University of Nevada, Reno

Uncertainty Quantification-Based Switching Control Method for Vision-Based Object Tracking in Unmanned Aerial Vehicles

A thesis submitted in partial fulfillment of the
requirements for the degree of Master of Science in
Mechanical Engineering

by

Antonio Fernández Castaño

Dr. Petros Voulgaris – Thesis Advisor

May, 2024

THE GRADUATE SCHOOL

We recommend that the thesis
prepared under our supervision by

Antonio Fernández Castaño

entitled

Uncertainty Quantification-Based Switching Control Method for Vision-Based Object Tracking in Unmanned Aerial Vehicles

be accepted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

Petros Voulgaris, Ph. D.
Advisor

Matteo Aureli, Ph. D.
Committee Member

Christos Papachristos, Ph. D.
Graduate School Representative

Markus Kemmelmeier, Ph.D., Dean
Graduate School

May, 2024

ABSTRACT

Ensuring safe aerial robotics has become paramount with the increasing utilization of UAVs (Unmanned Aerial Vehicles) in dense urban environments. Machine Learning-based object detection methods that process camera sensor images have become essential in planning and navigating UAVs through these environments. However, like any machine learning tool, the object detection model is prone to aleatoric and epistemic uncertainties. This thesis introduces a perception-control framework incorporating a switching controller to improve robustness against uncertain measurements processed by an object detector. The proposed mechanism enables switching between different controllers based on the level of uncertainty quantified. This approach aims to enhance the reliability and safety of UAV operations in challenging environments. We tested this framework using a simulated environment where a UAV tracks an object using the YOLOv5 object detector. Our results demonstrate that the proposed control framework can mitigate the uncertainties of the machine learning perception by decreasing the system gain from the sensor noise to the output of the vehicle dynamics.

DEDICATION

Dedicated to my father Antonio Fernández Monterrubio, my mother María Dolores Castaño Sánchez, and my brother Álvaro Fernández Castaño

ACKNOWLEDGEMENTS

I would like to thank many people who have advised, guided, supported, and inspired me to conclude this thesis. First, I would like to express my sincere gratitude to my advisor, Dr. Petros Voulgaris. Dr. Voulgaris' support and guidance were essential and encouraging throughout my research and coursework completion. His academic experience and charisma made a difference in the way that I developed myself through my research experience.

Secondly, I would like to thank Dr. Hyung-Jin Yoon for his valuable guidance and thoughtful feedback which has played an important role in this research. Dr. Yoon also exposed me to different areas of control theory with direct industry applications of my interest throughout our time together at UNR. Doubtlessly, he has been a pivotal person in my academic development. Thirdly, I would like to thank Dr. Matteo Aureli for all the knowledge that he has shared with me inside and outside of the classroom. I would also like to thank Christos Papachristos for his feedback as a valuable member of my thesis committee.

Next, I would like to thank my lab partners at the Learning Autonomy and Control Systems Lab, Caleb and Jessica. I am grateful to have shared this journey with them as well as having the chance to learn from their areas of expertise.

I extend a special thank you to my family, partner, and friends for their unwavering support, understanding, and care. They have been a constant source of inspiration throughout my academic career and personal life.

This work is supported by the National Aeronautics and Space Administration (NASA) under the University Leadership Initiative (ULI) grant 80NSSC22M0070 and the National Science Foundation (NSF) under the AWD-01-00003618: Collaborative Research: Towards Attack-Resilient Vision grant number GR15325.

The views, opinions, findings, and conclusions reflected in this work are solely those of the authors and do not represent the official policy or position of NASA or NSF.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 Overview

The history of Unmanned Aerial Vehicles (UAVs) has been closely tied to military applications in the first stages of their invention. The pioneer pilotless vehicles were developed for the First World War and afterwards deployed during the Vietnam War. Notwithstanding, these aerial vehicles were inspired by Leonardo da Vinci's aerial screw which dated in the late 1480s.

At present, UAVs are used for various purposes including search and rescue, military reconnaissance, scientific research, and even commercial applications like surveying, photography, or inspections [1]. Companies such as Uber and Aurora are pursuing research projects on UAVs transporting passenger and cargo delivery. Others such as Lockheed Martin remain concentrating their efforts on military applications, particularly in developing UAVs ability to fly with military-grade payloads as well as providing thermal infrared capabilities among other features to their camera systems. Likewise, there exist numerous companies that merely focus on drone technology in a certain industry, such as Equinox's Drones which specializes in agriculture.

In recent decades, the exponential growth in its market size is a clear indicator of the increasing popularity of Unmanned Aerial Vehicles (UAVs). According to SNS Insider

Research, this market was valued at US$ 32.17 Bn in 2022s and is projected to reach US$ 82.55 Bn by 2030 [2].

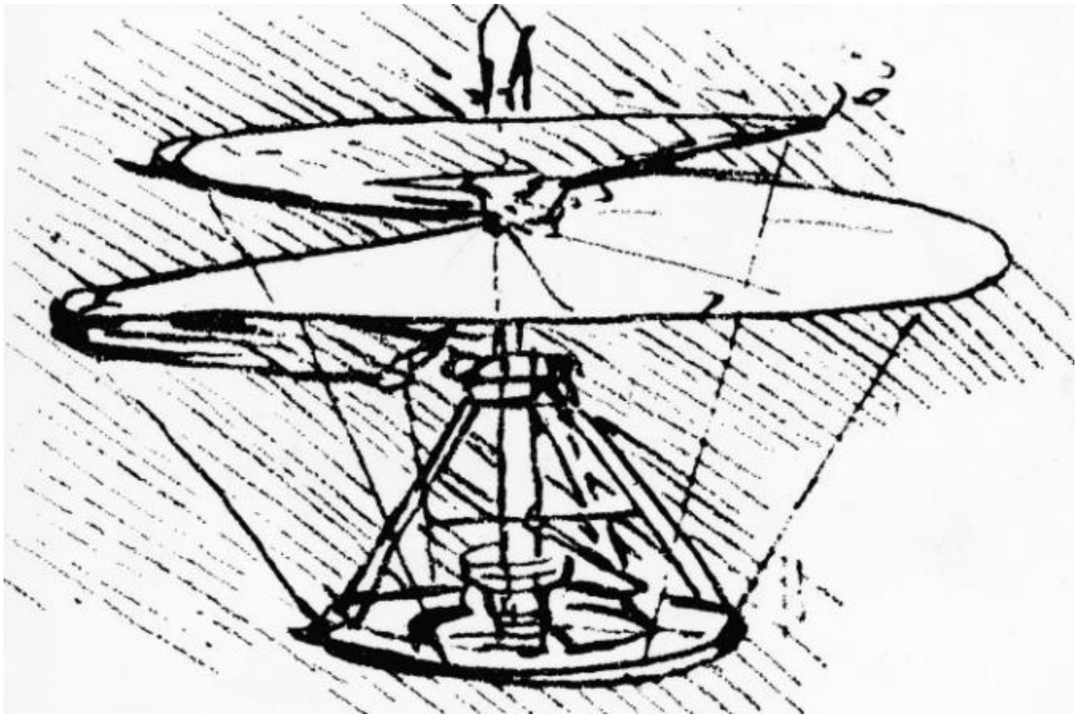


Figure 1. Leonardo da Vinci's "aerial screw" [3]

The rapid and substantial growth in the market size of Unmanned Aerial Vehicles (UAVs) in recent decades is linked to their evolving applications, with one notable trend being their utilization in last-mile package delivery within urban environments. Project initiatives from industry leaders are directed toward establishing safe and efficient delivery services that can seamlessly operate within the challenges of dense urban settings. UAVs emerge as a swift and flexible solution, proficient at navigating through congested cities to ensure the expedited and timely delivery of packages. An integral aspect of this application involves leveraging cameras and employing autonomous decision-making, guided by the uncertainties captured by these cameras. Ensuring the safety of UAV flights in the light of extreme weather conditions, unforeseen pedestrian movements, or aerial traffic demands the development of sophisticated Machine Learning algorithms. These algorithms play a pivotal role in controlling the behavior of UAVs based on real-time camera data, contributing to the overall reliability and success of last-mile delivery services in urban landscapes.

