ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

DEVELOPMENT AND TESTING NOVEL GUIDANCE ALGORITHMS FOR VISUAL DRONE INTERCEPTION

M.Sc. THESIS

Ahmet Talha ÇETİN

Department of Aeronautics and Astronautics

Aeronautics and Astronautics Programme

JUNE 2024

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

DEVELOPMENT AND TESTING NOVEL GUIDANCE ALGORITHMS FOR VISUAL DRONE INTERCEPTION

M.Sc. THESIS

Ahmet Talha ÇETİN
(511211106)

Department of Aeronautics and Astronautics

Aeronautics and Astronautics Programme

Thesis Advisor: Assoc. Prof. Dr. Emre Koyuncu

JUNE 2024

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

GÖRSEL DRON YAKALAMA İÇİN YENİ GÜDÜM ALGORİTMALARININ GELİŞTİRİLMESİ VE TESTİ

YÜKSEK LİSANS TEZİ

Ahmet Talha ÇETİN
(511211106)

Uçak ve Uzay Mühendisliği

Uçak ve Uzay Mühendisliği Programı

Tez Danışmanı: Assoc. Prof. Dr. Emre Koyuncu

HAZİRAN 2024

Ahmet Talha ÇETİN, a M.Sc. student of ITU Graduate School student ID 511211106 successfully defended the thesis entitled "DEVELOPMENT AND TESTING NOVEL GUIDANCE ALGORITHMS FOR VISUAL DRONE INTERCEPTION", which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Assoc. Prof. Dr. Emre Koyuncu** Istanbul Technical University	.....
Jury Members :	**Assist. Prof. Dr. Ramazan YENİÇERİ** Istanbul Technical University	.....
	Dr. Murad ABU-KHALAF Turkish Aerospace Industries	.....

Date of Submission : **24 May 2024**
Date of Defense : **13 June 2024**

To my lovely wife,

FOREWORD

I would like to thank, my thesis advisor Dr. Emre Koyuncu due to providing me full support during my university life and his help during thesis theory and implementation. It would have been very difficult to do this thesis without his vision and the facilities he provided.

I would like to express my gratitude, my precious wife Fulin who has always been by my side in my life since we met. I never forget her support and understanding during long lasting flight tests during this thesis implementation. This work and my life would be very difficult without her support.

I have to thank my mother and father for always being there for me in every decision I have made so far and for all the support they have given me in my life. I also thank my sister Rumeysa for being not only a sister but also one of my best friends.

I am grateful to my mother-in-law and father-in-law for helping and trusting me in everything since the day we met and for seeing me as their own son.

I would like to thank, Dr. Ramazan Yeniçeri for his help during the thesis and flight tests.

And finally, I would like to thank my friends Mehmet Zeki Paşaoğlu and Serkan Şenel, with whom we conducted the flight tests in this thesis. The sometimes tiring but always fun times we spent in the flight test area would have been very difficult without them. I would also like to thank Mustafa Demir for his help in the flight tests of the thesis and for being a big brother to me as I always value his ideas.

June 2024 Ahmet Talha ÇETİN

TABLE OF CONTENTS

ABBREVIATIONS

VIO	**:** Visual Inertial Odometry
MPC	**:** Model Predictive Controller
YOLO	**:** You Look Only Once
GPS	**:** Global Positioning System
UAV	**:** Unmanned Aerial Vehicle
PID	**:** Proportional Integral Derivative
IMU	**:** Inertial Measurement Unit
ROS	**:** Robot Operating System

LIST OF TABLES

Page

LIST OF FIGURES

Page

DEVELOPMENT AND TESTING NOVEL GUIDANCE ALGORITHMS FOR VISUAL DRONE INTERCEPTION

SUMMARY

The growing prevalence of UAVs has unlocked a multitude of opportunities across various fields, ranging from surveillance to delivery services. However, this surge in UAV usage has also brought about an increase in potential threats, including both deliberate malicious actions and accidental incidents. As a result, the necessity for efficient counter-drone systems to address the risks associated with unauthorized drone operations has become critically important. Moreover, the emergence of innovative drone technologies that operate independently of traditional methods like GPS and RF-links further limits the available countermeasures. This development has highlighted the significant challenge of using visual feedback to guide a quadrotor in intercepting rapidly moving targets, emphasizing the urgent need for robust security solutions.

In addition to traditional countermeasures, there is a growing need for non-GPS navigation systems in counter-drone strategies. These systems are essential to ensure that interception and neutralization efforts remain effective even when conventional navigation methods are disrupted or unavailable. This is highly practical, because of the counter drone systems generally have jamming and GPS-spoofing capability. Non-GPS navigation technologies, such as visual odometry and other sensor-based approaches, provide alternative means to maintain accurate tracking and engagement of rogue drones, ensuring comprehensive protection against increasingly sophisticated UAV threats.

This study tackles the challenge of guiding a quadrotor to intercept fast-moving targets visual and radar feedback by Visual Inertial Odometry or GPS respectively. Proposed system, designed as a counter UAV solution, utilizes onboard camera and radar information of the aerial threat. Target interception process has been divided into two parts. One is pre-terminal phase guidance where target information comes from radar feedback. Unless the target has not been seen at the camera, interceptor guided from radar feedback. Once the target is detected by the camera, the quadrotor switches to terminal phase guidance which is guiding counter drone to aerial target by visual feedback.

For pre-terminal guidance, two different algorithms were developed. A Model Predictive Control based guidance algorithm has been designed for pre-terminal guidance. For pre-terminal guidance, parallel interceptions (toward the head or back) provide robustness to inevitable visual processing latency in terminal phase compared to lateral engagements. By addressing these issues, the proposed methodology mainly utilizes Model Predictive Control (MPC) method with added terminal constraints

to satisfy engagement at the desired angle. While formulating the MPC, the objective function in the MPC is modified to reduce the interceptor's requirement for maneuvering at the end of the trajectory. MPC prediction horizon is calculated by considering vehicle limits to satisfy the feasibility of the problem.

Another method is we use Bezier Splines to guide the quadrotor. Since quadrotors has limited onboard computational power, MPC might not be practical for some cases. By ensuring continuity with Bezier Splines, the system determines the optimal interception direction (towards the head or tail) and calculates the time-to-go, considering in the target's position and velocity along with the interceptor's kinematic constraints. This method specifically addresses latency issues in target detection, crucial for intercepting high-speed targets effectively. Moreover, the delays introduced by target detection and localization pose significant challenges, particularly for small quadrotors with limited computational power. The proposed approach aims to achieve parallel engagement with the target's velocity vector, whether from the front or rear, thus minimizing delays and overcoming visual tracking difficulties before target is detected by onboard camera. This strategy reduces lateral acceleration within the image frame during the final stages of interception, resulting in smaller miss distances. This outcome is consistent with established guidance literature, which recognizes the advantages of reduced acceleration at the end of the interception path.

When the target is detected by camera using object detection algorithms, terminal phase guidance is initiated. For detecting aerial threats, the object detection algorithm You Only Look Once (YOLO) is used. Maintaining detection and tracking by camera can be interrupted due to limitations such as motion blur, noise in the image and getting out of the camera field of view. When detection is interrupted, Kalman Filter is used for prediction of the target. For image based guidance we utilized proportional guidance with some modifications. For this work we assume that no stabilizing mechanism that preserve orientation of the camera is used. Since no stabilizing mechanism is used for the camera, we formulized propotional guidance rules in roll and pitch stabilized frame in order not to being affected from camera orientation.

We employed two distinct navigation methods: GPS-based navigation and Visual Inertial Navigation for navigating towards to target at the pre-terminal phase. The well-established open-source ArduPilot platform was utilized for GPS-based navigation, while VINS-Mono was implemented for Visual Inertial Navigation. As for controllers, due to the differing frequencies of estimated odometry data from these systems, different position controllers were employed for each navigation solution. The ArduPilot built-in controller was utilized for GPS-based navigation, whereas a custom controller was designed and flight-tested for handling VIO feedback. The aforementioned navigation and control methods allowed us to compare and evaluate their performance in different scenarios. The GPS-based navigation provided a reliable and accurate solution in environments with clear GPS signals, while the Visual Inertial Navigation offered a robust alternative in situations where GPS signals were weak or unavailable. The custom controller designed for VIO feedback was optimized to handle the unique characteristics of visual inertial data, ensuring smooth and precise control of the quadrotor. Through this approach, we were able to develop a comprehensive navigation system that can adapt to various operational conditions,

enhancing the overall reliability and effectiveness of the quadrotor's guidance and control.

Finally, real world flight tests were conducted for assessing overall performance of the system. To evaluate the performance of the GPS-based and VIO-base navigation algorithms, interception flights tests were conducted separately and the performance of the guidance algorithm was assessed accordingly. In real-world flight tests, we tested the use of Bezier splines in the pre-terminal along and image-based visual servoing for the terminal phase. In doing so, we examined the use of GPS-based and VIO based navigation algorithms. Results show performance of the proposed methodology.

GÖRSEL DRON YAKALAMA İÇİN YENİ GÜDÜM ALGORİTMALARININ GELİŞTİRİLMESİ VE TESTİ

ÖZET

İnsansız Hava Araçlarının (İHA'ların) artan yaygınlığı, gözetimden teslimat hizmetlerine kadar çeşitli alanlarda birçok fırsatın kapısını aralamıştır. Ancak, İHA kullanımındaki bu artış, kasıtlı kötü niyetli eylemler ve kazara meydana gelen olaylar da dahil olmak üzere potansiyel tehditlerin de artmasına yol açmıştır. Sonuç olarak, izinsiz drone operasyonlarına karşı etkili karşı-drone sistemlerinin gerekliliği kritik bir önem kazanmıştır. Ayrıca, GPS ve RF bağlantıları gibi geleneksel yöntemlerden bağımsız olarak çalışan yenilikçi drone teknolojilerinin ortaya çıkışı, mevcut karşı önlemleri daha da sınırlamaktadır. Bu gelişme, bir quadrotorun hızlı hareket eden hedefleri yakalamasını görsel geri bildirimle yönlendirmenin önemli bir zorluk olduğunu vurgulamakta ve sağlam güvenlik çözümlerine olan acil ihtiyacı öne çıkarmaktadır.

Geleneksel karşı önlemlere ek olarak, karşı-drone stratejilerinde GPS dışı navigasyon sistemlerine duyulan ihtiyaç artmaktadır. Bu sistemler, geleneksel navigasyon yöntemlerinin kesintiye uğradığı veya kullanılamadığı durumlarda durdurma ve etkisiz hale getirme çabalarının etkili kalmasını sağlamak için gereklidir. Bu oldukça pratiktir, çünkü karşı-drone sistemleri genellikle karıştırma ve GPS aldatma yeteneklerine sahiptir. GPS dışı navigasyon teknolojileri, görsel odometri ve diğer sensör tabanlı yaklaşımlar gibi, istenmeyen drone'ların doğru bir şekilde takip edilmesini ve etkisiz hale getirilmesini sağlamak için alternatif yollar sunar, böylece giderek daha sofistike hale gelen İHA tehditlerine karşı kapsamlı bir koruma sağlar.

Bu çalışma, hızlı hareket eden hedefleri görsel ve radar geri bildirimiyle yakalamak için bir quadrotoru yönlendirme zorluğunu ele almaktadır. Önerilen sistem, karşı İHA çözümü olarak tasarlanmış olup, hava tehdidinin yerleşik kamera ve radar bilgilerini kullanmaktadır. Hedef yakalama süreci iki aşamaya ayrılmıştır. Birincisi, hedef bilgilerinin radar geri bildiriminden geldiği ön-terminal aşaması rehberliğidir. Hedef kamera tarafından görülene kadar, önleyici radar geri bildiriminden yönlendirilir. Hedef kamera tarafından tespit edildiğinde, quadrotor görsel geri bildirimle hava hedefine yönlendirilen terminal aşaması rehberliğine geçer.

Ön-terminal rehberliği için iki farklı algoritma geliştirilmiştir. Model Öngörülü Kontrol (MPC) tabanlı bir rehberlik algoritması ön-terminal rehberliği için tasarlanmıştır. Ön-terminal rehberliğinde, baş veya arka yöne doğru paralel kesişmeler, terminal aşamasında kaçınılmaz görsel işleme gecikmesine karşı yanal angajmanlara göre daha fazla dayanıklılık sağlar. Bu sorunları ele alarak, önerilen yöntem esas olarak, istenen açıyla angajmanı sağlamak için ek terminal kısıtlamaları olan Model Öngörülü Kontrol

(MPC) yöntemini kullanır. MPC formüle edilirken, MPC'nin hedef fonksiyonu, yolun sonunda önleyicinin manevra gereksinimini azaltmak için değiştirilir. MPC tahmin ufku, sorunun uygulanabilirliğini sağlamak için araç limitleri dikkate alınarak hesaplanır.

Bir diğer yöntem ise quadrotoru yönlendirmek için Bezier Eğrileri kullanmamızdır. Quadrotorların sınırlı yerleşik hesaplama gücü olduğu için, bazı durumlarda MPC pratik olmayabilir. Bezier Eğrileri ile sürekliliği sağlayarak, sistem optimal kesişme yönünü (baş veya kuyruk) belirler ve hedefin konumu ve hızı ile önleyicinin kinematik kısıtlamalarını dikkate alarak geçiş süresini hesaplar. Bu yöntem, hedef tespitindeki gecikme sorunlarını özellikle yüksek hızlı hedefleri etkili bir şekilde yakalamak için ele alır. Ayrıca, hedef tespiti ve yer belirleme sırasında oluşan gecikmeler, özellikle sınırlı hesaplama gücüne sahip küçük quadrotorlar için önemli zorluklar oluşturur. Önerilen yaklaşım, hedefin hız vektörüne paralel angajmanı baş veya arka tarafından sağlayarak, gecikmeleri en aza indirir ve hedef yerleşik kamera tarafından tespit edilmeden önce görsel izleme zorluklarını aşar. Bu strateji, yakalamanın son aşamalarında görüntü çerçevesinde yanal ivmeyi azaltarak daha küçük kaçırma mesafeleri elde edilmesini sağlar. Bu sonuç, yakalama yolunun sonunda ivmenin azaltılmasının avantajlarını tanıyan yerleşik rehberlik literatürü ile tutarlıdır.

Hedef, nesne tespit algoritmaları kullanılarak kamera tarafından tespit edildiğinde, terminal aşaması rehberliği başlatılır. Hava tehditlerini tespit etmek için, "You Only Look Once" (YOLO) nesne tespit algoritması kullanılmıştır. Kamera ile tespit ve takibi sağlamak, hareket bulanıklığı, görüntüdeki gürültü ve kameranın görüş alanının dışına çıkması gibi sınırlamalar nedeniyle kesintiye uğrayabilir. Tespit kesintiye uğradığında, hedefin tahmini için Kalman Filtresi kullanılır. Görüntü tabanlı rehberlik için bazı modifikasyonlarla orantılı rehberlik kullanılmıştır. Bu çalışmada, kameranın yönelimini koruyan bir dengeleme mekanizmasının kullanılmadığı varsayılmıştır. Kamera için herhangi bir dengeleme mekanizması kullanılmadığından, orantılı rehberlik kurallarını kamera yöneliminden etkilenmemek için roll ve pitch stabilize edilmiş çerçevede formüle ettik.

Hedefe doğru gezinmek için iki farklı navigasyon yöntemi kullandık: ön-terminal aşamasında hedefe doğru GPS tabanlı navigasyon ve Görsel Ataletli Navigasyon. GPS tabanlı navigasyon için iyi bilinen açık kaynak ArduPilot platformu kullanılırken, Görsel Ataletli Navigasyon için VINS-Mono uygulanmıştır. Kontrolörler açısından, bu sistemlerden gelen tahmini odometri verilerinin farklı frekansları nedeniyle, her bir navigasyon çözümü için farklı konum kontrolörleri kullanılmıştır. GPS tabanlı navigasyon için ArduPilot yerleşik kontrolör kullanılırken, VIO geri bildirimi ile başa çıkmak için özel bir kontrolör tasarlanmış ve uçuş testleri yapılmıştır.

Yukarıda belirtilen navigasyon ve kontrol yöntemleri, performanslarını farklı senaryolarda karşılaştırmamıza ve değerlendirmemize olanak sağladı. GPS tabanlı navigasyon, net GPS sinyallerine sahip ortamlarda güvenilir ve doğru bir çözüm sağlarken, Görsel Ataletli Navigasyon, GPS sinyallerinin zayıf veya mevcut olmadığı durumlarda sağlam bir alternatif sunmuştur. VIO geri bildirimi için tasarlanan özel kontrolör, görsel atalet verilerinin benzersiz özelliklerini işleyerek quadrotorun düzgün ve hassas kontrolünü sağladı. Bu yaklaşım sayesinde, çeşitli operasyonel

koşullara uyum sağlayabilen kapsamlı bir navigasyon sistemi geliştirebildik ve bu da quadrotorun rehberlik ve kontrolünün genel güvenilirliğini ve etkinliğini artırdı.

Son olarak, sistemin genel performansını değerlendirmek için gerçek dünya uçuş testleri gerçekleştirilmiştir. GPS tabanlı ve VIO tabanlı navigasyon algoritmalarının performansını değerlendirmek için kesişme uçuş testleri ayrı ayrı yapılmış ve rehberlik algoritmasının performansı buna göre değerlendirilmiştir. Gerçek dünya uçuş testlerinde, ön-terminal ve terminal aşaması için Bezier eğrileri ve görüntü tabanlı görsel servo kullanımı test edilmiştir. Bunu yaparken, GPS tabanlı ve VIO tabanlı navigasyon algoritmalarının kullanımı incelenmiştir. Sonuçlar, önerilen metodolojinin performansını göstermektedir.

1. INTRODUCTION

The rapid growth of unmanned aerial vehicles (UAVs) has opened up a wide range of applications across diverse fields, from surveillance to delivery services. However, this surge in UAV usage has also led to an increase in potential threats, including both deliberate malicious actions and accidental incidents. As a result, the need for efficient counter-drone systems to address the risks associated with unauthorized drone operations has become critically important. Moreover, the emergence of innovative drone technologies that operate independent of traditional methods like GPS and RF-links further limits the available countermeasures, emphasizing the urgent need for robust and urgent security solutions. Recent developments in computer vision brings significant technical opportunities, and associated challenges, of using visual feedback to guide a quadrotor in intercepting rapidly moving targets. This study proposes a comprehensive approach to guide a quadrotor using visual and radar feedback for intercepting fast-moving aerial targets, designed as a counter-UAV solution. The proposed system utilizes onboard camera and radar information to track and engage aerial threats, employing two distinct guidance algorithms for pre-terminal and terminal phases of the interception process. The effectiveness of the proposed methodology is evaluated through simulation tests and real-world flight tests, comparing GPS-based navigation and Visual Inertial Navigation (VIN) approaches to ensure robust and reliable counter-drone capabilities in the face of increasingly sophisticated UAV threats.

1.1 Purpose of Thesis

The primary purpose of this thesis is to develop and evaluate a robust system for guiding a quadrotor to intercept fast-moving aerial targets using visual and radar feedback. The proposed system is designed as a counter-UAV solution to address

the growing threat of unauthorized drone operations and the limitations of existing countermeasures.

