Czech Technical University in Prague

Faculty of Electrical Engineering
Department of Cybernetics
Multi-robot Systems

Interception Trajectory Planning for Unmanned Aerial Vehicles

Master's Thesis

Michal Pliska

Prague, May 2023

Study programme: Cybernetics and Robotics
Supervisor: Ing. Matouš Vrba

Acknowledgments

I would like to express my gratitude to my supervisor, Matouš Vrba, for his guidance and support throughout this thesis. I also appreciate the assistance of my lab colleagues, particularly Tomáš Bača, in addressing various issues during my research and experiments. Additionally, I am thankful for the friendly atmosphere in the lab. Finally, I acknowledge the support and encouragement from my parents and family in my academic pursuits.

Author statement for undergraduate thesis

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

Prague, May 26, 2023

MASTER‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name:	**Pliska Michal**	Personal ID number:	**483708**
Faculty / Institute:	**Faculty of Electrical Engineering**		
Department / Institute:	**Department of Cybernetics**		
Study program:	**Cybernetics and Robotics**		

II. Master’s thesis details

Master’s thesis title in English:

Interception Trajectory Planning for Unmanned Aerial Vehicles

Master’s thesis title in Czech:

Plánování odchytové trajektorie pro bezpilotní letouny

Guidelines:

Formulate the task of planning a trajectory for autonomously catching a flying vehicle by an UAV as a Model Predictive Control problem. Design and implement an MPC-based planner for this task and test it in realistic simulated experiments. Compare performance of the designed solution with other planning approaches such as non-linear optimization and proportional navigation in simulations. Evaluate these approaches using suitable metrics.

Bibliography / sources:

[1] Yanushevsky, Rafael, „Modern Missile Guidance,“ CRC Press, 2007.
[2] M. Guelman, „Proportional Navigation with a Maneuvering Target,“ in Transactions on Aerospace and Electronic Systems, vol. AES-8, no. 3, pp. 364-371, 1972
[3] M. Vrba, D. Heřt and M. Saska, „Onboard Marker-Less Detection and Localization of Non-Cooperating Drones for Their Safe Interception by an Autonomous Aerial System,“ in Robotics and Automation Letters 4(4):3402-3409, 2019.
[4] T. Báča, D. Heřt, G. Loianno, M. Saska and V. Kumar, „Model Predictive Trajectory Tracking and Collision Avoidance for Reliable Outdoor Deployment of Unmanned Aerial Vehicles,“ in International Conference on Intelligent Robots and Systems, 2018.
[5] C. Richter, A. Bry, and N. Roy, „Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments,“ in International Journal of Robotics Research, 2016.

Name and workplace of master’s thesis supervisor:

Ing. Matouš Vrba   Multi-robot Systems FEE

Name and workplace of second master’s thesis supervisor or consultant:

Date of master’s thesis assignment: **17.01.2023**   Deadline for master's thesis submission: **26.05.2023**

Assignment valid until: **22.09.2024**

Ing. Matouš Vrba Supervisor’s signature	prof. Ing. Tomáš Svoboda, Ph.D. Head of department’s signature	prof. Mgr. Petr Páta, Ph.D. Dean’s signature

III. Assignment receipt

The student acknowledges that the master’s thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master’s thesis, the author must state the names of consultants and include a list of references.

.

Date of assignment receipt   Student’s signature

© ČVUT v Praze, Design: ČVUT v Praze, VIC

Abstract

This thesis focuses primarily on the design, development, and testing of novel methods for autonomous interception of intrusive UAVs using another UAV. An advanced detection and tracking mechanism is implemented using light detection and range (LIDAR) measurements from the interceptor UAV with respect to its agile flight characteristics. After validation, the technique is augmented using the Interactive Multiple Model (IMM) filter algorithm, which shows superior tracking performance of highly maneuverable targets.

For the trajectory planning, various algorithms, including Proportional Navigation (PN) and Nonlinear Model Predictive Control (NMPC), are implemented and compared to a baseline method, navigation based on nonlinear optimization (NLOPT). The PN method is adapted specifically for drone interceptions, and NMPC is utilized with a simple objective to leverage the benefits of this type of control.

Through rigorous testing and comparison in simulations, PN and NMPC notably outperforms the baseline algorithm, with NMPC showing exceptional performance across all metrics. PN is also successfully tested against a human-piloted target in a real-world scenario, highlighting the practical applicability and effectiveness of these methods in tackling real-world UAV interception challenges. This marks a significant milestone, validating the theoretical foundations of this research and also underlining its practical value in addressing the growing issue of intrusive drones.

Keywords Autonomous Aerial Interception Systems, Estimation and Filtration, Trajectory Planning

Abstrakt

Tato diplomová práce se primárně soustředí na návrh, vývoj a testování nových metod pro autonomní zachycení nekooperujících dronů pomocí jiného dronu. Je implementován pokročilý detekční a sledovací mechanismus, který využívá měření LIDAR (Light Detection and Ranging) z interceptoru s ohledem na jeho k agresivní létání. Tato metoda byla dále rozšířena pomocí algoritmu Interactive Multiple Model (IMM) filtru, který prokázal vynikající schopnost sledovat rychle manévrující cíle.

Za účelem plánování odchytové trajektorie byly implementovány a porovnány různé algoritmy, včetně Proportional Navigation (PN) a Nonlinear Model Predictive Control (NMPC), s výchozí metodou, navigací založenou na nelineární optimalizaci (NLOPT). Metoda PN, určená původně pro řízení naváděných raket, je specificky adaptována pro zachycení dronů a NMPC je implementováno s jednoduchým cílem využít výhod tohoto typu řízení.

V důkladném testování a porovnání v simulacích, PN a NMPC ukázalo, že tyto metody významně předčí výchozí algoritmus, přičemž NMPC ukazuje nejlepší výsledky ve všech metrikách. PN je také úspěšně testována proti cíli řízeném lidským pilotem v reálném nasazení, což zdůrazňuje praktickou použitelnost a účinnost těchto metod při řešení skutečných výzev během odchytu dronu. To představuje významný milník, který potvrzuje teoretické základy této práce a zároveň zdúrazňuje jejich praktickou hodnotu při řešení rostoucího problému nekooperujících dronů.

Klíčová slova Autonomní Letecké Záchytné Systémy, Odhad a Filtrace, Plánování Trajektorie

Contents

Chapter 1

Introduction

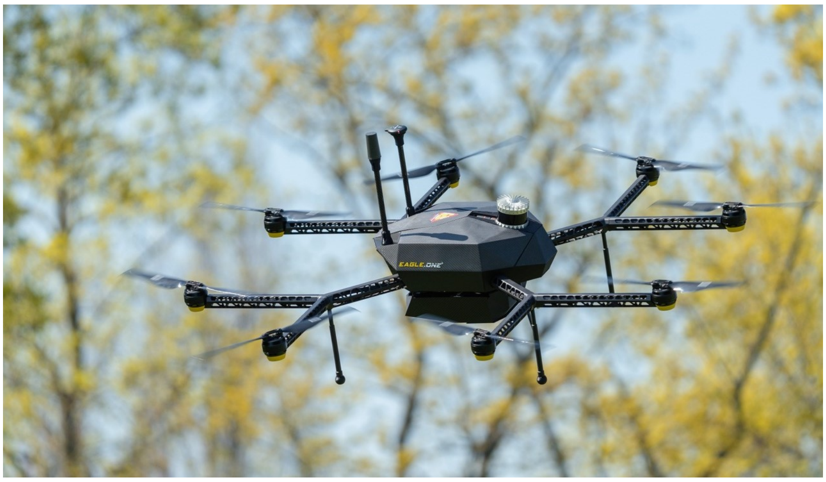


Figure 1.1: The UAVin the image (Eagle One,) aims to become a fully autonomous interceptor and is the main motivation for this work.

Over the past few years, there was tremendous progress in the development of multirotor UAVs. Advances in computational hardware and affordability made UAVs more accessible and capable than ever before. However, this progress has also introduced new challenges. As the number and capabilities of drones on the market continue to grow, there has been an increase in drone incidents, such as accidents, near-misses, and even intentional misuse. This leads to a negative perception of drones by the public [25]. With rising concerns about general safety, privacy, and misuse for criminal activities.

In December 2018, multiple drone sightings near London's Gatwick Airport caused significant disruption to air travel. The airport had to close its runway repeatedly over a period of three days, resulting in the cancellation or diversion of approximately 1,000 flights and affecting around 140,000 passengers. Despite extensive investigations, the perpetrators were never identified[1].

In August 2018, an alleged assassination attempt on the Venezuelan President Nicolás Maduro occurred during a military parade in Caracas. Two drones carrying explosives detonated near the parade, causing injuries to several military personnel. This incident demonstrated the potential for drones to be used in targeted attacks and raised concerns about their misuse for terrorist activities[2].

The number of such incidents is expected to increase as the field of drone technology

[1] http://en.wikipedia.org/w/index.php?title=Gatwick%20Airport%20drone%20incident&oldid=1138255439

[2] http://en.wikipedia.org/w/index.php?title=2018%20Caracas%20drone%20attack&oldid=1144142399

continues to advance, particularly with the development of more capable artificial intelligence (AI) and autonomous systems. To address this challenge, it is important to have effective methods for neutralizing rogue drones.

These methods can be divided into destructive and non-destructive approaches. Destructive approaches include shooting down the drone or jamming its communication or GPS signal, causing it to crash. Using firearms, specialized anti-drone projectiles, or anti-drone defense systems, such as net guns or lasers, can physically damage or disable the drone. Jamming its communication or GPS signal, making it difficult or impossible to distinguish the intended signal from the jamming signal, can cause a loss of control and potentially a crash. This method involves using electronic countermeasure devices designed to disrupt the signals between the drone and its operator. However, these methods can pose safety risks to the public and may not be suitable for all situations.

Speciffically trained birds of prey can be a more non-invasive method. Some countries, such as the Netherlands, have experimented with training birds of prey, including eagles, to capture drones in mid-air. Birds are trained to recognize drones as prey and snatch them out of the sky, making the drones inoperable.

Finally, drones equipped with nets or other capture mechanisms can be deployed to intercept and disable rogue drones. These drone-catching drones can physically capture the target drone, entangle it in a net, or even use tethered systems to tow it to a safe location.

The Tokyo Metropolitan Police has used drones to catch drones as a countermeasure against rogue drones. Police drones are equipped with a net that can be used to capture unauthorized or suspicious drones that fly in restricted areas. When police detect a rogue drone, they deploy the arial interception system to pursue it. Once the police drone is in proximity to the rogue drone, it releases the net to entangle and capture the target. After capturing the rogue drone, the police drone can safely transport it to a secure location on the ground, minimizing the risk of injury or damage to property.This method is considered non-destructive and safe, as it avoids causing harm to people or property during the process of neutralizing the rogue drone. A drawback of this method lies in its reliance on manual control by an operator, which places a significant burden on human resources and a limit on the potential deployment environments.

These examples highlight the need to address the issue of rogue UAVs. Their increasing accessibility presents escalating risks. This necessitates advanced, autonomous interception systems that can neutralize threats while safeguarding people and property.

1.1 Related works

In this section, we conduct a thorough review of the existing literature, specifically focusing on research that tackles sub-problems and challenges analogous to those outlined in our study.

1.1.1 Detection and Tracking

Accurate, fast, and robust detection and tracking of intruder UAV is essential for successful interception. Tracking in this work is based on the work presented in [4], which presents a robust and accurate approach to detect flying objects with an emphasis on use with dynamic aerial interception of an agile target using a 3D LiDAR sensor. The approach uses a novel 3D occupancy voxel mapping method for target detection and a cluster-based multiple hypothesis tracker. The proposed method outperforms the state-of-the-art onboard detection methods. The authors demonstrate the practical usability and performance of the system in both simulated and real-world experiments, using the Eagle One[3] platform (Fig. 1.1).

Another notable work [13] developed a relative localization system that operates without the need for markers or specialized equipment. Images from the onboard camera are passed to a Convolutional Neural Network (CNN) to detect nearby UAVs. This approach eliminated the need for specialized markers on the UAVs, enabling the localization of non-cooperating objects. The system is designed for real-time, onboard implementation on an UAVs platform, facilitating relative stabilization of multiple UAVs in a formation or swarm-like behavior when operating in a closed feedback loop with the UAVs' control systems. The authors demonstrate the viability and robustness of the proposed method through real-world experiments. The system proved suitable for robotic tasks that require relative location on-line, such as swarming, formation flight, or collision avoidance.

The work in [13] is a natural continuation of an approach presented in [16], where the authors present a novel approach for the fast localization of non-cooperating drones using a depth image from a stereo camera. The proposed algorithm is computationally efficient enough to process images online on an UAV with limited resources, making it suitable for use in the control feedback of an autonomous aerial intercepting system (AAIS). Through experiments, the authors demonstrate that their proposed method is faster, more precise, and more robust than a CNN-based method, with a longer detection range and better depth estimation. The method also shows high reliability in ranges up to 20 meters.

One notable research effort in the field of autonomous drone interception was inspired by the Mohamed Bin Zayed International Robotics Challenge (MBZIRC) 2020 [6]. The authors of this article developed a novel approach for autonomously catching fast-flying objects using an UAV. Their solution utilizes a 3D LiDAR sensor for target detection and predicts the target's trajectory to determine an optimal interception position. The interceptor UAV navigates into the interception position to safely approach the target, with continuous adjustments made based on the updated estimation and prediction of the target's motion. The system successfully caught the target object in a dedicated onboard net, earning the second place in Challenge 1 and the first place in the combined Grand Challenge of the MBZIRC 2020 competition. This research demonstrates the potential for practical applications in drone interception.

Detection is not the only challenge during interception. Due to the maneuvering of the intruder UAV, robust tracking is necessary. The maneuvering target tracking survey is conducted by X. Rong Li and Vesselin P. Jilkov [27]. The authors provide a detailed overview of multiple-model methods, which have become the dominant approach to maneuvering target tracking in the recent years. They divide the tracking algorithms into three main groups or generations to present the survey in a structured manner and help the readers to grasp the similarities and differences between the various methods.

The first generation of algorithms according to [27] is autonomous algorithms, where each model (model represents the mode of behavior of the target) operates independently without direct communication or cooperation. The filters compete with one another, each striving to offer the best representation of the target's state. Although this approach has some advantages, it may not be optimal in situations where cooperation between models could lead to improved performance. An example of this approach is Multiple Model Adaptive Estimattion (MMAE).

[3] `https://eagle.one/cs/`.

The second generation of algorithms, named by [27] as cooperating algorithms, addresses this limitation by involving a certain level of cooperation between the models or filters. Cooperation can take various forms, such as sharing information, exchanging estimates, or jointly optimizing an objective function. By exploiting the strengths of multiple models and filters, cooperating algorithms aim to enhance the tracking performance. The state-of-the-art is the Interactive Multi-Model Filter (IMM), used in this work.

Lastly, the third generation comprises variable structure algorithms, which allow the structure of the algorithm itself to vary or adapt over time based on the target's dynamics or other factors. This flexibility can lead to better performance, particularly when the target's behavior is highly non-linear or non-stationary. These methods are still an area of active research and may be challenging to deploy.

This comprehensive survey serves as a useful reference for researchers and practitioners working on tracking systems and applications that involve maneuvering targets.

1.1.2 Missile Guidance and Proportional Navigation

Missile guidance presents a similar problem to interceptor trajectory planning and therefore some missile guidance approaches can be adapted for autonomous aerial interception.

A well-established technique is Proportional Navigation (PN). In [31], the authors compare two classes of PN laws: pure proportional navigation (PPN), which commands the interceptor by acceleration in the direction perpendicular to its velocity vector, keeping the total velocity constant, and true proportional navigation (TPN), which commands the acceleration of the interceptor perpendicular to the line of sight (LOS) changing its total velocity. As a result, the authors argue that PPN is more practical than TPN and its generalizations and recommend further investigation into PPN as a guidance law.

From a qualitative point of view, PN is analyzed in [33]. The study determines the conditions for a missile to reach the target from any initial state and demonstrates the existence of a boundary for the required acceleration of the missile. When the target maneuvers with normal acceleration, the missile's trajectories are defined by a nonlinear time-varying system of differential equations, which cannot be solved analytically. Their approach makes strong assumptions and reduces this system of differential equations to a single, first-order, linear time-varying differential equation, which can be solved in analytic terms. However, this paper aims to show that qualitative methods can be applied to obtain a general solution when considering a planar pursuit.

1.1.3 Model Predicitive Control and Trajectory Planning

A higly influential work on trajectory planning in cluttered indoor environments is [23]. Richter, Bry, and Roy presented a novel approach to generate polynomial trajectories. Their method involves the joint optimization of polynomial path segments using an unconstrained quadratic program. This approach has the advantage of being numerically stable for high-order polynomials and a large number of segments, leading to more efficient trajectory planning.