Figure 2. Interconnected Drones by Computer Networks in Dense Urban Environment [4]

Our work aims to convey two objectives, the design of a controller capable of switching between two designs based on noise rejection and performance as well as an uncertainty quantification method integral to the switching process and correlated with the perception errors.

Consequently, we present an innovative approach by incorporating the integration of switching control and uncertainty quantification into a robust control framework for neural network objectors. Our contributions encompass the introduction of a novel switching robust control framework based on H_∞ synthesis. This framework is designed to elevate UAV performance by adeptly adapting to uncertainty levels within a vision-based object detector utilizing a deep neural network (DNN). Additionally, our exploration involves employing ensemble methods for uncertainty estimation in the vision-based object detector, particularly within virtual reality (VR) simulation environments. Within this VR setting, the object detector navigates through image frames captured in diverse weather conditions, including both clear and adverse scenarios. Subsequently, uncertainty quantification takes place, providing a foundation for efficient controller switching.

1.2 Literature Review

UAVs exhibit several advantages over manned aircraft, including a lower maintenance cost and agility. These vehicles are highly utilized in hazardous and challenging environments. The environments where UAVs are operated to fulfill these purposes often have a high potential for difficult-to-model accurate uncertainty, which is challenging for the safe operation of the aircraft. Consequently, effectively dealing with uncertainties is critical to the operation of autonomous UAVs since the implications of losing control can quickly become severe. To overcome these challenges caused by uncertainties, there exist research works on control methods to improve the robustness of UAVs.

In [5], the authors proposed a robust backstepping controller and a sliding mode controller for tracking a trajectory in a tandem rotor UAV in the presence of wind gusts. Similarly, a robust cascaded PID controller rooted in a parameter space approach (PSA) was introduced to guarantee robustness against uncertainty employing maximum sensitivity and Integral-Time-Absolute-Error for robustness analysis [6]. In [7], the authors present a robust multi-rotor nonlinear controller consisting of an inner nonlinear attitude controller based on dynamic inversion control and Lyapunov's second method to account for inertial uncertainty and disturbances, and an outer feedback position controller to ensure stability. Additionally, UAVs have a limited battery range, and hence disturbances such as wind can lead to a higher energy consumption limiting the UAV's applications among various other complications [8]. A controller based on finite-time Lyapunov stability was validated with a backstepping method for the position and the attitude of a small rotor-craft UAV exposed to bounded uncertainties and disturbances in [9]. Among the robust controller implementations in UAVs, several works use the H_∞ method to design a controller. A nonlinear H_∞ controller to stabilize rotational movements along with a model predictive controller (MPC) were implemented to solve the path following problem for a quad-rotor helicopter [10]. Adlakha et al. [11] executed an iterative learning filter which was formulated into an H_∞ optimal feedback controller design problem. Afterward, a robust filter is obtained by computing an optimization law to ensure the learning convergence. Likewise, an H_∞/S-plane controller in the Laplace domain for UAV longitudinal control is presented in [12], showcasing superior disturbance rejection and faster response compared to H_∞/PD and PID controllers. In [13], the authors experimentally tested the impact of a H_∞ loop sharing controller to stabilize speed, throttle, and yaw control.

Furthermore, there exist works on switching controller gains in UAVs. However, they do not perform the switching based on uncertainty quantification. A stochastic switching design for a quad-rotor UAV is proposed implementing a general observer-based controller framework that unveils the momentary availability of global positioning information to perform the switching technique [14]. Analogously, a switching mode control scheme for the hoovering control of a quad-rotor UAV was executed by alternating between two modes, controlling the three position coordinates, and controlling the three attitude angles respectively [15]. In [16], a communication protocol alongside a switched decentralized controller was put forth to control a nested multi-agent system.

In a like manner, the use of object detectors on UAVs has also been extensively studied. A comprehensive survey focused on Single Object Detection (SOD), Video Object Detection (VID), and Multiple Object Tracking (MOT), suggested that the integration of different sensor modalities in UAVs could enhance robustness and accuracy in object detection and tracking for Remote Sensing (RS) applications [17]. Deep learning-based object detection can also introduce challenges when operating in complex environments. Generally, deep learning detectors provide point estimates at test-time with no associated uncertainty quantification and are prone to trigger false positive detections. This issue can be solved by implementing an ensemble of convolutional neural network (CNN) architectures capable of producing a distributive prediction as opposed to point estimates at test time as in [18].

The perception data obtained from cameras and other sensors contain uncertainty and jeopardizes the vehicle's safe and efficient performance. Several factors such as altitude, camera angle, occlusion, and motion blur contribute to unsafe UAV operations. This effect is compounded by the viewpoint fluctuations in the images due to the dataset being recorded from a top view angle [19]. The authors in [20] addressed this problem by developing a detector that can classify trained and untrained objects through object features through a distance metric method. In [21], a YOLOv3 objected detection method for UAVs is presented to solve small object miss-detection and false detection by optimizing Resblock in Darknet by concatenating two Residual Networks and improving training data classification, anchor box confirmation by k-means, and hard data retaining. Likewise, uncertainty estimation and its corresponding mitigation are analyzed in related works. A method entailing adapting a Conditional Variational Autoencoder (CM-VAE) with a Monte Carlo Dropout (MCD) technique was implemented to approximate posterior distribution over encoder parameters to permit the capture of epistemic uncertainty in the perception encoder [22]. In [23], the authors included an additional branch to the neural network (NN) proposing a framework capable of predicting aleatoric uncertainty in the form of Gaussian distribution. Comparably, a multidisciplinary design optimization (MDO) is proposed to amend a system in which components are either parameterized with models or selected off-the-shelf to mitigate its uncertainty in [24].

1.3 Thesis Organization

This thesis is organized in four chapters. In Chapter 2, the problem statement is introduced as well as the proposed framework. The control method and terms implemented to design the controller such as H_∞ controller synthesis, H_∞ norm, and System Identification toolbox are also presented. Additionally, this chapter explores object detection using cameras and its uncertainty and further describes the operational mechanism of the neural network-based object detection method and clarifies its functioning. Chapter 3 presents the simulation environment, controller design and simulations results. The simulation results for equal weather conditions, varying weather conditions and switching controller are shown as well as their discussion. Finally, Chapter 4 consists of the conclusion and the future work of this project. Likewise, an appendix is included containing the stability analysis of the noise-robust controller and high-performance controller designs.

This work has been adapted from the title Uncertainty Quantification-Based Switching Control Method for Vision-Based Object Tracking in Unmanned Aerial Vehicles published by myself, Caleb Patton, Hyung Jin Yoon and Petros Voulgaris in the AIAA SCITECH 2024 Forum Control Techniques for AAM Autonomy Session [25].

Chapter 2

PROBLEM STATEMENT

2.1 Introduction

This chapter initiates by outlining the problem statement of this work. Our objective is to quantify the uncertainty from the camera perception and implement a switching controller guided by this uncertainty to decrease the system gain from the sensor noise to the output of the vehicle dynamics. To accomplish this, the chapter explores the needed background for the controller design, introducing H_∞ controller synthesis, H_∞ norm, and the System Identification toolbox. Likewise, it presents object detection using cameras and its uncertainty estimation, providing insight into how object detectors function and detailing the quantification of uncertainty from camera perception using neural networks. Finally, it presents the proposed framework designed to address the stated problem synthesizing and integrating the array of concepts introduced throughout this chapter.