The specific objectives of this research are:

1. To develop two distinct guidance algorithms for the pre-terminal and terminal phases of the interception process: Since most of the guidance algorithms are developed for missiles, interceptors are assumed to have initial velocity and can be changed slightly in angles. Assumptions such as having initial velocity and slight change in angles, makes guidance problem challenging for quadrotor.

2. To implement a visual guidance algorithm for the terminal phase. First of all, a visual detection algorithm to track aerial threats is needed. Then, a guidance algorithm that works with Line-of-Sight camera feedbacks and associated latency due to hardware limitation of the quadrotor is needed.

3. A system that works with/out GPS when GPS is not available. In this regards, we compare and evaluate the performance of two navigation solutions: a. GPS-based navigation b. Visual Inertial Odometry

4. To design and test custom position controllers for each navigation solution to ensure optimal performance and seamless integration with the guidance algorithms.

5. To conduct comprehensive simulation tests and real-world flight tests to assess the overall performance of the system, including its ability to effectively intercept fast-moving aerial targets using both GPS-based navigation and Visual Inertial Navigation.

By achieving these objectives, this thesis aims to contribute to the development of advanced counter-drone systems capable of addressing the growing threat of unauthorized drone operations and the limitations of existing countermeasures. The proposed system's ability to guide a quadrotor using visual and radar feedback, while employing robust guidance algorithms and navigation solutions, is expected to enhance the effectiveness of counter-UAV efforts in various applications, such as critical infrastructure protection, public safety, and military operations.

1.2 Literature Review

The UAV impact guidance problem has been extensively investigated in the literature from various perspectives. The authors in [2], adapt Proportional Navigation methods to quadrotor systems by incorporating thrust vector control. Finding an optimal P gain that works effectively for all flight scenarios can be challenging. Gain scheduling can be made but does not satisfy impact angle requirements. In [3], a time optimal interception trajectory is calcualted using indirect optimal control techniques. In [4], the authors employed model predictive control (MPC) to address the quadrotor state interception problem. Model Predictive Control is a control technique that involves solving optimization problems at each control step. While MPC offers advantages in terms of its ability to handle complex dynamics and constraints, it typically requires higher computational resources compared to simpler control methods. In [5], authors utilized image-based visual servoing (IBVS) in combination with Kalman Filtering. In [6], authors applied IBVS with Reinforcement Learning to solve the target-tracking problem. Authors in [7], propose a visual feedback approach to track an object. In [8], authors introduce position-based visual servoing (PBVS) based on determining the interception point using prior knowledge of the target trajectory's shape. In work conducted by Strydom et al. (2015) [9], a stereo camera system is employed to implement Position-Based Visual Servoing (PBVS) for unmanned aerial vehicles (UAVs). The proposed method feeds target position reference to the position controller.In [10], authors employed model predictive control methods with the target's position as a reference; however, their method does not incorporate impact angle constraints.

In this work, visual inertial odometry is used as an alternative navigation solution. In the field of visual-inertial odometry and simultaneous localization and mapping (SLAM), numerous approaches have been proposed and developed over the years. One of the foundational methods is the Multi-State Constraint Kalman Filter (MSCKF), an extended Kalman Filter (EKF) initially introduced in [11], which has since seen many variations and improvements. The MSCKF leverages the advantages of the

Kalman Filter in handling nonlinear systems and provides a robust framework for state estimation. In [12], the Optimized Keyframe-based Visual-Inertial SLAM (OKVIS), is described. OKVIS employs a keyframe-based and optimization-based sliding window estimator that minimizes landmark reprojection errors to achieve accurate state estimation. This method combines the strengths of keyframe-based techniques with optimization to handle visual and inertial data effectively. The ROVIO (Robust Visual Inertial Odometry) algorithm, presented in [13], extends the Kalman Filter framework by tracking both 3D landmarks and image patch features. This dual-tracking capability allows ROVIO to maintain robust and accurate state estimation even in challenging environments with dynamic changes and varying lighting conditions. VINS-Mono, detailed in [1], takes a different approach by utilizing a nonlinear optimization-based sliding window estimator. It incorporates pre-integrated IMU factors to refine the state estimation process, providing high accuracy in visual-inertial navigation tasks. This method's reliance on nonlinear optimization helps in effectively handling the complex relationships between visual and inertial measurements. SVO (Semi-direct Visual Odometry) combined with the Multi-Sensor Fusion (MSF) framework, as proposed in [14] and [15], presents a loosely-coupled configuration where a visual odometry pose estimator is fused with an extended Kalman Filter. This fusion, as discussed in [16], allows the system to integrate visual pose estimates with inertial sensor data, resulting in improved state estimation accuracy and robustness.

Lastly, SVO combined with GTSAM (Georgia Tech Smoothing And Mapping), as highlighted in [8], offers a lightweight visual odometry frontend paired with a full-smoothing backend provided by the incremental smoothing and mapping algorithm iSAM2 [9]. This combination leverages the efficiency of SVO for fast visual odometry and the robustness of GTSAM for comprehensive state estimation and mapping.

2. QUADROTOR INTERCEPTION GUIDANCE

2.1 Aerial Target Detection

2.1.1 Object Detection Algorithm

The YOLO algorithm is a widely recognized method for object detection that utilizes a single convolutional neural network (CNN). It works by partitioning the input image into a grid and employing the CNN across the entire image to predict both the class and position of objects within each grid cell. The algorithm then consolidates the predictions from all the grid cells to produce the final output. YOLO is renowned for its speed and precision, making it a favored choice for real-time object detection tasks in applications like autonomous vehicles and surveillance systems.

Target tracking is conducted by executing a detection algorithm on consecutive frames within the video stream captured by the camera. Thus, the performance of target tracking is contingent upon the efficacy of the object detection algorithm. Throughout the project, the following performance metrics, which align with the literature, are monitored for object detection performance:

Precision: This parameter measures the accuracy of a detection if one is made, indicating how much the detection coincides with a correct result. For example, a precision rate of 90 % means that if a "drone" object is detected in a video frame, there is a 90 % likelihood that the detected object is indeed a "drone". However, this metric does not provide information about whether all "drone" objects have been detected.

Recall (Sensitivity): This parameter indicates whether all target objects in the frames processed by the detection algorithm were detected, regardless of the precision of the detection. For instance, a recall rate of 90 % signifies that among 100 frames each containing a "drone" object, the algorithm correctly identified the "drone" in 90 of these frames.

2.1.2 Kalman Filter for Image

When detection is interrupted Kalman Filter is used for predicting the target on image frame.

$$\mathbf{x}_{k|k-1} = \mathbf{F}_k \mathbf{x}_{k-1|k-1} + \mathbf{B}_k \mathbf{u}_k \tag{2.1}$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T + \mathbf{Q}_k \tag{2.2}$$

In the update step, the measurement residual $\mathbf{y}_k$ is computed using the observation vector $\mathbf{z}_k$ and the observation model $\mathbf{H}_k$, as shown in Equation 2.3. The residual covariance $\mathbf{S}_k$ is calculated next, as per Equation 2.4. The Kalman gain $\mathbf{K}_k$ is then derived using Equation 2.5, which is essential for updating the state estimate $\mathbf{x}_{k|k}$ and the error covariance matrix $\mathbf{P}_{k|k}$, as shown in Equations 2.6 and 2.7, respectively.

$$\mathbf{y}_k = \mathbf{z}_k - \mathbf{H}_k \mathbf{x}_{k|k-1} \tag{2.3}$$

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k \tag{2.4}$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T \mathbf{S}_k^{-1} \tag{2.5}$$

$$\mathbf{x}_{k|k} = \mathbf{x}_{k|k-1} + \mathbf{K}_k \mathbf{y}_k \tag{2.6}$$

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1} \tag{2.7}$$

Where $\mathbf{x}_k$ is the state vector at time k, $\mathbf{P}_k$ is the error covariance matrix at time k, $\mathbf{F}_k$ is the state transition model, $\mathbf{Q}_k$ is the process noise covariance matrix, $\mathbf{H}_k$ is the observation model, $\mathbf{z}_k$ is the observation vector at time k, $\mathbf{R}_k$ is the measurement noise covariance matrix, $\mathbf{y}_k$ is the innovation or measurement residual, $\mathbf{S}_k$ is the innovation covariance, $\mathbf{K}_k$ is the Kalman gain, and $\mathbf{I}$ is the identity matrix.

States for the Kalman Filter are horizontal and vertical pixel positions and velocities, respectively. The observation vector consists of pixel positions from the object detection algorithm. When detection is started after interruption more than a specified time period, filter is being reset.

2.2 Pre-Terminal Guidance with Model Predictive Control

The methodology exploits a predictive guidance strategy that depends on model predictive control methods to solve the impact guidance problem.

The overall method can be explained as follows: first average velocities along the trajectory are calculated, after that interception direction is selected with the strategy explained in the Section 2.2.2. Thereafter, according to interception direction, time-to-go values are calculated and problem feasibility is checked with minimum time-to-go values. With calculated time-to-go values and problem states, MPC problem is solved and first control input is applied to the system. Summarized methodology is given in the Figure 2.1

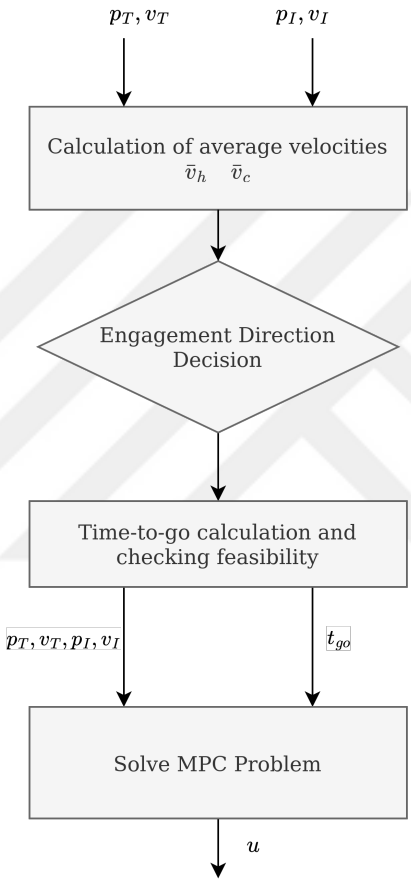

Figure 2.1 : Flowchart of the Proposed Interception Process

2.2.1 MPC formulation

Let $\boldsymbol{x}_I = [\boldsymbol{p}_I, \boldsymbol{v}_I]$, $\boldsymbol{x}_T = [\boldsymbol{p}_T, \boldsymbol{v}_T] \in R^6$ denotes the states of the interceptor and target UAVs. $\boldsymbol{p}_I$, $\boldsymbol{p}_T$ are position vectors in the global cartesian coordinates for interceptor and target UAVs, respectively, and $\boldsymbol{v}_I$, $\boldsymbol{v}_T$ are velocity vectors in cartesian coordinates. Control inputs are acceleration commands in global cartesian coordinates $\boldsymbol{u} = \boldsymbol{a}_I \in R^3$. By concatenating interceptor and target state variables, the augmented state $\boldsymbol{x}$ is defined

as in 2.8.

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_I \\ \boldsymbol{x}_T \end{bmatrix} \tag{2.8}$$

Kinematic equations can be written as follows:

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\boldsymbol{p}}_I \\ \dot{\boldsymbol{v}}_I \\ \dot{\boldsymbol{p}}_T \\ \dot{\boldsymbol{v}}_T \end{bmatrix} = \begin{bmatrix} \boldsymbol{v}_I \\ \boldsymbol{a}_I \\ \boldsymbol{v}_T \\ \boldsymbol{0} \end{bmatrix} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \tag{2.9}$$

Note that, for the optimization, it is assumed that the target preserves the velocity vector until engagement. Since obtaining target acceleration information is not practical for most applications, it has not been included in the optimization problem. As target velocity will be updated every time the NMPC problem is solved, the proposed strategy can be used with accelerating targets. System kinematics discretized into N steps over time horizon t_{go} of size $dt = t_{go}/N$ using Runge-Kutta 4^{th} Order. Resulting discrete system kinematics are denoted as $\boldsymbol{f}_d$, and discrete state and control variables are denoted as $\boldsymbol{x}_k$ and $\boldsymbol{u}_k$, respectively.

The problem aims to find acceleration commands to satisfy collision at a specified impact angle, and time-to-go t_{go}. Desired engagement is parallel to the target velocity vector. To ensure that, terminal constraint $\boldsymbol{v}_I(t_{go}) \times \boldsymbol{v}_T(t_{go}) = \boldsymbol{0} \in R^3$ is added. Optimization problem formulation made as follows:

$$\begin{aligned} \underset{u(\cdot)}{\text{minimize}} \quad & \textstyle\sum_{k=0}^{N-1} \boldsymbol{u}_k^\top R(k) \boldsymbol{u}_k \\ \text{subject to} \quad & \boldsymbol{x}_{k+1} = \boldsymbol{f}_d\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right) \\ & \boldsymbol{x}_k(0) = \boldsymbol{x}_{\text{init}} \\ & \boldsymbol{p}_I\left(t_{go}\right) = \boldsymbol{p}_T\left(t_{go}\right) \\ & \boldsymbol{v}_I\left(t_{go}\right) \times \boldsymbol{v}_T\left(t_{go}\right) = \boldsymbol{0} \in R^3 \\ & \boldsymbol{u}_{\min} \leq \boldsymbol{u_k} \leq \boldsymbol{u}_{\max} \end{aligned} \tag{2.10}$$

Where $R(k)$ is the input weights matrix, a function of timestep k that penalizes subsequent acceleration inputs more to ensure the interceptor requires less maneuver at the end of the trajectory. Input weights matrix determined as:

$$R(k) = \text{diag}\left((k+1)^3, (k+1)^3, (k+1)^3\right) \tag{2.11}$$

This nonlinear optimization problem is solved at every sampling time with calculated t_{go}, and the first control input is given to the system. Implementation of the MPC made with CasADi ([17]) and problem solved with interior-point methods ([18]).

2.2.2 Interception Direction Decision

Engagement is desired to be parallel to target velocity; thereby, two solutions are possible for this problem: a) Head-to-head engagement and b) chasing engagement (engagement from back). Head-to-head engagement is where target and interceptor velocities are opposite when engagement occurs. In chasing engagement, the target and interceptor velocities are in the same direction. The decision is made by considering the target position, velocity, and interceptor kinematic limits. The main logic behind the decision is that if target approaches to the interceptor and interceptor kinematic limits could satisfy the head-to-head engagement, head-to-head engagement is favored. Chasing engagement is preferred for receding targets and targets that could not be engaged with head-to-head engagement.

Formulation can be made as follows, if relative position vector is defined as $\boldsymbol{r} = \boldsymbol{p}_T - \boldsymbol{p}_I$, scalar projection of $\boldsymbol{r}$ onto $\boldsymbol{v}_T$ can be defined as $|\text{proj}_{\boldsymbol{v}_T} \boldsymbol{r}|$. When target approaches i.e. $\frac{\boldsymbol{r}}{|\boldsymbol{r}|} \cdot \frac{\boldsymbol{v}_T}{|\boldsymbol{v}_T|} < 0$, it is looked that if target can be hit by head-to-head by specified average velocity $\bar{v}_h$ i.e. conservative condition in 2.34 holds head-to-head engagement will be preffered.

$$\frac{|\text{proj}_{\boldsymbol{v}_T} \boldsymbol{r}| + \sqrt{|\boldsymbol{r}|^2 - |\text{proj}_{\boldsymbol{v}_T} \boldsymbol{r}|^2}}{\bar{v}_h + |\boldsymbol{v}_T|} < \frac{|\text{proj}_{\boldsymbol{v}_T} \boldsymbol{r}|}{|\boldsymbol{v}_T|} \tag{2.12}$$

When this condition for head-to-head engagement is not satisfied or when target recedes $\frac{\boldsymbol{r}}{|\boldsymbol{r}|} \cdot \frac{\boldsymbol{v}_T}{|\boldsymbol{v}_T|} \geq 0$ chasing engagement will be preferred.

2.2.3 Time-to-Go Estimation

To ensure the feasibility of the optimization problem formalized above, time-to-go must be selected considering the target vehicle's kinematics and the limits of the interceptor vehicle. For chasing engagement time-to-go is calculated as in 2.35 and head-to-head engagement time-to-go is calculated as in 2.36.

$$t_{go}^c = \frac{|\boldsymbol{p}_T - \boldsymbol{p}_I|}{|\bar{v}_c \boldsymbol{u}_T - \boldsymbol{v}_T|} \tag{2.13}$$

$$t_{go}^h = \frac{|\boldsymbol{p}_T - \boldsymbol{p}_I|}{|\bar{v}_h \boldsymbol{u}_T + \boldsymbol{v}_T|} \tag{2.14}$$

Where $\bar{v}_c$ and $\bar{v}_h$ is the average velocity along the interception trajectory for the chasing and head-to-head engagement, respectively, and $\boldsymbol{u}_T$ is the unit target velocity vector. $\bar{v}_c$ and $\bar{v}_h$ are specified as in the 2.37 with tunable parameters N_c and N_h.

$$\bar{v}_c = N_c|\boldsymbol{v}_T|, \quad \bar{v}_h = N_h|\boldsymbol{v}_T| \tag{2.15}$$

N_c and N_h are only tunable parameters of this method and they specifies proportion of the interceptor speed to the target speed. Calculations at 2.35 and 2.36 can result

Algorithm 1 Minimum Time-to-Go Estimation

1: **input:** $\boldsymbol{v}_T, \boldsymbol{v}_I, \bar{v}_c, \bar{v}_h, v_{max}, a_{max}$
2: **output:** $t_{go,min}$
3: $t_{go,min} \leftarrow 0$
4: **if** Chasing Engagement **then**
5: $\quad \boldsymbol{v}_{avg} \leftarrow \bar{v}_c \frac{\boldsymbol{v}_T}{|\boldsymbol{v}_T|}$
6: **else if** Head-to-Head Engagement **then**
7: $\quad \boldsymbol{v}_{avg} \leftarrow -\bar{v}_h \frac{\boldsymbol{v}_T}{|\boldsymbol{v}_T|}$
8: **end if**
9: **for** $i \leftarrow 1$ to 3 **do**
10: $\quad$ **if** $|2(\boldsymbol{v}_{avg})_i - (\boldsymbol{v}_I)_i| \leq v_{max}$ **then**
11: $\quad\quad t_{go,i} \leftarrow \left|\frac{2((\boldsymbol{v}_{avg})_i - (\boldsymbol{v}_I)_i)}{a_{max}}\right|$
12: $\quad$ **else if** $|2(\boldsymbol{v}_{avg})_i - (\boldsymbol{v}_I)_i| > v_{max}$ **then**
13: $\quad\quad t_1 \leftarrow \left|\frac{v_{max} - (\boldsymbol{v}_I)_i}{a_{max}}\right|$
14: $\quad\quad t_{go,i} \leftarrow t_1 + \frac{\left(2(\boldsymbol{v}_{avg})_i - v_{max} - (\boldsymbol{v}_I)_i\right)t_1}{2\left(v_{max} - (\boldsymbol{v}_{avg})_i\right)}$
15: $\quad$ **end if**
16: $\quad t_{go,min} \leftarrow \max\{t_{go,i}, t_{go,min}\}$
17: **return** $t_{go,min}$

in infeasible time-to-go estimation when the interceptor cannot achieve a specified average velocity along the trajectory. To overcome this problem, minimum time-to-go $t_{go,min}$ estimation is needed considering interceptor acceleration and maximum velocity limits. This time-to-go can be calculated using interceptor vehicle limits as given in the Algorithm 2. In the algorithm $t_{go,i}$ denotes the time-to-go value of i^{th} axis and $(.)_i$ denotes the i^{th} element of velocity vector. Indices represent elements of vector in the xyz order. In the algorithm, the minimum time required to achieve a specified average velocity is calculated for each axis within the range of lines 9 to 16. During these calculations, two conditions are considered. The first condition occurs at lines 12 to 14, where the vehicle needs to reach its maximum velocity in order to satisfy

the average velocity requirement. The second condition arises at lines 10 to 11, where the vehicle does not need to attain its maximum velocity to meet the average velocity requirement. Consequently, the minimum time-to-go will be determined by the axis that exhibits the lowest time-to-go value among all the axes considered. Finally, for chasing engagement time-to-go is selected as in the 2.38.

