditions.

5.1 Simulation Setup

The simulation environment for developing and evaluating the MPCC algorithm was implemented in Python, utilizing CasADi¹ [15] for symbolic problem formulation and the ACADOS² [16] python interface for nonlinear program solving.

For the MPCC, a prediction horizon of $N = 40$ steps was chosen, with a time step of $\Delta T = 0.2$ seconds, resulting in a look-ahead time of 8 seconds. This values showed to give a good performance, balancing between computational efficiency and long range prediction. The desired airspeed v_d is set to 20 m/s.

To ensure realistic behavior of the UAV model, the following constraints were implemented:

$$\begin{aligned} 15 \text{ m/s} &\leq v_x^{\mathcal{B}} \leq 23 \text{ m/s}, \\ 0.0 \text{ m/s} &\leq v_y^{\mathcal{B}} \leq 0.0 \text{ m/s}, \\ -0.4 \text{ m/s}^2 &\leq u_x \leq 0.4 \text{ m/s}^2, \\ -20 \text{ m/s}^2 &\leq u_y \leq 20 \text{ m/s}^2, \\ -\pi/3 \text{ rad/s} &\leq u_{\dot{\psi}} \leq \pi/3 \text{ rad/s}, \\ 0 &\leq v_{\theta} \leq 50, \\ 0 &\leq \hat{\theta} \leq \theta_{\max} \end{aligned}$$

These constraints represent limitations on velocity, longitudinal and lateral acceleration, yaw rate, and path parameter progression, respectively.

The constraints implemented in the simulation might appear overly permissive at first glance. However, they are designed to accommodate extreme scenarios that could arise during flight. Let's examine one such scenario to illustrate the necessity of these constraints: Consider a UAV flying along a circular path under the following conditions:

¹<https://web.casadi.org>

²https://docs.acados.org/python_interface/index.html

- Body-frame velocity: $v_x^B = 20 \text{ m/s}$, $v_y^B = 0 \text{ m/s}$ (assuming no side-slip)
- Tailwind: $w = 20 \text{ m/s}$
- Circle radius: $R = 100 \text{ m}$

Using the wind, ground, and airspeed relationship from equation (3.1), the maximum ground speed will be:

$$v_g = v_x^B + w = 20 + 20 = 40 \text{ m/s} \quad (5.1)$$

To maintain this circular path, the UAV must generate a centripetal acceleration:

$$a_{centripetal} = \frac{v_g^2}{R} = \frac{40^2}{100} = 16 \text{ m/s}^2 \quad (5.2)$$

Assuming a coordinated turn, this acceleration corresponds to a bank angle of:

$$\phi = \arctan \frac{a_{centripetal}}{g} \approx 60^\circ \quad (5.3)$$

It's evident that, considering the high wind scenarios, high accelerations, and consequently high bank angles, have to be archived in order to follow paths.

The cost function weights were tuned to balance the various control objectives:

$q_c = 50.0$	(contouring error),
$q_l = 20.0$	(lag error),
$r_1 = 1.0$	(longitudinal acceleration change),
$r_2 = 10.0$	(lateral acceleration change),
$r_3 = 10.0$	(yaw rate change),
$q_v = 100.0$	(path progression),
$r_v = 5.0$	(airspeed error)

These weights were chosen to prioritize path following accuracy (contouring and lag errors) while maintaining smooth control inputs and desired airspeed. The balance between progress and airspeed regulation in the cost function is critical for optimal path following. It's essential that the progress weight exceeds the airspeed weight. Otherwise, the controller may prioritize maintaining a constant airspeed over progressing along the path, leading to suboptimal behavior. Figure 5.1 illustrates this issue.

In the scenario depicted, the controller prioritizes maintaining a constant airspeed over progress along the path, resulting in the UAV effectively turning and hovering in place when facing strong tailwinds. While this behavior satisfies the airspeed constraint, it clearly compromises the primary objective of path following. As a consequence, the prediction horizon wraps around the circular path, causing the UAV's trajectory to deviate significantly from the intended route.

To provide a clear overview of the simulation parameters, Table 5.1 summarizes the key settings used in the MPCC implementation. These parameters, including cost function weights, MPCC configuration, and system constraints, are consistently applied across all scenario analyses unless explicitly stated otherwise.

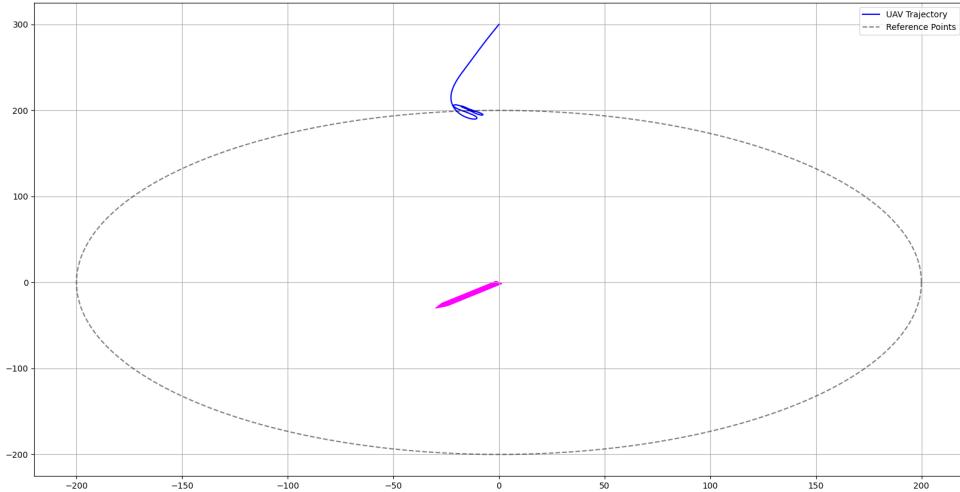


Figure 5.1: Path following results with insufficient incentive to progress along the path. The wind vector is represented by the pink arrow, while the desired airspeed is approximately 20 m/s.