Another innovation in their work is a technique for automatically allocating time to each segment of the trajectory, which allows for adjusting quadrotor speeds along the path. The time allocation is based on a single parameter determining aggressiveness, subject to actuator constraints. By automatically determining the appropriate time allocation for each segment, the algorithm can adapt to different levels of aggressiveness and achieve smooth, dynamically feasible trajectories.

The use of polynomial trajectories, combined with the differentially flat representation of the quadrotor, eliminates the need for computationally intensive sampling and simulation in the high-dimensional state space of the vehicle during motion planning. This results in faster generation of high-quality trajectories compared to purely sampling-based planning methods. However, this approach sacrifices the guarantee of asymptotic convergence to the global optimum that sampling-based methods provide.

Recently, a novel approach to enable a UAV to land autonomously on an oscillating platform, such as an Unmanned Surface Vehicle (USV), is presented in [1]. The primary contributions of the study include the development of a prediction pipeline that uses pose estimation data from an AprilTag marker and the Fast Fourier Transform (FFT) to analyze and predict the periodic motion of the USV induced by the waves, and the implementation of a Model Predictive Control (MPC) algorithm that optimizes landing trajectories while considering the oscillating platform's motion.

Both numerical and realistic simulations using Gazebo were conducted to validate the performance of the proposed system, highlighting that the novel MPC-NE algorithm achieves better landing results with lower tilt angles upon touchdown and shorter solution times compared to the state-of-the-art SH-MPC. The results of the study showed that the proposed approach is highly effective in predicting the motion of the USV and ensuring safe and successful landings even under challenging conditions. In the real-world experiments, the UAV was able to land within 50 seconds of acquiring the required FFT accuracy, with tilt angles upon touchdown less than 5° (0.09 rad).

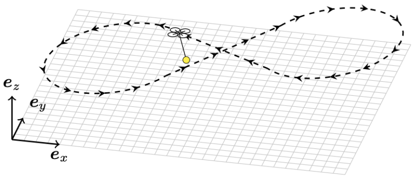


Figure 1.2: Depiction of the UAV interception problem from MBZIRC addressed in the [6].

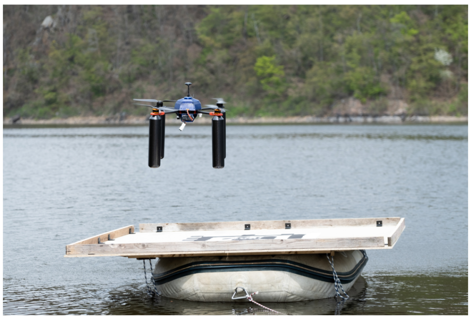


Figure 1.3: figure
Illustration of an autonomous UAV landing on an oscillating platform as presented in [1].

A mathematical model of the physical system plays a key role in MPC. However, modeling all phenomena can be difficult or even unfeasible. In this research paper [10], Torrente et al. propose an innovative approach to improve the position tracking of a quadrotor by integrating Gaussian Processes (GPs) with Model Predictive Control (MPC). The primary focus of the study is to compensate for aerodynamic effects that often hinder the quadrotor's performance. The authors present a data-driven MPC method, in which GPs are used to augment the nominal dynamics of a quadrotor. The GP models are trained on previously recorded flight data to predict acceleration errors of the nominal model, given the current velocity in the body frame. The resulting system significantly improves the quadrotor's positional tracking accuracy, both in simulations and in real-world experiments.

The paper demonstrates that their GP-augmented MPC method outperforms a state-of-the-art linear drag model in various experiments. In real-world experiments, their approach exhibits up to a 50% improvement in tracking performance compared to the linear drag model. Furthermore, the GP-augmented controller is particularly effective in situations where linear drag models fail to capture the nonlinearity of aerodynamic effects.

This work has important implications for future research, as the authors suggest that leveraging the fast fitting time of their GP models can enable real-time adaptation of the dynamic model to varying conditions, such as wind disturbance or battery voltage. Additionally, the predicted uncertainty in their method can be used to plan safe, agile trajectories close to obstacles.

1.2 Problem statement

Based on the detection and uncertainty analysis related to the measurement uncertainty of a maneuvering UAV with an onboard LIDAR system, as described in [4], this work aims to enhance the original tracking method by dynamically scaling the measurement covariances. Additionally, an Interacting Multiple Model (IMM) filter will be incorporated to enable effective tracking of even highly maneuverable UAVs. These improvements will be comprehensively evaluated and compared with the original method using several established metrics.

For interception trajectory planning, Proportional Navigation will be scrutinized and enhanced to better suit UAV interception scenarios. In addition, non-linear model predictive control (NMPC) will be employed to manage the UAV and address the interception problem. Rigorous tests will be performed in simulations to assess these algorithms and to benchmark them against the original solution using various metrics. Proportional Navigation will also undergo real-world testing. This comprehensive approach aims to provide robust and reliable solutions to the challenges posed by UAV interceptions.

1.3 System overwiev

The proposed interception solution's pipeline is illustrated in Fig. 1.4. The components of the problem addressed in this thesis are highlighted in red, while the functionalities of the MRS UAV control system [7], [17] are marked in green. This work mainly focuses on 3D tracking and interception planning.

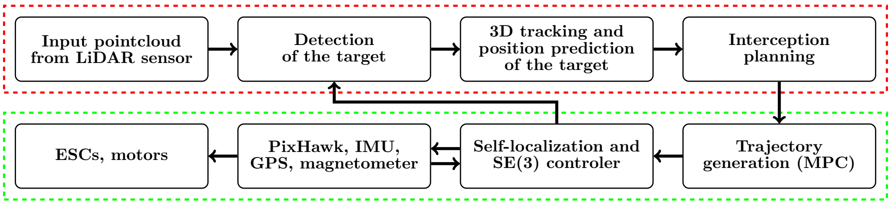

Input pointcloud from LiDAR sensor
Detection of the target
3D tracking and position prediction of the target
Interception planning
ESCs, motors
PixHawk, IMU, GPS, magnetometer
Self-localization and SE(3) controler
Trajectory generation (MPC)

Figure 1.4: Overview of the interception pipe.

Chapter 2

Detection and Tracking

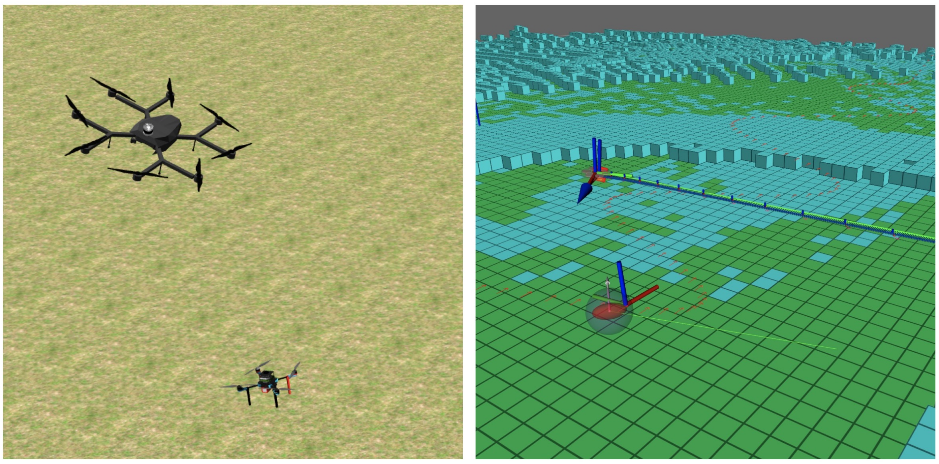


Figure 2.1: The picture on the left displays a Gazebo, used for simulations. The picture on the right exhibits the tracker implemented in this chapter. It is worth noting the broader covariance of the target position due to the yawing of the interceptor.

The performance of the intercepting algorithm is heavily influenced by the performance of the tracking algorithm and the state estimation. Specifically, if the intruder UAV is not visible or the estimate of its state is inaccurate, interception becomes either impossible or ineffective. Furthermore, considering the interception pipeline (shown in Fig.1.4) in a linear way, where the drone is first detected and later intercepted, one might assume that detection is not affected by the interception. However, as our goal is to create a fast and agile interceptor, tracking is in practice disrupted by highly agile maneuvers.

While it is possible to develop intercepting algorithms capable of addressing specific limitations of detection, doing so could compromise interceptor performance. Moreover, it is essential to design intercepting algorithms that are as independent of detection as possible, as this would result in more general and applicable outcomes in the end.

For all these reasons, I decided to enhance the original pipeline 2.1, proposed in [4]. The biggest challenges involved dealing with the high uncertainty of the interceptors' pose during interceptor maneuvers and estimating the future trajectory of a hostile drone without knowing its control inputs. These problems are addressed in the following subsections.

2.1 Flying object detection

The method proposed in [4] utilizes a 3D LiDAR sensor to track the intruder drone. LiDAR operates by emitting laser pulses and measuring the time it takes for the light to reflect back after encountering an object. This technique offers numerous advantages, including high accuracy and robustness under various environmental conditions. Most notably, it enables the detection of even small objects and accurately determines their range despite the presence of a noisy background.

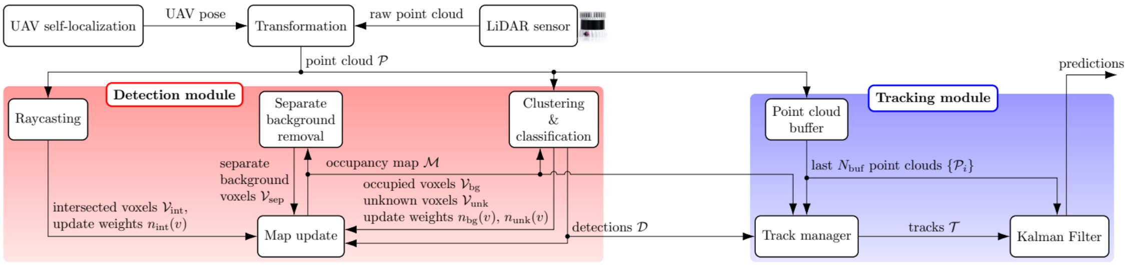

UAV self-localization
UAV pose
Transformation
raw point cloud
LiDAR sensor
point cloud $\mathcal{P}$
predictions
Detection module
Raycasting
Separate background removal
Clustering & classification
separate background voxels $\mathcal{V}_{\text{sep}}$
occupancy map $\mathcal{M}$
occupied voxels $\mathcal{V}_{\text{bg}}$
unknown voxels $\mathcal{V}_{\text{unk}}$
update weights $n_{\text{bg}}(v)$, $n_{\text{unk}}(v)$
intersected voxels $\mathcal{V}_{\text{int}}$,
update weights $n_{\text{int}}(v)$
Map update
detections $\mathcal{D}$
Tracking module
Point cloud buffer
last N_{buf} point clouds $\{\mathcal{P}_i\}$
Track manager
tracks $\mathcal{T}$
Kalman Filter

Figure 2.2: Overview schematic of the detection and tracking pipeline. Illustration taken from [4].

For this purpose, a novel and innovative approach based on a 3D occupancy voxel mapping method for target detection is proposed in [4]. The approach is further enhanced by a cluster-based multiple hypothesis tracker, which effectively compensates for delays and sporadic false detections. This makes the detector an excellent candidate for agile multi-robot interactions, specifically in the domain of autonomous aerial interception.

The method of [16] is based on a LiDAR sensor mounted on a UAV and it has already been successfully implemented in practice, specifically on the prototype Autonomous Aerial Interception System (AAIS) platform, named Eagle One[1] (as shown in Fig 1.1).

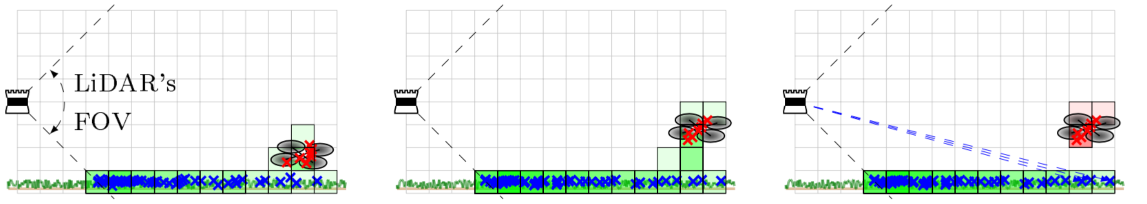
An overview of the detection system is presented in Fig. 2.2. The main element of the system is a **Detection module** that relies on an occupancy voxel map $\mathcal{M}$. Scans from the LiDAR are transformed to a static world frame based on the UAV's self-localization pipeline. The transformed scans P are used to iteratively update $\mathcal{M}$. Typically, the map $\mathcal{M}$ contains voxels classified as unknown (no information provided yet), occupied, free, and sometimes uncertain. However, this method implements a key new state called "tentative occupied," that allows the detection of flying objects based on the clustering of voxels classified as such. The idea is illustrated in Fig. 2.3.

A tracking algorithm, which will be the subject of further discussion later, is used to associate subsequent detections corresponding to the same objects. It receives the latest filtered point clouds $\mathcal{P}$, detections $\mathcal{D}$, and the latest occupancy map M.

The tracker keeps a buffer of the last $\mathcal{N}_{\text{buf}}$ point clouds sorted by the time of acquisition. When a new set of detections $\mathcal{D}$ is obtained, the corresponding point cloud is selected from the buffer. Each detection from $\mathcal{D}$ is then tracked through subsequent point clouds in the buffer to the latest one using a Kalman Filter (KF)-based multi-target tracking algorithm.

The detections $\mathcal{D}$ and map $\mathcal{M}$ are typically several iterations older than the newest point cloud in the buffer. Based on the uncertainty of the track, which is predicted to the time of the next point cloud using information from the older $\mathcal{M}$ and $\mathcal{D}$, part of the newer point cloud is selected around the predicted position of the track. Points within the selection are clustered based on minimizing the inter-cluster distance. The centers of the clusters are tried for association and used to update already existing tracks, using prediction uncertainty

[1] `https://eagle.one/cs/`.



(a) Situation at time t_0.

(b) Situation at time t_1.

(c) Situation after raycasting (the relevant rays are the blue dashed lines).

Figure 2.3: The diagram illustrates the dynamic evolution of voxel classifications within map $\mathcal{M}$ as an initially stationary object suddenly begins to move. The implementation of a unique "tentative occupied" state allows for the efficient detection of this motion. Illustration taken from [4].

once again. This is another key feature of this approach and should be properly highlighted because it allows the processing and finding of track updates practically in all point clouds, including the most recent one, while allowing a robust detection method based on a voxel map with relatively high delay.

The provided solution is capable of detecting flying objects up to 10 times per second, based on the LiDAR's rate. The quality of the tracks, based on the detections provided with known uncertainties, is highly dependent on the tracking algorithm used.

2.2 Drone Dynamics and Tracking Algorithms

After successful detection, the target needs to be tracked, with its state estimated. There are several methods that can be utilized for this purpose. To comprehend these options fully, we first study the dynamics of drones, given the assumption that we are tracking another drone.

2.2.1 Target model

The nonlinear dynamics of a UAV can generally be defined as follows:

$$\begin{aligned} \dot{\mathbf{r}} &= \mathbf{v}, && (2.1)\\ m\dot{\mathbf{v}} &= f\mathbf{R}\mathbf{e}_3 + mg\mathbf{e}_3, && (2.2)\\ \dot{\mathbf{R}} &= \mathbf{R}\hat{\mathbf{\Omega}}, && (2.3)\\ \mathbf{J}\dot{\mathbf{\Omega}} + \mathbf{\Omega} \times \mathbf{J}\mathbf{\Omega} &= \mathbf{M}, && (2.4) \end{aligned}$$

where $\mathbf{r} = [x, y, z]^T$ denotes the position, $\mathbf{R}(\phi, \theta, \psi) \in SO(3)$ represents the orientation of the UAV, and $\mathbf{\Omega} \in \mathbb{R}^3$ is the angular velocity in the body-fixed frame. The hat map $\hat{\cdot} : \mathbb{R}^3 \to SO(3)$ is defined by the condition $\hat{\mathbf{x}}\mathbf{y} = \mathbf{x} \times \mathbf{y}$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^3$. The mass of the UAV is given by m, $\mathbf{J}$ denotes the inertia matrix, $\mathbf{M}$ represents the total moment, and g is the gravitational acceleration. Lastly, f refers to the net thrust produced by the propellers.