2.2 Control System Synthesis

2.2.1 H_∞ Controller Synthesis

A H_∞ controller is a type of controller that can directly enforce frequency domain specifications to the control loop of a given system.

Considering the following LTI system in discrete time, for $t \in \mathbb{R}_{\geq 0}$:

$$\dot{x}(t+1) = Ax(t) + B_w w(t) + B_u u(t), \qquad x(0) = x_0$$

$$z(t) = C_z x(t) + D_{zw} w(t) + D_{zu} u(t)$$

$$y(t) = C_y x(t) + D_{yw} w(t) + D_{yu} u(t),$$

where $w(t) \in \mathbb{C}^{m_1 \times 1}$ is an eogenous disturbance input which contains the reference signal and perhaps other signals such as noise, $u(t) \in \mathbb{C}^{m_2 \times 1}$ is a control input computed by the controller K, $z(t) \in \mathbb{C}^{p_1 \times 1}$ is the designed performance output, and $y(t) \in \mathbb{C}^{p_2 \times 1}$ is the measured output. The values m_1, p_1, m_2, and p_2 represent the number of measurement outputs, control inputs, disturbance inputs, and control inputs respectively.

Likewise, the closed-loop system can be expressed as:

$$\dot{x}(t+1)_{cl} = \begin{bmatrix} A & B_w \\ C_y & A_c \end{bmatrix} \begin{bmatrix} x \\ x_c \end{bmatrix} + \begin{bmatrix} B_u \\ D_{yu} \end{bmatrix} u(t) + \begin{bmatrix} B_w \\ D_{yw} \end{bmatrix} w(t),$$

$$z(t)_{cl} = \begin{bmatrix} C_z & D_{zw} \end{bmatrix} \begin{bmatrix} x \\ x_c \end{bmatrix} + D_{zu} u(t),$$

$$y(t)_{cl} = \begin{bmatrix} C_y & D_{yw} \end{bmatrix} \begin{bmatrix} x \\ x_c \end{bmatrix} + D_{yu} u(t)$$

Thus, the final controller K subsequent to the addition of the new state vector $x_K = \begin{bmatrix} x \\ x_c \end{bmatrix}$ is represented as follows:

$$\dot{x}(t+1) = A_K x_K(t) + B_K u(t)$$

$$y_K(t) = C_K x_K(t) + D_K u(t)$$

where A_K, B_K, C_K, D_K are augmented matrices. Our framework intends to utilize two sets of these augmented matrices depending on the quantified uncertainty of the environment.

We conducted several simulations operating the UAV autonomously and controlling it with a joystick for data collection purposes. Subsequently, system identification from [26] was utilized to obtain the dynamics of the UAV at the point of operation from the collected data. Likewise, the optimal H_∞ optimal controller was designed following the H_∞ optimal controller documentation provided by [27]. Higher noise gain weights were inputted to higher uncertainty conditions and vice versa for low uncertainty conditions for the sake of minimizing the H_∞ norm of the system.

2.2.2 H_∞ Norm

The H_∞ norm is a measure of the gain of a linear time-invariant system. It represents the largest possible gain the system can apply to a signal while meeting a given performance specification. Likewise, it assesses the system's sensitivity to external disturbances and uncertainties. The H_∞ norm is expressed as follows:

$$\|G\|_\infty = \max_{d \neq 0} \frac{\|e\|_2}{\|d\|_2} = \max_{\omega} \left(G(j\omega)\right) : \omega \in \mathcal{R} \tag{1}$$

where $G(j\omega)$ is the transfer function of the system evaluated at frequency ω and σ is the singular value of the matrix $G(j\omega)$. A space of such function is denoted as the so-called Hardy space. The main objective of the designed controller is minimizing the maximum H_∞ norm. Thus, obtaining a low H_∞ norm guarantees bounding the system against uncertainties and the system’s robustness.

We compute the H_∞ norm of the closed-loop system in order to measure and analyze the sensitivity of the system to uncertainty and disturbances after applying our controller design as Section 2.4 Proposed Framework states. Our goal is to minimize the input-output ratio in Equation (1).

2.2.3 System Identification

The System Identification Toolbox from [26], equips users with MATLAB functions, Simulink blocks and a dedicated app, facilitating dynamic system modeling, time-series analysis, and forecasting. Among its various applications, System Identification allows to acquire insights into the dynamic relationships of a measured data to process models, establish transfer functions and state-space models in continues or discrete time, utilizing time or frequency domain data. It offers black-box modeling and grey-box modeling. Likewise, it is a powerful tool to estimate nonlinear dynamics using Hammerstein-Wiener and Nonlinear ARX models with machine learning techniques such as Gaussian Processes (GP) or Support Vector Machines (SVM) [27]. It also permits the user to have the capability to carry out real-time parameters and state estimation. The System Identification Toolbox implements the system algorithm in [28].

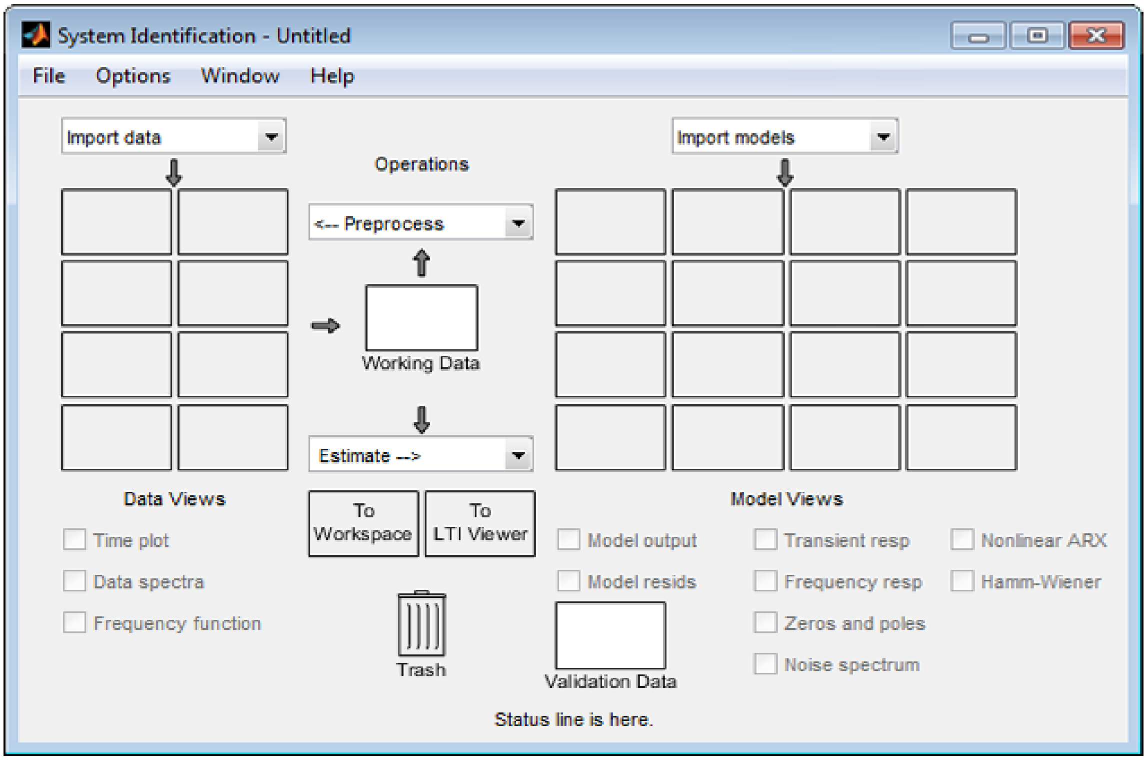


Figure 3. Importing Data Arrays into the System Identification App [27]

2.3 Object Detection Using Camera & its Uncertainty Estimation

2.3.1 Object Detection

To enable our system to comprehend and process the simulated environment, we employ the YOLOv5 object detection model, an enhanced version of YOLO [29] sourced from [30]. Let's provide a concise overview of the YOLOv5 interface. Given an image frame xt with dimensions 448×448 and RGB channels, the detector network, denoted as YOLO(·), transforms the image x_t into three tensor outputs, represented as $y_t = \text{YOLO}(x_t)$, where the output comprises three tensors, namely $y_t = (y_{t,1}, y_{t,2}, y_{t,3})$. These tensors produce grid maps, where each two-dimensional coordinate index is denoted as an anchor point. The indices (anchor points) on the grid map are associated with feature values.