Table 5.1: MPCC Parameters and Constraints

Cost Function Weights	
Contouring error q_c	50.0
Lag error q_l	20.0
Longitudinal acceleration r_1	1.0
Lateral acceleration r_2	10.0
Yaw rate r_3	10.0
Path progression q_v	100.0
Airspeed error r_v	5.0
MPCC Parameters	
Prediction horizon N	40
Time step ΔT	0.2 s
Look-ahead time T_f	8 s
Desired airspeed v_d	20 m/s
Constraints	
Body-frame x-velocity v_x^B	[15, 23] m/s
Body-frame y-velocity v_y^B	0 m/s
Longitudinal acceleration u_x	[-0.4, 0.4] m/s ²
Lateral acceleration u_y	[-20, 20] m/s ²
Yaw rate $u_{\dot{\psi}}$	[- $\pi/3$, $\pi/3$] rad/s
Path progression rate v_θ	[0, 50]
Path parameter $\hat{\theta}$	[0, θ_{\max}]

5.1.1 ACADOS

ACADOS was selected as the core framework for solving the optimal control problem. This open-source software package is not a solver itself, but rather a collection of algorithms and tools that facilitate the formulation and solution of optimal control problems.

Primarily, ACADOS is designed for rapid solutions of optimal control problems. The availability of a Python interface allows for seamless integration with our simulation environment, facilitating rapid coding.

Three key settings in ACADOS have been important in making the MPCC implementation work.

Firstly, the Sequential Quadratic Programming (SQP) method was chosen as the primary algorithm for solving our nonlinear optimization problem. SQP is particularly well-suited for our application due to its ability to handle nonlinear constraints effectively and its favorable convergence properties[17].

Secondly, partial condensing was employed to integrate the state dynamics into the cost function. This choice was motivated by the relatively long prediction horizon used in our formulation. While full condensing can be beneficial for shorter horizons, it increased the computational burden.

Lastly, the High-Performance Interior Point Method (HPIPM) was selected as the Quadratic Programming (QP) solver for each SQP iteration. Extensive numerical experiments[18] have demonstrated HPIPM's reliability in solving challenging QPs.

Algorithm 1 outlines the high-level implementation of the MPCC approach.

Algorithm 1 Model Predictive Contouring Control Implementation

```

1: Initialize  $X_0, T_{sim}, \Delta t, path, model$ 
2:  $X_{warm}, U_{warm} \leftarrow \text{WarmStart}(X_0, model, path)$ 
3:  $X_{current} \leftarrow X_0$ 
4:  $t \leftarrow 0$ 
5: while  $t < T_{sim}$  do
6:    $X_{opt}, U_{opt} \leftarrow \text{SolveMPCC}(X_{warm}, U_{warm}, X_{current}, model, path, sqp_rti)$ 
7:    $u_{apply} \leftarrow U_{opt}[0]$ 
8:    $x_{next} \leftarrow \text{IntegrateSystem}(x_{current}, u_{apply}, \Delta t)$ 
9:    $X_{warm}, U_{warm} \leftarrow \text{ShiftHorizon}(X_{opt}, U_{opt}, \Delta t, model)$ 
10:   $t \leftarrow t + \Delta t$ 
11: end while

```

The warm start procedure is a critical component of the MPCC implementation, significantly enhancing the solver's efficiency and convergence. It aims to find a good initial estimate for the optimal states and control inputs before the main control loop begins. This is achieved by iteratively solving through SQP the MPCC problem while maintaining a constant initial state until the cost function converges to a minimum.

The warm start procedure ensures an optimal or near-optimal initial solution, significantly enhancing the MPCC algorithm's performance. By iteratively solving the MPCC problem during warm start, the solution converges towards an optimal state before entering the main control loop. This approach results in minimal state changes between subsequent iterations.

These small inter-iteration changes create ideal conditions for employing SQP_RTI (Sequential Quadratic Programming with Real-Time Iteration) within the main

Algorithm 2 Warm Start Procedure for MPCC

Require: $X_0, model, path, N, \epsilon, iter_{max}$

- 1: Initialize $X_{warm} \leftarrow X_0$
- 2: Initialize $U_{warm} \leftarrow [0, \dots, 0]$ ▷ N zero vectors
- 3: $cost_{prev} \leftarrow \infty$
- 4: $iter \leftarrow 0$
- 5: **while** $iter < iter_{max}$ **do**
- 6: $X_{opt}, U_{opt}, cost \leftarrow \text{SolveMPCC}(X_{warm}, U_{warm}, X_0, model, path, sqp_full)$
- 7: **if** $|cost - cost_{prev}| < \epsilon$ **then**
- 8: **break** ▷ Convergence achieved
- 9: **end if**
- 10: $X_{warm} \leftarrow X_{opt}$
- 11: $U_{warm} \leftarrow U_{opt}$
- 12: $cost_{prev} \leftarrow cost$
- 13: $iter \leftarrow iter + 1$
- 14: **end while**
- 15: **return** X_{warm}, U_{warm}

loop. SQP_RTI is particularly effective in this scenario as it performs only one quadratic programming (QP) problem per sampling time, exploiting the similarity between subsequent problems in the receding horizon approach. This method significantly reduces computational load.

The final step in each iteration is to integrate the system state using the first control input determined by the MPCC. This integration is performed using the Explicit Runge-Kutta method provided by ACADOS. The method applies the equations of motion described in Section 3.2.2 within the function *IntegrateSystem* from 1 to compute the next state.

Subsequently, the prediction horizon is shifted to prepare for the next iteration. This shift is accomplished through linear interpolation between the penultimate and final states and inputs of the current optimal solution. The process involves removing the first state and input from the horizon and appending a new state and input at the end. This updated horizon then serves as the initial guess for the subsequent optimization iteration.

5.2 Scenario Analysis

To evaluate the MPCC's performance, a series of scenarios involving circular paths of varying radii under different wind conditions will be examined. The test environment is designed to progress from simple to challenging scenarios.

The circular path is chosen for its unique property of exposing the UAV to wind from all directions as it traverses the circuit. The scenarios are parameterized as follows:

- Wind direction: Constant from North-East
- Wind magnitude: Ranging from 0 m/s to 25 m/s
- Path radius: Varying from 50 m to 100 m
- Initial Condition: $E = 150$ m, $N = 0$ m, $v_x^B = 20$ m/s, $v_y^B = 0$ m/s and $\psi = 0$

Figure 5.2 illustrates the scenario setup, depicting the circular path and the wind direction.