$$t_{go} = \max\{t_{go}^{c}, t_{go,min}\} \tag{2.16}$$

For head-to-head engagement time-to-go is selected as in the 2.39.

$$t_{go} = \max\{t_{go}^{h}, t_{go,min}\} \tag{2.17}$$

2.3 Pre-Terminal Guidance with Bezier Splines

The general procedure can be described as follows: the average velocities are initially computed along the trajectory. Subsequently, the interception direction is determined based on the strategy outlined in Section 2.3.2. Following that, the time-to-go values are calculated based on the chosen interception direction, and the feasibility of the problem is assessed by comparing the time-to-go values with the minimum required values. Then, using specified average velocities and calculated time-to-go values, trajectories are calculated every-time step to ensure continuity using Bezier Splines, then sampled inputs will be given to the position controllers. The described methodology is given in Figure 2.2

2.3.1 Trajectory Representation with Bezier Splines

Trajectories are represented with 6^{th} degree Bezier Splines. Representation of the trajectory is given as

$$p = C_p^T N \tag{2.18}$$

Where C_p is the matrix of control points of the Bezier curve as given in the equation 2.19.

$$C_p = \begin{bmatrix} C_{p_1} & C_{p_2} & C_{p_3} & C_{p_4} & C_{p_5} & C_{p_6} & C_{p_7} \end{bmatrix}^T \tag{2.19}$$

N discrete basis functions in matrix format are given in the equation 2.23. R is the Bezier curve basis function. For specified time-to-go t_{go}, $\tau_i = t_i/t_{go}$ is be used for

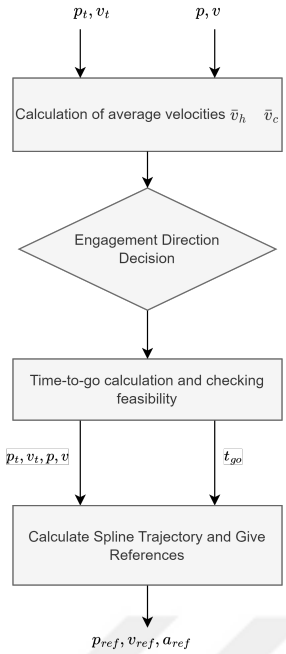

Figure 2.2 : Pursuit Guidance Strategy

parameterize trajectory at time t_i.

$$N = \begin{bmatrix} R(0) & R(\tau_1) & \dots & R(\tau_i) & \dots & R(\tau_N) \end{bmatrix} \tag{2.20}$$

Velocity v and acceleration a profiles can be obtained by the derivative of the Bezier curves.

$$v = \frac{1}{t_{go}} C_p^T \dot{N}, \quad a = \frac{1}{t_{go}^2} C_p^T \ddot{N} \tag{2.21}$$

Where $\dot{N}$ and $\ddot{N}$ are the first and second derivatives of discrete basis functions in a matrix format which can be written as

$$\dot{N} = \begin{bmatrix} \dot{R}(0) & \dot{R}(\tau_1) & \dots & \dot{R}(\tau_i) & \dots & \dot{R}(\tau_N) \end{bmatrix} \tag{2.22}$$

$$\ddot{N} = \begin{bmatrix} \ddot{R}(0) & \ddot{R}(\tau_1) & \dots & \ddot{R}(\tau_i) & \dots & \ddot{R}(\tau_N) \end{bmatrix} \tag{2.23}$$

Basis functions of Bezier curves and their derivatives are given in the equation 2.26.

$$R(\tau) = \begin{bmatrix} (1-\tau)^6 \\ 6\tau(1-\tau)^5 \\ 15\tau^2(1-\tau)^4 \\ 20\tau^3(1-\tau)^3 \\ 15\tau^4(1-\tau)^2 \\ 6\tau^5(1-\tau) \\ \tau^6 \end{bmatrix} \tag{2.24}$$

$$\dot{R}(\tau) = \begin{bmatrix} -6(1-\tau)^5 \\ -6(6\tau-1)(1-\tau)^4 \\ -30\tau(3\tau-1)(1-\tau)^3 \\ -60\tau^2(2\tau-1)(1-\tau)^2 \\ 30\tau^3(3\tau^2-5\tau+2) \\ 6(5-6\tau)\tau^4 \\ 6\tau^5 \end{bmatrix} \tag{2.25}$$

$$\ddot{R}(\tau) = \begin{bmatrix} 30(1-\tau)^4 \\ -60(\tau-1)^3(3\tau-1) \\ 30(15\tau^2-10\tau+1)(1-\tau)^2 \\ -120\tau(5\tau^3-10\tau^2+6\tau-1) \\ 30\tau^2(15\tau^2-20\tau+6) \\ -60\tau^3(3\tau-2) \\ 30\tau^4 \end{bmatrix} \tag{2.26}$$

For given initial position, velocity, and acceleration values, control points are calculated as in Equation 2.29below.

$$C_{p_1} = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} \tag{2.27}$$

$$C_{p_2} = \frac{t_{go}}{6} \begin{bmatrix} \dot{x}_0 \\ \dot{y}_0 \\ \dot{z}_0 \end{bmatrix} + C_{p_1} \tag{2.28}$$

$$C_{p_3} = \frac{t_{go}^2}{30} \begin{bmatrix} \ddot{x}_0 \\ \ddot{y}_0 \\ \ddot{z}_0 \end{bmatrix} - C_{p_1} + 2 \cdot C_{p_2} \tag{2.29}$$

Then using the position, velocity, and acceleration at the estimated time of the collision, control points are calculated as given in Eq. 2.32.

$$C_{p_7} = \begin{bmatrix} x_\tau \\ y_\tau \\ z_\tau \end{bmatrix} \tag{2.30}$$

$$C_{p_6} = - \begin{bmatrix} \dot{x}_\tau \\ \dot{y}_\tau \\ \dot{z}_\tau \end{bmatrix} \cdot \frac{t_{go}}{6} + C_{p_7} \tag{2.31}$$

$$C_{p_5} = \begin{bmatrix} \ddot{x}_\tau \\ \ddot{y}_\tau \\ \ddot{z}_\tau \end{bmatrix} \cdot \frac{t_{go}^2}{30} + 2 \cdot C_{p_6} - C_{p_7} \tag{2.32}$$

Only free control point determined by equation 2.33

$$C_{p_4} = \frac{C_{p_3} + C_{p_5}}{2} \tag{2.33}$$

2.3.2 Interception Direction Decision

Interception direction is specified by utilizing current position $p = [x, y, z]$ and velocity $v = [v_x, v_y, v_z]$ values obtained from odometry, as well as the position of the target vehicle ($p_t = [x_t, y_t, z_t]$) and its velocity ($v_t = [v_{x_t}, v_{y_t}, v_{z_t}]$ information. In the designed system, there are two different options for hitting the target in the desired flight direction: frontal interception (approaching the target from the direction it is moving towards) or rear interception (approaching the target from the direction it is moving away from).

If the relative position vector is defined as $r = p_t - p$, the scalar projection of r onto v_t can be denoted as $|\operatorname{proj} v_t r|$. When the target approaches, indicated by $\frac{r}{|r|} \cdot \frac{v_t}{|v_t|} < 0$, it is assessed whether the target can be hit head-to-head with a specified average velocity $\bar{v}_h$, as determined by the condition in 2.34.

$$\frac{|\operatorname{proj}_{v_t} r| + \sqrt{|r|^2 - |\operatorname{proj}_{v_t} r|^2}}{\bar{v}_h + |v_t|} < \frac{|\operatorname{proj} v_t r|}{|v_t|} \tag{2.34}$$

If the condition for head-to-head engagement as specified in Equation 2.34 is not satisfied or when the target recedes, which can be expressed mathematically as $\frac{r}{|r|} \cdot \frac{v_t}{|v_t|} \geq 0$, chasing engagement will be preferred.

2.3.3 Time-to-Go Estimation

To guarantee the practicality of the online trajectory generation using splines, it is essential to choose an appropriate time-to-go that takes into account the kinematics of the interceptor and target vehicle and the limitations of the interceptor vehicle. When it comes to chasing engagement, the time-to-go is determined using the formula specified in equation 2.35, while for head-to-head engagement, the time-to-go is calculated according to equation 2.36.

$$t_{go}^{c} = \frac{|p_t - p|}{|\bar{v}_c u_t - v_t|} \tag{2.35}$$

$$t_{go}^{h} = \frac{|p_t - p|}{|\bar{v}_h u_t + v_t|} \tag{2.36}$$

The average velocities along the interception trajectory for chasing engagement ($\bar{v}_c$) and head-to-head engagement ($\bar{v}_h$) are determined. The unit target velocity vector is

denoted as u_t. The values of $\bar{v}_c$ and $\bar{v}_h$ are defined in equation 2.37, where Nc and N_h are adjustable parameters that determine their specific values.

$$\bar{v}_c = N_c|v_t|, \quad \bar{v}_h = N_h|v_t| \tag{2.37}$$

N_c and N_h are tunable parameters of this method, and they specify the proportion of the interceptor speed to the target speed. The calculations provided in equations

Algorithm 2 Minimum Time-to-Go Estimation

input: $v_t, v, \bar{v}_c, \bar{v}_h, v_{max}, a_{max}$
output: $t_{go,min}$
$t_{go,min} \leftarrow 0$
if Chasing Engagement **then**
    $v_{avg} \leftarrow \bar{v}_c \frac{v_t}{|v_t|}$
else if Head-to-Head Engagement **then**
    $v_{avg} \leftarrow -\bar{v}_h \frac{v_t}{|v_t|}$
end if
for $i \leftarrow 1$ to 3 **do**
    if $|2(v_{avg})_i - (v)_i| \leq v_{max}$ **then**
        $t_{go,i} \leftarrow \left| \frac{2((v_{avg})_i - (v)_i)}{a_{max}} \right|$
    else if $|2(v_{avg})_i - (v)_i| > v_{max}$ **then**
        $t_1 \leftarrow \left| \frac{v_{max} - (v)_i}{a_{max}} \right|$
        $t_{go,i} \leftarrow t_1 + \frac{\left(2(v_{avg})_i - v_{max} - (v)_i\right)t_1}{2\left(v_{max} - (v_{avg})_i\right)}$
    end if
    $t_{go,min} \leftarrow \max\{t_{go,i}, t_{go,min}\}$
return $t_{go,min}$

2.35 and 2.36 may yield infeasible estimates for the time-to-go when the interceptor is unable to achieve the specified average velocity along the trajectory. To address this issue, it is necessary to estimate the minimum time-to-go ($t_{go,min}$) by considering the limitations of interceptor acceleration and maximum velocity. The Algorithm 2 outlines the process of calculating this time-to-go using the interceptor vehicle limits. In the algorithm, $t_{go,i}$ represents the time-to-go value for the i^{th} axis, and $(.)_i$ denotes the i^{th} element of the velocity vector, with indices corresponding to the elements in the order of XYZ. The algorithm calculates the minimum time required to achieve the specified average velocity for each axis. During these calculations, two conditions are taken into account. The first condition occurs when the vehicle must reach its maximum velocity to meet the average velocity requirement. The second condition

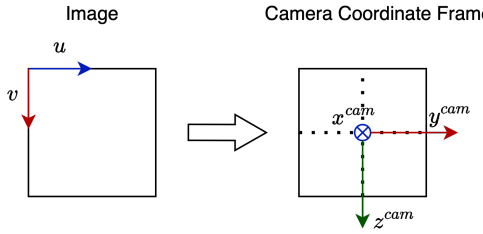

Figure 2.3 : Image plane and camera coordinate frame

arises when the vehicle does not need to reach its maximum velocity to satisfy the average velocity requirement. As a result, the minimum time-to-go is determined based on the axis that yields the lowest time-to-go value among all the considered axes. In the case of chasing engagement, the time-to-go is selected according to equation 2.38.

$$t_{go} = \max\{t_{go}^{c}, t_{go,min}\} \tag{2.38}$$

For head-to-head engagement, time-to-go is selected as in the 2.39.

$$t_{go} = \max\{t_{go}^{h}, t_{go,min}\} \tag{2.39}$$

2.4 Terminal Phase Guidance Algorithm

Pixel values from the detection algorithm are converted into line of sight (LOS) angles within a roll-stabilized coordinate system, and efforts are made to zero these line of sight angles. The reason for operating within a roll-stabilized coordinate system is to eliminate any coupling effects that might arise due to the vehicle's roll angle. The camera's field of view (FOV) is limited, and because the vehicle dynamics cause the vehicle to move forward with a nose-down pitch angle, the camera is positioned on the pitch axis to look upwards. Thus, the pitch angle can achieve larger values. Pixel values from the detection algorithm are first transferred from image coordinates to the camera coordinate axis.

A 3-dimensional vector is created at these coordinates. With u as the horizontal pixel value and v as the vertical pixel value, the created 3-dimensional vector is given in the Equation 2.40

$$\begin{pmatrix} x_{\text{cam}} \\ y_{\text{cam}} \\ z_{\text{cam}} \end{pmatrix} = \begin{pmatrix} f \\ \frac{(u-p_h/2)\cdot d}{p_h} \\ \frac{(v-p_v/2)\cdot d}{p_v} \end{pmatrix} \tag{2.40}$$

Where f is the focal length, d is the size of the sensor inside the camera, and p_h and p_v are the horizontal and vertical pixel counts, respectively. The sensor size d, focal length, and the field of view FOV are related by the following Equation:

$$d = 2f \tan\left(\frac{FOV}{2}\right) \tag{2.41}$$

Subsequently, the vector in camera coordinates is transformed into a stabilized coordinate frame against rolling and pitching:

$$\begin{pmatrix} x_{rps} \\ y_{rps} \\ z_{rps} \end{pmatrix} = R_{rs}^{rps} R_{b}^{rs} R_{cam}^{b} \begin{pmatrix} x_{cam} \\ y_{cam} \\ z_{cam} \end{pmatrix} \tag{2.42}$$

where R_{cam}^{b} is the rotation matrix from camera coordinate frame to body coordinate frame, R_{b}^{rs} is the rotation matrix from body coordinate frame to roll-stabilized coordinate frame, and R_{rs}^{rps} is the rotation matrix from roll-stabilized coordinate frame to both roll and pitch-stabilized coordinate frame. This coordinate transformation ensures that the references are given independent of the vehicle's orientation. The pitch angle θ_c is the camera's upward looking angle along the pitch axis due to mounting angle.

$$R_{cam}^{b} = \begin{bmatrix} \cos\theta_c & 0 & \sin\theta_c \\ 0 & 1 & 0 \\ -\sin\theta_c & 0 & \cos\theta_c \end{bmatrix} \tag{2.43}$$

$$R_{b}^{rs} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \tag{2.44}$$

$$R_{\mathrm{rs}}^{\mathrm{rps}} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \tag{2.45}$$

After this stage, the horizontal λ_h and vertical λ_v line of sight (LOS) angles on the roll-stabilized and pitch-stabilized axis are found as follows:

$$\lambda_h = \arctan\left(\frac{y^{rps}}{x^{rps}}\right) \tag{2.46}$$

$$\lambda_v = \arctan\left(\frac{z^{rps}}{x^{rps}}\right) \tag{2.47}$$

After this stage, the goal is to zero out the LOS angles. References for roll and pitch angles, yaw angular velocity, and climb rate will be given to the vehicle. The reference for the roll angle is

$$\phi_{\text{ref}} = K_{p,\text{roll}}\sigma(\lambda_h) \tag{2.48}$$

The reference for yaw angular velocity is

$$\omega_{z_{\text{ref}}} = K_{p,\text{yaw}}\sigma(\lambda_h) \tag{2.49}$$

where σ is the signed square root function.

$$\sigma(x) = (\text{sign}\, x)\sqrt{x^2} \tag{2.50}$$

The climb rate reference is

$$v_{\text{climbrate}_{\text{ref}}} = K_{p,\text{climbrate}}(\lambda_v) \tag{2.51}$$

The pitch angle reference will be used as the pitch angle θ_{tf} transitioning from pre-terminal to terminal phase navigation. The pitch angle reference is

$$\theta_{\text{ref}} = \theta_{\text{tf}} \tag{2.52}$$

3. NAVIGATION AND CONTROL ARCHITECTURE

3.1 Navigation

Two design has been considered for navigation solution. One of them is using GPS navigation to estimate global position assuming GPS is not jammed or spoofed. Another navigation solution is to determine position by onboard camera and IMU sensors which is Visual Inertial Odometry.

3.1.1 Visual Inertial Navigation

Visual Inertial Odometry (VIO) is an advanced method utilized for estimating the position and orientation of a moving object by integrating data from cameras and inertial sensors. VIO algorithms leverage a combination of visual and inertial measurements to predict the movement of the camera and correct errors in inertial measurements. This capability enables accurate location and orientation estimations even in environments where GPS is unavailable or unreliable, using just the camera and inertial sensors.

Among the VIO algorithms, one that achieves high accuracy is the VINS-Mono algorithm. VINS-Mono, a monocular version of the Visual-Inertial System (VINS) algorithms, is a visual and inertial simultaneous localization and mapping algorithm. It employs only one camera to estimate the system's position and orientation and operates in real-time. This speed and efficiency make it suitable for applications requiring quick decision-making, such as autonomous navigation.

The VINS-Mono algorithm predicts the system's motion by tracking features across consecutive video frames and uses inertial sensor measurements to correct errors in visual measurements. It also utilizes visual measurements to create a map of the environment, which helps enhance the accuracy of motion predictions.