Each tensor output consists of four dimensions, i.e., $y_{t,i} \in \mathbb{R}^{B\times S_i\times S_i\times D}$, where the subscript i denotes the tensor output index, B = 5 represents the number of bounding boxes for each anchor point, S_i signifies the number of anchor points in both horizontal and vertical directions on the rectangular image ($S_1 = 56$, $S_2 = 28$, $S_3 = 14$) and D denotes the number of features linked to the corresponding bounding box at the anchor point. These features include the bounding box coordinates and object classification probabilities [31]. The high-dimensional output empowers YOLO(·) to consider a diverse range of object classes, bounding box shapes, and coordinates. For instance, the outputs $y_{t,1}$ include numerous bounding boxes at $B \times S_1 \times S_1$ with B bounding box templates. This high-dimensional output undergoes filtering through non-max suppression [30], resulting in object detection bounding boxes with increased confidence.

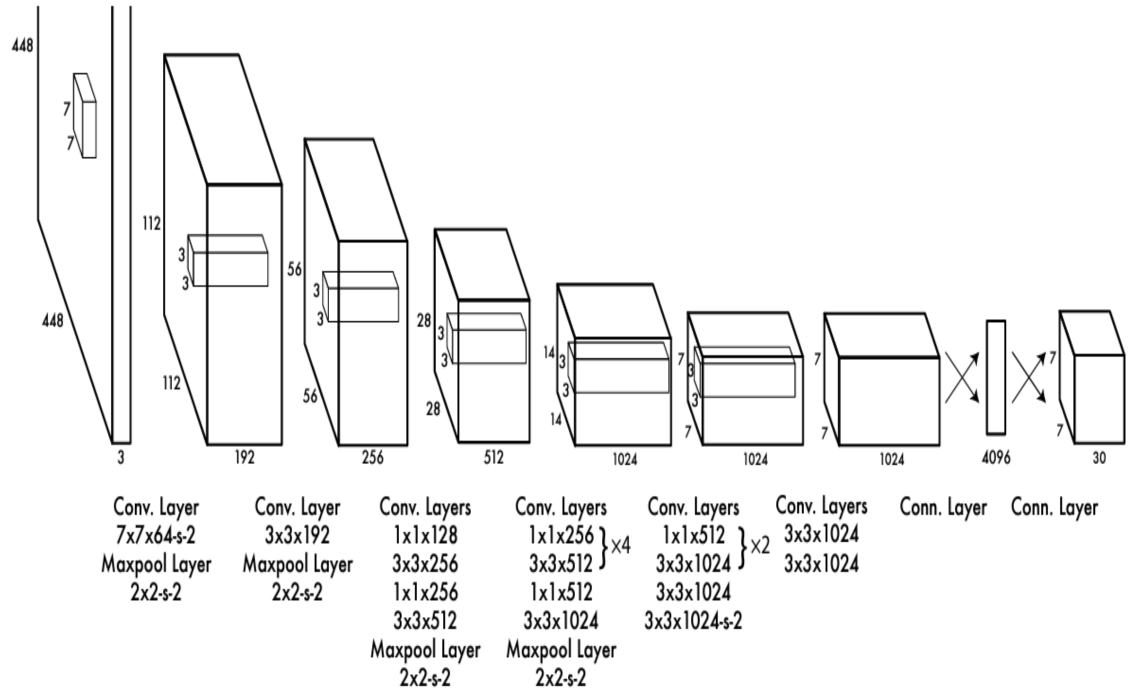


Figure 4. Architecture of YOLO [29]

YOLO outperforms traditional object detector methods such as Faster R-CNN (Region-based Convolutional Neural Network) in terms of efficiency and speed. YOLO is a single-pass object detection algorithm, and hence it processes the entire image in one forward pass tough the neural network. It predicts bounding boxes and class probabilities by segmenting the image into a grid. Similarly, It is able to manage objects of various sizes within a single grid cell more efficiently due to its grid segmentation since it foresees bounding boxes at different scales. In contrast, traditional methods perform this process in two steps by first generating region proposal and subsequently predicting the bounding boxes and class probabilities. Likewise, there exist improved versions of YOLOv5 as illustrated in Figure 5, including advancements such as YOLOv8 which was released in January of 2023.

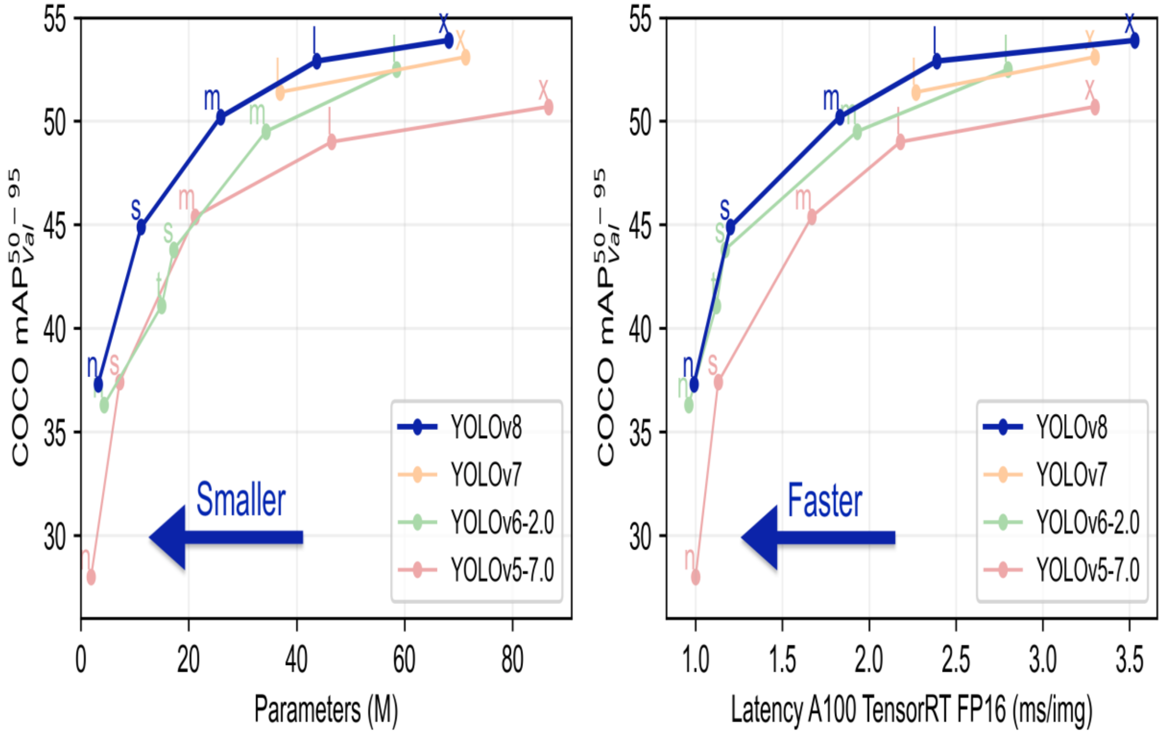


Figure 5. Comparison of the latest YOLO versions as of January 2024 [30]

As illustrated in Figure 5, YOLOv8 demonstrates superior performance compared to its predecessors when assessing the effectiveness of various object detection models. This evaluation considers the influence of different parameter variables or configurations on the models' performance in the COCO dataset. Similarly, it outperforms its previous versions in terms of latency. The utilization of YOLOv5 for our simulator environment stems from its straightforward implementation since it is built on the PyTorch framework as well as its ability to efficiently meet the specific object detector parameters required for this work.