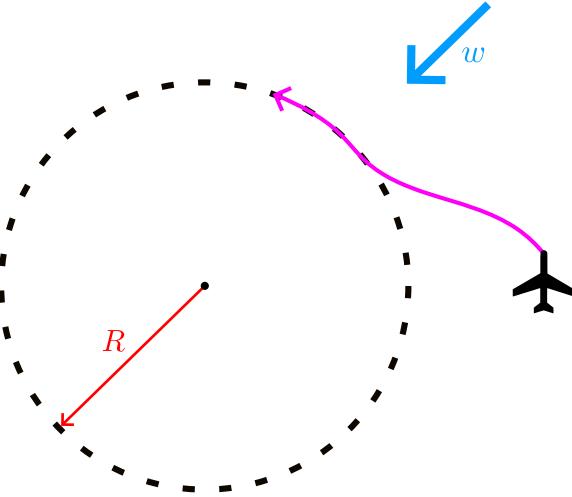


Figure 5.2: Scenario Analysis

This range of conditions enables the assessment of the MPCC's performance across various operational scenarios. Larger radii with lower wind speeds require less lateral acceleration and present smaller disturbances relative to the UAV's airspeed. Smaller radii combined with higher wind speeds necessitate greater lateral accelerations and involve wind disturbances that are more significant relative to the UAV's airspeed. In extreme cases, the controller may opt to fly backwards along the path when proceeding forward would exceed the feasible acceleration regions, as shown in Section 3.1.

5.3 Big Radius Conditions

5.3.1 Scenario 1: Large Radius - No Wind

This scenario establishes a baseline for the MPCC performance under ideal conditions:

- Wind speed: 0 m/s & Path radius: 100 m

Trajectory Analysis

Figure 5.3 demonstrates that the UAV follows the circular path with high precision. In the absence of wind, the plane maintains a tangential orientation to the path throughout its trajectory.

Control Inputs and states

Regarding the control inputs and velocity states, it can be observed in Figure 5.3 that v_x^B remains within bound as it attempts to accelerate rapidly to reach the path. In contrast, the v_y^B velocity remains approximately 0.0 throughout the entire simulation due to the enforced zero side-slip condition. Control inputs remain stable around the cruising point.

5.3.2 Scenario 2: Large Radius - High Wind

This scenario tests the MPCC's performance under high wind conditions

- Wind Speed = 14 m/s & Path radius: 100 m

Trajectory Analysis

As illustrated in Figure 5.4, the UAV demonstrates precise path-following precision despite the strong wind. When encountering crosswind conditions, the controller adapts the UAV's orientation to counteract the wind effect.

Control Inputs and States

The control inputs and state variables, shown in Figure 5.4, exhibit characteristics similar to Scenario 1, with predominantly smooth transitions. However, a notable distinction is observed in the longitudinal acceleration command (u_x). The plot reveals that u_x reaches its maximum value, indicating that the controller commands full throttle to overcome the wind resistance and maintain the desired path.

5.4 Small Radius Conditions

5.4.1 Scenario 3: Small Radius - High Wind

This scenario presents a more challenging environment for the MPCC, combining a tighter turning radius with high wind conditions. The increased difficulty arises from the mismatch between constant ground speed requirements and the heightened acceleration demands of the smaller radius.

- Wind speed: 14 m/s & Path radius: 50 m

Trajectory Analysis

Figure 5.5 reveals a notable initial overshoot as the UAV enters the circular path, likely due to excess entry speed. However, the controller reduces speed and achieves precise path following for the remainder of the trajectory.

Control Inputs and states

In the control input is particularly evident in the u_y plot of Figure 5.5, reveals aggressive acceleration maneuvers necessitated to counteract the initial overshoot.

5.4.2 Scenario 4: Small Radius - Maximum Wind

This scenario pushes the MPCC to its limits by introducing wind speeds approaching the UAV's maximum airspeed of 23 m/s, representing the most challenging conditions for controlled flight and path following.

- Wind speed: 20 m/s & Path radius: 50 m

Trajectory Analysis

Figure 5.6 demonstrates the MPCC's ability to maintain reasonable path adherence despite the challenging wind conditions. Notably, the controller opts for a counterintuitive but effective strategy: allowing backward motion along the path to reduce ground speed, thereby decreasing the required lateral acceleration and enabling tighter path following, as seen in 3.1

Control Inputs and States

During the downwind segment, the controller minimizes forward acceleration u_x , allowing the wind to drive the UAV along the path while controlling yaw to maintain the desired orientation.

5.4.3 Scenario 5: Small Radius - Excess Wind

This scenario exceeds the capabilities of the controller, as the controller is subjected to stronger winds than the airspeed. This results in the creation of no-fly zones, which are determined by the initial conditions. These no-fly zones are inaccessible because the plane's airspeed cannot overcome the speed of the wind.

- Wind speed: 25 m/s & Path radius: 50 m

Trajectory Analysis

In the trajectory, it's evident that the plane can't progress to the upper path of the circle. Consequently, the controller minimizes the contour error by crossing the circle and eventually finding an alternative solution: crossing the circle completely until the other side. At that point, the controller aggressively fights against the wind until it comes directly perpendicular to the path, then the airspeed limitation causes the plane to drift away from the path.

Control Inputs and States

In the control side, it's evident that the throttle remains fully engaged until the plane begins to drift away. At that point, the plane loses control and cannot take any further action.