VINS-Fusion, an extension of the VINS-Mono algorithm, uses both monocular and stereo cameras to estimate the system's position and orientation. It combines the benefits of monocular and stereo visual-inertial odometry by tracking visual features with the monocular camera and estimating depth information with the stereo camera. This integration improves the accuracy of motion predictions and environmental mapping. Designed to operate in real-time, this design comprises several core components:

- *Feature Tracking:* This design uses the monocular camera to track features in the environment and matches these features across video frames to predict the system's motion. For each new image, the existing features are tracked using the KLT sparse optical flow algorithm . Concurrently, new corner features are detected to ensure that a minimum number of features are maintained in each image. The detector enforces a uniform feature distribution by setting a minimum separation of pixels between neighboring features. The two-dimensional (2-D) features are first undistorted and then projected onto a unit sphere after passing through an outlier rejection process. The outlier rejection is performed using RANSAC with a fundamental matrix model.
- *Inertial Measurement Unit (IMU):* This work uses IMU data, integrated over time, to predict the system's motion.
- *Stereo Depth Estimation:* This work can use the stereo camera to estimate the depth of features in the environment, enhancing the accuracy of motion predictions.
- *Non-linear Optimization:* This design employs a non-linear optimization algorithm to minimize errors between visual and inertial measurements, improving motion predictions and environmental mapping.
- *Loop Closure:* This design uses loop closure detection to enhance the system's location estimates when the camera revisits a previously seen location.

In this work, the VINS-Fusion algorithm was employed. The system incorporated a monocular camera and an IMU sensor, and since only odometry information was required, the mapping and loop closure features of the VINS-Fusion algorithm were disabled.

The operating principles of the VINS-Fusion algorithm used in this work are illustrated in the flowchart provided in Figure 3.1.

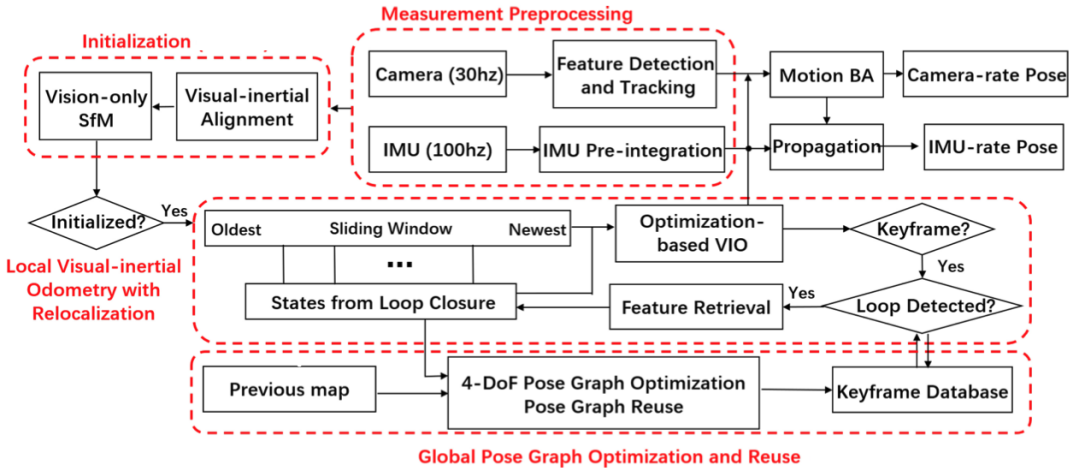

Figure 3.1 : Block diagram illustrating the full pipeline of the VINS-Mono [1]

3.2 Controllers

3.2.1 GPS-aided Navigation Controllers

A decoupled control structure is utilized for flight control. Higher-level controllers are responsible for position and velocity control, as well as lower-level controllers are responsible for angle and angular velocity control. The higher-level controllers employ a P+P+PID structure with feedforward, while the lower-level controllers utilize a P+PID structure with feedforward In addition to the vertical position and velocity controller, compensation for hover state is implemented through hover gas compensation. The horizontal and vertical position controllers can be observed in 3.2 and 3.3, respectively. The acceleration values obtained from the high-level controllers are converted into an angle and angular velocity references, which form the references for the lower-level controllers. The normalized reference values from the lower-level controller are distributed to the motors according to the vehicle configuration by using motor mixing. The block diagram of the lower-level controller is shown in Figure 3.4.

3.2.2 Visual Inertial Navigation Position Controllers

Position controllers have been designed to follow the odometry information obtained from VIO and the reference position and speed values coming from the guidance algorithm. Additionally, during vehicle navigation, position and speed tracking is

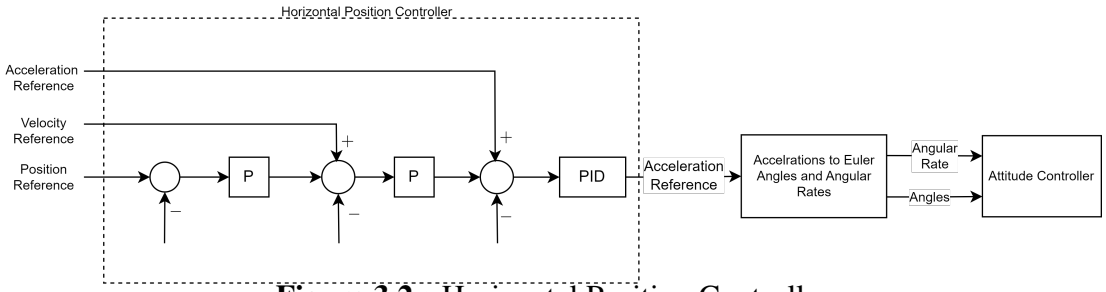

Figure 3.2 : Horizontal Position Controller

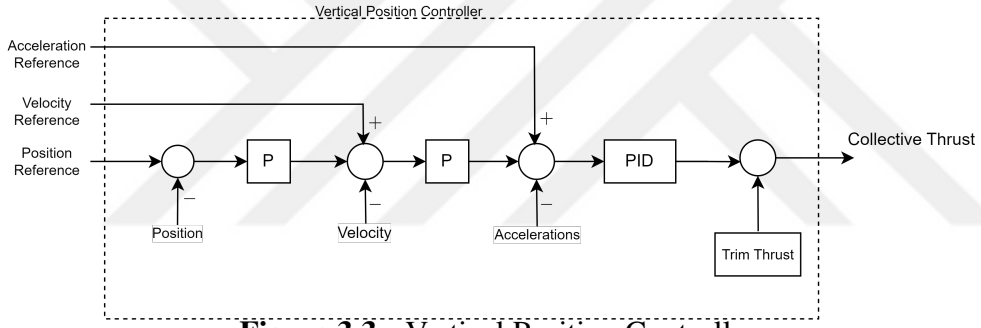

Figure 3.3 : Vertical Position Controller

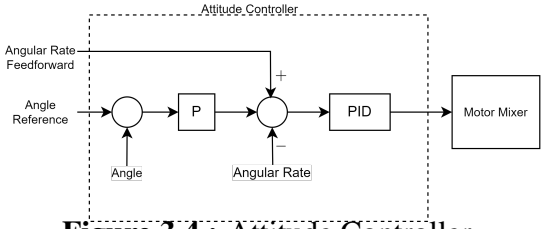

Figure 3.4 : Attitude Controller

provided for the transition to the visual guidance system, and a yaw angle reference has been created with the requirement that the yaw angle should face the direction of travel.

The outputs of the guidance algorithm are the reference position in the inertial coordinate axis ($p_{\text{ref}} = [x_{\text{ref}}, y_{\text{ref}}, z_{\text{ref}}]$), speed ($v_{\text{ref}} = [\dot{x}_{\text{ref}}, \dot{y}_{\text{ref}}, \dot{z}_{\text{ref}}]$), and acceleration ($a_{\text{ref}} = [\ddot{x}_{\text{ref}}, \ddot{y}_{\text{ref}}, \ddot{z}_{\text{ref}}]$) values. The position and speed values of the vehicle will come from VIO. These references and measurements are used to design a position controller. The position controller is designed as two channels, a horizontal position channel and a vertical position channel.

For the horizontal position channel, the control input is defined as the horizontal acceleration command in the inertial coordinate axis, $u_h = [u_{h,x}, u_{h,y}]^T$. This control input and the dynamics between it and the speed can be considered as a first-order system. The time constant and gain of this dynamic have been found from the Simulink model and then verified with flight tests. The dynamics between the elements of the control input and the speed are

$$\frac{v_{h,i}(s)}{u_{h,i}(s)} = \frac{1.9}{2s+1} \tag{3.1}$$

Including the kinematic relationship between speed and position for the dynamics between the control input and the measured position p, we find

$$\frac{p_{h,i}(s)}{u_{h,i}(s)} = \frac{1.9}{s(2s+1)} \tag{3.2}$$

A controller that includes an integrator must be designed to be resilient against disturbances (wind, gusts, etc.). The inclusion of a derivative is necessary to allow for speed reference provision and speed reference tracking. Acceleration feedforward is added to improve reference tracking. The topology of the cascaded P+PI controller can be seen in Figure 3.5.

The speed error control input has maximum and minimum value saturations added, and integral windup in saturation cases is prevented using the clamping method. The change in the acceleration reference has been shaped not to exceed the maximum jerk.

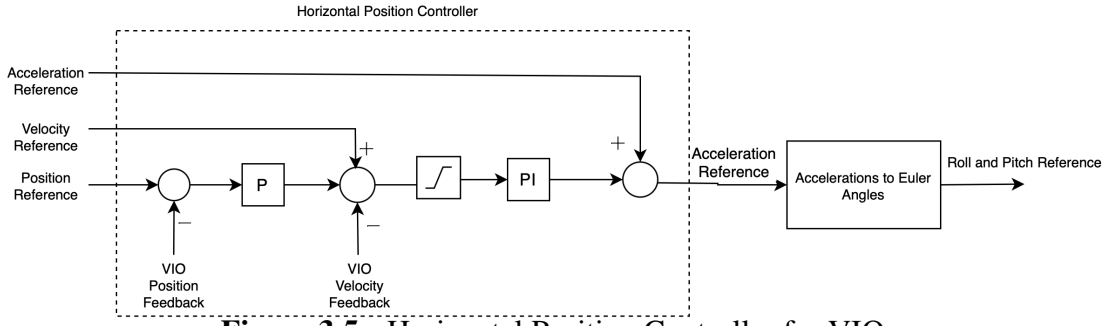

Figure 3.5 : Horizontal Position Controller for VIO

The found u'_h as in equation 3.3 in the inertial axis serves as a reference, and this reference needs to be converted to roll ϕ and pitch θ angles for feeding into the autopilot.

$$u_h^{v1} = R_i^{-1} u'_h \tag{3.3}$$

If defined as such, the reference angles will be calculated and provided to the autopilot as in the equation 3.4.

$$\begin{aligned} \theta_{\text{ref}} &= \arctan\left(-\frac{u_{h,x}^{v1}}{g}\right) \\ \phi_{\text{ref}} &= \arctan\left(\frac{\cos(\theta_{\text{ref}}) u_{h,y}^{v1}}{g}\right) \end{aligned} \tag{3.4}$$

The yaw angle reference is also chosen so that the vehicle continuously faces the target. Accordingly, the yaw angle reference is given in the equation 3.5.

$$\psi_{\text{ref}} = \text{atan2}(y_{\text{ref}} - y, x_{\text{ref}} - x) \tag{3.5}$$

The yaw angle reference is also passed through a low-pass filter before being fed into the autopilot. The filter's cut-off frequency is designed at 5 Hz.

In the vertical position channel, the incoming vertical speed reference is converted into a climb rate and fed directly into the autopilot, thus using the autopilot's altitude controller. The climb rate reference is given in the equation 3.6.

$$v_{\text{climb rate ref}} = -K_c \dot{z} \tag{3.6}$$

4. SIMULATION AND HARDWARE IMPLEMENTATION

4.1 6-DOF Quadrotor and Onboard Sensors Simulation

4.1.1 Dynamics

The dynamic model of the quadrotor is described by the following set of equations , 4.1, 4.2, 4.3, 4.4 :

$$\dot{p} = R_B^I v^B \tag{4.1}$$

$$\dot{q} = \frac{1}{2} q \otimes \Omega \tag{4.2}$$

$$m\dot{v}^B + m\Omega \times v^B = F_{prop} + F_{aero} + F_{grav} \tag{4.3}$$

$$I_{cm}\dot{\Omega} + \Omega \times I_{cm}\Omega = M_{prop} + M_{gyro} + M_{aero} \tag{4.4}$$

where $p = [x\ y\ z]^T$ is the position of the vehicle in the inertial frame, $q = [q_0\ q_1\ q_2\ q_3]^T$ represents the attitude of the vehicle, $v^B = [u\ v\ w]^T$ is the airspeed in the body frame, $\Omega = [p\ q\ r]^T$ is the angular velocity of the vehicle in the body frame, m is the mass in kg, $I_{cm} \in \mathbb{R}^{3\times3}$ represents the moment of inertia, and R_B^I is the rotation matrix from the body frame to the inertial frame.

Thrust is calculated as in equation 4.5:

$$T_i = C_t \omega_i^2 \tag{4.5}$$

where T_i is the thrust produced by the i^{th} motor's propeller, C_t is the thrust coefficient, and ω_i is the rotational speed of the i^{th} motor. The internal dynamics of the motor are modeled as a first-order lag. The total thrust for the four propellers is expressed in equation 4.6:

$$F_{prop} = \sum_{i=1}^{4} T_i = C_t(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \tag{4.6}$$

The drag force acting on the vehicle is given in 4.7 as:

$$F_{aero} = -C_d v^B |v^B| \tag{4.7}$$

where C_d is the body frame drag coefficient in $N/(m/s)^2$. F_{grav} is the force due to gravitational acceleration, where g is the gravitational acceleration in m/s^2.

The rotation of the propellers produces a reaction torque counter to the direction of rotation. C_m is the moment coefficient of the propeller in $N/(m/s)^2$, and d is the distance between the rotor and the center of mass in meters. The propeller torque in the equation 4.8, M_{prop}, is given by:

$$M_{prop} = \begin{bmatrix} -\frac{\sqrt{2}}{2}dC_t & -\frac{\sqrt{2}}{2}dC_t & \frac{\sqrt{2}}{2}dC_t & \frac{\sqrt{2}}{2}dC_t \\ \frac{\sqrt{2}}{2}dC_t & -\frac{\sqrt{2}}{2}dC_t & -\frac{\sqrt{2}}{2}dC_t & \frac{\sqrt{2}}{2}dC_t \\ -C_m & C_m & -C_m & C_m \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \tag{4.8}$$

Defining I_r as the moment of inertia of the motor and propeller, p and q as angular velocities in the body frame. The aerodynamic moment due to the angular velocity of the vehicle and changes in the rotation axis and rotational speed of the motor causes a gyroscopic moment, M_{gyro}, shown in the equation 4.9

$$M_{gyro} = \sum_{i=1}^{4} (-1)^{i+1} \begin{bmatrix} I_r p \omega \\ I_r q \omega \\ I_r \dot{\omega} \end{bmatrix} \tag{4.9}$$

The aerodynamic moment can be calculated as in equation :

$$M_{aero} = -C_{dm}\Omega|\Omega| \tag{4.10}$$

where C_{dm} is the drag moment coefficient.

Quadrotor dynamics are simulated in the MATLAB/Simulink environment for finding tunable parameters of the control and guidance algorithms.

4.1.2 Target on pinhole camera simulation

Target simulated in inertial coordinates. Then, using camera's intrinsic and extrinsic parameters target projection onto image is simulated. Assuming target's inertial coordinates are $p_T^I = [x_T^I \; y_T^I \; z_T^I]^T$, target position on the camera frame is $p_T^{cam} = [x_T^{cam} \; y_T^{cam} \; z_T^{cam}]^T$ is calculated by equation 4.11.

$$p_T^{cam} = R_b^{cam} R_I^b p_T^I \tag{4.11}$$

Where R_b^{cam} defined as rotation matrix from body to camera is given at equation 4.12.

$$R_b^{cam} = \begin{bmatrix} \cos\theta_c & 0 & -\sin\theta_c \\ 0 & 1 & 0 \\ \sin\theta_c & 0 & \cos\theta_c \end{bmatrix} \tag{4.12}$$

$R_I^b = (R_b^I)^T$ and rotation matrix from body to inertial coordinates, R_b^I is given at the equation 4.13.

$$R_b^I = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \tag{4.13}$$

By using positions in the camera coordinate frame, simulated pixels that projected onto image frame. Used coordinate frame is mentioned in the the Section 2.4 at the Figure 2.3. Horizontal and vertical pixel locations p_u and p_v are given at the equations 4.14 and 4.15, respectively.

$$p_u = \frac{f \cdot y_T^C \cdot p_{horz}}{x_T^C \cdot d_{horz}} + o_{horz} \tag{4.14}$$

$$p_v = \frac{f \cdot z_T^C \cdot p_{vert}}{x_T^C \cdot d_{vert}} + o_{vert} \tag{4.15}$$

p_{horz} and p_{vert} are total horizontal and vertical pixels, respectively. d_{horz} and d_{vert} are sensor size at horizontal and vertical directions, and o_{horz} and o_{vert} are offset due to difference at the accepted reference points of the camera and image coordinate frames.

Visibility is checked by whether calculated p_u and p_v are within the limits of the pixels. If both of the $0 \leq p_u \leq p_{horz}$ and $0 \leq p_v \leq p_{vert}$ conditions are satisfied then target is visible.

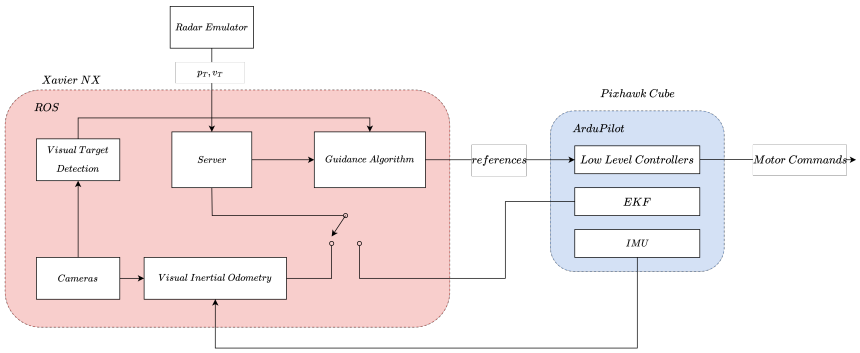

Figure 4.1 : Hardware and software architecture of the system

4.2 Hardware Implementation

To test the algorithms in flight tests, test platform's system architecture is given at the Figure 4.1. As a mission computer Xavier NX is used. Pixhawk Cube is selected as a autopilot hardware and ArduPilot [19] is used for software of the autopilot due to proven reliability and versatility. Robot Operating System (ROS) is used in the mission computer [20], since ease of interoperability between different tasks in the mission computer and also ease of integration with ArduPilot. Data communication between autopilot and mission computer made using MAVLink protocol [21].