2.3.2 Uncertainty Estimation with Object Detection

Classical Deep Learning Models exhibit an inherent determinism, providing precise predictions based on available training data. Notwithstanding, this deterministic characteristic frequently neglects uncertainties prevalent in real-world scenarios. To address this limitation, uncertainty estimation methods, including Bayesian approaches and Monte Carlo Dropout, are employed within the object detection pipeline. Our proposed framework for uncertainty estimation acquires perspectives from various related works. In [32], the authors introduced a Stereo Multi-State Constraint Kalman Filter (S-MSCKF), offering superior computational efficiency compared to traditional stereo visual-inertial odometry methods. Additionally, [33] proposed uncertainty estimation through probabilistic filtering implementing a Bayesian approach, providing reliable estimates of camera states in a monocular visual-inertial system (VINS). Similarly, [34] implemented a Deep Inference for Covariance Estimation (DICE) approach, utilizing a deep neural network to predict sensor measurement covariance from raw data. In [35], Tsuei et al. highlighted inaccuracies in covariance estimates within an Extended Kalman Filter (EKF) and demonstrated correction through learning a nonlinear map from empirical ground truth to estimated values. Furthermore, [36] presented an ensemble network for object detection in UAV images, aiming to reduce the high false negative rate of multi-stage detectors by integrating a multi-stage method with a single-stage approach.

Our uncertainty estimation with object detection approach employs ensemble methods to estimate uncertainty. Initially, we divided the complete dataset into five subsets, as illustrated in Figure 6. Subsequently, we trained five YOLO networks, each on one of the subsets shown in the figure. With these five fully trained YOLO networks, we can perform five inferences for a new image, which involves object detection. The discrepancy among these five inferences serves as a measure for epistemic uncertainty, specifically uncertainties arising from limited data. The idea behind this uncertainty estimation method is as follows: if a new image closely matches images in the training set, the inference is likely to be accurate. Furthermore, if similar images to the new one are present in all five training subsets from Figure 6, there will be conformity among the five inferences. In contrast, if the new image is rare and similar images are found only in one of the subsets, disagreement among the inferences is expected.

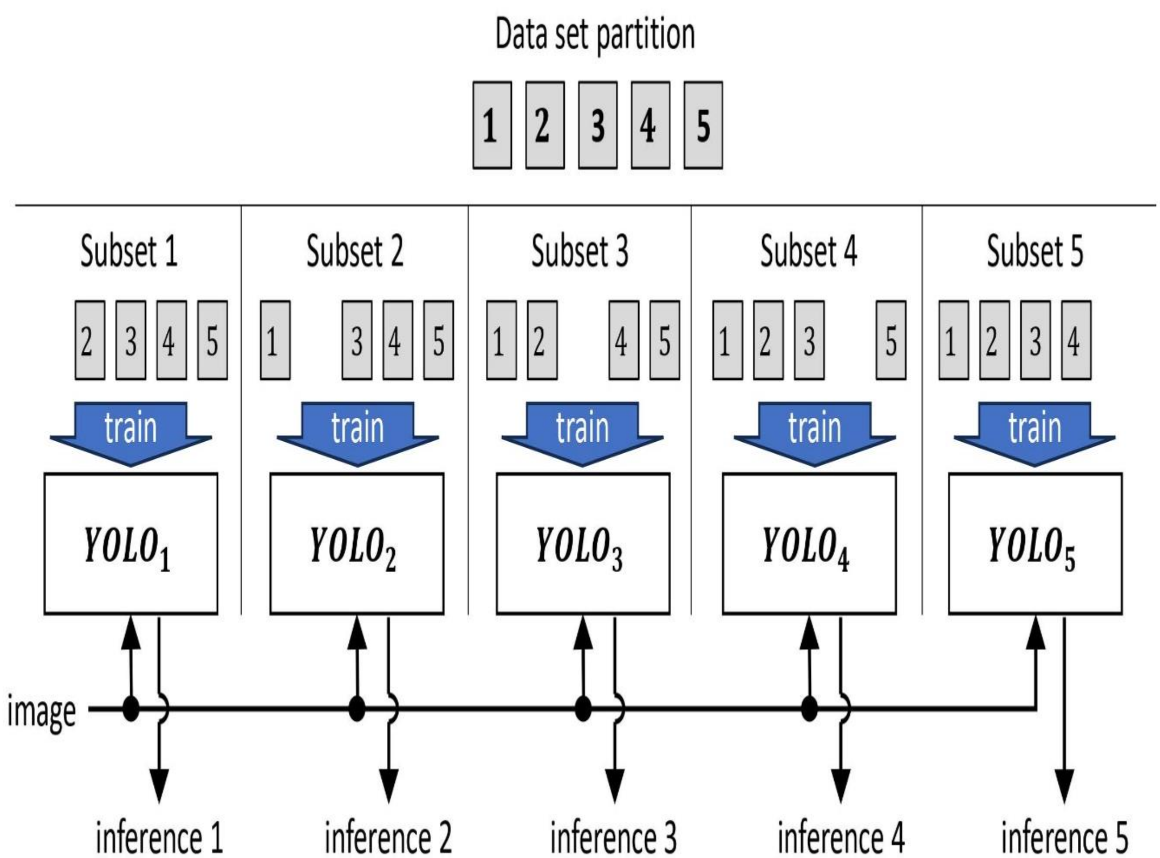


Figure 6. Ensemble Five Method Architecture

In our approach to object tracking, we utilized the predictions from the five YOLO networks by computing their average output. Subsequently, we applied the non-max suppression, as detailed in Section 2.3.1, to the averaged results of the five YOLO networks. Thus, this facilitates the determination of the bounding boxes of the detected objects in our simulations.

2.4 Proposed Framework

The proposed framework strives to adapt to adverse weather conditions by switching to a robust controller from the initial high-performance controller. This decision-making process relies on uncertainty assessments derived from the perception model. The framework consists of two primary elements: (1) a switch connecting a noise-robust controller and a high-performance controller, incorporating uncertainty quantification and (2) uncertainty quantification correlated with perception errors, as illustrated in Figure 7. The subsequent subsections outline the previous two main components.