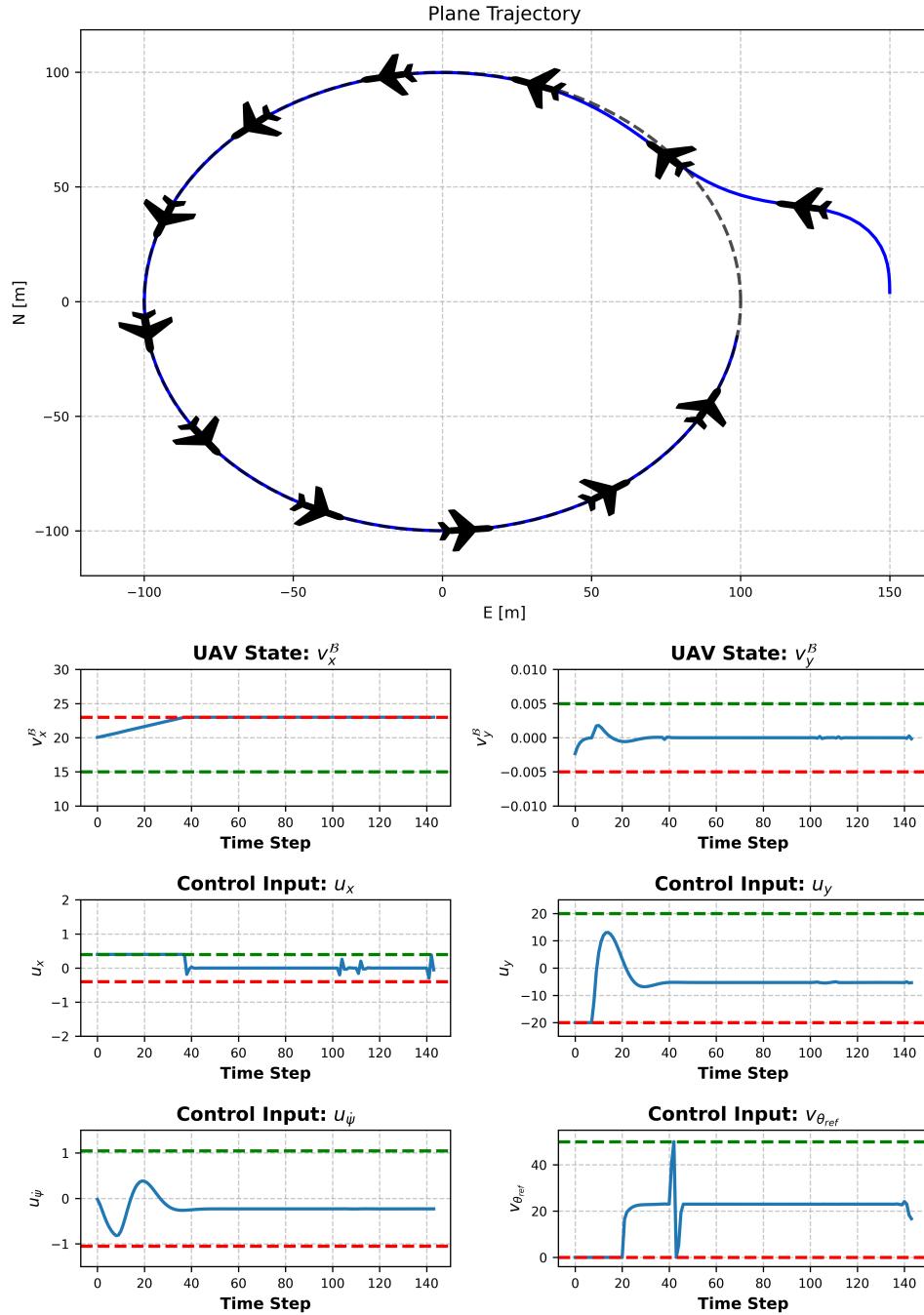


Figure 5.3: Scenario 1: Large Radius (100 m) - No Wind. Observe precision path following, tangential orientation, and stable control inputs. Note v_x^B acceleration to reach the path and v_y^B maintained at approximately 0.0.