Significant aspects to consider when dealing with UAVs are the jerk $\mathbf{j}$ and the snap $\mathbf{s}$. These terms represent the third and fourth derivatives of the position, respectively. Minimizing

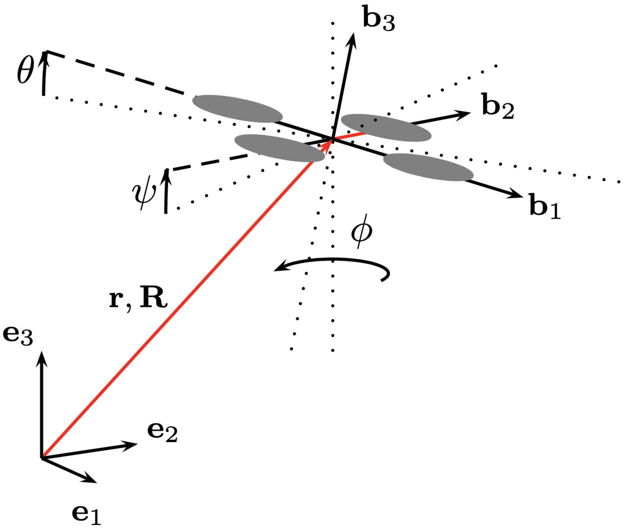


Figure 2.4: The world coordinate frame w in which the position and orientation of the UAV is expressed by translation $\mathbf{r} = [x, y, z]^T$ and rotation $\mathbf{R}(\phi, \theta, \psi)$. Taken from [17].

these variables is usually desired to ensure smooth and efficient trajectories, as doing so reduces mechanical stress, conserves energy, and enhances overall safety. These observations can potentially be leveraged during the tracking process.

Traditionally, maneuvering targets are perceived as objects capable of abruptly changing their acceleration. However, due to the inherent dynamics of drones, abrupt changes in acceleration are not feasible. A suitable analogy is that just as an airplane cannot instantly reverse its velocity, a drone's dynamics only allows for discontinuous changes in angular acceleration (when neglecting the dynamics of propellers).

We assumed that when a UAV is flying at a constant height, the acceleration is proportional to tilt, the jerk is approximately equal to the tilt rate, and the snap is proportional to the tilt acceleration, which can be changed discontinuously. While this example is a considerable simplification of the real dynamics, it can provide the necessary insight into the dynamic behavior of a UAV.

For this reason, a non-linear model was considered to capture this and other phenomena inherent in UAV dynamics. However, during initial experiments, even the Extended Kalman Filter (EKF) [28] or the Unscented Kalman Filter (UKF) [28] struggled with a maneuvering target UAV due to high measurement uncertainty and agile maneuvers of the interceptor UAV. This issue may largely arise from the fact that the measurements include only positional information. Note that these simulations were done using MATLAB [5].

Therefore, we decided to opt for a simpler LTI model to address the complexities and uncertainties inherent in drone tracking. Despite this decision, the deeper understanding of drone dynamics we gained from this exploration is invaluable for future refinements and adaptations of the model.

2.2.2 Kalman Filter

A classical approach to tracking a Linear Time-Invariant (LTI) system is the Kalman Filter (KF) [35]. The KF is a linear, recursive estimator that provides optimal state estimation for linear systems with additive Gaussian noise.

The KF operates in two main steps: prediction and correction. The filter equations for the prediction phase are as follows:

$$\begin{aligned}\hat{\mathbf{x}}_{k|k-1} &= \mathbf{F}_k\hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_k\mathbf{u}_k, && (2.5)\\ \mathbf{P}_{k|k-1} &= \mathbf{F}_k\mathbf{P}_{k-1|k-1}\mathbf{F}_k^T + \mathbf{Q}_k. && (2.6)\end{aligned}$$

The filter equations for the correction phase are as follows:

$$\begin{aligned}\mathbf{S}_k &= \mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T + \mathbf{R}_k && (2.7)\\ \mathbf{K}_k &= \mathbf{P}_{k|k-1}\mathbf{H}_k^T\mathbf{S}_k^{-1}, && (2.8)\\ \mathbf{v}_k &= \mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k|k-1}, && (2.9)\\ \hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\mathbf{v}_k, && (2.10)\\ \mathbf{P}_{k|k} &= (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k|k-1}. && (2.11)\end{aligned}$$

Here, $\mathbf{F}_k$ is the state transition matrix, $\mathbf{B}_k$ is the control input matrix, $\mathbf{u}_k$ is the control input vector, $\mathbf{H}_k$ is the measurement matrix, $\mathbf{Q}_k$ is the process noise covariance, and $\mathbf{R}_k$ is the measurement noise covariance. $\hat{\mathbf{x}}_{k|k-1}$ and $\hat{\mathbf{x}}_{k|k}$ represent the state vectors, $\mathbf{P}_{k|k-1}$ and $\mathbf{P}_{k|k}$ are the state covariance matrices, $\mathbf{K}_k$ is the Kalman gain, $\mathbf{S}_k$ is the innovation covariance, and $\mathbf{v}_k$ is the innovation. The subscript notation $k|k-1$ denotes the estimate at time step k given information up to time step $k-1$, while $k|k$ denotes the estimate at time step k given information up to time step k.

The measurement likelihood can be computed as a multivariate Gaussian probability density function using the innovation and innovation covariance:

$$p(\mathbf{z_k}|\hat{\mathbf{x}}_{k|k-1}) = \frac{1}{\sqrt{(2\pi)^n|\mathbf{S}_k|}}\exp\left(-\frac{1}{2}\mathbf{v}_k^T\mathbf{S}_k^{-1}\mathbf{v}_k\right), \qquad (2.12)$$

where n is the dimension of the measurement vector $\mathbf{z}_k$, and $|\mathbf{S}_k|$ denotes the determinant of the innovation covariance matrix $\mathbf{S}_k$. The optimal task that the KF solves is the estimation of the true state of a linear system with Gaussian noise while minimizing the mean squared error. For Gaussian distributions, the logarithm of the likelihood function is proportional to the negative of the squared error. Therefore, maximizing the likelihood is equivalent to minimizing the squared error [35]. This makes the KF an optimal linear estimator in the presence of Gaussian uncertainties.

Two significant observations were made when considering the tracking of agile, non-cooperative UAV using a linear KF.

Firstly, the agility of the drone, coupled with the lack of information regarding its control signals, introduces a considerable degree of uncertainty in the state estimates. This is indeed a crucial factor, as the performance of the tracking algorithm hinges on how effectively it can handle this inherent uncertainty. It is especially significant for algorithms like the Kalman Filter, which assumes a linear and Gaussian nature of the system's dynamics and noise characteristics.

Secondly, for the standard linear Kalman Filter, the time evolution of the state covariance matrix $\mathbf{P}$, which is used in the calculation of the Kalman gain $\mathbf{K}$, is primarily determined by the process noise covariance $\mathbf{Q}$, the measurement noise covariance $\mathbf{R}$, and the initial state covariance $\mathbf{P_0}$. Therefore, in scenarios where the target starts to maneuver abruptly (leading to high innovation), the Kalman gain $\mathbf{K}$ may remain relatively constant. This could be counter-intuitive, as one might expect the filter to be more responsive to such significant changes in the target's dynamics. An illustrative situation is depicted on Fig. 2.5, where simulation of a 2D moving target tracked using KF with constant velocity model is provided.

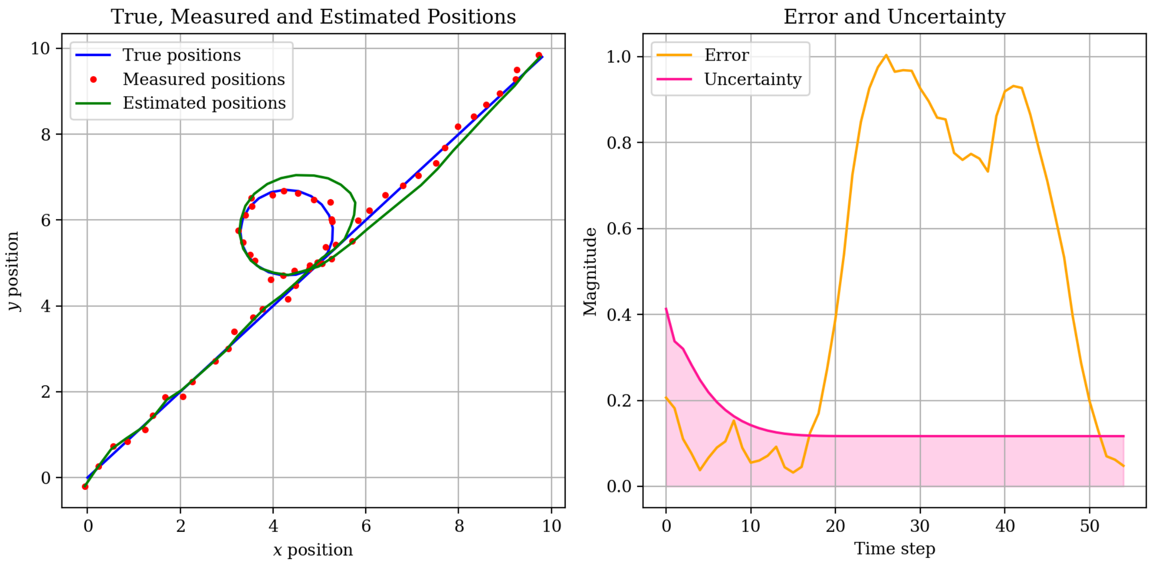


Figure 2.5: Simulation of a moving target in 2D tracked using a KF with constant velocity model. It demonstrates how the uncertainty remains relatively constant despite abrupt turning oh the target, highlighting a potential limitation when tracking a maneuvering drone.

2.3 Understanding Measurement Uncertainty

An exploration of literature in the domain of LiDAR technology [3], as well as an analysis of outcomes from competitions as presented in [2], [14], [9] or [6], could lead one to have confidence in the tried and tested design of mounting a LiDAR sensor on a drone. However, in these scenarios, the LiDAR sensor is deployed primarily for algorithms such as the Iterative Closest Point (ICP) [29], where the drone typically moves slowly, without abrupt directional changes or fast maneuvers.

Preliminary experiments revealed that even minor inaccuracies in the drone's orientation estimation during agile maneuvers can markedly impact the system's accuracy. This effect is especially noticeable when the target is at a considerable distance and the drone's orientation is uncertain. In subsequent sections, we will formally scrutinize this observation and discuss its implications. It is worth highlighting that the remainder of this section largely paraphrases [4], a work being conducted concurrently with this thesis.

2.3.1 LiDAR Measurement Uncertainty

Let us consider a single point $\boldsymbol{p}_m$ that is measured by a LiDAR sensor with a (unit) direction vector $\overrightarrow{\boldsymbol{d}}$ and a corresponding range l_m. The pose of the sensor in a static world frame $\mathcal{W}$ is represented by a translation vector $\boldsymbol{t}_m$ and rotation matrix $\mathbf{R}_m$. The measured point $\boldsymbol{p}_m$ can be expressed as a function of the measured range, translation, and rotation as

$$\boldsymbol{p}_m = l_m \mathbf{R}_m \overrightarrow{\boldsymbol{d}} + \boldsymbol{t}_m. \tag{2.13}$$

If we had absolute accuracy in the measurements of the sensor's pose and range, we could obtain the corresponding ground-truth point $\boldsymbol{p}_{gt}$ as

$$\boldsymbol{p}_{gt} = l_{gt} \mathbf{R}_{gt} \overrightarrow{\boldsymbol{d}} + \boldsymbol{t}_{gt} \tag{2.14}$$

where $l_{gt}, \boldsymbol{t}_{gt}$, and $\mathbf{R}_{gt}$ represent the ground-truth (noise-less) range and pose.

We can model the relation between the measured values $l_m, \boldsymbol{t}_m, \mathbf{R}_m$ and their corresponding ground-truth values as

$$l_{gt} = l_m + l_n, \tag{2.15}$$

$$\boldsymbol{t}_{gt} = \boldsymbol{t}_m + \boldsymbol{t}_n, \tag{2.16}$$

$$\mathbf{R}_{gt} = \mathbf{R}_m \mathbf{R}_n = \mathbf{R}_m \mathbf{R}_z(\gamma_n) \mathbf{R}_y(\beta_n) \mathbf{R}_x(\alpha_n), \tag{2.17}$$

where $l_n, \boldsymbol{t}_n$, and $\alpha_n, \beta_n, \gamma_n$ represent the unobservable measurement noise. We define a vector $\boldsymbol{w}$ that is assumed to be drawn from a multivariate Gaussian distribution as

$$\boldsymbol{w} = \begin{bmatrix} l_n & \boldsymbol{t}_n^\top & \alpha_n & \beta_n & \gamma_n \end{bmatrix}^\top, \tag{2.18}$$

$$\boldsymbol{w} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\boldsymbol{w}}), \tag{2.19}$$

where $\boldsymbol{\Sigma}_{\boldsymbol{w}}$ is a known covariance matrix of the measurement noise.

In practice, the true position of $\boldsymbol{p}_{gt}$ is unknown, but its probability distribution can be estimated given the measured point $\boldsymbol{p}_m$ and the measurement uncertainty $\boldsymbol{\Sigma}_{\boldsymbol{w}}$. Although an analytical solution may not be practical, the probability distribution can be approximated using linearization from equations (2.15)-(2.17) as a Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. The transformation of the known covariance $\boldsymbol{\Sigma}_{\boldsymbol{w}}$ of the random variable $\boldsymbol{w}$ to the covariance $\boldsymbol{\Sigma}$ can be derived as

$$\boldsymbol{\Sigma} = \mathbf{J} \boldsymbol{\Sigma}_{\boldsymbol{w}} \mathbf{J}^\top, \quad \mathbf{J} = \left. \frac{\partial \boldsymbol{p}_{gt}}{\partial \boldsymbol{w}} \right|_{\boldsymbol{w}=\mathbf{0}} \tag{2.20}$$

Using the substitutions

$$\dot{\mathbf{R}}_{\alpha_n} = \left. \frac{\partial \mathbf{R}_x}{\partial \alpha_n} \right|_{\alpha_n=0}, \tag{2.21}$$

$$\dot{\mathbf{R}}_{\beta_n} = \left. \frac{\partial \mathbf{R}_y}{\partial \beta_n} \right|_{\beta_n=0}, \tag{2.22}$$

$$\dot{\mathbf{R}}_{\gamma_n} = \left. \frac{\partial \mathbf{R}_z}{\partial \gamma_n} \right|_{\gamma_n=0}, \tag{2.23}$$

the Jacobian $\mathbf{J}$ evaluates to

$$\mathbf{J} = \begin{bmatrix} \mathbf{R}_m \overrightarrow{\boldsymbol{d}} & \mathbf{I} & l_m \mathbf{R}_m \dot{\mathbf{R}}_{\alpha_n} \overrightarrow{\boldsymbol{d}} & l_m \mathbf{R}_m \dot{\mathbf{R}}_{\beta_n} \overrightarrow{\boldsymbol{d}} & l_m \mathbf{R}_m \dot{\mathbf{R}}_{\gamma_n} \overrightarrow{\boldsymbol{d}} \end{bmatrix} \tag{2.24}$$

The mean $\boldsymbol{\mu}$ of the linearized distribution is the expected value of $\boldsymbol{p}_{gt}$, which can be obtained from equation (2.13 as

$$\boldsymbol{\mu} = \mathrm{E}\left[\boldsymbol{p}_{gt}\right] = \boldsymbol{p}_{gt}\big|_{\boldsymbol{w}=\boldsymbol{0}} = \boldsymbol{p}_{m}. \tag{2.25}$$

The approximated probability density function of $\boldsymbol{p}_{gt}$ is then

$$f_{\boldsymbol{p}_{gt}}(\boldsymbol{p}) \approx f_{\mathcal{N}}(\boldsymbol{p}, \boldsymbol{\mu}, \boldsymbol{\Sigma}), \tag{2.26}$$

which represents a multivariate normal probability density function with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$. This principle is illustrated in Fig. 2.8.

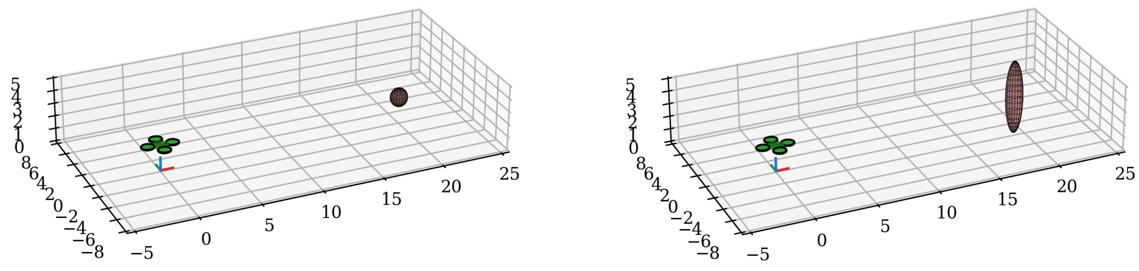


Figure 2.6: $\alpha_n = 0°$.