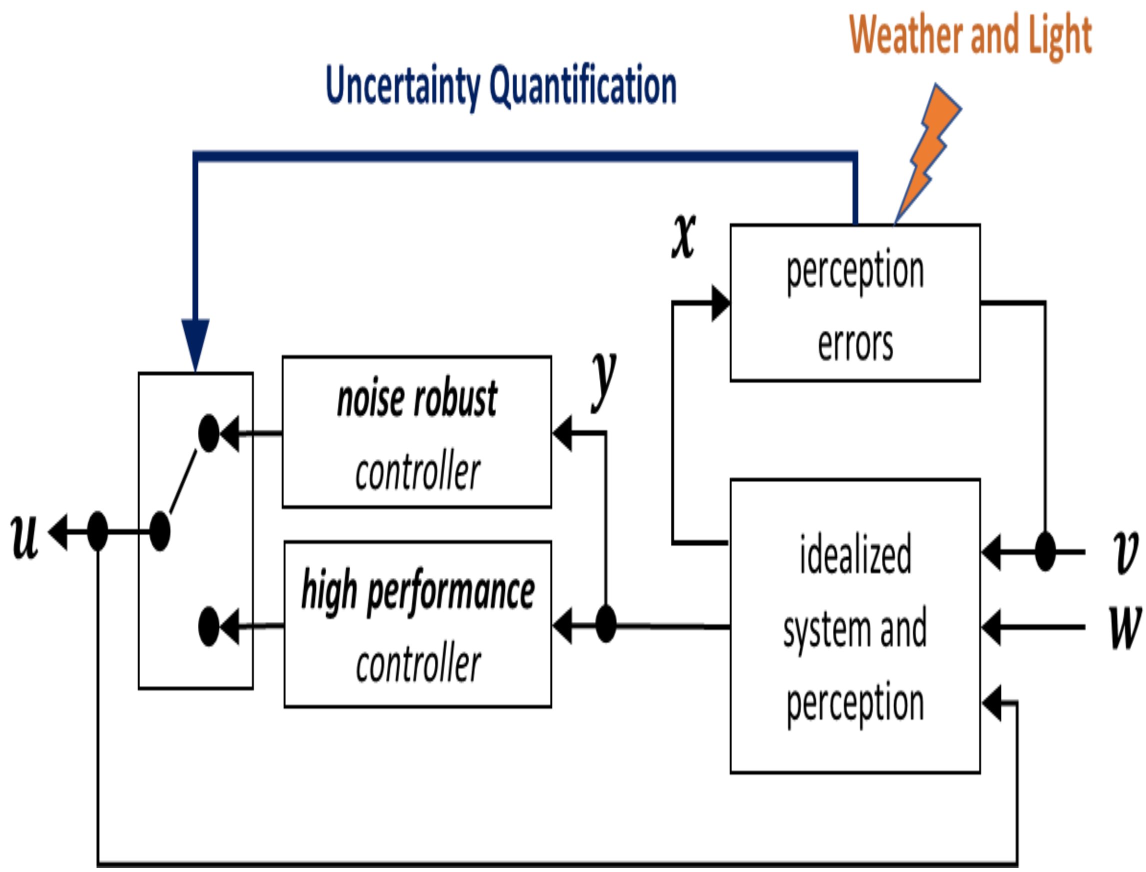


Figure 7. Uncertainty Quantification-Based Switching Control Framework

2.4.1 Controller Design

In formulating the controllers, we take into account a discrete-time linear time-invariant model to address the dynamics of the UAV system, outlined as follows:

$$\begin{aligned} x(k+1) &= Ax(k) + B(u(k) + w(k)) \\ y(k) &= Cx(k) + Du(k) + v(k) \end{aligned} \tag{3}$$

Here, k represents the time step index, w(k) signifies the disturbance at time step k influencing the system, such as factors like wind gusts, while v(k) denotes the sensor noise affecting perception under adverse conditions like intense sunlight, rain, or snow. The discretization of the model is essential as object perception is executed within digital computers.

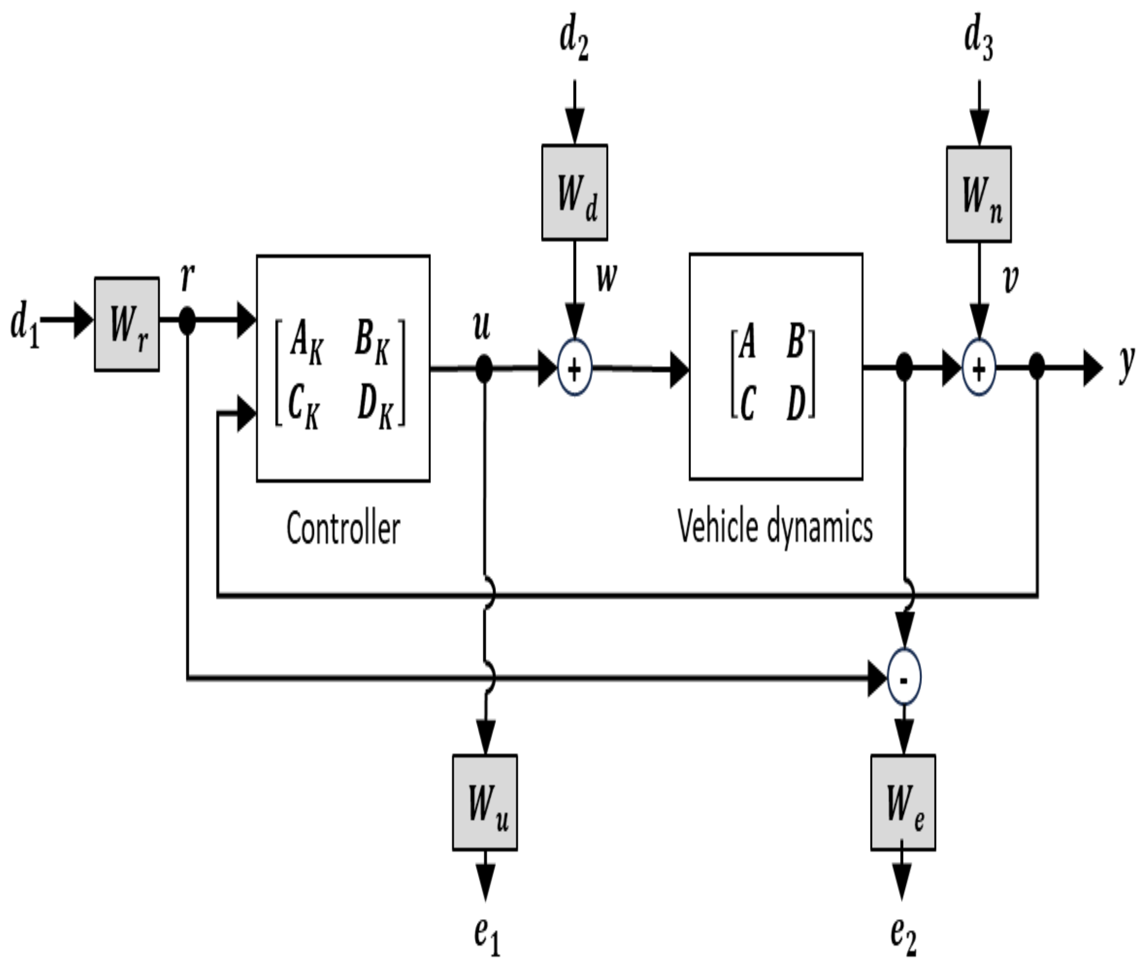


Figure 8. Inputs d = (d_1, d_2, d_3) and outputs e = (e_1, e_2) of the controlled system for H_∞ controller synthesis.

The design of the two controllers involves minimizing the system norm, with inputs represented by d_1, d_2, d_3 and outputs denoted as e_1 and e_2, as illustrated in Figure 8. To implement H_∞ controller synthesis, the system norm is specified in the following manner:

$$\|G\|_\infty = \max_{d \neq 0} \frac{\|e\|_2}{\|d\|_2} \quad (2)$$

where, G signifies the transfer function of the controlled system, and e forms the vector combininge$_1$ and e_2, while d represents the concatenation of d_1, d_2 and d_3. The vehicle dynamics in Equation (3) are illustrated in Figure 8, showcasing its matrices A, B, C, and D. Likewise, the controller synthesis will establish the dynamics of the optimized controller, with its matrices A_K, B_K, C_K, and D_K.