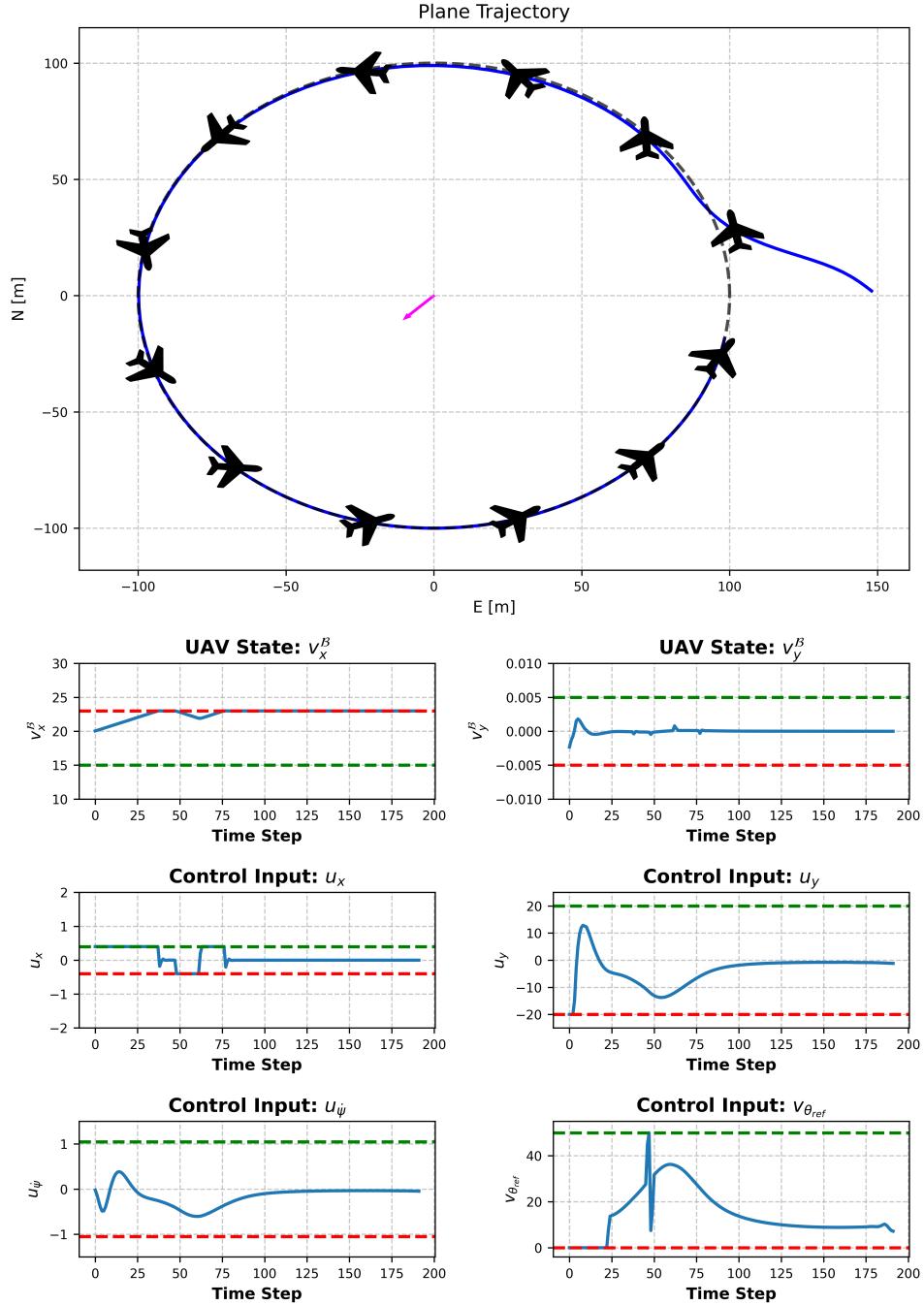


Figure 5.4: Scenario 2: Large Radius (100 m) - High Wind (14 m/s). Note precise path following despite strong wind, adaptive UAV orientation in crosswind, and u_x reaching maximum for wind resistance.

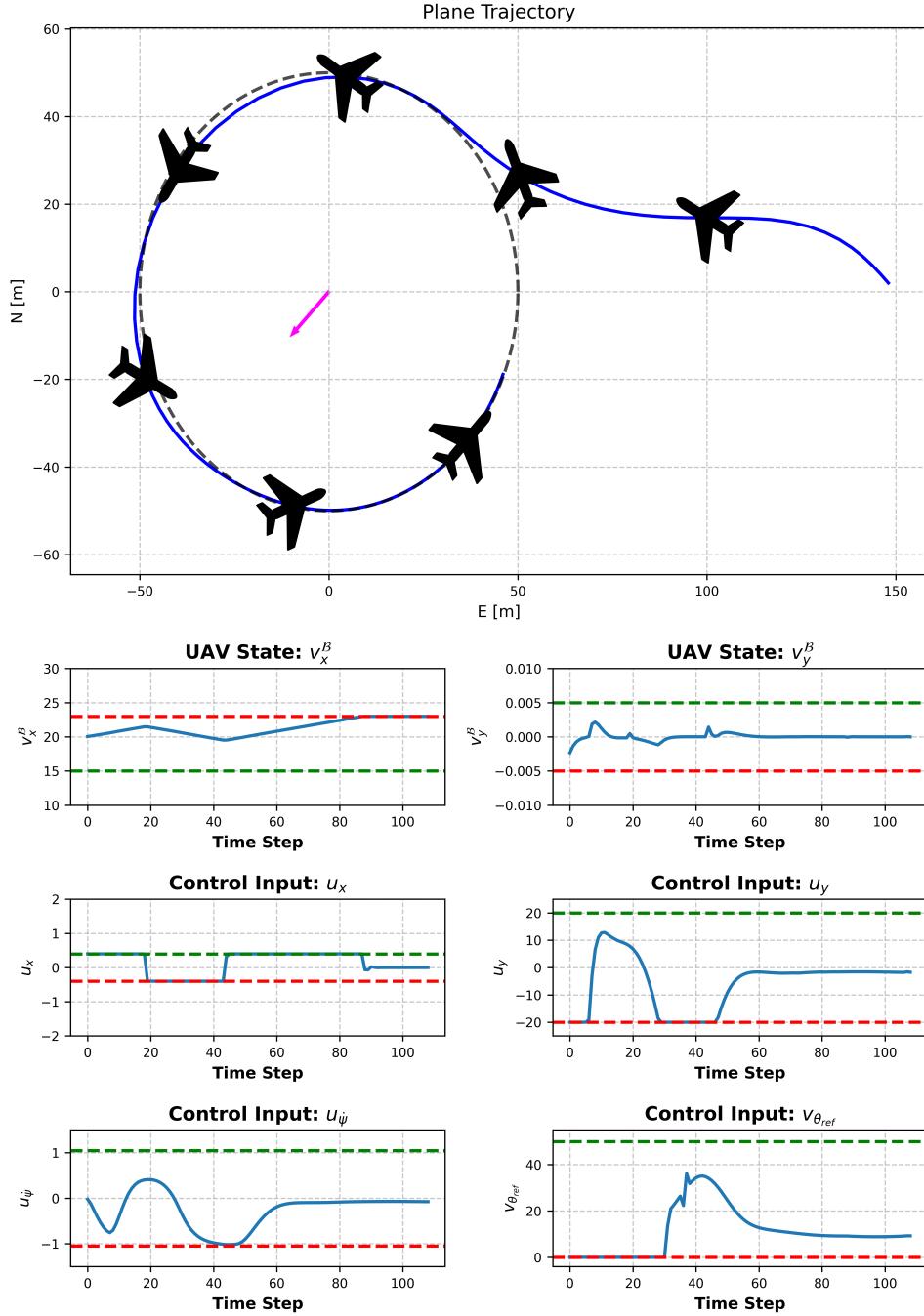


Figure 5.5: Scenario 3: Small Radius (50 m) - High wind (14 m/s). Observe initial overshoot due to excess entry speed, followed by precise path following. Note aggressive u_y maneuvers to counteract overshoot.