Two camera is used at the system: one camera for monocular visual inertial navigation and another for aerial target detection. A server node has been created that provides autopilot status and data communication with the autopilot. The server also provides radio communication with the radar or a device that emulates the radar. It parses and interprets the incoming target data and feeds the target position and speed to the guidance algorithm. For the flight tests, the target vehicle sends its position and speed information to the ground station, and the ground station sends this message to the interceptor vehicle. In this way, the radar measuring the position and speed of the target vehicle was emulated. Depending on whether GPS is active or not, the interceptor's odometry information was obtained from visual inertial odometry or GPS-based navigation estimates in ArduPilot's EKF. Requirements for pre-terminal phase guidance and terminal phase guidance is fed into Guidance Algorithm node and references were fed into autopilot.

5. RESULTS

5.1 Simulation Results

5.1.1 Pre-Terminal guidance with MPC Simulation Results

For the purpose of validating the algorithm, simulations are carried out to test scenarios where the target is approaching and scenarios where the target is moving away. The proposed algorithm is also evaluated on both maneuvering and non-maneuvering targets. Although the proposed algorithm does not utilize information about the target's acceleration, it is still tested on targets that do have acceleration. The target's kinematics are defined by the equation 5.1, while the interceptor's kinematics are defined by the equation 5.2.

$$\dot{\boldsymbol{x}}_T = \begin{bmatrix} \dot{\boldsymbol{p}}_T \\ \dot{\boldsymbol{v}}_T \end{bmatrix} = \begin{bmatrix} \boldsymbol{v}_T \\ \boldsymbol{a}_T \end{bmatrix} \tag{5.1}$$

$$\dot{\boldsymbol{x}}_I = \begin{bmatrix} \dot{\boldsymbol{p}}_I \\ \dot{\boldsymbol{v}}_I \end{bmatrix} = \begin{bmatrix} \boldsymbol{v}_I \\ \boldsymbol{a}_I \end{bmatrix} \tag{5.2}$$

Four simulation scenarios were analyzed to demonstrate the effectiveness of the approach. The initial two scenarios involve targets moving at constant velocities, one receding and the other approaching, while the remaining two scenarios involve targets with accelerations, again one receding and one approaching. The maximum and minimum acceleration values are represented as $\boldsymbol{a}_{max} = (18, 18, 18)$ and $\boldsymbol{a}_{min} = (-18, -18, -9.81)$ respectively. The values chosen for the simulations are denoted as N_c and N_h are :

$$N_c = 1.5 \quad N_h = 1 \tag{5.3}$$

The simulations are terminated when the Euclidean distance between the target and interceptor positions is less than 0.2 meters. At the beginning of the simulations, the interceptor is stationary, meaning it has no initial velocity. For the simulation experiments, the value of N was set to 10. The Model Predictive Control (MPC) is

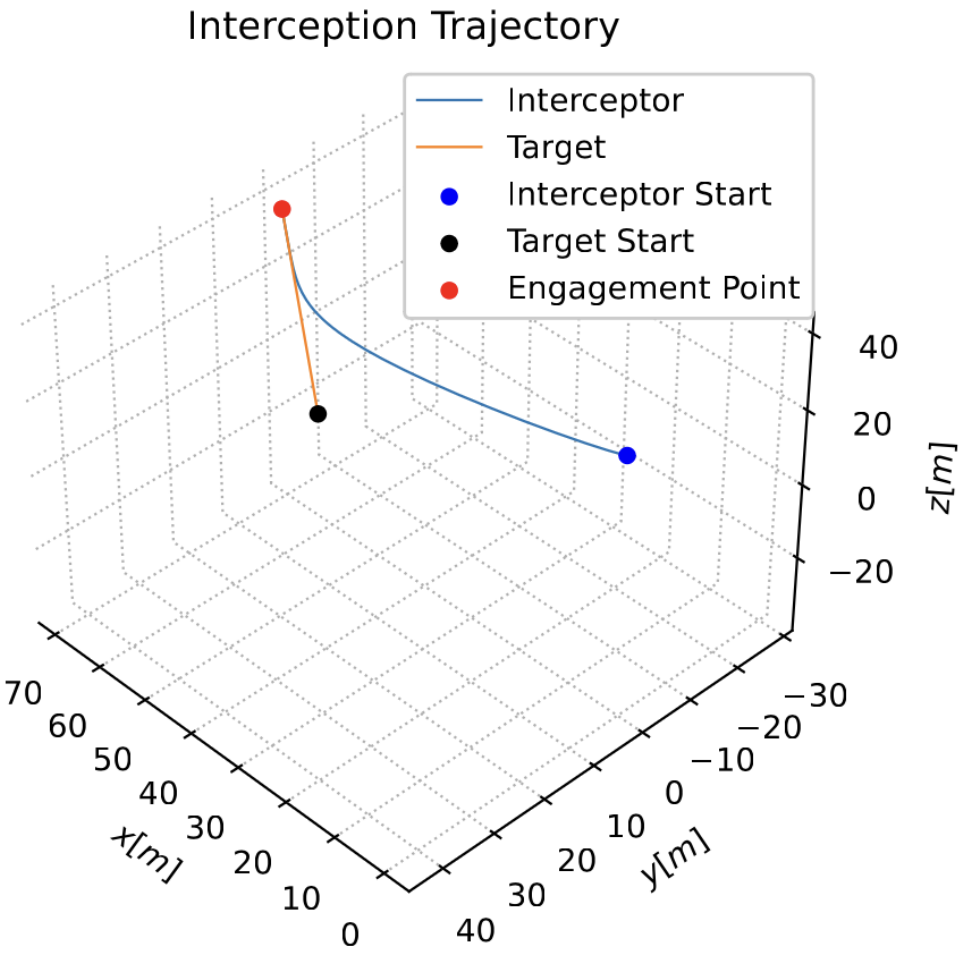

Figure 5.1 : Chase interception trajectory for target with constant velocity

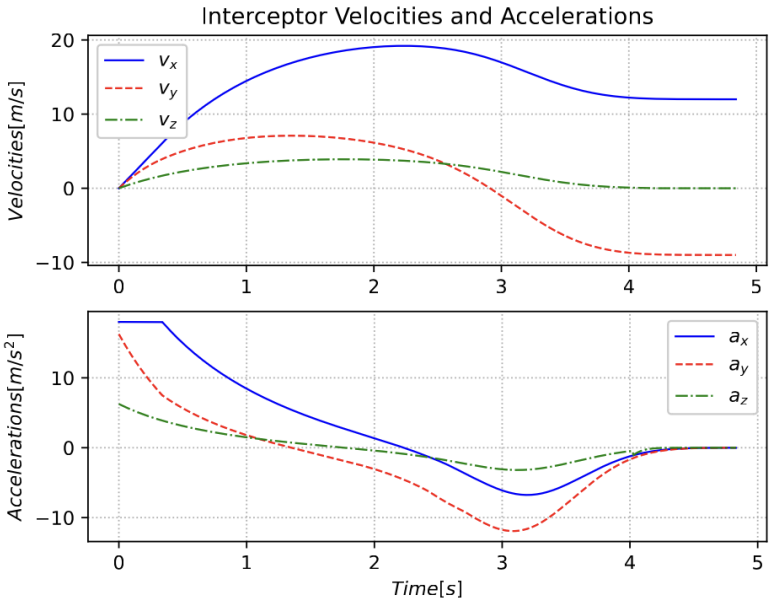

Figure 5.2 : Velocity and acceleration profiles of chasing interception for target with constant velocity

solved at a frequency of 50 Hz, and the first control input computed by the MPC is applied to the system.

The first test case for the proposed method involves a receding non-maneuvering target. The target has an initial position of $\boldsymbol{p}_T(0) = (30,30,40)^T$ and a constant velocity of $\boldsymbol{v}_T(0) = (8,-6,0)^T$. The interceptor's initial position is set to $\boldsymbol{p}_I(0) = (0,0,30)^T$. The 3D position of the interception chase is shown in Figure 5.1. The velocity and acceleration profiles for the interceptor are presented in Figure 5.2. As expected for a constant velocity target scenario, the accelerations approach zero.

Another scenario is the non-maneuvering approaching target. Target's initial position is $\boldsymbol{p}_T(0) = (-70,50,40)^T$ and velocity is $\boldsymbol{v}_T(0) = (7,-7,0)^T$. Interceptor's initial position is $\boldsymbol{p}_I(0) = (0,0,30)^T$. 3D-Position of head-to-head interception is given in the Figure 5.3. Velocity and acceleration profiles for interceptor are given in the Figure 5.4. Accelerations go to zero, as expected for a constant velocity target scenario.

Although target acceleration information is not used in guidance law, the proposed method intercepts maneuvering targets as shown in 3D-Position in the Figure 5.5. Target initial position is $\boldsymbol{p}_T(0) = (0,30,40)^T$ and initial velocity is $\boldsymbol{v}_T(0) = (8,5,0)^T$

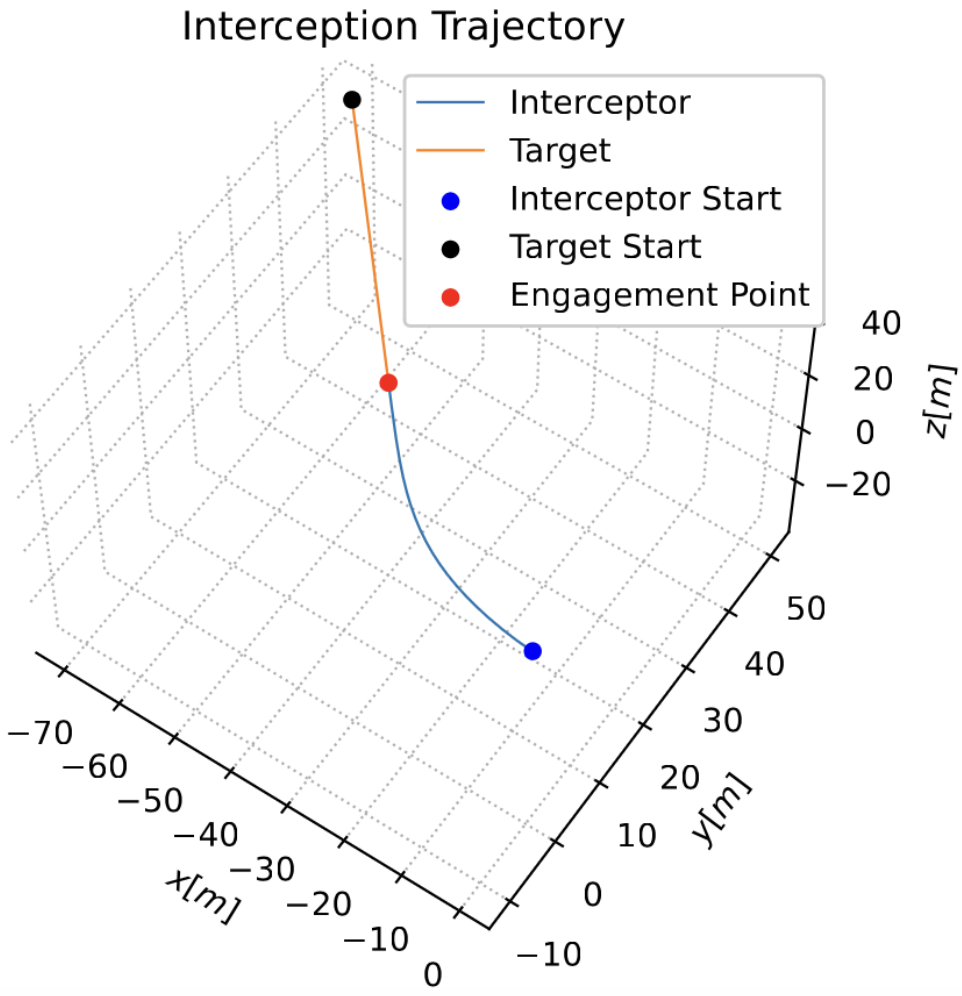

Figure 5.3 : Head-to-head interception trajectory for target with constant velocity

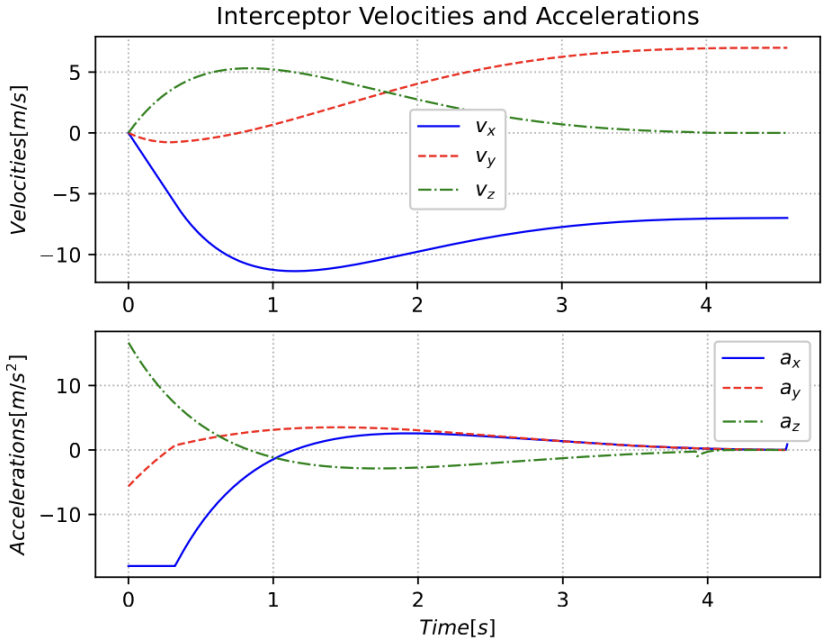

Figure 5.4 : Velocity and acceleration profiles of head-to-head interception for target with constant velocity

with radial acceleration $2m/s^2$ in X-Y plane. Initial position for the interceptor is $\boldsymbol{p}_I(0) = (0,0,30)^T$. Velocities and accelerations for this scenario can be seen at the Figure 5.6. Engagement is occurred at 6.98^{th} seconds. In another case for this scenario, the target has an acceleration component at the tangent to the velocity, which means the target is increasing its speed during flight. 3D-Position in the Figure 5.7, velocity and acceleration profiles are in the Figure 5.8. Initial position of the interceptor and the target are $\boldsymbol{p}_I(0) = (0,0,30)^T$ and $\boldsymbol{p}_T(0) = (10,30,40)^T$, respectively. Initial velocity of the interceptor is $\boldsymbol{v}_T(0) = (5,0,0)^T$, tangential and radial accelerations are $3m/s^2$ and $-1m/s^2$, respectively. Target has a velocity limit of 10 m/s.

Finally, the head-to-head engagement for maneuvering target scenario is assessed on the simulations. Target initial position is $\boldsymbol{p}_T(0) = (50,50,30)^T$ and initial velocity is $\boldsymbol{v}_T(0) = (-7,-7,0)^T$ with radial acceleration $2m/s^2$ in X-Y plane. Interceptor's initial position is $\boldsymbol{p}_I(0) = (0,0,30)^T$. 3D-Position graph can be seen at the Figure 5.9. Velocities and accelerations for this scenario can be seen at the Figure 5.10. In another instance of this situation, the target exhibits an acceleration component aligned with the tangent to its velocity vector. This implies that the target is increasing its speed

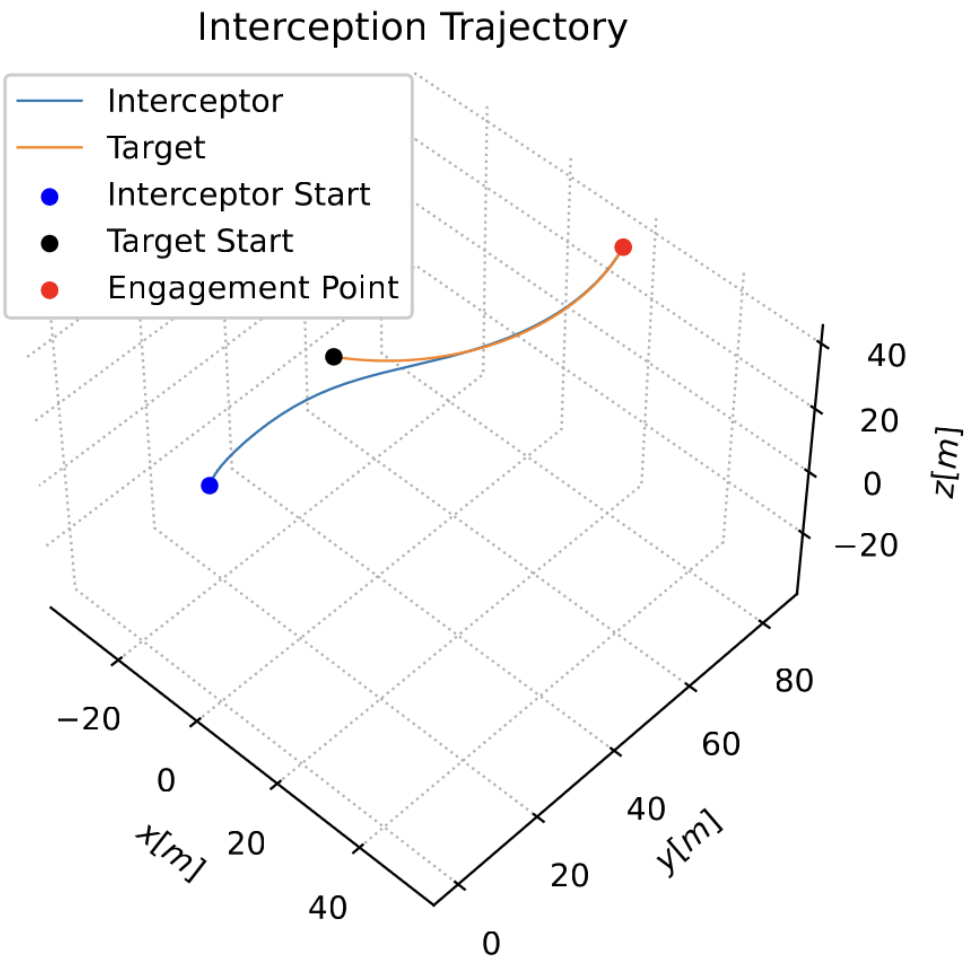

Figure 5.5 : Chasing interception trajectory for target with radial acceleration

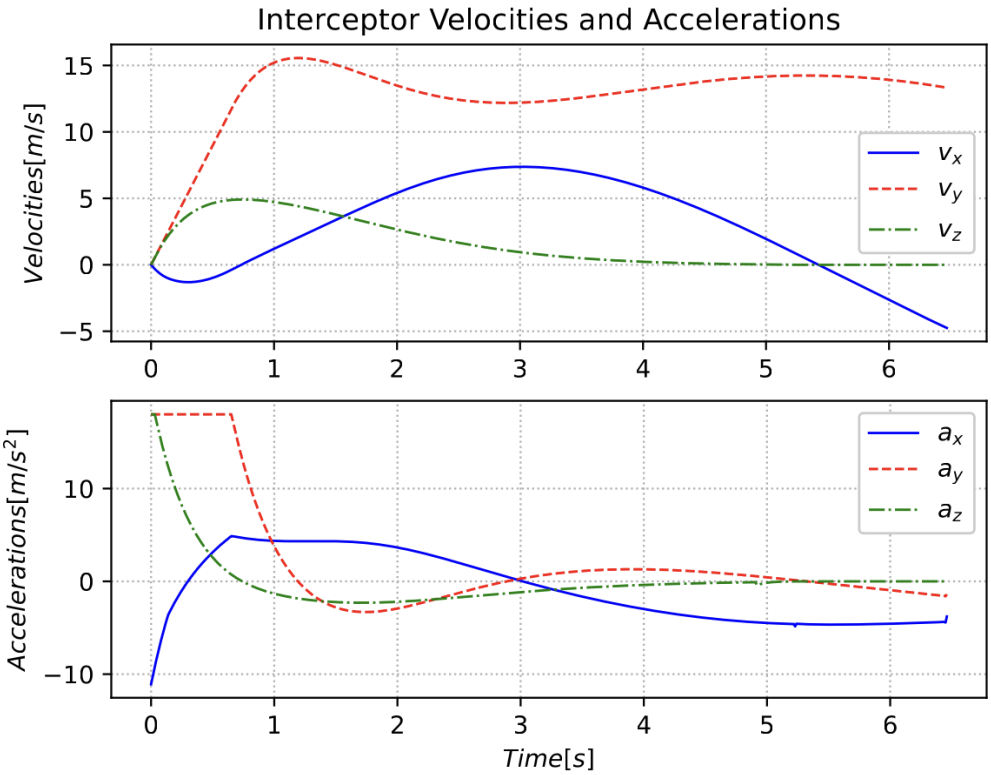

Figure 5.6 : Velocity and acceleration profiles of chasing interception for target with radial acceleration

during its trajectory. The three-dimensional position can be observed in Figure 5.11, while the velocity and acceleration profiles are depicted in Figure 5.12. Initially, the interceptor's position and the target's position are $\boldsymbol{p}_I(0) = (0,0,30)^T$ and $\boldsymbol{p}_T(0) = (70,70,40)^T$, respectively. The interceptor's initial velocity is $\boldsymbol{v}_T(0) = (-3,-3,0)^T$, and the tangential and radial accelerations are $3m/s^2$ and $-1m/s^2$, respectively. The target has a maximum velocity limit of 10 m/s.