The H_∞ controller synthesis aims to minimize the input-output ratio in Equation (2), expressed as $\frac{\|e\|_2}{\|d\|_2}$. Illustrated in Figure 8, the signal d_1 is assigned the weight W_r and becomes the reference r for optimization using H_∞ controller synthesis. For the disturbance w, the signal d_2 obtains the weight W_r and W_n is assigned for the noise v. The weights allow us to tailor the controller's focus, enhancing performance against either noise or disturbance. Adjusting the weights, such as setting a higher value for W_n than W_d, directs the synthesized controller to prioritize reducing the impact of noise (d_3) on the overall output e over disturbance (d_2). Similarly, modifications to W_u and W_e enable configuring the controller's objective by emphasizing either control effort reduction or error mitigation.

The configuration of the noise robust controller in Figure 7 involves assigning a higher weight to noise, denoted as W_n, for the H_∞ controller synthesis. On the other hand, the high-performance controller is established by implementing a smaller weight to noise, emphasizing a greater focus on disturbance rejection.

2.4.2 Object Detection Using Camera and its Uncertainty Estimation

We implemented the proposed control framework in a basic tracking task, where a UAV moves toward a specified target object using a deep neural network (DNN) for object detection, as shown in Figure 9. In the illustration, the object detector supplies the bounding box delineating the target object. For instance, in Figure 9(b), the object detector associates a green box with the blue car.

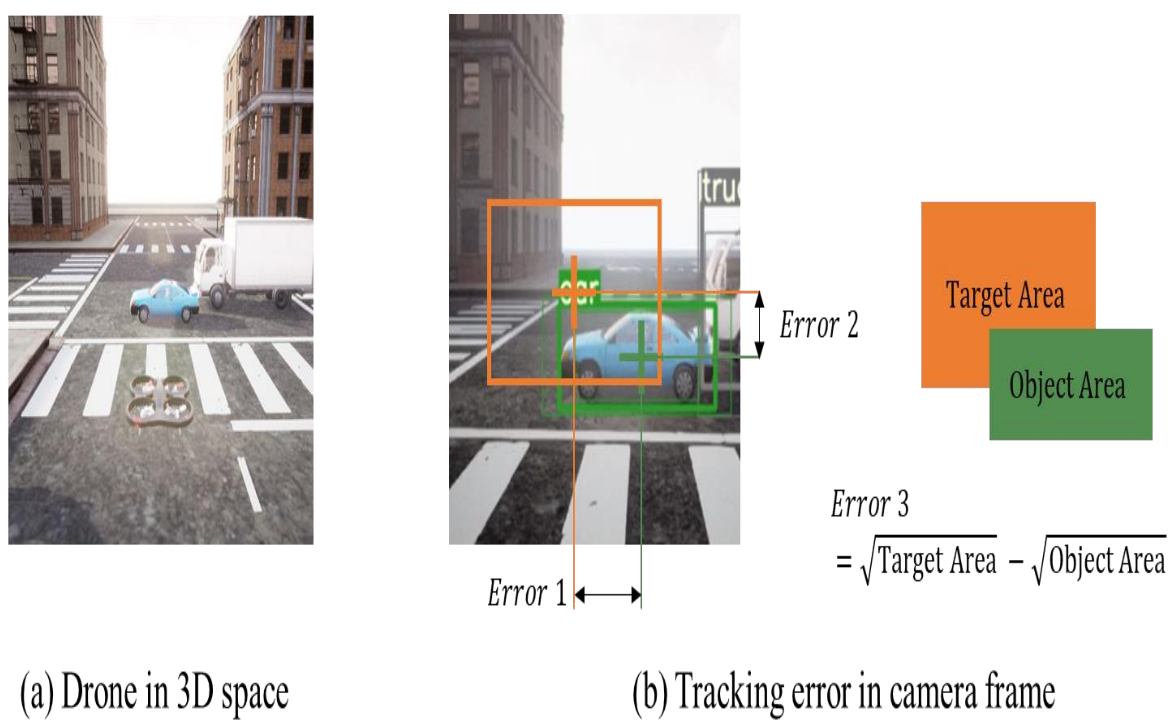


Figure 9. Tracking the object (blue car) using the object bounding box in the camera frame.

One approach to tracking a designated object using a camera frame involves aligning the center of the bounding box with the center of the camera frame. The controller can also maintain the distance to the target by utilizing the size of the bounding box. Therefore, the output vector y in Figure 8 includes both the center coordinates and the size of the bounding box. The reference vector r incorporates the center coordinate of the camera frame and the desired bounding box size. Consequently, the tracking errors illustrated in Figure 3(b), specifically Error 1, Error 2, and Error 3, represent error vectors to be minimized, defined as e_2 in Figure 8. Error 1 and Error 2 measure the spatial misalignment between the center of the bounding boxes in the x and y directions respectively. Error 3 complements these errors by measuring the distance towards the target utilizing the bounding box size. By maintaining the size of the bounding box in the camera frame, the distance to the target object is preserved. Expressing the bounding box size in terms of area and then taking the square root, the error becomes a linear function of the distance to the target since the target distance is expressed as a reference point. This component of the error guarantees that the error remains proportional to the distance to the target, ensuring the measurement's consistency across different object sizes. Likewise, the control signal u in Figure 8 serves as the actuator command, representing the desired velocity vector sent to the UAV's flight controller for the drone shown in Figure 9(a).

Likewise, as introduced in Section 2.3.2 Uncertainty Estimation with Object Detection, the controller in Figure 8 uses the averaged standard deviations of the coordinates obtained from the ensemble methods of the five YOLO networks, focusing on the selected bounding boxes determined by non-max suppression.

(a) Normal (b) Rain (c) Snow

Figure 10. Object Detections using YOLOv5 with different weather conditions

Figure 10 presents the average detection and its associated uncertainty (standard deviation). The disparity between bounding boxes represented by thick lines (average) and thin lines (average plus/minus standard deviation) visually demonstrates the discrepancy among the five YOLO networks. Similarly, Figure 10 illustrates that adverse weather conditions, such as rain or snow, contribute to increased discrepancy attributed to uncertainties.

Chapter 3

APPLICATION OF FRAMEWORK

3.1 Introduction

This chapter first introduces the simulation environment conditions as well as the controller design process. Subsequently, upon the gathering of data to perform system identification, three different experiments are carried out. In the first instance, different weights in the noise channel as tested to identify the noise robust controller and high performance controller respectively. Afterward, both controllers are tested in varying weather conditions to study their impact in terms of error reduction as well as analyze their impact in the uncertainty over time. Finally, the proposed switching idea is presented and justified as the best controller design.

3.2 Experiment Design

We tested the proposed framework in a simulation environment. We developed the 3D virtual reality environments for the simulations, employing AIRSIM plug-in [37]. The vehicle dynamics in AIRSIM are typical quad-copter dynamics that incorporate rigid body motion and actuator delays. The controller commands target velocities, i.e., longitudinal, lateral, and vertical velocity, to keep the target's bounding box centered in the camera view. Consequently, the vehicle moves toward the target object. The vehicle simulation, Ensemble five of the YOLO network, and the switching controller run on a desktop computer with AMD Ryzen 9 7900 CPU, 64GB RAM, and NVIDIA GeForce RTX 4060 GPU. The integration of the simulation, Ensemble five, and controllers were connected with ROS [39] to allow simultaneous computations with the simulation and time-stamped data collection. The sampling time for the simulation was 0.1 seconds.