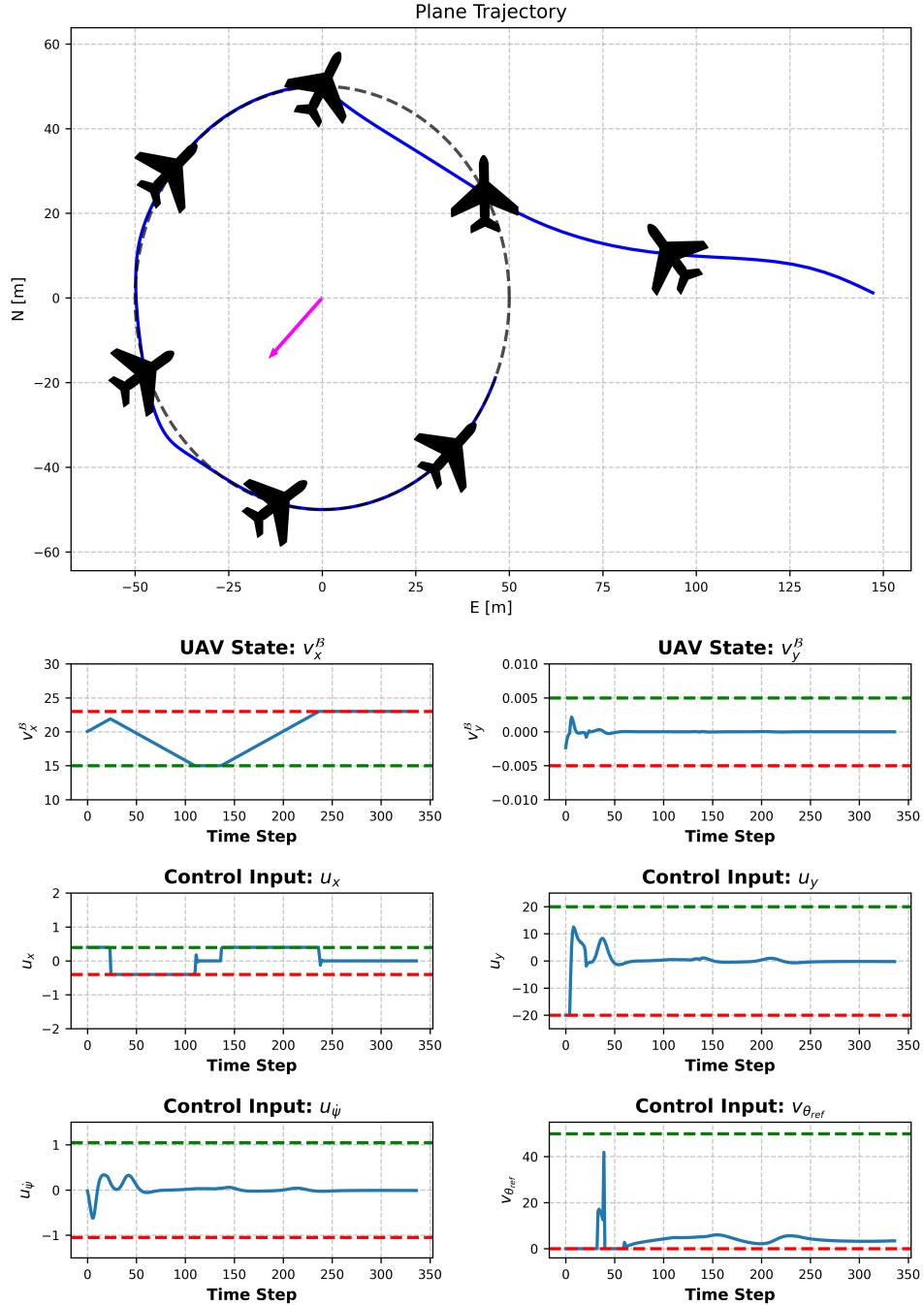


Figure 5.6: Scenario 4: Small Radius (50 m) - Maximum Wind (20 m/s). Note reasonable path adherence despite challenging conditions. Observe backward motion strategy in downwind segments to reduce ground speed and less lateral acceleration.