Results obtained from the simulations show the ability of the presented predicted guidance strategy. Modifying the input weight matrix $R(k)$ by adding time weights for the MPC cost satisfies the requirement for less acceleration input at the end of the interception trajectory. This property of the method allows having smaller miss distances. As can be seen from the results, interceptor speed converges to $\bar{v}_c$ or $\bar{v}_h$ for chasing or head-to-head engagement, respectively. Even though the target is assumed to have constant velocity in the MPC problem, the presented method can satisfy interception with targets with changing velocity.

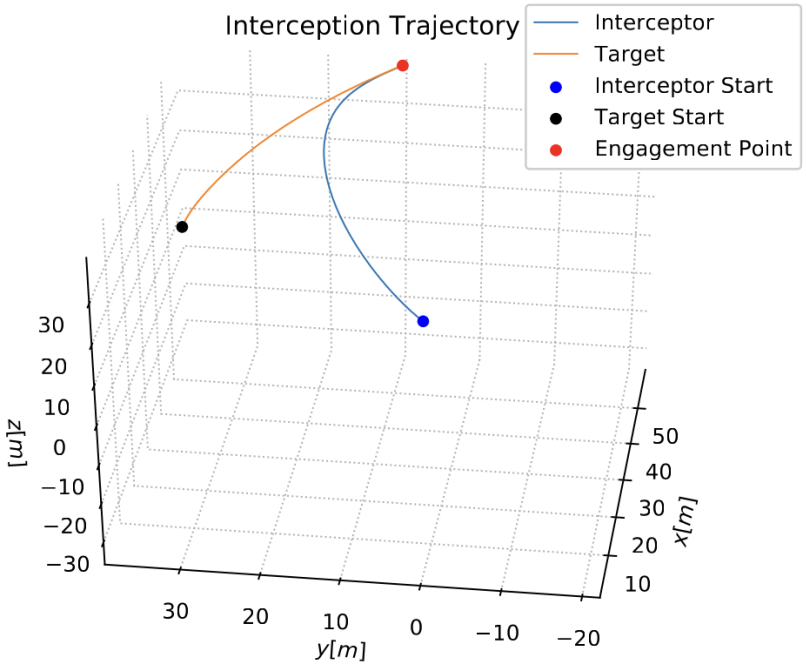

Figure 5.7 : Chasing interception trajectory for target with radial and tangential acceleration

Proposed guidance procedure considering that the visual interception problem demands a parallel impact angle enabling effective visual tracking and robustness to processing latency. We have exploited MPC with added terminal constraints to satisfy engagement at the desired angle. The cost input weight matrix is used in a time-weighted form to reduce the interceptor's requirement for maneuvering at the end of the trajectory, and time-to-go values are calculated considering vehicle limits. Simulation results demonstrate the effectiveness of the proposed method for intercepting both maneuvering and non-maneuvering targets.

5.1.2 Pre-terminal guidance with Bezier Splines

For algorithm validation purposes, simulation studies are conducted for approaching and receding target interception scenarios for both maneuvering and non-maneuvering targets. Simulations are conducted using explained dynamical model and flight controller. Four simulation scenarios were considered to show the method's performance: receding and approaching maneuvering and non-maneuvering targets.

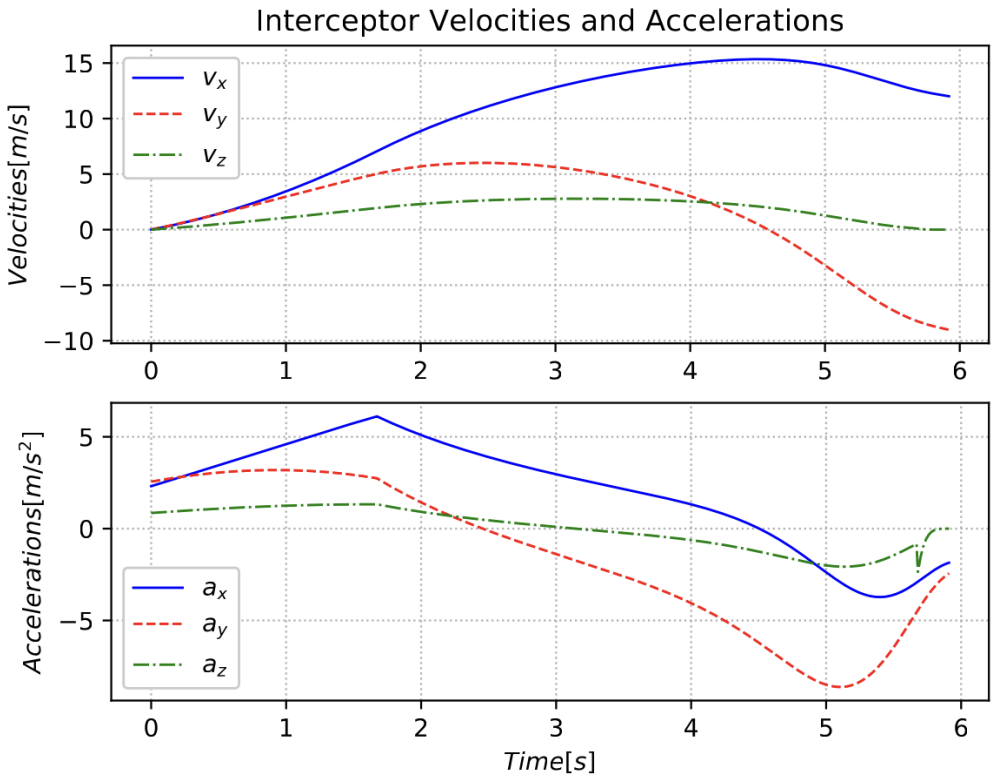

Figure 5.8 : Velocity and acceleration profiles of chasing interception for target with radial and tangential acceleration

The simulations are terminated once the Euclidean distance between the positions of the target and interceptor becomes smaller than 0.2 meters. Initially, the interceptor is stationary without any velocity. The trajectory is planned, and the system receives sampled reference positions, velocities, and accelerations at each sampling time of the guidance controller.

First scenario is the non-maneuvering approaching target. The target begins at the initial coordinates $p_t(0) = (50, 80, 50)^T$ with an initial velocity of $v_t(0) = (-5, -3, -2)^T$. Simultaneously, the interceptor starts at the position $p(0) = (0, 0, 50)^T$. 3D-Position of head-to-head interception is given in the Figure **??**. Velocity and acceleration profiles for an interceptor are given in Figure 5.14.

Secondly, the proposed method is tested on receding non-maneuvering target scenarios. The target starts from the initial position $p_t(0) = (50, 50, 50)^T$ and has an initial velocity of $v_t(0) = (-5, 5, 2)^T$. At the same time, the interceptor initiates its movement from the position $p(0) = (0, 0, 50)^T$. 3D-Position of chasing interception

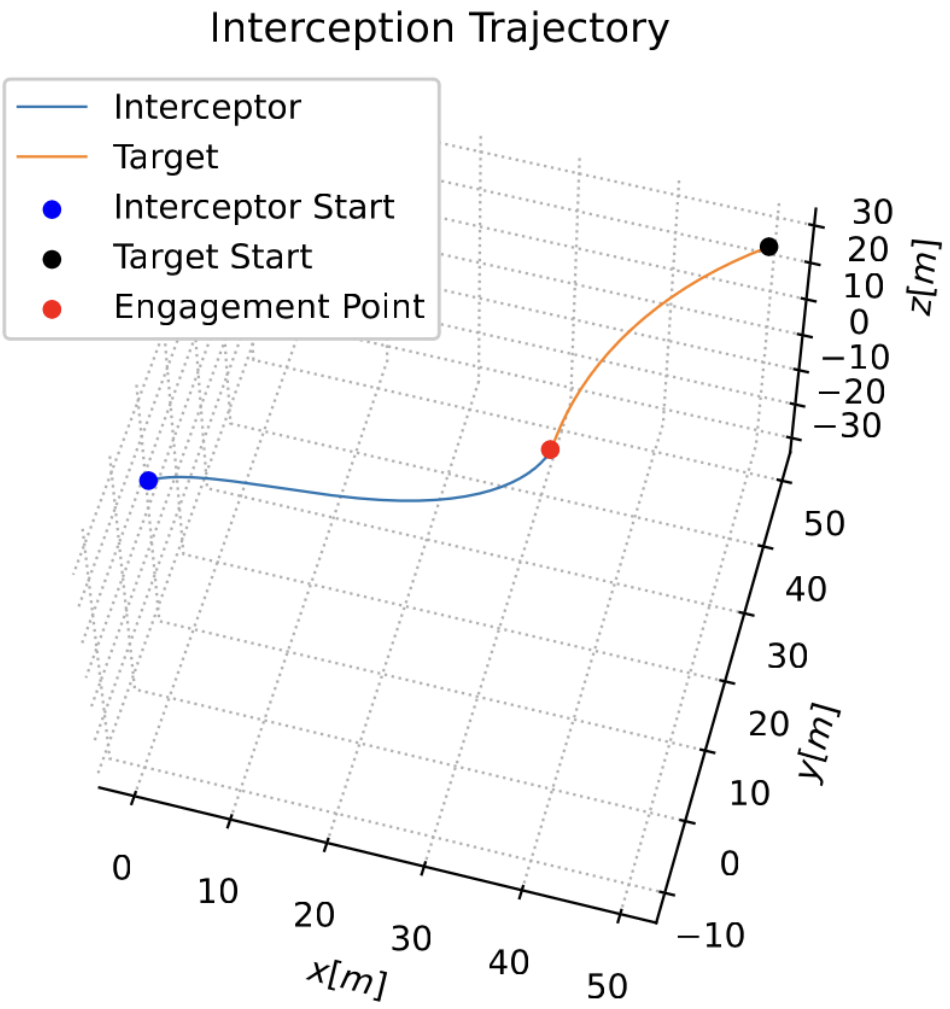

Figure 5.9 : Head-to-head interception trajectory for target with horizontal radial acceleration

is given in the Figure 5.15. Velocity and acceleration profiles for the interceptor are given in Figure 5.16.

Thirdly, the proposed method is tested on maneuvering target scenarios. The target starts from the initial position $p_t(0) = (50,50,50)^T$ and has an initial velocity of $v_t(0) = (5,5,0)^T$. Target has $1m/s^2$ radial acceleration in xy-plane and $-1m/s^2$ radial acceleration in xz-plane. Interceptor's initial position is $p(0) = (0,0,30)^T$. 3D-Position of chasing interception is given in the Figure 5.17. Velocity and acceleration profiles for the interceptor are given in Figure 5.18.

Lastly, the proposed approach undergoes testing in scenarios involving maneuvering targets. The target commences its motion from the initial coordinates $p_t(0) =$

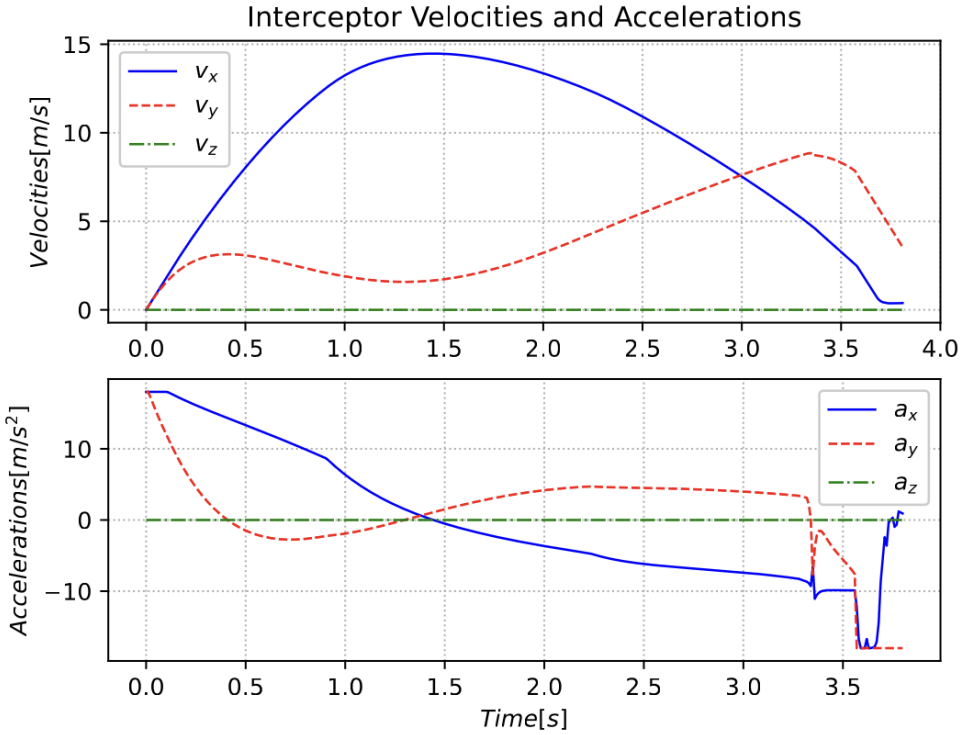

Figure 5.10 : Velocity and acceleration profiles of head-to-head interception for target with radial acceleration

$(-80,50,50)^T$ with an initial velocity of $v_t(0) = (0,-7,0)^T$. The target exhibits a radial acceleration of $1m/s^2$ in the xy-plane. The interceptor initiates its motion from the position $p(0) = (0,0,30)^T$. The three-dimensional position of the intercepting pursuit is depicted in Figure 5.19. Velocity and acceleration profiles for the interceptor are illustrated in Figure 5.20.

5.1.3 Terminal Phase Simulation

For simulating terminal phase, MATLAB based simulations are conducted for full 6-DoF model. Gains of the guidance algorithm are tuned in the simulation environment by batch simulations. Then, tuned guidance gains are simulated in ArduPilot Software-In-The-Loop (SITL). Simulation results can be seen at the Figures 5.21, 5.22 and 5.23.

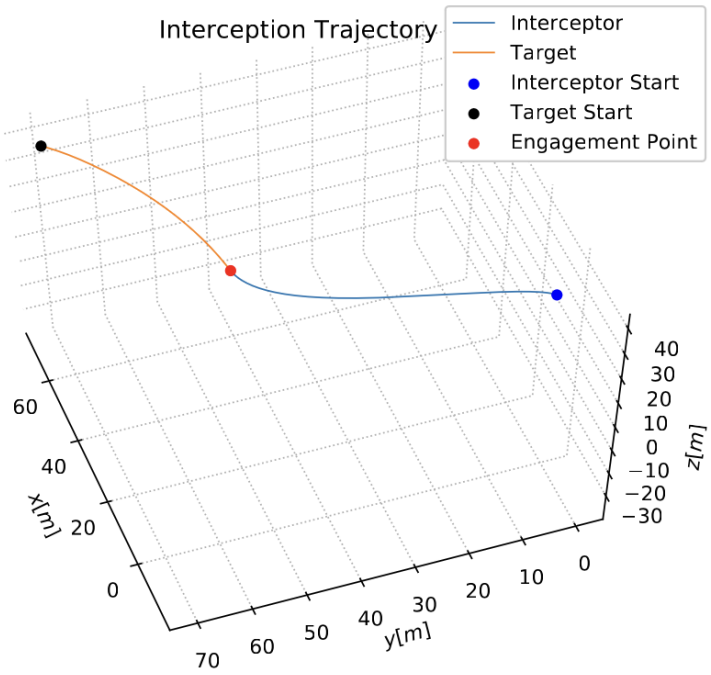

Figure 5.11 : Head-to-head interception trajectory for target with tangential and radial acceleration

Table 5.1 : VIO Navigation Root Mean Square Errors wrt GPS Navigation.