Figure 2.7: $\alpha_n = 8°$.

Figure 2.8: Illustration of orientation uncertainty's impact on LiDAR measurement precision. Both scenarios involve a 20-meter distance to the measurement point.

Unfortunately, the current session of the MRS UAV system [7] does not provide covariance estimates of drone orientation. Despite significant effort, additional attempts to extract these estimates from the PixHawk flight controller[2] have also been unsuccessful. As an alternative, these covariances were approximated using angular rates, which were adjusted using a calibrated coefficient. Despite this simplification, the method has been proven to significantly enhance tracking performance, as demonstrated in section 2.5.3.

2.4 Tracking a Maneuvering target

In a typical scenario, KF incorporates known input into the system. However, in the case of a non-cooperative drone, this information is unknown. For simplicity, let us consider the constant velocity (CV) model. Setting a lower process noise $\mathbf{Q}$, we can achieve excellent tracking performance when the target is moving in a straight line. However, this might result in poor performance or even loss of tracking during maneuvers. On the contrary, setting a higher process noise might increase the overall error. Alternatively, models such as constant acceleration (CA) or constant jerk (CJ) could be used to achieve effective tracking with low error during maneuvers. Still, the issue of high noise during linear movement would persist. The solution is to utilize multiple models simultaneously and switch between them intelligently.

2.4.1 Interacting Multiple Model filter

The Interacting Multiple Model (IMM) filter is a sophisticated, probabilistic filtering technique used for estimation problems with multiple dynamic models. This filter is particularly useful in situations where the actual motion of a target or system may abruptly change

[2] `https://docs.px4.io/main/en/`.

or is governed by different modes. IMM operates by running multiple filters in parallel, each assuming a different dynamic model, and then combining their results based on model probabilities. These probabilities are updated at each time step using a Markov chain approach, effectively capturing the likelihood of each model given the observed data. This allows the IMM filter to provide more accurate and reliable estimates than a single-model approach, particularly in the context of tracking targets with complex, multimodal behavior [27].

The IMM method operates through a sequence of a mixing phase and a filtering phase. Each phase is executed as follows:

Mixing

In the mixing phase, the state estimates of model i, denoted as $\hat{\mathbf{x}}^{(i)}_{k-1|k-1}$, and covariances, denoted as $\mathbf{P}^{(i)}_{k-1|k-1}$, from the previous time step are combined based on a Markov chain model, which is used to describe the probability p_{ji} of transitioning from model j to model i. This process generates new mixed estimates and covariances for each filter, represented as $\hat{\mathbf{x}}^{(i)}_{k|k-1}$ and $\mathbf{P}^{(i)}_{k|k-1}$ respectively. The equations for the mixing phase can be described mathematically as:

$$\mu^{(i)}_{k|k-1} = \sum_{j=1}^{N} p_{ji}\mu^{(j)}_{k-1}, \tag{2.27}$$

$$\mu^{(j|i)}_{k-1} = \frac{p_{ji}\mu^{(j)}_{k-1}}{\mu^{(i)}_{k|k-1}}, \tag{2.28}$$

$$\hat{\mathbf{x}}^{(i)}_{k|k-1} = \sum_{j=1}^{N} \hat{\mathbf{x}}^{(j)}_{k-1|k-1}\mu^{(j|i)}_{k-1}, \tag{2.29}$$

$$\mathbf{P}^{(i)}_{k|k-1} = \sum_{j=1}^{N} \mu^{(j|i)}_{k-1}\left[\mathbf{P}^{(j)}_{k-1|k-1} + (\hat{\mathbf{x}}^{(j)}_{k-1|k-1} - \hat{\mathbf{x}}^{(i)}_{k|k-1})(\hat{\mathbf{x}}^{(j)}_{k-1|k-1} - \hat{\mathbf{x}}^{(i)}_{k|k-1})^T\right], \tag{2.30}$$

where $\mu^{(i)}_{k}$ is the probability of mode i at time step k. We can view the mixing step of the IMM filter as a preparatory process, in which each model is updated independently of its current probability. The new state for each model is constructed based on the likelihood of all models transitioning to it, under the assumption that this model is correct. Later, the final estimate incorporates the actual likelihood of each model.

Filtering

In the filtering phase, each filter performs a standard predict-update cycle using the mixed estimates and covariances from the mixing phase. This process produces updated estimates and covariances for each filter, denoted as $\hat{\mathbf{x}}^{(i)}_{k|k}$ and $\mathbf{P}^{(i)}_{k|k}$ respectively. The equations for the filtering phase can be described mathematically as:

$$\hat{\mathbf{x}}_{k+1|k}^{(i)} = \mathbf{A}_k^{(i)} \hat{\mathbf{x}}_{k|k}^{(i)} + \mathbf{B}_k^{(i)} u_k^{(i)}, \tag{2.31}$$

$$\mathbf{P}_{k+1|k}^{(i)} = \mathbf{A}_k^{(i)} \mathbf{P}_{k|k}^{(i)} \left(\mathbf{A}_k^{(i)}\right)^T + \mathbf{B}_k^{(i)} \mathbf{Q}_k^{(i)} \left(\mathbf{B}_k^{(i)}\right)^T, \tag{2.32}$$

$$\mathbf{K}_k^{(i)} = \mathbf{P}_{k|k-1}^{(i)} \left(\mathbf{H}_k^{(i)}\right)^T \left(\mathbf{S}_k^{(i)}\right)^{-1}, \tag{2.33}$$

$$\hat{\mathbf{x}}_{k|k}^{(i)} = \hat{\mathbf{x}}_{k|k-1}^{(i)} + \mathbf{K}_k^{(i)} \left(\underline{y}_k - \mathbf{H}_k^{(i)} \hat{\mathbf{x}}_{k|k-1}^{(i)}\right), \tag{2.34}$$

$$\mathbf{P}_{k|k}^{(i)} = \mathbf{P}_{k|k-1}^{(i)} - \mathbf{K}_k^{(i)} \mathbf{H}_k^{(i)} \mathbf{P}_{k|k-1}^{(i)}. \tag{2.35}$$

Here, $\mathbf{K}_k^{(i)}$ is the Kalman gain, $\mathbf{y}_k$ is the measurement, $\mathbf{H}$ is the measurement matrix, and $\mathbf{I}$ is the identity matrix. $\mathbf{A}_k^{(i)}$ is the state transition matrix, $\mathbf{B}_k^{(i)}$ is the control-input matrix, $\mathbf{u}_k^{(i)}$ is the control vector, $\mathbf{Q}_k^{(i)}$ is the process noise covariance, and $\mathbf{S}_k^{(i)}$ is the innovation covariance. The predict and update steps are performed for each model independently. The final step is the mode probability update:

$$L_k^{(i)} = \mathcal{N}\left(\underline{y}_k \underline{y}_{k|k1}^{(i)}; 0, \mathbf{S}_k^{(i)}\right), \tag{2.36}$$

$$\mu_k^{(i)} = \frac{\mu_{k|k-1}^{(i)} L_k^{(i)}}{\sum_{j=1}^{r} \mu_{k|k-1}^{(j)} L_k^{(j)}}, \tag{2.37}$$

Where $L_k^{(i)}$ is the probability of model i being correct given the corresponding measurement $\mathbf{y}_k$, as in equation (2.12). The estimates from all the filters are then combined into a single estimate using a weighted sum, where the weights are determined by the updated mode probabilities:

$$\hat{\mathbf{x}}_{k|k} = \sum_{i=1}^{r} \mu_k^{(i)} \hat{\mathbf{x}}_{k|k}^{(i)}, \tag{2.38}$$

$$\mathbf{P}_{k|k} = \sum_{i=1}^{r} \mu_k^{(i)} \left[\mathbf{P}_{k|k}^{(i)} + (\hat{\mathbf{x}}_{k|k} - \hat{\mathbf{x}}_{k|k}^{(i)})(\hat{\mathbf{x}}_{k|k} - \hat{\mathbf{x}}_{k|k}^{(i)})^T\right], \tag{2.39}$$

The estimated state obtained from the IMM algorithm serves as the final output for the interception algorithms. A comparison of the performance between a simple KF and an IMM in an example problem is shown in Fig.2.9 (Compare with Fig. 2.5).

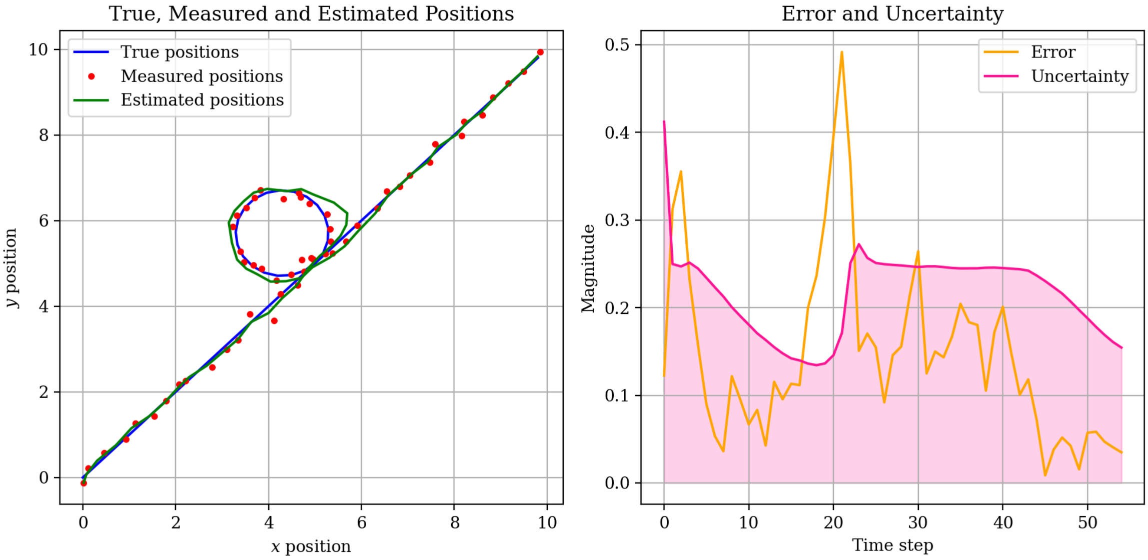


Figure 2.9: This simulation presents a 2D moving target tracked using an IMM filter, featuring two CV models with distinct process noise levels. The simulation illustrates how uncertainty increases during abrupt target maneuvers. Compare with Fig. 2.5.

2.4.2 Model Selection and Noise Characterization

In general, it is recommended to keep the number of models reasonably low, as mentioned in the survey[27]. For the sake of simplicity, no models tracked with the EKF or UKF are included in the proposed approach. Instead, several linear models are considered: Constant Position (CP), Constant Velocity (CV), Constant Acceleration (CA), and Constant Jerk (CJ). Through experimentation, it was found that the best results were achieved with the CV and CA models. It should be noted that tuning the IMM filter can be particularly challenging. Despite efforts to automate the process using MATLAB's "fmincon" function with collected data, transferring these results to the simulation was unsuccessful. Therefore, manual tuning was applied. For the Constant Velocity model, the system is described by the following state transition matrix:

$$\mathbf{A}_{\mathrm{CV}}[k] = \begin{bmatrix} \mathbf{I} & \Delta t_{[k]}\mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \tag{2.40}$$

where $\mathbf{I} \in \mathbb{R}^{3\times3}$ is an identity matrix, $\mathbf{0} \in \mathbb{R}^{3\times3}$ is a zero matrix, and $\Delta t_{[k]}$ is the duration since the previous time step $k-1$. For the Constant Acceleration model, state transition matrix is:

$$\mathbf{A}_{\mathrm{CA}}[k] = \begin{bmatrix} \mathbf{I} & \Delta t_{[k]}\mathbf{I} & \frac{1}{2}\Delta t_{[k]}^2\mathbf{I} \\ \mathbf{0} & \mathbf{I} & \Delta t_{[k]}\mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}. \tag{2.41}$$

In both models, the input matrix $\mathbf{B}$ is an empty matrix, because the target's control inputs are unknown in our non-cooperative context. Only the position states are directly observed in this model. Hence, the measurement matrix $\mathbf{H}$ is:

$$\mathbf{H} = \begin{bmatrix} \mathbf{I}_{3\times3} & \cdots & \mathbf{0}_{3\times3} \end{bmatrix}. \tag{2.42}$$

The model matrices (from equations (2.40), (2.41), and (2.42)) define the dynamics and observation models for our IMM estimation. Originally [4], the process noise was set using diagonal matrices scaled by coefficients:

$$\mathbf{Q} = \mathbf{I}_{n \times n} \cdot \sigma^2, \tag{2.43}$$

where $\mathbf{I}_{n \times n}$ is an identity matrix of size $n \times n$, and σ^2 is the noise variance. However, analytically derived models that are slightly more sophisticated provide more accurate results in simulation. The process noise $\mathbf{Q}_{\text{CV}}$ for the constant velocity (CV) model is calculated using the following formula:

$$\mathbf{Q}_{\text{CV}} = \sigma_a^2 \begin{bmatrix} \frac{\Delta t^3}{3} & \frac{\Delta t^2}{2} \\ \frac{\Delta t^2}{2} & \Delta t \end{bmatrix}, \tag{2.44}$$

where σ_a^2 is the variance of the acceleration noise in the constant velocity model. This is a measure of how much the velocity changes randomly over time. Δt is the time step. The process noise $\mathbf{Q}_{\text{CA}}$ for the constant acceleration (CA) model is given by:

$$\mathbf{Q}_{\text{CA}} = \sigma_j^2 \begin{bmatrix} \frac{\Delta t^5}{20} & \frac{\Delta t^4}{8} & \frac{\Delta t^3}{6} \\ \frac{\Delta t^4}{8} & \frac{\Delta t^3}{3} & \frac{\Delta t^2}{2} \\ \frac{\Delta t^3}{6} & \frac{\Delta t^2}{2} & \Delta t \end{bmatrix}, \tag{2.45}$$

where σ_j^2 is the variance of the jerk noise in the constant acceleration model. This denotes the measure of how the acceleration changes randomly over time. A higher process noise places less emphasis on the process model, causing the Kalman filter to rely more on the measurements for state correction. The measurement noise dynamically scaled covariance matrix $\mathbf{R}$ was previously discussed in 2.3.1. Each model applies its own tuning coefficient to multiply the angular rates (these rates approximate the uncertainty in position), and the $n \times n$ matrix $\mathbf{R_0}$ is added as a constant uncertainty. This mainly accounts for the fact that the drone is not a point and is only partially visible from certain angles, causing floating center estimation. In mathematical terms, it can be represented as:

$$\mathbf{R_0} = \mathbf{I}_{n \times n} \cdot c_R, \tag{2.46}$$

where $\mathbf{I}_{n \times n}$ is a $n \times n$ identity matrix and c_R is a scaling coefficient tuned for each model.

2.5 Tracker performance

In this section, we study the performance of the original KF used for tracking, the KF enhanced by the improved measurement covariance matrix 2.3.1, and finally, the IMM algorithm.

2.5.1 Evaluation Metrics

To compere performance of various filters, we consider several metrics:

- **Track Interruption Count (TIC)** and **Tracking Continuity Percentage (TCP)**: TIC counts the instances when target tracking was interrupted (target was lost), and TCP indicates the percentage of total duration during which the drone was successfully tracked.

- **Root Mean Squared Error (RMSE)**: The RMSE is the square root of the mean squared error (MSE) and is expressed in the same units as the state variables. The RMSE is given by:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k)^2}. \tag{2.47}$$

  A lower RMSE indicates a better filter performance.
- **Normalized Estimation Error Squared (NEES)**: The NEES assesses the consistency of the filter by comparing the estimation error with the filter's covariance matrix. The NEES is defined as:

$$\text{NEES}_k = (\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k)^T \boldsymbol{P}_k^{-1}(\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k), \tag{2.48}$$

  where $\boldsymbol{P}_k$ is the filter's covariance matrix at the time step k. An average NEES (ANEES) over all time steps provides a measure of the filter's consistency.

2.5.2 Testing procedure

To comprehensively test the tracker's capabilities under varying conditions, we propose a systematic approach consisting of progressively more challenging scenarios. For the interceptor, which tracks the target, we consider three stages: Hovering Interceptor (HI), Flying Interceptor (FI), and Maneuvering Interceptor (MI).

For each of these interceptor stages, the target goes through three sub-stages: Target Moving (TM), Target Flying (TF), and Target Maneuvering (TMa). This results in a total of nine combinations, as shown in the Table 2.1.