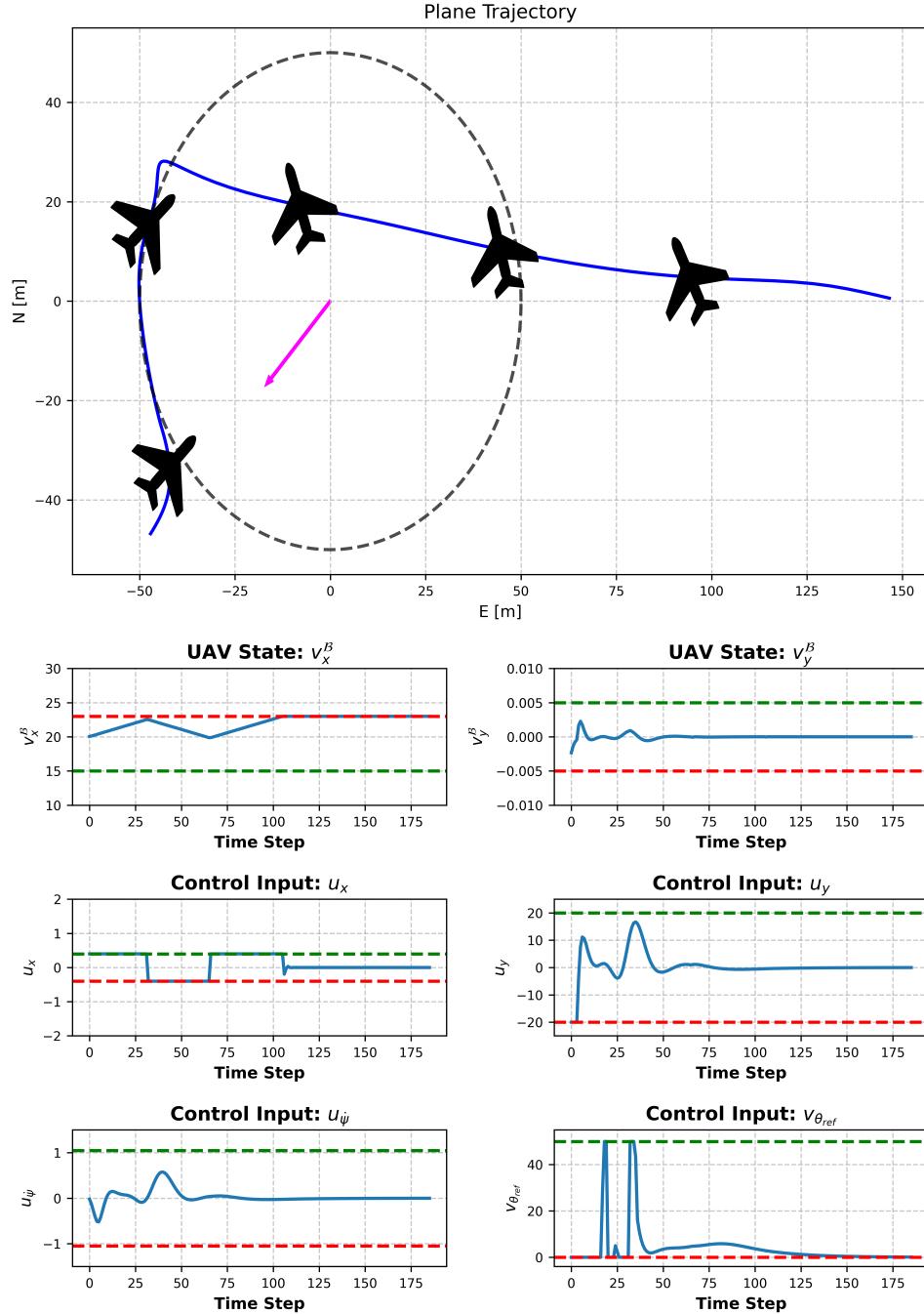


Figure 5.7: Scenario 5: Small Radius (50 m) - Excess Wind (24 m/s). Observe creation of no-fly zones due to wind exceeding airspeed. Note the controller's strategy: crossing the circle to minimize contour error, aggressive wind resistance, and eventual loss of control.

5.5 Arbitrary conditions and paths

While previous sections focused on circular paths and specific initial conditions, this section demonstrates the controller's capability in handling more complex scenarios.

The controller exhibits high precision when following arbitrary paths, even in challenging conditions such as high wind speeds. Figure 5.8 illustrates the controller's performance on an arbitrary path under a significant wind speed of 14 m/s.

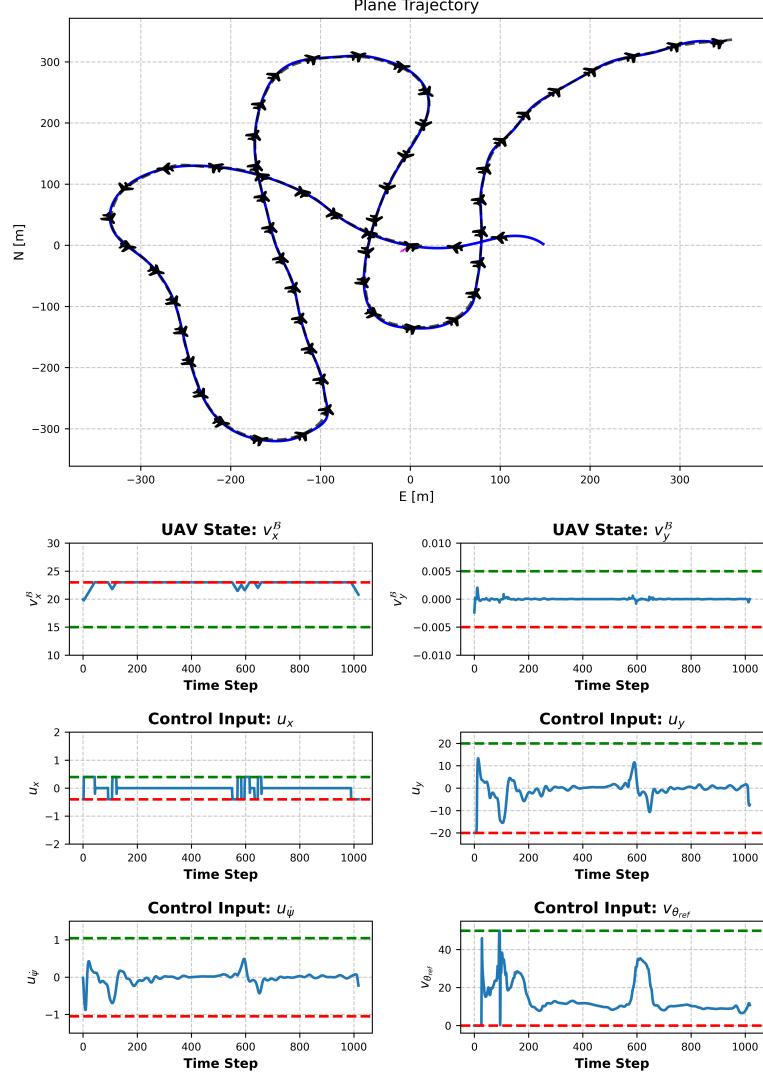


Figure 5.8: Scenario 6: Arbitrary Path - High Wind (14 m/s). Observe the controller's maintaining path adherence despite strong wind, compensating more in high-curvature regions. Note slight overshoots in sharp turns and tighter tracking in straight segments.

The controller also shows ability to converge to the desired path from a wide range of initial conditions. Noticeable from the 5.9 is the wrapping already mentioned on 5.1 on the top left quarter, where the plane has tailwind sometimes does not have enough incentive to progress. Addressing this issue will require a more comprehensive analysis of the control law's behavior in tailwind conditions.

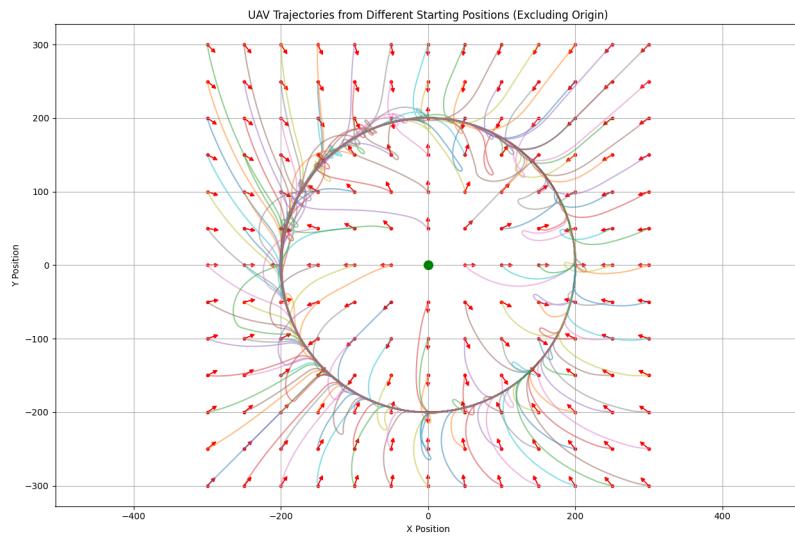


Figure 5.9: Evaluation of different initial conditions with a 14 m/s wind coming from the North-East and desired airspeed of 20 m/s. Note the wrapping issue in the top left quarter of the circle.