Direction	RMS Error (m)
North Direction	5.8196
East Direction	5.7843
Down Direction	3.2855

5.2 Flight Tests

5.2.1 Visual inertial odometry navigation flight tests

Different flight scenerios are conducted to assess the performance of the VIO. More than 20 test flights are made. Root Mean Square error between VIO estimations and GPS estimations are given in the Table 5.1. For testing VIO and controllers, the waypoint following trajectory has been created. 4 waypoints are defined where they

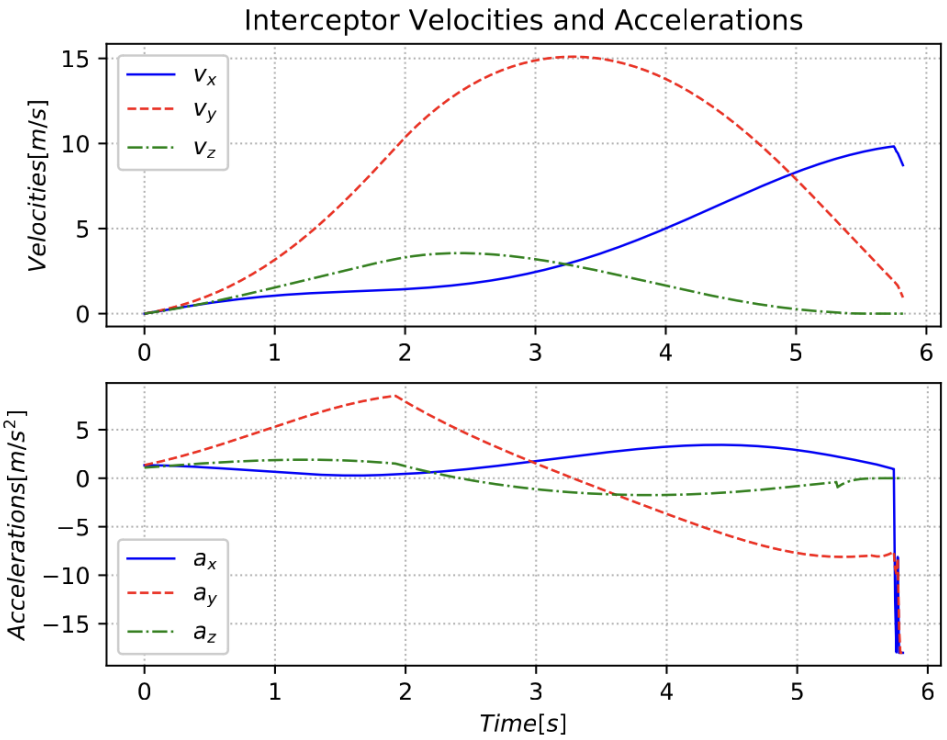

Figure 5.12 : Velocity and acceleration profiles of head-to-head interception for target with tangential and radial acceleration

are given in the equation 5.4.

$$\begin{aligned} \boldsymbol{wp_1} &= (0,\ 0,\ -100)^T \\ \boldsymbol{wp_2} &= (100,\ 0,\ -100)^T \\ \boldsymbol{wp_3} &= (100,\ -100,\ -100)^T \\ \boldsymbol{wp_4} &= (0,\ -100,\ -100)^T \\ \boldsymbol{wp_5} &= (0,\ 0,\ -100)^T \end{aligned} \tag{5.4}$$

For these waypoints, trajectory generated for given waypoints. VIO and GPS-based EKF estimations are given at the Figures 5.24 and 5.25. Controller tracking performance can be seen at the Figure 5.26. Reference positions and position feedbacks from VIO are given at the upper side Figure 5.26. Reference velocities and and velocity feedbacks from VIO are given at the lower side of the Figure 5.26.

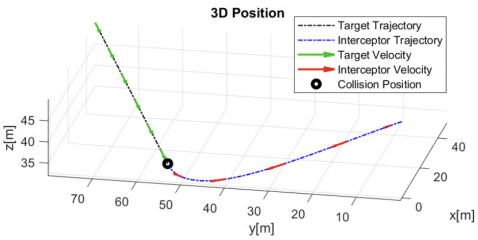

Figure 5.13 : Positions of Head-to-head interception trajectory for non-maneuvering target

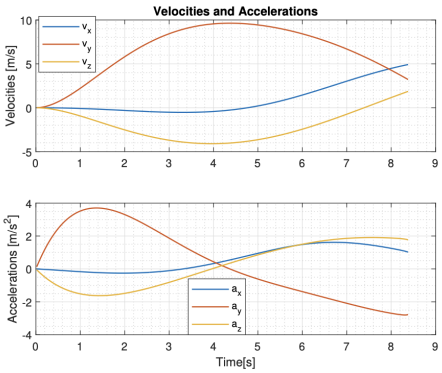

Figure 5.14 : Velocity and acceleration profiles of head-to-head interception for non-maneuvering target

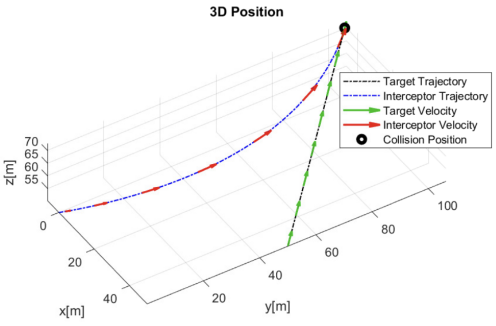

Figure 5.15 : Chasing interception trajectory for non-maneuvering target

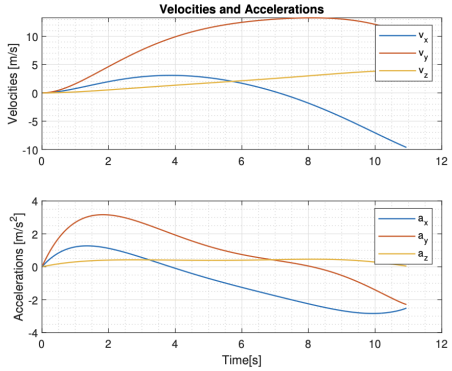

Figure 5.16 : Velocity and acceleration profiles of chasing interception for non-maneuvering target

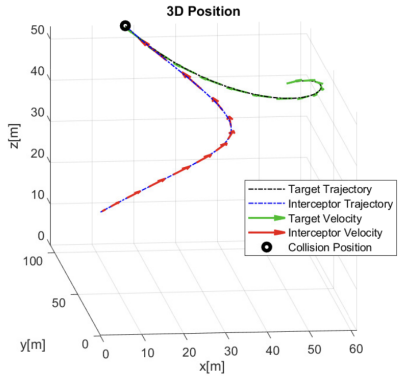

Figure 5.17 : Chasing interception trajectory for maneuvering target

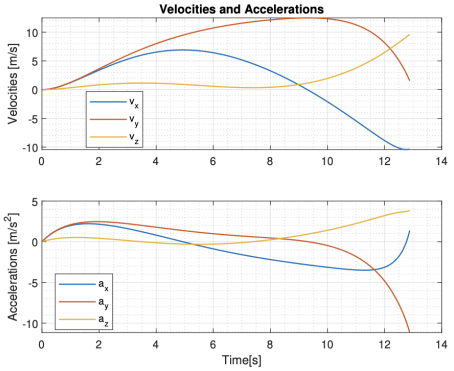

Figure 5.18 : Velocity and acceleration profiles of chasing interception for target with radial acceleration

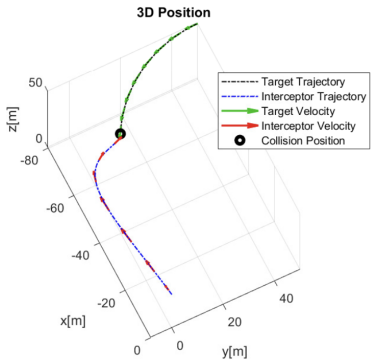

Figure 5.19 : Head-to-head interception 3D position trajectory for maneuvering target

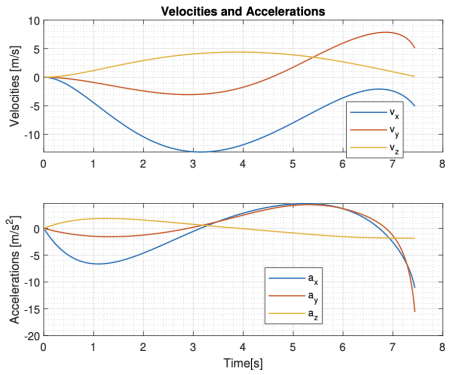

Figure 5.20 : Velocity and acceleration profiles of head-to-head interception for maneuvering target

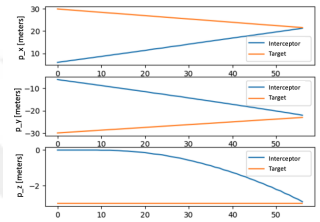

Figure 5.21 : SITL Results of the Terminal Phase Simulation Scenerio 1

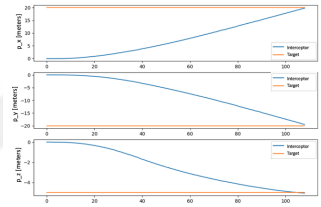

Figure 5.22 : SITL Results of the Terminal Phase Simulation Scenerio 2

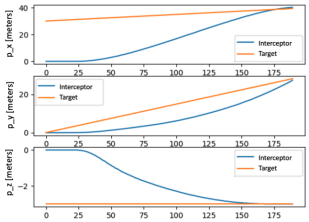

Figure 5.23 : SITL Results of the Terminal Phase Simulation Scenerio 3

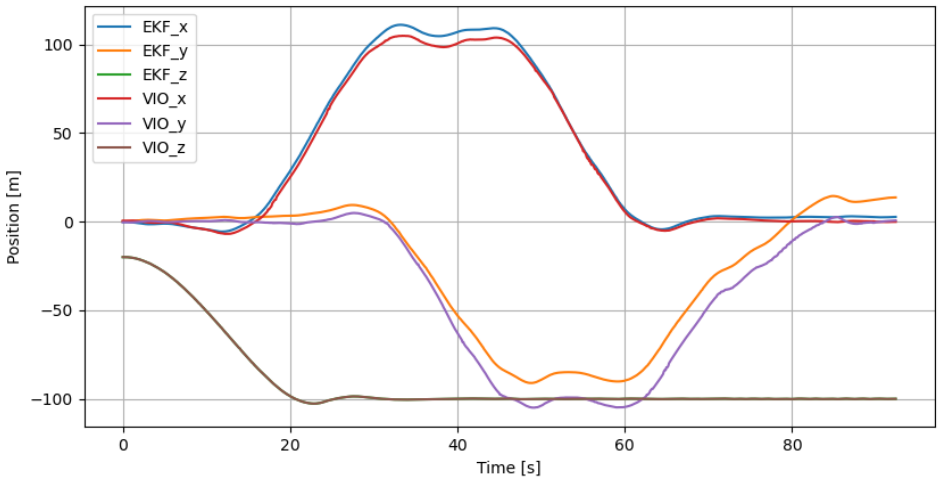

Figure 5.24 : VIO and EKF Position Estimations Comparison

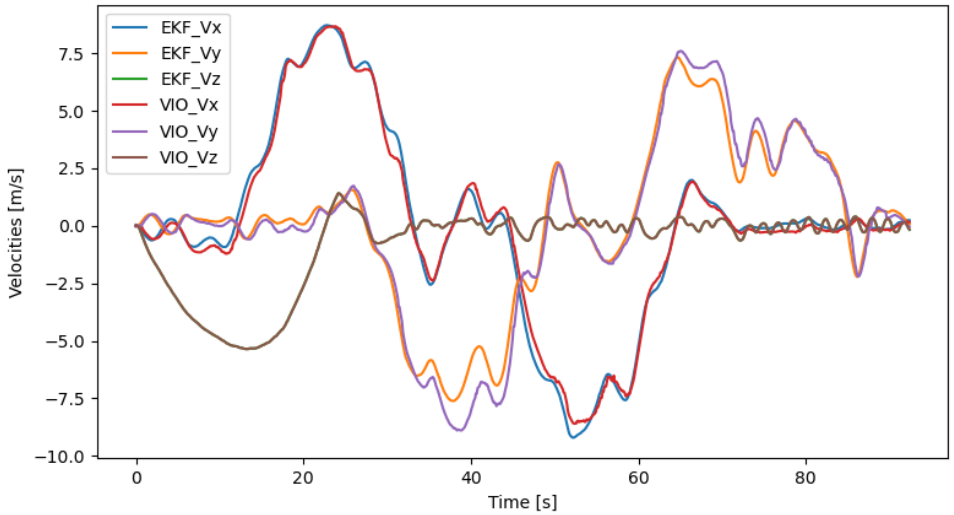

Figure 5.25 : VIO and EKF Velocity Estimations Comparison

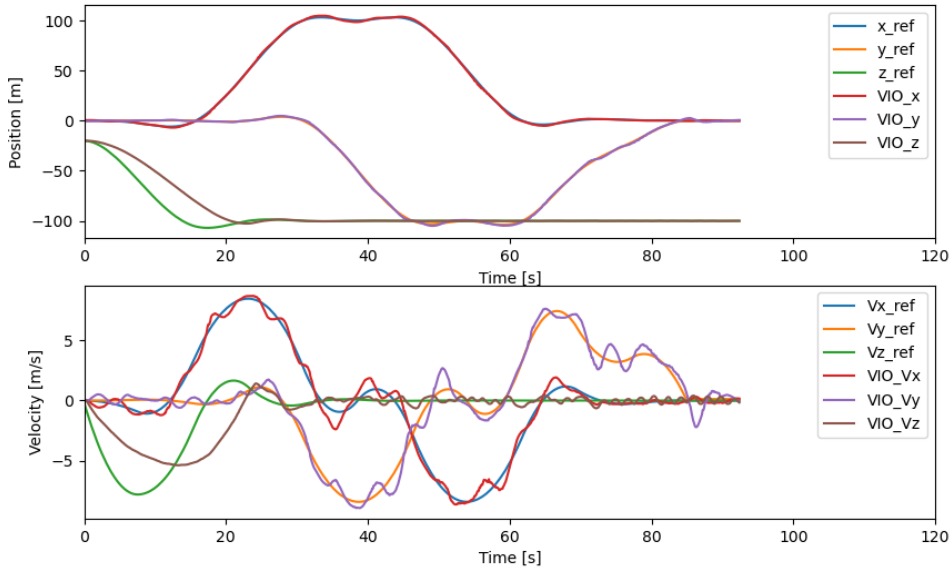

Figure 5.26 : Controller tracking performance

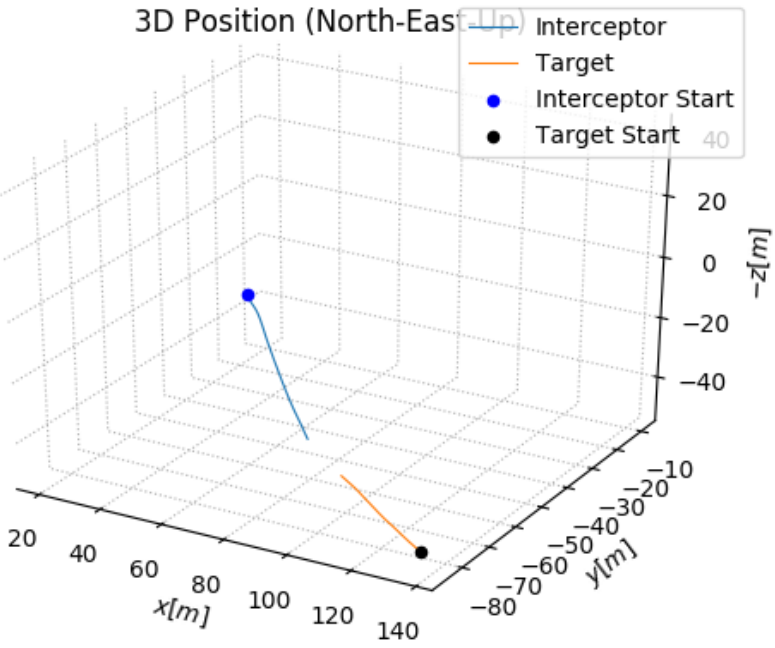

Figure 5.27 : Guidance with GPS 3D Position

5.2.2 Interception flight tests

Guidance and navigation algorithm integrated system is tested on flight. Since MPC requires high computational power only Bezier Spline based guidance algorithm is tested. Both GPS-based navigation and VIO navigation algorithms are tested for pre-terminal phases. For the flight tests, target position and velocity data comes at 5 Hz. GPS-based Navigation and terminal phase results are given at the Figures 5.27, 5.28, 5.29 and 5.27.

VIO-based Navigation and terminal phase results are given at the Figures 5.31, 5.32, 5.33 and 5.31.