Table 2.1: Different Interceptor and Target Scenarios

	Target Moving	**Target Flying**	**Target Maneuvering**
Hovering Interceptor	HI-TM	HI-TF	HI-TMa
Flying Interceptor	FI-TM	FI-TF	FI-TMa
Maneuvering Interceptor	MI-TM	MI-TF	MI-TMa

These scenarios are designed to gradually increase in complexity, allowing us to thoroughly assess the tracker's robustness and reliability under diverse and challenging conditions.

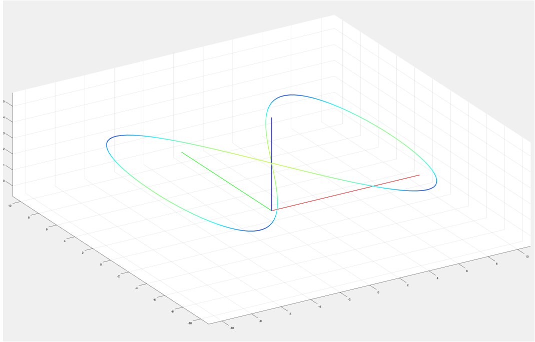


Figure 2.10: Flying Interceptor (FI) - The interceptor is smoothly moving along a predictable path.

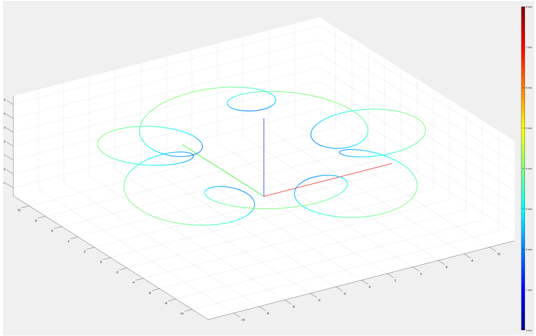


Figure 2.11: Maneuvering Interceptor (MI) - The interceptor is performing complex and abrupt maneuvers.

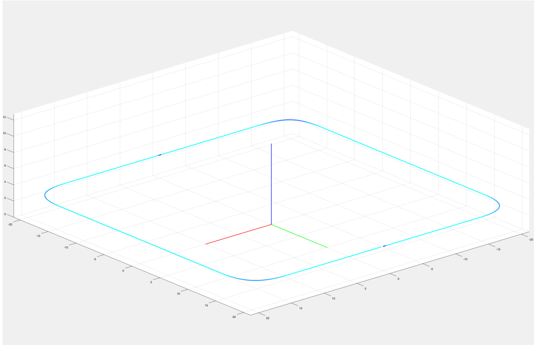


Figure 2.12: Target Moving (TM) - The target is in motion, moving along an arbitrary path.

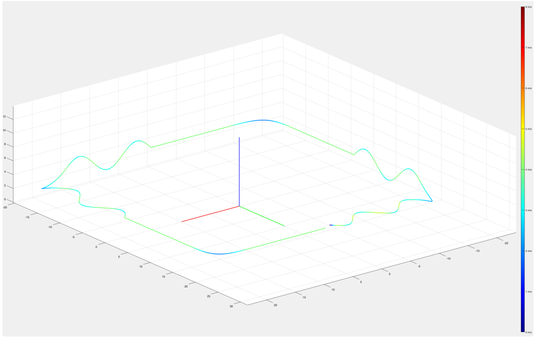


Figure 2.13: Target Flying (TF) - The target is flying, indicating controlled movement, likely in a horizontal direction.

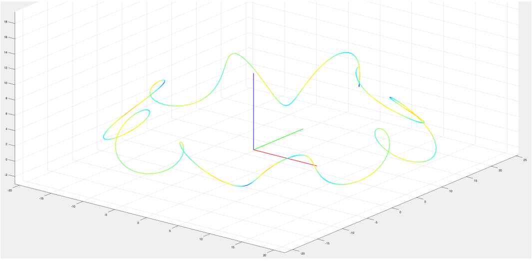


Figure 2.14: Target Maneuvering (TMa) - The target is actively performing complex and abrupt maneuvers.

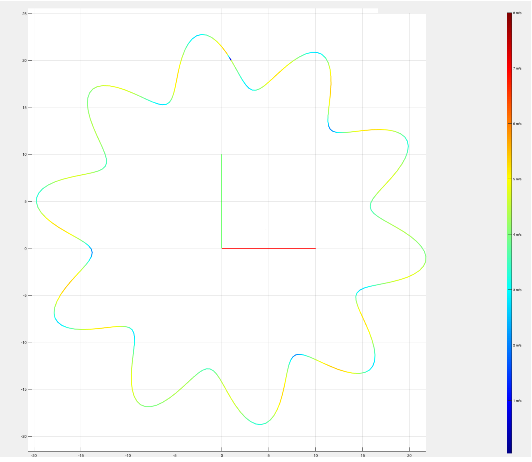


Figure 2.15: View on Target Maneuvering (TMa) from top.

2.5.3 Results

The performance metrics for the three different filters: Constant Velocity Kalman Filter (CVKF), Constant Velocity Filter with Enhanced Measurement Covariance (CV_EMV), and Interacting Multiple Model (IMM) Filter, are quite distinct. In general, IMM outperforms the other two across the considered scenarios.

Looking at the Constant Velocity Kalman Filter (Table 2.2), we observe that the filter performs reasonably well, with track confirmation percentage (TCP) generally above 70%. However, the root mean square error (RMSE) and average normalized estimation error squared (ANEES) for position and velocity are high, indicating that the estimated states deviate significantly from the true states.

For the CVKF with enhanced measurement covariance (Table 2.3), the tracking performance improves notably. The track confirmation percentage is higher than with the standard CVKF, reaching up to 98.13% in the Hovering Interceptor Target Moving scenario. The RMSE and ANEES metrics also decrease, showing a more accurate estimation of the target states.

Table 2.2: Performance metrics for the Constant Velocity Kalman Filter

		TIC	TCP (%)	RMSE Pos.	RMSE Vel.	ANEES Pos.	ANEES Vel.
HI	**TM**	8	93.75	1.01	1.12	247.51	968.60
	TF	7	81.69	1.96	1.22	353.94	922.85
	TMa	12	94.68	1.44	1.99	343.90	382.47
FI	**TM**	7	87.95	0.92	0.92	179.49	815.46
	TF	7	87.48	0.94	1.30	123.68	1169.39
	TMa	14	91.55	0.91	1.89	77.63	353.64
MI	**TM**	12	71.83	1.28	0.89	222.96	849.23
	TF	12	86.23	1.14	1.19	271.44	1329.89
	TMa	14	82.94	1.17	2.05	170.36	469.92

Table 2.3: Performance metrics for the Constant Velocity Filter with Enhanced Measurement Covariance

		TIC	TCP (%)	RMSE Pos.	RMSE Vel.	ANEES Pos.	ANEES Vel.
HI	**TM**	4	98.13	0.79	1.10	34.87	54.14
	TF	6	85.94	0.94	1.27	45.40	64.29
	TMa	11	81.09	1.08	1.98	66.87	199.72
FI	**TM**	3	95.31	0.72	0.88	25.08	32.68
	TF	7	86.56	0.80	1.21	28.85	61.74
	TMa	8	90.16	1.01	1.89	53.26	167.98
MI	**TM**	7	75.47	1.14	0.97	39.87	34.66
	TF	7	79.38	1.96	1.16	52.18	39.19
	TMa	10	82.94	1.28	1.97	69.23	167.35

The IMM filter (Table 2.4) demonstrates superior performance in all scenarios. The track Interruption Count (TIC) is generally lower than in the previous filters. TCP reaches 100% in the Hovering Interceptor Target Flying scenario, indicating highly reliable tracking. Furthermore, the RMSE and ANEES metrics are the lowest, showing that IMM provides the most accurate estimations among the three filters. In conclusion, the IMM filter demonstrates superior tracking performance under various conditions and therefore is the most robust choice for UAV tracking tasks.

Table 2.4: Performance metrics for the Interacting Multiple Model (IMM) Filter

		TIC	TCP (%)	RMSE Pos.	RMSE Vel.	ANEES Pos.	ANEES Vel.
HI	**TM**	2	98.59	0.44	0.58	10.65	3.31
	TF	0	100.00	0.40	0.84	10.21	7.16
	TMa	2	98.44	0.53	1.51	15.26	18.89
FI	**TM**	2	93.91	0.57	0.44	13.29	2.16
	TF	4	88.44	0.72	0.90	25.00	8.04
	TMa	3	88.44	0.66	1.69	16.44	23.83
MI	**TM**	3	75.63	1.42	0.56	87.46	3.75
	TF	3	94.38	1.52	0.99	40.33	10.75
	TMa	4	85.60	1.15	2.05	46.47	36.75

Chapter 3

Interception trajectory planning

After successful detection of the target drone, the next step is to intercept it. In this chapter, we explore and compare three different approaches to this challenge. First, a custom intercepting algorithm created prior by the MRS group is investigated. This algorithm uses a combination of a carefully chosen interception point and non-linear optimization (NLOPT) to plan the trajectory 3.1. Secondly, we investigate Proportional Navigation (PN). This is a method traditionally used homing missiles. We cover its variations, its strengths and weaknesses, and any improvements made to the original method 3.2. Lastly, we use of the nonlinear model predictive control for trajectory planning 3.3. These methods are compared in a realistic simulation with autonomous aerial interceptor.

3.1 Navigation Based on NLOPT

Prior to the research outlined in this thesis, a preliminary solution was developed by the MRS group. This method primarily relies on careful interception point selection, rapidly reaching this point, and subsequently aligning the trajectory with the estimated trajectory of the target.

The trajectory calculation in this method employs NLOPT [22], [24], which is a versatile, free, and open source library for non-linear optimization. NLOPT offers a unified interface for multiple optimization routines. It generates feasible trajectories using polynomial methods (e.g., splines). An example of this trajectory is shown in Fig. 3.1. Although this approach is functional, it bears a significant drawback as it relies heavily on the accuracy of the prediction of the target during the selection of the interception point. Any error in the estimation of the target's trajectory can lead to suboptimal results or a potential failure of the system. In the context of this work, the method of the MRS group will serve as the baseline solution against which the effectiveness of the new proposals will be evaluated.

3.2 Proportional Navigation

Proportional navigation (PN) is a well-established guidance method in the field of homing missiles [19]. It relies on the knowledge of the target's position and velocity. Since its inception, numerous variations and enhancements have been introduced. A number of these variations, and a new enhancement tailored to the drone interception scenarios is proposed.

3.2.1 Pure Proportional Navigation

A crucial concept to consider in PN is the Line of Sight (LOS) (see Fig. 3.2), which is a line between an observer and the target. The fundamental idea of PN comes from the

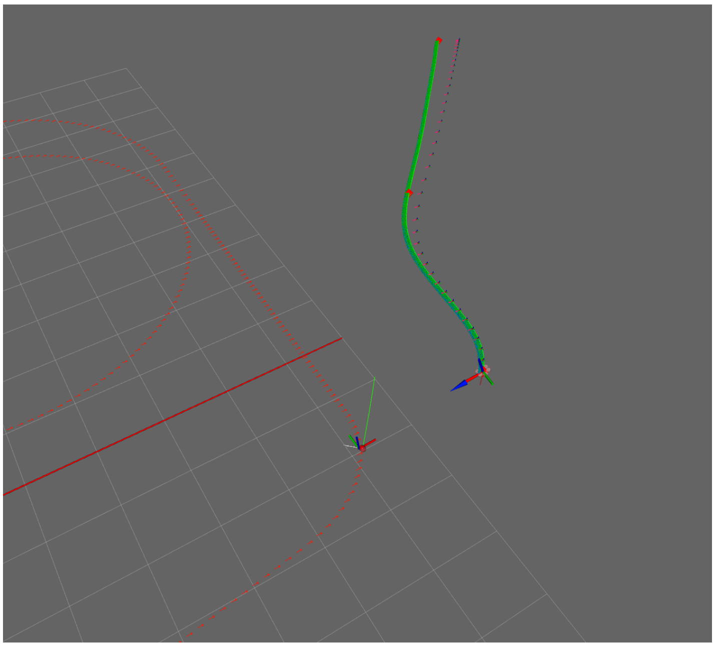


Figure 3.1: Representation of the interception point selection (first red waypoint) and trajectory alignment strategy developed by the MRS group.

observation that if the orientation of the LOS remains constant, the interceptor and the target are on a collision course. To utilize this observation, the rate of change of the LOS, denoted as $\dot{\lambda}$, is defined. The control law of PPN outputs the desired acceleration of the interceptor based on this rate of change. This is given by

$$a_i = N \cdot V_c \cdot \dot{\lambda}, \tag{3.1}$$

where a_i is the commanded acceleration to incerteptor perpendicular to velocity vector, N is a constant gain, typically set to 3 or 4 in most applications, and V_c is the interceptor velocity. In Eq. (3.1), we observe that the acceleration is determined proportionally to the LOS rate and the interceptor velocity, both multiplied by the gain N.

PPN commands acceleration perpendicular to the missile's velocity, thus maintaining its total velocity constant. This approach is suitable for missiles with control solely via aerodynamic surfaces. However, it's not quite suitable for drone interception as it introduces complexity in selecting the appropriate speed, which can complicate the interception process. Therefore, this variant is not employed.

3.2.2 True Proportional Navigation

In contrast to PPN, True Proportional Navigation (TPN) commands the interceptor perpendicular to its LOS. This method allows for changes in velocity magnitude. Despite these characteristics, TPN still encounters specific issues. Consider, for instance, a scenario where, after an unsuccessful interception attempt, the LOS rate remains zero because the interceptor was initially on a collision course with the target. In this situation, the interceptor would not initiate a second attempt, which is adequate in the context of missiles but not in the case of drones, which may necessitate multiple attempts.

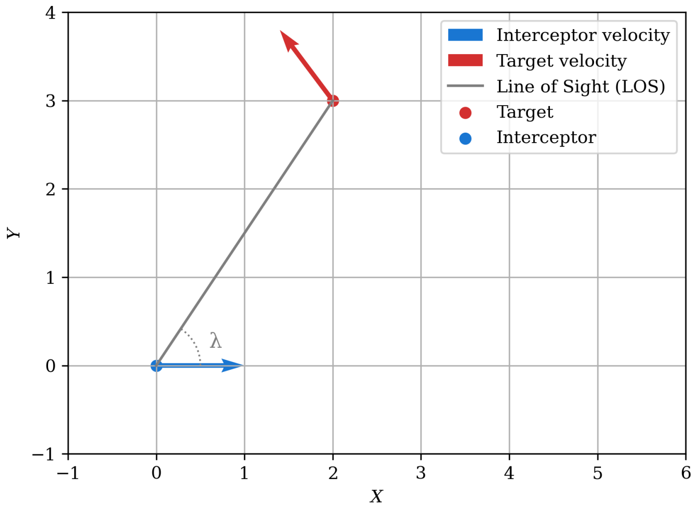


Figure 3.2: An illustration of the fundamental concepts of Proportional Navigation (PN), specifically showcasing the Line of Sight (LOS).

Secondly, if both the target and the interceptor are stationary, the LOS also does not change, therefore no engagement occurs. This is not a problem for missiles as TPN is typically employed only during the homing phase when the missile is already in proximity to the target. However, for drone interceptions, where multiple attempts may be desirable, this limitation can become problematic.

3.2.3 Optimal Guidance and Zero-Effort Miss (ZEM) Formulation of Proportional Navigation

The interception problems can be viewed through the lens of optimal control theory [26], [32], [30]. Assume that the target is non-maneuvering, and the control applied to the interceptor is the acceleration. This forms a system of equations

$$\dot{\boldsymbol{z}} = \begin{bmatrix} \dot{\boldsymbol{z}}_1 \\ \dot{\boldsymbol{z}}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{0} & \boldsymbol{I} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} \boldsymbol{z}_1 \\ \boldsymbol{z}_2 \end{bmatrix}}_{\boldsymbol{z}} + \underbrace{\begin{bmatrix} \mathbf{0} \\ -\boldsymbol{I} \end{bmatrix}}_{\mathbf{B}} \boldsymbol{u}, \quad \boldsymbol{u} = \boldsymbol{a}_i, \tag{3.2}$$

where $\boldsymbol{z}_1 = \boldsymbol{r}_t - \boldsymbol{r}_i$ represents the relative position between the target and the interceptor, with $\boldsymbol{r}_t$ denoting the target's position and $\boldsymbol{r}_i$ representing the interceptor's position. $\boldsymbol{z}_2 = \boldsymbol{v}_t - \boldsymbol{v}_i$ signifies the relative velocity between the target and the interceptor, with $\boldsymbol{v}_t$ as the target's velocity and $\boldsymbol{v}_i$ as the interceptor's velocity. $\mathbf{I}$ is a 3x3 identity matrix, and $\mathbf{0}$ is a 3x3 zero matrix. Finally, $\boldsymbol{u}$ is the control input of the system, with acceleration acting as the controlled quantity. In the equation, a_i is the acceleration vector of the interceptor.