Chapter 6

Conclusion and Outlook

6.1 Limitations of the Work

This work, while comprehensive in its approach to developing and testing the MPCC, has several limitations that should be acknowledged. Primarily, the results are based on simulations which do not capture all the real-world flying conditions or the physics of a UAV. The wind models used in our simulations, still represent simplifications of real atmospheric conditions as it is assumed to have a known constant magnitude and direction. More complex wind patterns, present additional challenges not fully addressed in this work.

Furthermore, the study assumes ideal tracking performance and actuator responses. In practice, sensor noise, actuator lag, and other hardware limitations could impact controller performance.

6.1.1 Future Work

The development and evaluation of this controller have highlighted several areas for improvement in future iterations. A key priority is the incorporation of more complex and realistic wind and UAV models, including varying wind magnitudes and directions, as well as introducing uncertainty in the model.

Testing the controller in a more advanced simulation environment is also crucial. This environment should incorporate a wider range of model mismatches. Such testing would help bridge the gap between controlled experiments and practical applications, ensuring the controller's effectiveness in actual flight conditions. A testing pipeline in SiL (Software in the Loop) would need to be setup by linking the controller to *PX4* and ultimately testing on a real platform.

The controller itself requires further refinement in several aspects. Developing a separated velocity controller could ensure more consistent velocity tracking, addressing issues observed in Chapter 5. The current controller, while penalizing deviations from the desired airspeed as defined in Section 4.2.2, still tends to prioritize path tracking over maintaining consistent velocity. This balancing act between airspeed constraints and path progress is an ongoing issue that leads to trajectory "wrinkling" in various situations, as seen in Figure 5.1.

Special attention should be given to improving the controller's performance in tail-wind scenarios, where it sometimes struggles to maintain forward progress.

6.2 Concluding Remarks

The constraint-aware Model Predictive Contouring Control framework developed in this thesis demonstrates significant potential for enhancing fixed-wing UAV path following in high wind conditions. Through the evaluations, it has been shown that the MPCC can effectively balance path accuracy, forward progress, and strict adherence to safety constraints and initial conditions across a range of challenging scenarios.

Key achievements include the controller's ability to maintain precision in circular and arbitrary paths under various wind conditions while always respecting defined operational limits, its adaptability in extreme winds by allowing backward motion within safe parameters and its robustness across different initial conditions, which is not possible with simpler methods like L_1 .

However, this work also reveals areas for further development. These include refining the controller's performance in tailwind scenarios, improving constant airspeed following, and addressing the occasional "wrinkling" effect observed in certain conditions, all while maintaining the strict constraint awareness. Future work should focus on incorporating more complex wind models, testing in advanced simulation environments and ultimately transitioning to real-world flight tests where physical limitations become even more critical.

In conclusion, this constraint-aware MPCC framework represents a promising approach for fixed-wing UAV path following. While there's room for refinement, the method effectively balances safety constraints with path accuracy. A key challenge remains in reconciling the controller's emphasis on path progress with the desire for constant airspeed in UAV operations.

Bibliography

- [1] Anthropic, “Claude,” ai assistant developed by Anthropic, accessed through web interface or API.
- [2] 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*, vol. 30, no. 6, pp. 1718–1728, 2007.
- [3] S. Park, J. Deyst, and J. How, “A new nonlinear guidance logic for trajectory tracking,” 08 2004.
- [4] T. Stastny, “L1 guidance logic extension for small uavs: handling high winds and small loiter radii,” *CoRR*, vol. abs/1804.04209, 2018.
- [5] 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)*, 2017, pp. 4254–4261.
- [6] Y. Kang and J. K. Hedrick, “Linear tracking for a fixed-wing uav using non-linear model predictive control,” *IEEE Transactions on Control Systems Technology*, vol. 17, no. 5, pp. 1202–1210, 2009.
- [7] J. Osborne and R. Rysdyk, *Waypoint Guidance for Small UAVs in Wind*.
- [8] D. Reinhardt and T. A. Johansen, “Nonlinear model predictive attitude control for fixed-wing unmanned aerial vehicle based on a wind frame formulation,” in *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2019, pp. 503–512.
- [9] T. J. Stastny, A. Dash, and R. Siegwart, *Nonlinear MPC for Fixed-wing UAV Trajectory Tracking: Implementation and Flight Experiments*.
- [10] C. Chen and J. Tan, “Path following for uav using nonlinear model predictive control,” *Journal of Physics: Conference Series*, vol. 2530, no. 1, p. 012021, jun 2023.
- [11] T. Faulwasser, B. Kern, and R. Findeisen, “Model predictive path-following for constrained nonlinear systems,” in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, 2009, pp. 8642–8647.
- [12] D. Lam, C. Manzie, and M. Good, “Model predictive contouring control,” in *49th IEEE Conference on Decision and Control (CDC)*, 2010, pp. 6137–6142.
- [13] A. Liniger, A. Domahidi, and M. Morari, “Optimization-based autonomous racing of 1:43 scale rc cars,” *Optimal Control Applications and Methods*, vol. 36, no. 5, p. 628–647, Jul. 2014.

- [14] A. Romero, S. Sun, P. Foehn, and D. Scaramuzza, “Model predictive contouring control for time-optimal quadrotor flight,” 2022.
- [15] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, “CasADI – A software framework for nonlinear optimization and optimal control,” *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.
- [16] R. Verschueren, G. Frison, D. Kouzoupis, J. Frey, N. van Duijkeren, A. Zanelli, B. Novoselnik, T. Albin, R. Quirynen, and M. Diehl, “acados – a modular open-source framework for fast embedded optimal control,” *Mathematical Programming Computation*, 2021.
- [17] P. T. Boggs and J. W. Tolle, “Sequential quadratic programming,” *Acta Numerica*, vol. 4, p. 1–51, 1995.
- [18] G. Frison and M. Diehl, “Hpipm: a high-performance quadratic programming framework for model predictive control,” 2020.