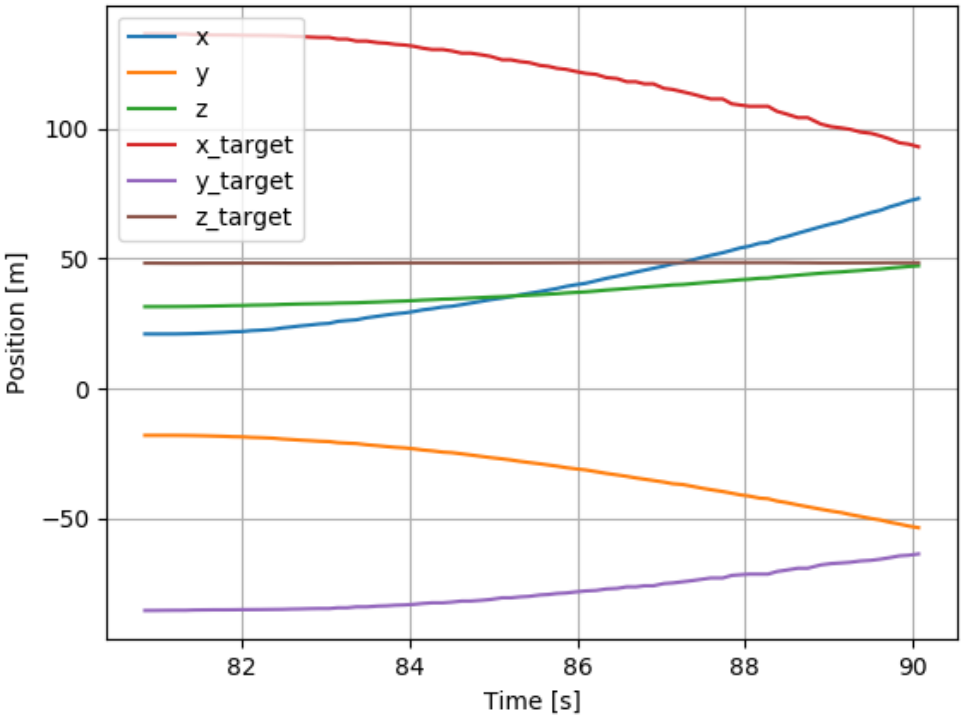

Figure 5.28 : Guidance with GPS Positions

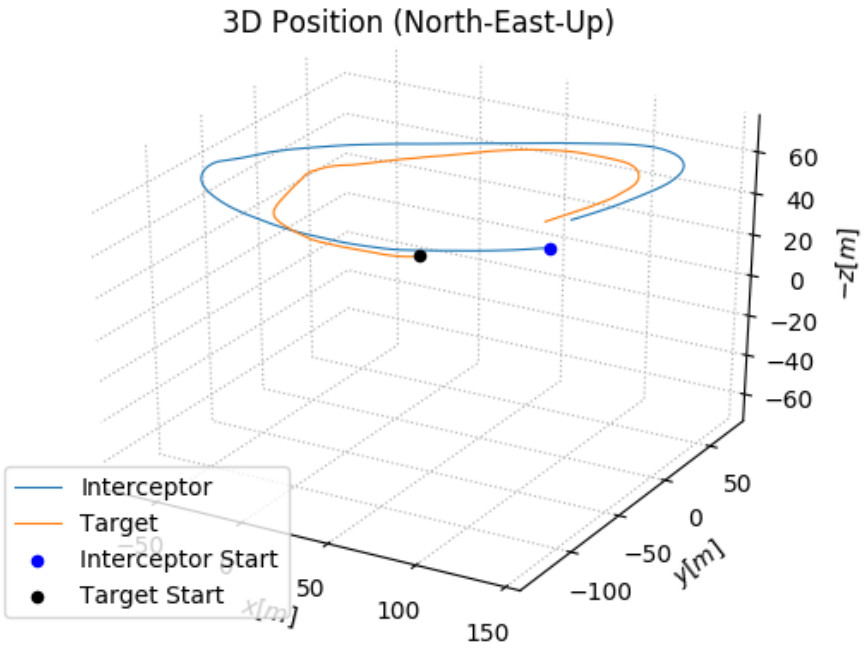

Figure 5.29 : Guidance with GPS 3D Position

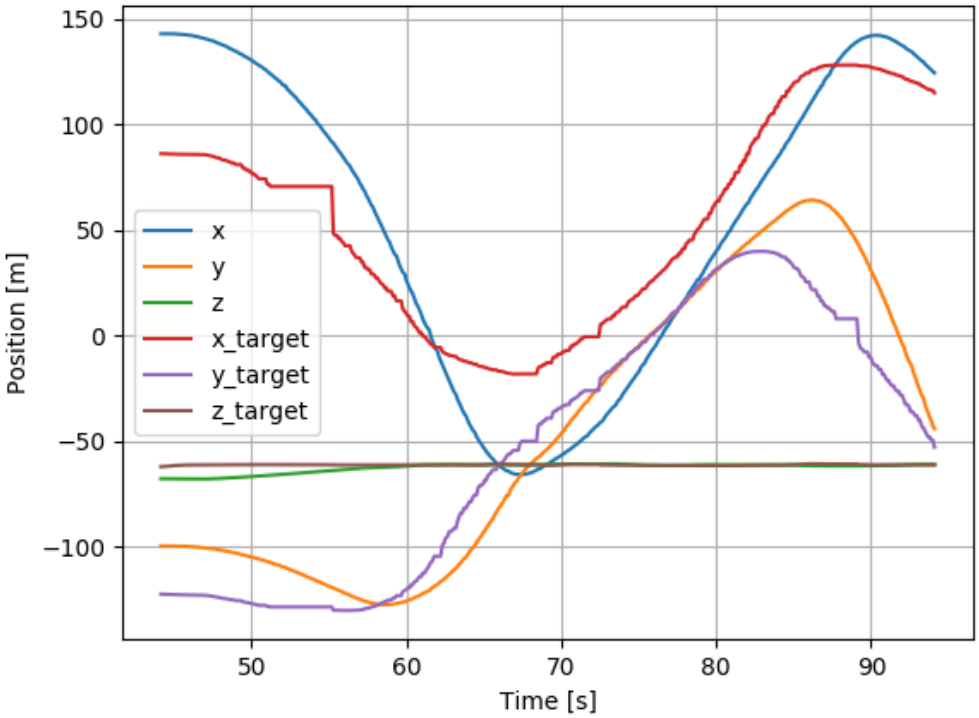

Figure 5.30 : Guidance with GPS Positions

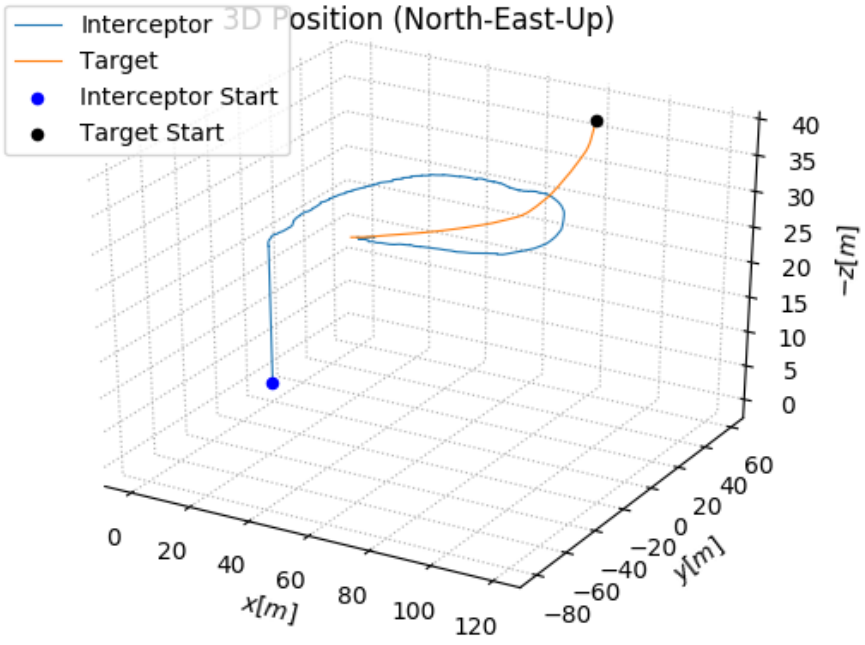

Figure 5.31 : Guidance with VIO 3D Position

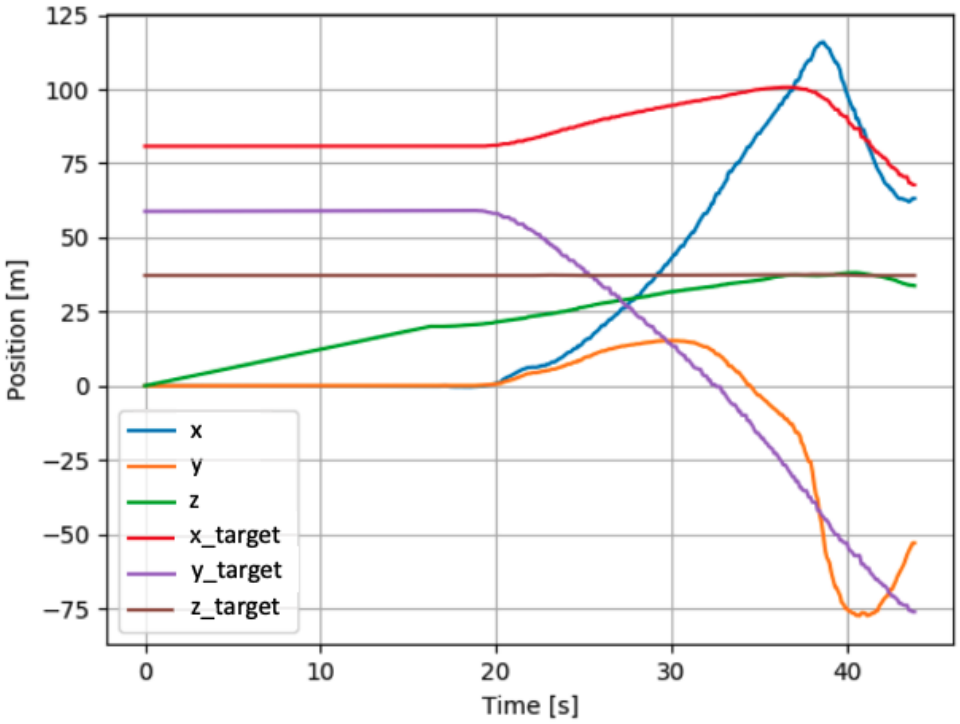

Figure 5.32 : Guidance with VIO Positions

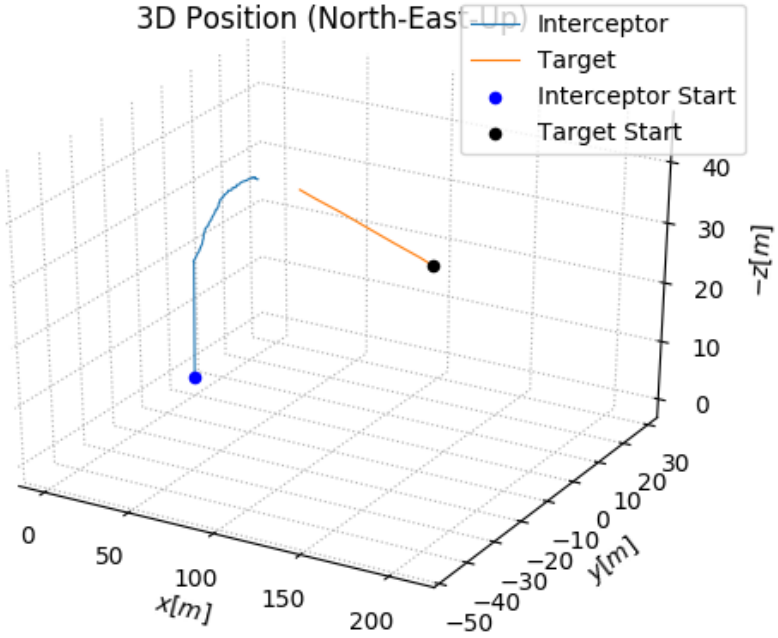

Figure 5.33 : Guidance with VIO 3D Position

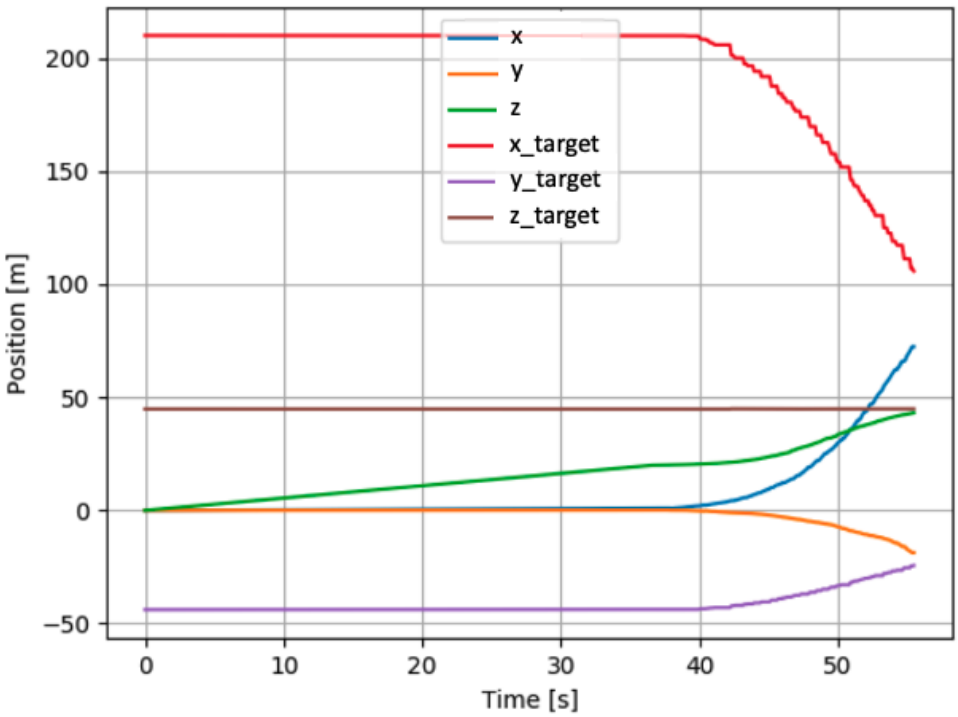

Figure 5.34 : Guidance with VIO Positions

5.3 Discussion

For this study, end-to-end quadrotor impact guidance algorithms are developed.

Firstly, we have proposed a MPC based pre-terminal guidance approach that takes into account the requirements of the visual interception problem. Specifically, the guidance procedure aims to achieve a parallel impact angle, which enables effective visual tracking and robustness to processing latency. To accomplish this, we have utilized Model Predictive Control (MPC) with added terminal constraints to ensure the desired engagement angle is satisfied. The cost input weight matrix in the MPC formulation is used in a time-weighted form to reduce the interceptor's need for maneuvering at the end of the trajectory. Additionally, the time-to-go values are calculated while considering the vehicle's kinematic limits to ensure the feasibility of the problem. Simulation results have demonstrated the effectiveness of the proposed method in intercepting both maneuvering and non-maneuvering targets.

Then, we used Bezier Splines to employ a shrinking horizon planning strategy that incorporates additional terminal constraints to ensure the engagement occurs at the desired angle. Simulation results demonstrate the effectiveness of the proposed approach in intercepting both maneuvering and non-maneuvering targets. The suggested methodology enables parallel engagements. By adjusting the final conditions of the Bezier Curve, the proposed planning based approach minimizes the acceleration input required at the end of the interception trajectory, resulting in smaller miss distances.

For visual guidance part Proportional Navigation based approach is utilized with image feedback. Guidance rule is defined in roll-pitch stabilized frame in order to prevent coupling from camera. Aerial target detection is done by YOLO algorithm and we used Kalman Filter to compansate missing detections.

REFERENCES

[1] **Qin, T.**, **Li, P. and Shen, S.** (2018). Vins-mono: A robust and versatile monocular visual-inertial state estimator, *IEEE Transactions on Robotics*, *34*(4), 1004–1020.

[2] **Moreira, M.**, **Papp, E. and Ventura, R.** (2019). Interception of non-cooperative UAVs, *2019 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, IEEE, pp.120–125.

[3] **Hehn, M. and D'Andrea, R.** (2012). Real-time trajectory generation for interception maneuvers with quadrocopters, *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, pp.4979–4984.

[4] **Mueller, M.W. and D'Andrea, R.** (2013). A model predictive controller for quadrocopter state interception, *2013 European Control Conference (ECC)*, IEEE, pp.1383–1389.

[5] **Srivastava, R.**, **Maity, A.**, **Lima, R. and Das, K.** (2020). Range estimation and visual servoing of a dynamic target using a monocular camera, *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, IEEE, pp.1309–1316.

[6] **Srivastava, R.**, **Lima, R.**, **Das, K. and Maity, A.** (2019). Least square policy iteration for ibvs based dynamic target tracking, *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, IEEE, pp.1089–1098.

[7] **Teuliere, C.**, **Eck, L. and Marchand, E.** (2011). Chasing a moving target from a flying uav, *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, pp.4929–4934.

[8] **Barišić, A.**, **Petric, F. and Bogdan, S.** (2021). Brain over Brawn–Using a Stereo Camera to Detect, Track and Intercept a Faster UAV by Reconstructing Its Trajectory, *arXiv preprint arXiv:2107.00962*.

[9] **Strydom, R.**, **Thurrowgood, S.**, **Denuelle, A. and Srinivasan, M.V.** (2015). Uav guidance: a stereo-based technique for interception of stationary or moving targets, *Towards Autonomous Robotic Systems: 16th Annual Conference, TAROS 2015, Liverpool, UK, September 8-10, 2015, Proceedings 16*, Springer, pp.258–269.

[10] **Srivastava, R.**, **Lima, R. and Das, K.** (2022). Aerial Interception of Non-Cooperative Intruder using Model Predictive Control, *2022 American Control Conference (ACC)*, IEEE, pp.494–499.

[11] **Mourikis, A.I. and Roumeliotis, S.I.** (2007). A multi-state constraint Kalman filter for vision-aided inertial navigation, *Proceedings 2007 IEEE international conference on robotics and automation*, IEEE, pp.3565–3572.

[12] **Leutenegger, S.**, **Furgale, P.**, **Rabaud, V.**, **Chli, M.**, **Konolige, K. and Siegwart, R.** (2013). Keyframe-based visual-inertial slam using nonlinear optimization, *Proceedings of Robotis Science and Systems (RSS) 2013*.

[13] **Bloesch, M.**, **Omari, S.**, **Hutter, M. and Siegwart, R.** (2015). Robust visual inertial odometry using a direct EKF-based approach, *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, IEEE, pp.298–304.

[14] **Forster, C.**, **Zhang, Z.**, **Gassner, M.**, **Werlberger, M. and Scaramuzza, D.** (2016). SVO: Semidirect visual odometry for monocular and multicamera systems, *IEEE Transactions on Robotics*, *33*(2), 249–265.

[15] **Lynen, S.**, **Achtelik, M.W.**, **Weiss, S.**, **Chli, M. and Siegwart, R.** (2013). A robust and modular multi-sensor fusion approach applied to mav navigation, *2013 IEEE/RSJ international conference on intelligent robots and systems*, IEEE, pp.3923–3929.

[16] **Faessler, M.**, **Fontana, F.**, **Forster, C.**, **Mueggler, E.**, **Pizzoli, M. and Scaramuzza, D.** (2016). Autonomous, vision-based flight and live dense 3D mapping with a quadrotor micro aerial vehicle, *Journal of Field Robotics*, *33*(4), 431–450.

[17] **Andersson, J.A.**, **Gillis, J.**, **Horn, G.**, **Rawlings, J.B. and Diehl, M.** (2019). CasADi: a software framework for nonlinear optimization and optimal control, *Mathematical Programming Computation*, *11*(1), 1–36.

[18] **Wright, S.J.** (1997). *Primal-dual interior-point methods*, SIAM.

[19] **ArduPilot Development Team et al.** *ArduPilot*, `https://www.ardupilot.org`.

[20] **Stanford Artificial Intelligence Laboratory et al.** *Robotic Operating System*, `https://www.ros.org`.

[21] **Lorenz Meier et al.** *MAVLink*, `https://mavlink.io/en/`.

CURRICULUM VITAE

Name SURNAME: Ahmet Talha ÇETİN

EDUCATION:

- **B.Sc.:** 2021, Istanbul Technical University, Aeronautical and Astronautical Faculty, Aeronautical Engineering Department

PROFESSIONAL EXPERIENCE AND REWARDS:

- 2021-Present: Research Assistant at Aviation Institute
- 2019-Present: Istanbul Technical University Aerospace Research Center

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- **Çetin A.T.**, Koyuncu E. (2023).Model Predictive Control-Based Guidance with Impact Angle Constraints for Visual Quadrotor Interception. *9th International Conference on Control, Decision and Information Technologies (CoDIT)*, (pp. 1-6). IEEE.
- **Çetin A.T.**, Koyuncu, E. (2024). Visual Pursuit Guidance Strategy with Shrinking Horizon Replanning for Drones. In *AIAA SCITECH 2024 Forum* (p. 0955).

OTHER PUBLICATIONS, PRESENTATIONS AND PATENTS:

- Çatak, A., Çetin, A. T., Koyuncu, E. (2023). Model Uncertainty-aware Adaptive Controller Design with Online Parameter Identification. AIAA SCITECH 2023 Forum.
- Yeniçeri, R., Koyuncu, E., Şenel, S., Paşaoğlu, M. Z., Çetin, A. T., Ösken, İ., ... Toksöz, M. A. (2023, June). A Multi-drone System for Formation Flight and Solo Attack. In 2023 10th International Conference on Recent Advances in Air and Space Technologies (RAST) (pp. 01-06). IEEE.
- Selim, A., Cetin, A. T., Ozkol, I., Koyuncu, E. (2024). Stochastic Trajectory and Robust Controller Optimization via Contractive Optimal Control. In AIAA SCITECH 2024 Forum (p. 2069).
- Selim, A., Cetin, A. T., Ozkol, I., Koyuncu, E. (2024). Stochastic Optimal Control under Non-Gaussian Uncertainties via Entropy Minimization and Dynamical Indicators. In AIAA SCITECH 2024 Forum (p. 2072).
- Sedaki, H., Koyuncu, E., Karabeyoglu, M. A., Gurer, A., Cetin, A. T. (2024). Cold Gas Thruster and Controller Development for Satellite Attitude Control. In AIAA SCITECH 2024 Forum (p. 1256).