The aim is to determine the control law $u(t)$, for $t_0 \leq t \leq T$, that minimizes the cost function

$$J\left(\mathbf{z}_0, u(\cdot), t_0\right) = \frac{1}{2}\mathbf{z}^\top(T)\mathbf{Q_f}\mathbf{z}(T) + \frac{1}{2}\int_{t_0}^{T} \mathbf{u}^2(t)dt, \tag{3.3}$$

subject to the dynamics defined by Eq. (3.2), with $\mathbf{z}(t_0) = \mathbf{z}_0$ and $t_0 \leq t \leq T$. The cost matrix $\mathbf{Q}_f$ is defined as:

$$\mathbf{Q}_f = \begin{bmatrix} b & 0 \\ 0 & c \end{bmatrix}, \tag{3.4}$$

where the b component indicates the weight given to minimizing the miss distance, and c corresponds to the emphasis on velocity matching at the point of interception.

This interception problem can be regarded as a quadratic optimization problem [30]. Leveraging optimal control theory [32] and solving the associated Riccati equation, the general solution [26] for the control input can be expressed as

$$\mathbf{u}^* = \frac{3}{t_{go}^2}\left[\frac{\left(1 + \frac{c}{2}t_{go}\right)\mathbf{z}_1 + \left(\frac{c}{3}t_{go} + 1 + \frac{c}{b}t_{go}^2\right)t_{go}\mathbf{z}_2}{1 + \frac{3}{bt_{go}}\left(1 + ct_{go}\right) + \frac{c}{4}t_{go}}\right], \tag{3.5}$$

where t_{go} denotes the time-to-go until interception. Note that the above control law adjusts the acceleration of the interceptor based on both the current relative position ($\mathbf{z}_1$) and the relative velocity ($\mathbf{z}_2$) between the target and interceptor, as well as the time-to-go. The the time-to-go, t_{go}, is estimated as the ratio $\frac{||\mathbf{z}_1||}{||\mathbf{z}_2||}$. This ensures a dynamic adjustment mechanism.

However, if we neglect the importance of meeting velocity, and consider the limiting case of $b \to \infty$, the control law simplifies to

$$\lim_{b\to\infty, c\to 0}\left\{\mathbf{u}^*\left(t_{go}; b, c\right)\right\} = \frac{3\left(\mathbf{z}_1 + t_{go}\mathbf{z}_2\right)}{t_{go}^2}, \tag{3.6}$$

This simplification leads to the control law

$$\mathbf{u}^* = 3\frac{ZEM}{t_{go}^2}, \tag{3.7}$$

where ZEM signifies the Zero-Effort Miss distance. While it may not be immediately apparent that the optimal control law in Eq. (3.5) and the PN law in Eq. (**??**) share a similar structure, it can be demonstrated [26] that they are indeed equivalent to each another, and the optimal gain, N, is 3.

This formulation allows for multiple attempts. However, it doesn't initiate engagement when the relative velocity z_2 is zero. Furthermore, during initial simulations, it was observed that the second engagement occurs slowly, and that for certain trajectories (e.g., circular), this control methodology can lead to the interceptor stalling in a fixed position behind a turning target.

To address these limitations, I propose a modification to the control law. The key idea is to produce higher control inputs when the target and interceptor are futher apart or not moving. This is accomplished by introducing a new term to the control law:

$$\mathbf{a}_i = N \cdot \frac{\mathbf{z}_1 + \mathbf{z}_2 t_{go}}{t_{go}^2} - \frac{1}{H}\|\mathbf{z}_1\|^2 \frac{\mathbf{z}_1}{\|\mathbf{z}_1\|}. \tag{3.8}$$

where H serves as a design parameter that balances the importance between the distance to target and the prediction accuracy when utilizing Proportional Navigation (PN). Fig. 3.3 shows vector field of desired acceleration for PN, while Fig. 3.4 shows how the field changes with this modification.

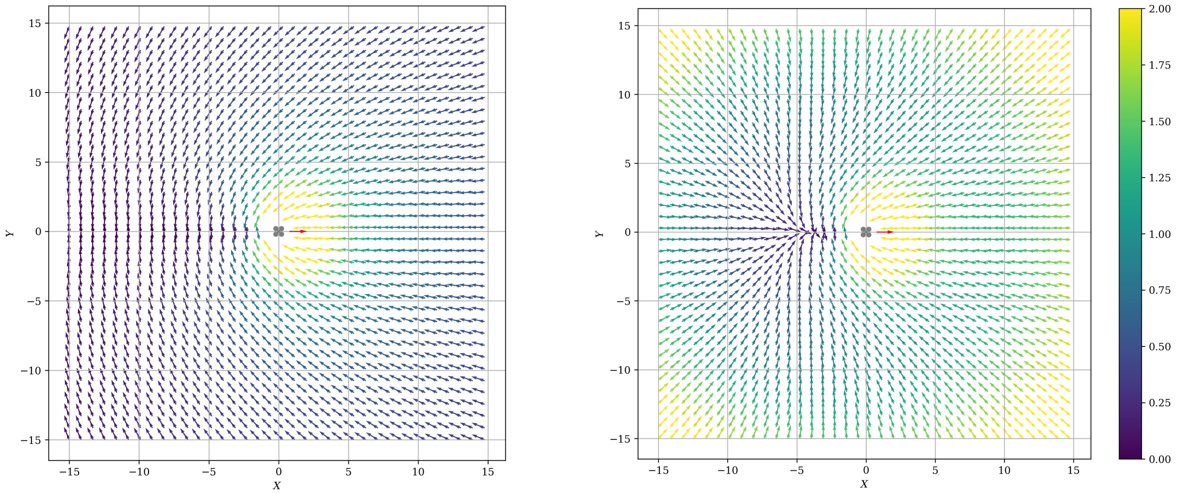


Figure 3.3: Simulation of Proportional Navigation (PN) in the form of Zero-Effort Miss (ZEM).

Figure 3.4: Enhanced Proportional Navigation implementation with improved interception strategy.

Figure 3.5: Comparison of original and Enhanced PN. Colors code desired acceleration. Target UAV is in origin. Acceleration of interceptor in hypothetical positions is denoted using arrows.

The proposed modification in eq. (3.8) ensures that the control input increases with the distance between the interceptor and target and decreases when they are stationary. This creates a more responsive and robust control system, preventing the interceptor from becoming trapped in a constant position behind a maneuvering target. The desired acceleration is adjusted to adhere to the interceptor's limits, and using the current state of the interceptor, a trajectory is sampled. This trajectory is subsequently published to the pipeline as referenced in 3.16. Simulation of first two interception attempts is depicted in Fig. 3.10.

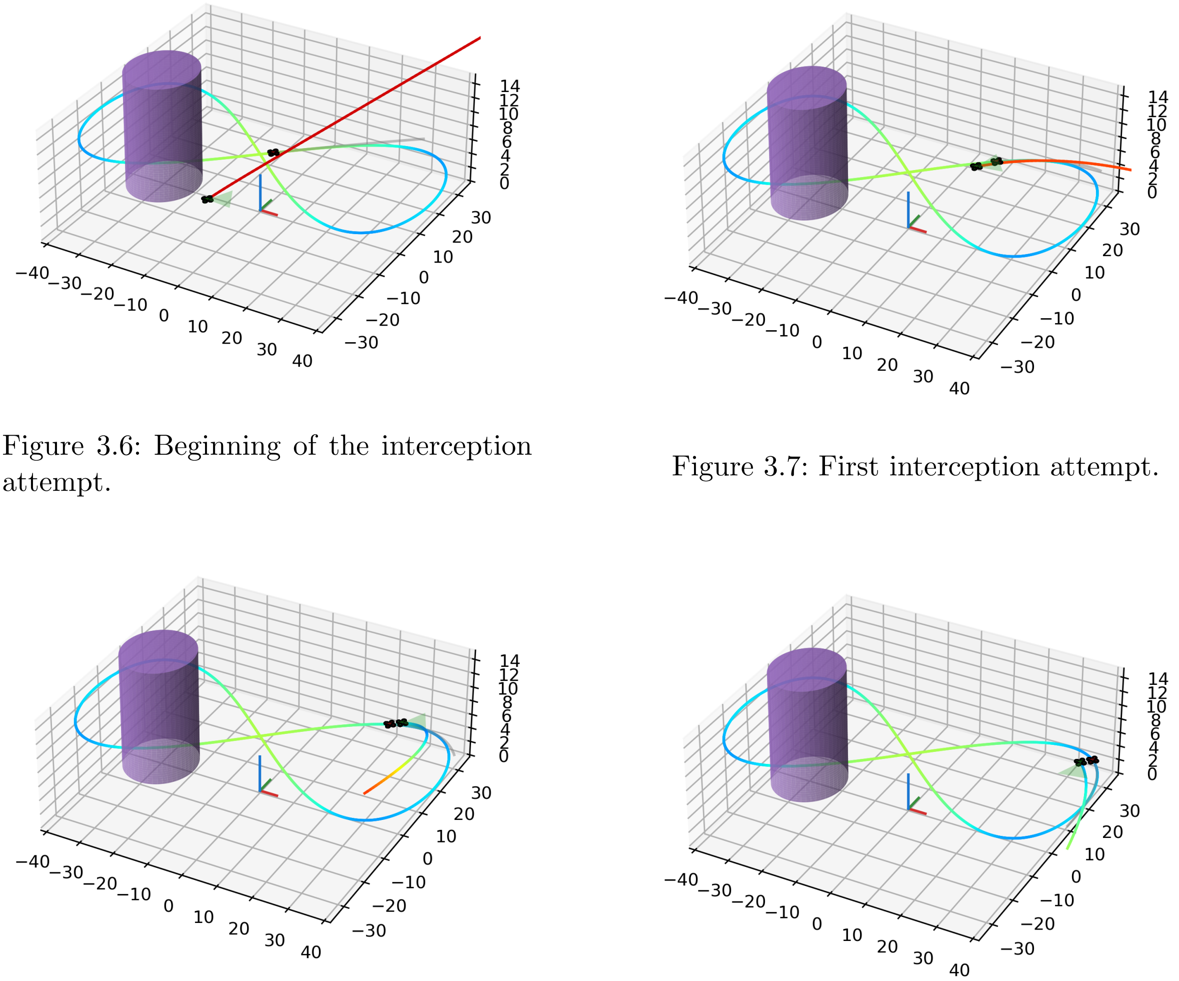


Figure 3.6: Beginning of the interception attempt.

Figure 3.7: First interception attempt.

Figure 3.8: After the first interception attempt.

Figure 3.9: Second interception attempt.

Figure 3.10: Proportional Navigation (PN) Simulation. Color codes velocity. Maximal velocity (red) is 12 m/s.

3.3 Nonlinear Model Precitive Control

Nonlinear Model Predictive Control (NMPC) is a control strategy that solves an open-loop optimal control problem at each sampling time instant, using the current state of the system as the initial condition, over a finite horizon, to obtain an optimal control sequence. This sequence is determined by minimizing a cost function, subject to the system's dynamics and other constraints. The first control input of the optimal sequence is then implemented on the system.

There are various strategies to solve this problem, and one such strategy is the Direct Single Shooting method [34]. Single Shooting is a direct method for solving optimal control problems where the control sequence is parameterized, and the system dynamics are integrated forward in time for each potential control sequence. The control sequence that minimizes the discrepancy at the end of the prediction horizon, the so-called "shooting gap", is selected as the optimal control input.

The Single Shooting method involves solving the following optimization problem:

$$\begin{gathered}\underset{u(k)}{\text{minimize}} \quad J(\mathbf{u}(k), \mathbf{x}(0)), \quad \forall k \in \{0, 1, \ldots, N\} \\ \text{subject to} \\ \mathbf{x}(0) \text{ is given}, \quad \forall k, \\ f(\mathbf{u}(k), \mathbf{x}(0)) \leq 0, \quad \forall k, \\ g(\mathbf{u}(k), \mathbf{x}(0)) = 0, \quad \forall k, \\ \mathbf{u}(k) \text{ lies within defined constraints}, \quad \forall k, \end{gathered} \tag{3.9}$$

In this formulation, the system dynamics are intentionally omitted as all states are expressed as a function of $\mathbf{u}$ and $\mathbf{x}_0$. Constraints on the states $\mathbf{x}(k)$ can be represented using the functions f and g.

The optimization problem is solved over a prediction horizon of length N, with k ranging from 0 to N. The control sequence $\mathbf{u}(k)$, defined for each k within this interval, is typically initialized with an estimate, and the optimization algorithm iteratively refines this estimate to find the sequence that minimizes the objective function $J(\mathbf{u}(k), \mathbf{x}(0))$. This procedure continues until a specified termination condition, such as a maximum number of iterations or a tolerable level of precision, is met.

3.3.1 NMPC control problem formulation

For the purpose of formulating control problem to be solved by the NMPC, both, the interceptor and target are modeled as point-mass up to their acceleration. This is a good compromise between fidelity and computational time. The objective function is to minimizing predicted distances over the control horizon.

The dynamics of this discrete-time Linear Time-Invariant (LTI) system can be represented in a state-space form. Let $\mathbf{x}[k] \in \mathbb{R}^{15}$ represent the state of the system, and $\mathbf{u}[k] \in \mathbb{R}^3$ be the control input. The system is defined as follows:

$$\mathbf{x}[k+1] = \mathbf{A}\mathbf{x}[k] + \mathbf{B}u[k] \tag{3.10}$$

where the state transition matrix $\mathbf{A} \in \mathbb{R}^{15\times15}$ and the control input matrix $\mathbf{B} \in \mathbb{R}^{15\times3}$ are given by:

$$\mathbf{A} = \begin{bmatrix} \mathbf{I} & \Delta t\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \Delta t\mathbf{I} \\ \mathbf{I} & \Delta t\mathbf{I} & \frac{\Delta t^2}{2}\mathbf{I} \\ \mathbf{0} & \mathbf{I} & \Delta t\mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}, \tag{3.11}$$

$$\mathbf{B} = \begin{bmatrix} \frac{\Delta t^2}{2}\mathbf{I} \\ \Delta t\mathbf{I} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}. \tag{3.12}$$

Here, $\mathbf{I}$ is the 3×3 identity matrix, $\mathbf{0}$ is a 3×3 zero matrix, and Δt is the time step of our discrete-time system. Even though that model is linear, NMPC was chosen because of trial experimentation with objective functions and its possibility to include obstacle avoidance in the future.

To define constraints on the velocities and accelerations, let us denote the maximum velocities as $\mathbf{v}_{\mathrm{x\ max}}$, $\mathbf{v}_{\mathrm{y\ max}}$, $\mathbf{v}_{\mathrm{z\ max}}$ and maximum acceleration as $\mathbf{a}_{\mathrm{x\ max}}$, $\mathbf{a}_{\mathrm{y\ max}}$, $\mathbf{a}_{\mathrm{z\ max}}$. All parameters for NMPC control problem are in table 3.1. The optimal control problem is then:

$$\min_{\mathbf{u}} \left\|\mathbf{x}[N]_{1:3} - \mathbf{x}[N]_{7:9}\right\|^2 + \sum_{k=0}^{N-1} \left\|\mathbf{x}[k]_{1:3} - \mathbf{x}[k]_{7:9}\right\|^2, \tag{3.13}$$

subject to

$$\mathbf{x}[0] = \mathbf{x}_0, \tag{3.14}$$

$$\mathbf{x}[k+1] = \mathbf{A}\mathbf{x}[k] + \mathbf{B}\mathbf{u}[k], \quad k = 0, \dots, N-1, \tag{3.15}$$

$$-v_{\mathrm{x\ max}} \le \mathbf{x}[k]_4 \le v_{\mathrm{x\ max}}, \quad k = 1, \dots, N, \tag{3.16}$$

$$-v_{\mathrm{y\ max}} \le \mathbf{x}[k]_5 \le v_{\mathrm{y\ max}}, \quad k = 1, \dots, N, \tag{3.17}$$

$$-v_{\mathrm{z\ max}} \le \mathbf{x}[k]_6 \le v_{\mathrm{z\ max}}, \quad k = 1, \dots, N, \tag{3.18}$$

$$-a_{\mathrm{x\ max}} \le \mathbf{u}[k]_1 \le a_{\mathrm{x\ max}}, \quad k = 0, \dots, N-1, \tag{3.19}$$

$$-a_{\mathrm{y\ max}} \le \mathbf{u}[k]_2 \le a_{\mathrm{y\ max}}, \quad k = 0, \dots, N-1, \tag{3.20}$$

$$-a_{\mathrm{z\ max}} \le \mathbf{u}[k]_3 \le a_{\mathrm{z\ max}}, \quad k = 0, \dots, N-1. \tag{3.21}$$

The formulation provided above, while not directly applicable to the single shooting method, is used for its superior clarity. However, it will subsequently be transformed for compatibility with the single shooting method using OpEn [12].

Description	**Parameter**	**Value**
Prediction horizon length	N	20
Time step	Δt	0.4 s
Maximum velocity	$v_{\mathrm{max},x}$, $v_{\mathrm{max},y}$, $v_{\mathrm{max},z}$	8 m/s, 8 m/s, 3 m/s
Maximum acceleration	$a_{\mathrm{max},x}$, $a_{\mathrm{max},y}$, $a_{\mathrm{max},z}$	3 m/s^2, 3 m/s^2, 2 m/s^2

Table 3.1: Optimization parameters used for the NMPC control problem.

To solve the optimal control problem using NMPC, several off-the-shelf optimizers were considered. One suitable choice is the OpEn optimizer [12], which utilizes the Proximal Averaged Newton-type method (PANOC) [21]. OpEn has been successfully applied to control UAVs and ground robots in previous works [8], [11], [15], [18]. It is capable of handling challenging optimization problems involving nonlinearities and constraints, making it appropriate for our current problem. Using an optimizer like OpEn can significantly boost the computational efficiency of the NMPC algorithm, making it suitable for real-time applications. During initial experiments with doMPC[1] based on [20], I found that OpEn offered superior performance speed. Simulation of first two interception attempts is depicted in Fig. 3.15.

[1] `https://www.do-mpc.com/en/latest/`.

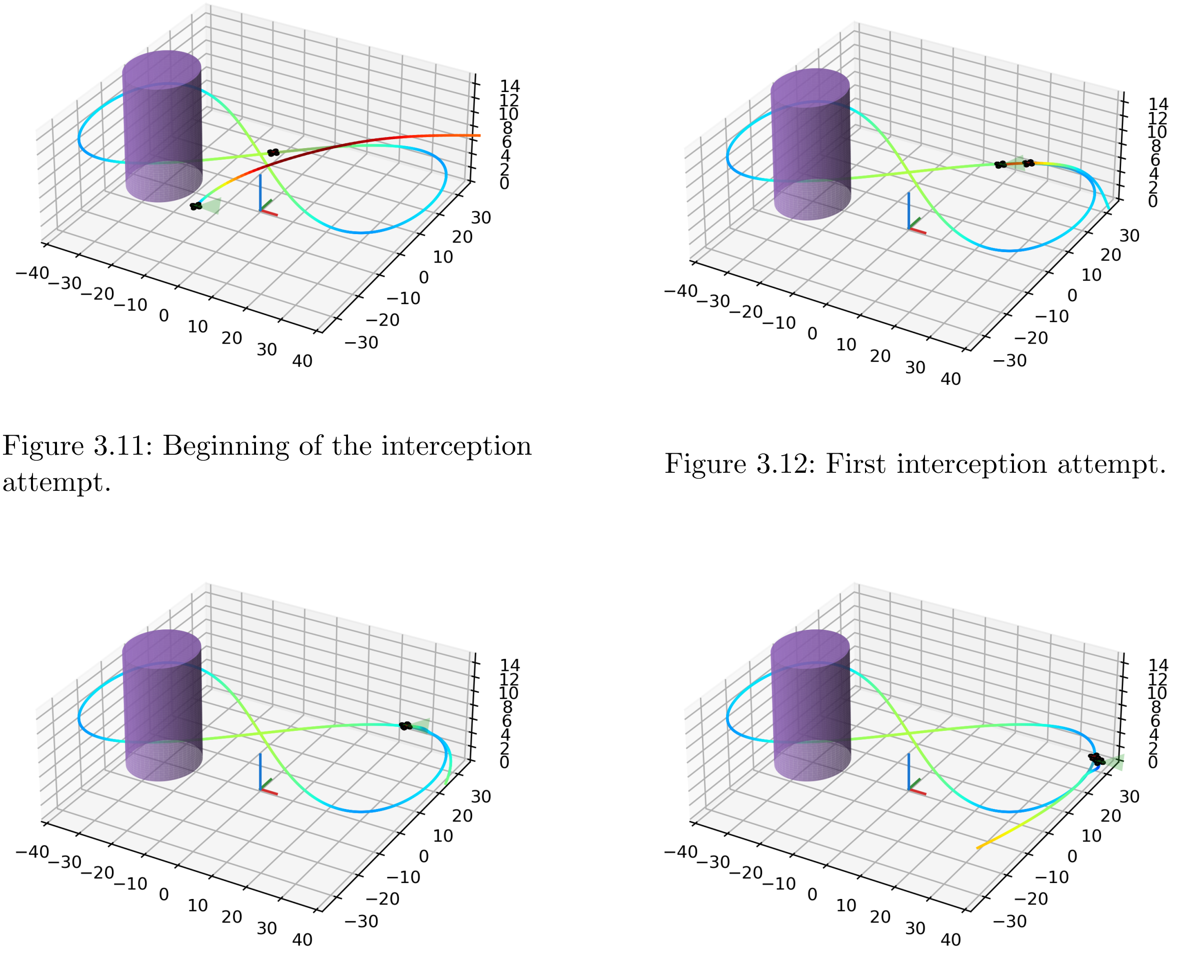


Figure 3.11: Beginning of the interception attempt.

Figure 3.12: First interception attempt.

Figure 3.13: After the first attempt.

Figure 3.14: Second interception attempt.

Figure 3.15: Simulation of NMPC. Color codes velocity. Maximal velocity (red) is 12 m/s.

3.4 Heading control

For all types of control, the heading of the interceptor is commanded to face the target at all times. The desired heading is dynamically adjusted based on the relative positions of the interceptor and the target. This concept can be formulated mathematically as:

$$\eta_d = \mathrm{atan2}\left(y_t - y_i, x_t - x_i\right), \tag{3.22}$$

where η_d is the desired heading, (x_i, y_i) are the coordinates of the interceptor, and (x_t, y_t) are the coordinates of the target. Here, the atan2 function is used instead of the standard atan function to account for the full range of possible angles (0 to 2π), ensuring that the correct quadrant for the angle is chosen based on the signs of $x_t - x_i$ and $y_t - y_i$.This continuous adjustment of the interceptor's heading ensures that it always faces the target, which is crucial for accurate and effective interception.

3.5 UAV Control

The planned trajectories are subsequently published to the MRS (Multi-Robot Systems) control system [7], [17]. This stack is designed to process all aspects of the planned trajectories, including heading, to ensure a stable and consistent flight trajectory, which is executed by the system. An overview of the MRS control stack and how it interfaces with the planned trajectories is illustrated in Figure 3.17.

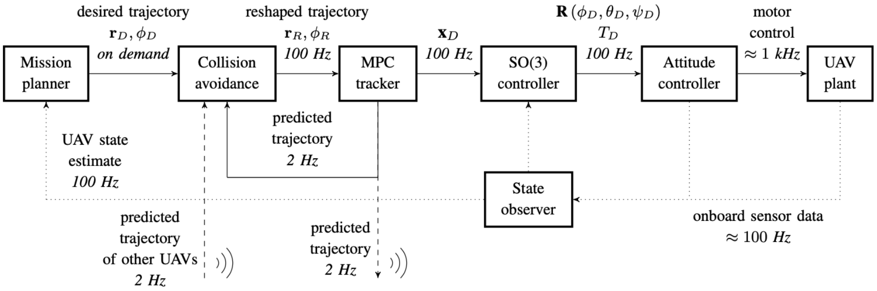


Figure 3.16: Schematic representation of the MRS control stack interacting with the planned trajectories. The diagram illustrates how the optimal trajectories are published to the stack, which then processes them to ensure stable flight control of the UAV. All of the proposed navigations are inside of mission logic. Collision avoidance is offline during all interception attempts.

Using the MRS control stack, we ensure that the UAV follows the optimal trajectory as closely as possible, while also considering real-world constraints and flight dynamics. This approach combines the mathematical optimality of our trajectory planning with robust, field-tested UAV control strategies. It allows the UAV to effectively follow the desired trajectory and aim towards the target, ultimately ensuring successful interception.

3.6 Results

This section provides a summary of the experimental evaluation of the functionality and effectiveness of the entire interception pipeline. The approach and methods employed in this work have been rigorously tested in a simulation and further validated with a real-world experiment.

3.6.1 Simulations

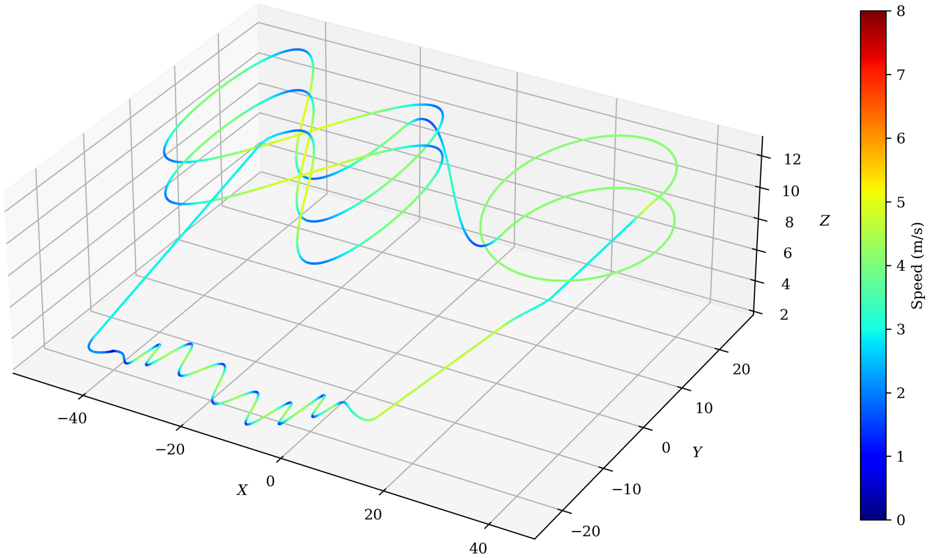


Figure 3.17: Visualization of the testing trajectory color-coded according to velocity.

Trajectory of the target used for testing, where the color map encodes the velocity, is illustrated on Fig. 3.17. One round of the trajectory takes approximately 400 seconds. For each intercepting algorithm, two rounds were simulated. The interceptor, target, and lidar models align closely with reality. MRS UAV system [7], [17] with Gazebo[2] serves as the simulation tool.

An interception attempt is defined as closing the distance between the UAVs to less than two meters. The maximum duration for each attempt is set to 4 seconds to avoid counting near-flight as a multiple interception attempts. The results of these simulations are summarized in table 3.2. The first column shows the number of attempts during the whole simulation. The second column is the average miss distance, and its standard deviation is in the last column. All attempts are ploted also in Figs. 3.18 - 3.23.

Method	**Attempts**	**Mean Distance (m)**	**Std Distance (m)**
NLOPT	11	1.54	0.30
PN	40	0.85	0.51
NMPC	39	0.25	0.22

Table 3.2: Results of the simulated interception attempts.

Each dot represents one interception attempt. To gain better insight, a color map is used to represent the miss distance, ranging from 0 meters (represented in blue) to 2 meters (represented in red).

[2] `https://gazebosim.org/home`

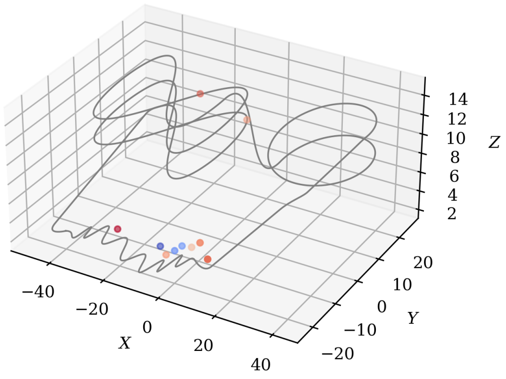


Figure 3.18: Simulation results of the navigation based on NLOPT method. Color map represents the miss distance in meters.

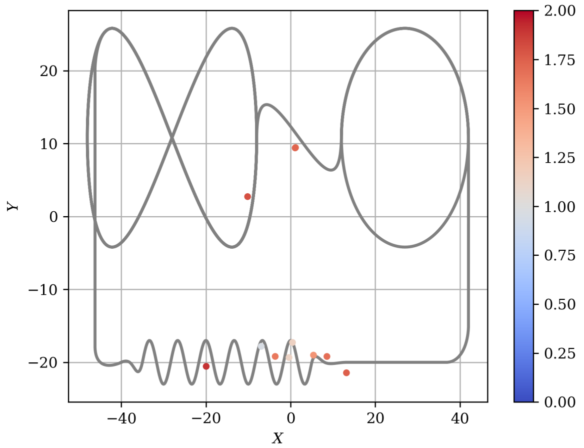


Figure 3.19: Top view. All points exhibit high miss distances. The navigation system is unable to execute interception attempts in certain passages.

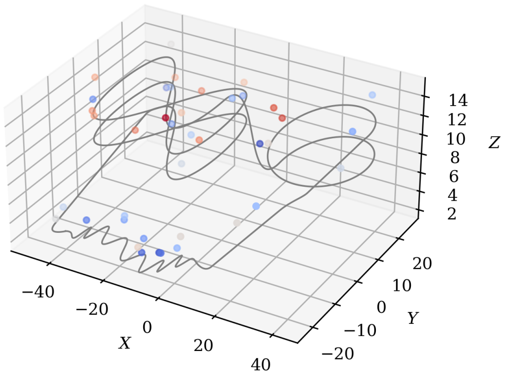


Figure 3.20: Simulation results of the PN method. Color map represents the miss distance in meters.

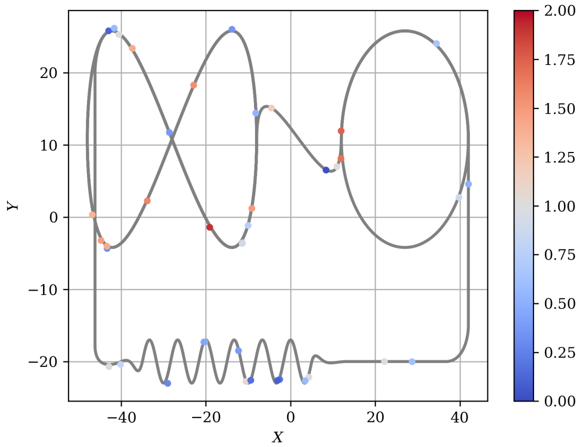


Figure 3.21: Top view. It can be observed that the mean color of points changes in relation to the target's motion mode.

Based on the presented results, all the tested methods - Navigation Based on NLOPT, PN, and NMPC - were able to conduct multiple interception attempts, with varying degrees of success. NLOPT showed the least promising results among the three methods. It had the lowest number of attempts and the highest mean and standard deviation of the miss distance. This indicates a lower level of accuracy and consistency in the interception attempts. While it was able to execute interceptions, its performance appears somewhat random rather than systematic, making it less reliable for the task at hand.

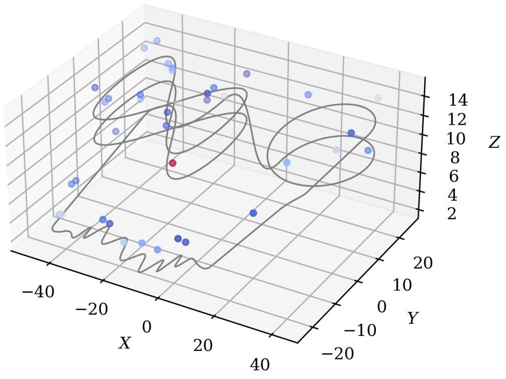


Figure 3.22: Simulation results of the NMPC method. Color map represents the miss distance in meters.

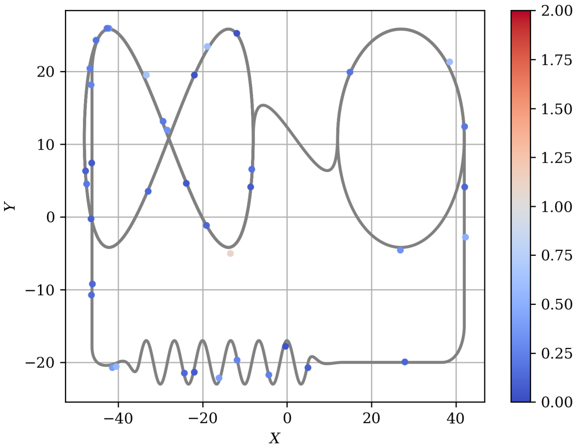


Figure 3.23: Top view. The miss distance remains consistently low during all target motion modes.

In contrast, the PN based method performed notably better than NLOPT. It managed significantly more attempts with a considerably lower mean miss distance, showcasing a higher degree of precision. However, its standard deviation was higher than that of NMPC, indicating a slightly lower consistency.

The NMPC method demonstrated superior effectiveness, registering the smallest mean and standard deviation of the miss distance. This suggests a higher level of precision and consistency compared to the other methods. The robust performance of the NMPC algorithm underscores its promising potential for successful application in interception tasks.

In conclusion, while NLOPT managed to conduct successful interceptions, its performance was more sporadic than systematic, marking it as the least reliable among the methods tested. The PN method displayed commendable performance, highlighting its capability for interception tasks, despite having slightly less consistency than the NMPC. However, the outstanding performance of the NMPC algorithm in terms of both precision and consistency points to its potential as the most promising method for such tasks. The encouraging results in the simulated environment lay a solid foundation for further exploration and validation of these methods in real-world scenarios, with particular focus on the NMPC algorithm.

3.6.2 Real-world test

The Proportional Navigation (PN) algorithm was also tested in a real-world environment. The test involved two platforms: Eagle One[3], which was manually airborned and then controlled autonomously using the interception algorithm, and a DJI Mavic[4], which was piloted by humans and used as an agile target.

[3] https://eagle.one/cs/.

[4] https://www.dji.com/cz/mavic.

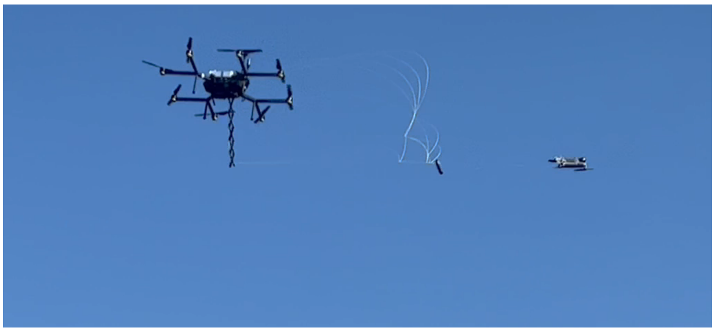


Figure 3.24: Photo of the successful interception attempt by Eagle One catching DJI Mavic

The Eagle One is equipped with a net intended to catch the target drone by entangling its rotors. The mid-flight deployment of the net is shown in Fig.3.25.

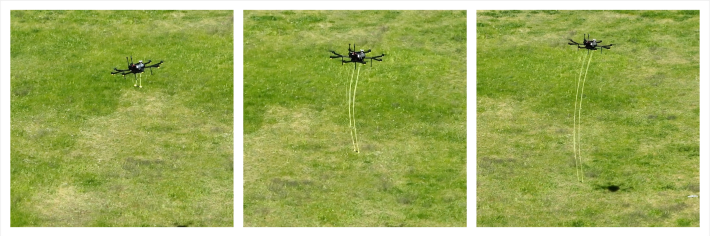


Figure 3.25: Deployment of the net on Eagle One.

The PN based trajectory planning proved to be successful in catching the drone. The Fig.3.26 showcase the progression of a successful interception attempt (Fig.3.24).

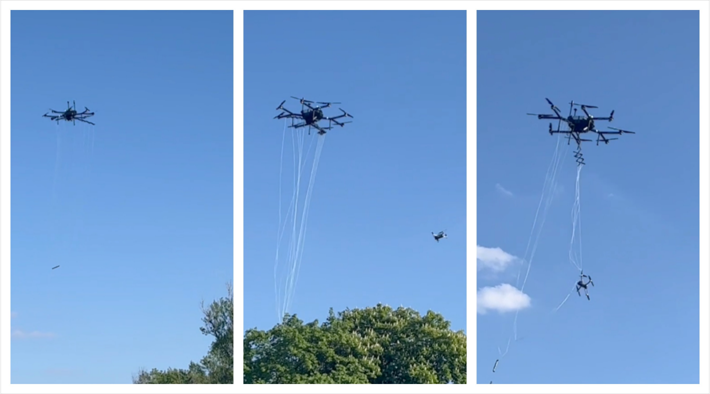


Figure 3.26: The progression of a successful interception attempt.

After a successful interception, Eagle One has the capability to release the caught drone and attempt another interception. Fig. 3.27 shows the net being dropped.

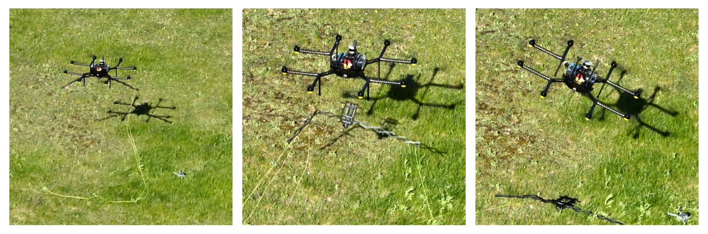


Figure 3.27: Eagle One dropping the caught drone.

A video showcasing the real-world experiment can be viewed **here**[5].

In conclusion, the success of the PN algorithm in real-world tests is a significant accomplishment. This achievement not only validates the effectiveness of the methods in a controlled simulated environment, but also their applicability and robustness in real-world conditions. It underscores the potential of these methods for effective use in practical drone interception tasks, thus paving the way for further advancements in this field.

[5] `https://www.youtube.com/watch?v=XCsJ5hSPWT4`.

Chapter 4

Conclusion and future work

This thesis focused on the subject of autonomous aerial interception of intruding UAVs using a separate UAV for this operation. To address this, a novel method of detection and tracking was extended and enhanced through an in-depth analysis of LIDAR measurement uncertainty, capturing the flying characteristics of the intercepting drone. After confirming its functionality, the method was further improved by implementing the Interactive Multiple Model (IMM) filter algorithm, which demonstrated superior tracking of highly maneuverable targets. All the considered tracking methods were rigorously tested in simulation and compared using various metrics, with IMM exhibiting superior performance.

For interception trajectory planning, several algorithms were employed. A custom planning method based on NLOPT served as the baseline method for comparison with new methods introduced in this thesis, namely the modification of Proportional Navigation (PN) and planner based on Nonlinear Model Predictive Control (NMPC). PN was thoroughly analyzed, and several in-depth observations were documented. Furthermore, the PN was adapted to suit drone-related interceptions. Model Predictive Control was utilized with a straightforward objective, exploiting the benefits of this type of control.

All methods were rigorously tested and compared in a simulation using multiple metrics. Among these methods, PN and NMPC significantly outperformed the baseline algorithm based on NLOPT, with NMPC demonstrating superior performance across all metrics.

PN was also tested in the real world cathing a human-piloted target, leading to a successful interception. This successful transition from simulation to real-world application represents a significant milestone, demonstrating the overall effectiveness and practicality of the methods developed in this thesis. The accomplishment not only validates the theoretical developments but also underscores their real-world applicability and potential for addressing the emerging challenges of drone interception.

Future research directions include the enhancement of the IMM filter with the addition of more models, possibly nonlinear ones, utilizing an Unscented Kalman Filter (UKF) and taking full advantage of the drone's dynamic properties. This advancement could significantly improve the accuracy and efficiency of the interception process. Moreover, the automatic tuning of filters through data-driven methods could be explored, given the challenging nature of manual tuning observed in this work.

In terms of the NMPC approach, further experiments could be conducted with various objectives. For instance, minimizing the time when the target drone is in occlusion under the interceptor drone might be an interesting area to investigate. This could provide additional insights into the optimal control of the interception process. Also, it should be possible to incorporate obstacle avoidance, minimizing risks during interception. In addition, integration of obstacle avoidance strategies within the NMPC framework could be considered to further enhance the safety and reliability of drone interception.

Furthermore, a more extensive investigation into target drone behavior could be beneficial. In this work, the target drone moved along unknown yet given trajectories and was only tested against active evasion maneuvers in real-world scenarios. Future work could entail the implementation of more behavioral modes based on observed target behavior. This could provide a richer understanding of interception dynamics, potentially leading to more robust and versatile interception methods.

Chapter 5

References

[1] P. M. Gupta, Pairet, T. Nascimento, and M. Saska, “Landing a uav in harsh winds and turbulent open waters,” *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 744–751, 2023. DOI: 10.1109/LRA.2022.3231831.

[2] M. Petrlik, P. Petracek, V. Kratky, *et al.*, “UAVs Beneath the Surface: Cooperative Autonomy for Subterranean Search and Rescue in DARPA SubT,” *Field Robotics*, vol. 3, pp. 1–68, 2023. DOI: https://doi.org/10.55417/fr.2023001.

[3] V. Pritzl, M. Vrba, C. Tortorici, R. Ashour, and M. Saska, “Adaptive estimation of uav altitude in complex indoor environments using degraded and time-delayed measurements with time-varying uncertainties,” *Robotics and Autonomous Systems*, vol. 160, p. 104 315, 2023, ISSN: 0921-8890. DOI: https://doi.org/10.1016/j.robot.2022.104315.

[4] M. Vrba, V. Walter, and M. Saska, *On onboard lidar-based flying object detection*, 2023. arXiv: 2303.05404 [cs.RO].

[5] T. M. Inc., *Matlab version: 9.13.0 (r2022b)*, Natick, Massachusetts, United States, 2022. [Online]. Available: https://www.mathworks.com.

[6] M. Vrba, Y. Stasinchuk, T. Báča, *et al.*, “Autonomous capture of agile flying objects using UAVs: The MBZIRC 2020 challenge,” *Robotics and Autonomous Systems*, vol. 149, p. 103 970, 2022, ISSN: 0921-8890. DOI: https://doi.org/10.1016/j.robot.2021.103970. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889021002396.

[7] T. Baca, M. Petrlik, M. Vrba, *et al.*, “The MRS UAV System: Pushing the Frontiers of Reproducible Research, Real-world Deployment, and Education with Autonomous Unmanned Aerial Vehicles,” *Journal of Intelligent & Robotic Systems*, vol. 102, no. 26, pp. 1–28, 1 2021. DOI: 10.1007/s10846-021-01383-5. [Online]. Available: https://link.springer.com/article/10.1007/s10846-021-01383-5.

[8] B. Lindqvist, P. Sopasakis, and G. Nikolakopoulos, “A scalable distributed collision avoidance scheme for multi-agent UAV systems,” in *Int Conf on Intelligent Robots and Systems (IROS)*, 2021.

[9] P. Petracek, V. Kratky, M. Petrlik, T. Baca, R. Kratochvil, and M. Saska, “Large-scale exploration of cave environments by unmanned aerial vehicles,” *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7596–7603, 2021. DOI: 10.1109/LRA.2021.3098304.

[10] G. Torrente, E. Kaufmann, P. Foehn, and D. Scaramuzza, *Data-driven mpc for quadrotors*, 2021. arXiv: 2102.05773 [cs.RO].

[11] S. S. Mansouri, C. Kanellakis, E. Fresk, *et al.*, “Subterranean mav navigation based on nonlinear MPC with collision avoidance constraints,” in *IFAC World Congress*, Berlin, Germany, 2020.

[12] P. Sopasakis, E. Fresk, and P. Patrinos, “OpEn: Code generation for embedded nonconvex optimization,” in *IFAC World Congress*, Berlin, Germany, 2020.

[13] M. Vrba and M. Saska, “Marker-less micro aerial vehicle detection and localization using convolutional neural networks,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2459–2466, 2020, ISSN: 2377-3766. DOI: 10.1109/LRA.2020.2972819.

[14] T. Rouček, M. Pecka, P. Čížek, *et al.*, "Darpa subterranean challenge: Multi-robotic exploration of underground environments," in *Modelling and Simulation for Autonomous Systems*, 2019, pp. 274–290, ISBN: 978-3-030-43890-6. DOI: 10.1007/978-3-030-43890-6_22. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-43890-6_22.

[15] E. Small, P. Sopasakis, E. Fresk, P. Patrinos, and G. Nikolakopoulos, "Aerial navigation in obstructed environments with embedded nonlinear model predictive control," in *European Control Conference (ECC)*, 2019, pp. 3556–3563.

[16] M. Vrba, D. Heřt, and M. Saska, "Onboard marker-less detection and localization of non-cooperating drones for their safe interception by an autonomous aerial system," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3402–3409, 2019, ISSN: 2377-3766. DOI: 10.1109/LRA.2019.2927130.

[17] T. Baca, D. Hert, G. Loianno, M. Saska, and V. Kumar, "Model predictive trajectory tracking and collision avoidance for reliable outdoor deployment of unmanned aerial vehicles," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 6753–6760. DOI: 10.1109/IROS.2018.8594266.

[18] A. Sathya, P. Sopasakis, R. Van Parys, A. Themelis, G. Pipeleers, and P. Patrinos, "Embedded nonlinear model predictive control for obstacle avoidance using panoc," in *European Control Conference (ECC)*, 2018, pp. 1523–1528.

[19] R. Yanushevsky, *Modern Missile Guidance.* Sep. 2018, ISBN: 9781351202954. DOI: 10.1201/9781351202954.

[20] S. Lucia, A. Tatulea-Codrean, C. Schoppmeyer, and S. Engell, "Rapid development of modular and sustainable nonlinear model predictive control solutions," *Control Engineering Practice*, vol. 60, 51–62, Mar. 2017. DOI: 10.1016/j.conengprac.2016.12.009.

[21] L. Stella, A. Themelis, P. Sopasakis, and P. Patrinos, "A simple and efficient algorithm for nonlinear model predictive control," in *IEEE Conference on Decision and Control (CDC)*, 2017, pp. 1939–1944.

[22] C. Richter, A. Bry, and N. Roy, "Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments," in *Robotics Research*, Springer, 2016, pp. 649–666.

[23] 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*, vol. 34, no. 7, pp. 969–1002, 2015. DOI: 10.1177/0278364914558129. [Online]. Available: https://doi.org/10.1177/0278364914558129.

[24] M. Burri, H. Oleynikova, M. W. Achtelik, and R. Siegwart, "Real-time visual-inertial mapping, re-localization and planning onboard mavs in unknown environments," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015.

[25] R. Clothier, D. Greer, D. Greer, and A. Mehta, "Risk perception and the public acceptance of drones," *Risk Analysis*, vol. 35, Feb. 2015. DOI: 10.1111/risa.12330.

[26] N. Palumbo, R. Blauwkamp, and J. Lloyd, "Modern homing missile guidance theory and techniques," *Johns Hopkins APL Technical Digest (Applied Physics Laboratory)*, vol. 29, Jan. 2010.

[27] X. Rong Li and V. Jilkov, "Survey of maneuvering target tracking. part v. multiple-model methods," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 4, pp. 1255–1321, 2005. DOI: 10.1109/TAES.2005.1561886.

[28] S. Julier and J. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004. DOI: 10.1109/JPROC.2003.823141.

[29] S. Rusinkiewicz and M. Levoy, "Efficient variants of the icp algorithm," in *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*, 2001, pp. 145–152. DOI: 10.1109/IM.2001.924423.

[30] P. Dorato, C. Abdallah, and V. Cerone, *Linear Quadratic Control: An Introduction.* Krieger Publishing Company, 2000, ISBN: 9781575241562. [Online]. Available: https://books.google.cz/books?id=n1EcnwEACAAJ.

[31] U. S. Shukla and P. R. Mahapatra, “The proportional navigation dilemma-pure or true?” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 26, pp. 382–392, 1990.

[32] D. Kirk, *Optimal Control Theory: An Introduction.* Prentice-Hall, 1976. [Online]. Available: https://books.google.cz/books?id=b1lDwAEACAAJ.

[33] M. Guelman, “Proportional navigation with a maneuvering target,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-8, no. 3, pp. 364–371, 1972. DOI: 10.1109/TAES.1972.309520.

[34] G. A. Hicks and W. H. Ray, “Approximation methods for optimal control synthesis,” *The Canadian Journal of Chemical Engineering*, vol. 49, no. 4, pp. 522–528, 1971. DOI: https://doi.org/10.1002/cjce.5450490416. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cjce.5450490416. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/cjce.5450490416.

[35] R. E. Kalman, “A New Approach to Linear Filtering and Prediction Problems,” *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, Mar. 1960, ISSN: 0021-9223. DOI: 10.1115/1.3662552. eprint: https://asmedigitalcollection.asme.org/fluidsengineering/article-pdf/82/1/35/5518977/35_1.pdf. [Online]. Available: https://doi.org/10.1115/1.3662552.