3.3 Controller Design for Experiment

We used hinfsyn and 4sid in MATLAB control system toolbox [26]. The hinfsyn is an implementation of the standard H_∞ synthesis [39–41]. Likewise, the 4sid implements the system identification algorithm in [28]. We collected data for system identification at a sampling rate of 10 Hertz to use the H_∞ synthesis. The center coordinates, and the bounding box size for the target object detection is the output vector y, and the image frame center coordinates and the desired bounding box size are the reference vector r. The control input, i.e., the UAV's velocity command, was collected as u for the system identification. For the data collection, we had five manual flights toward the target. Each flight takes about 3 to 5 minutes. The A, B, C, and D determined by the system identification have dimensions as 6×6, 6×3, 3×6, and 3×3 respectively. We set the low number state at 6 for the three-dimensional translation motion since the data is low-dimensional for a system with many unmonitored components, including onboard flight controllers that take the velocity command, UAV dynamics, and object detection deep neural network.

Upon the determination of the A, B, C, and D matrices by 4sid, we used hinfsyn to determine the controller dynamics, i.e., A_K, B_K, C_K, and D_K that minimizes the H_∞ norm.

The A_K, B_K, C_K, and D_K determined by the controller synthesis have dimensions as 12×12, 12×6, 3×12, and 3×6 respectively. As shown in Figure 8 in Section 2.4 Proposed Framework, the controller uses both the reference r and the output y. As a result, B_K and D_K have six columns.

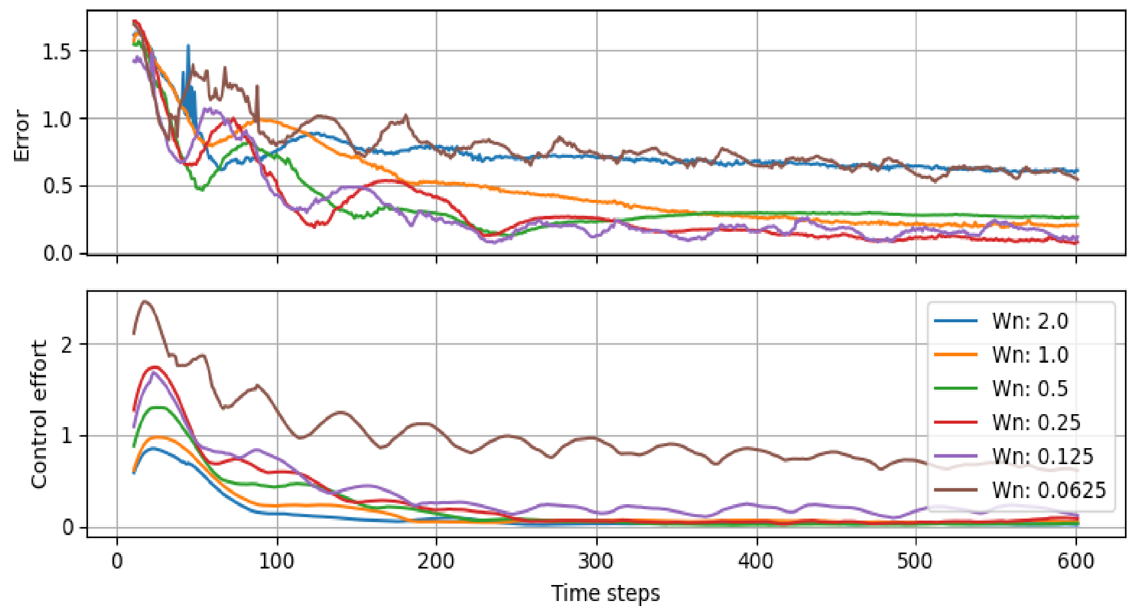
Regarding the weights on the input d and the output e in Figure 8, we used fixed weights, but the noise weight, i.e., W_n as follows. For the tracking error, we used weight varying over frequency, i.e., W_e has dc-gain at 2.5 and high-frequency gain at 0.01. The weight focuses on the steady-state error because the sampling rate is at 10 Hz. The same weights are applied to the three error components (horizontal, vertical, and distance). We used constant weights as $W_r = 1$, $Wd = 0.1$, and $W_u = 0.01$ for the other weights. Finally, for the noise weight, W_n, we tried different values in the set, {2, 1, 0.5, 0.25, 0.125, 0.0625}.

3.4 Results

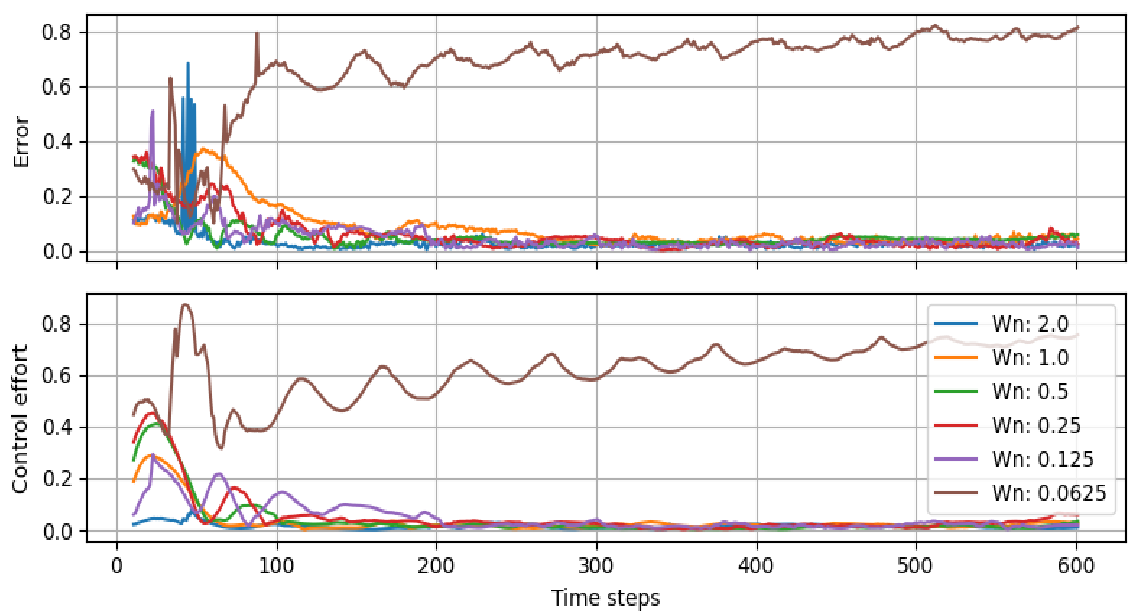
3.4.1 Determination of the Noise-Robust and High-Performance Controllers in Clear Weather Conditions

We expect the controller with the smallest W_n to perform well in good clear weather conditions since it has the least consideration in noise. Furthermore, it should be noted that it is likely that the controller with the greatest W_n would move slowly but be the most robust against the noise. For the purpose of validating these hypotheses, we ran five flight simulations with random initial positions where the target is still in the camera frame.

In the first instance, we tested the six controllers determined by H_∞ synthesis using the noise weight W_n in {2, 1, 0.5, 0.25, 0.125, 0.0625} on clear weather condition. We plotted both the average of the trajectories in Figure 11a and the standard deviation of the trajectories in Figure 11b to assess the variance of the trajectories given different noise weights. In Figure 11a and 11b, Error is calculated as the 2-norm of the error vector of [Error 1, Error 2, Error 3] shown in Figure 9 in Section 2.4 Proposed Framework. Control effort is calculated as the 2-norm of the vector of the three-dimensional velocity command, i.e., longitudinal, lateral, and vertical velocity. The figure shows that the trajectory with $W_n = 2.0$ (blue) has the large steady-state Error with the least control effort. As W_n decreases from $W_n = 2.0$ to $W_n = 0.0625$, the steady-state error decreases until $W_n = 0.25$. After $W_n = 0.25$, the Error increases, which is against our expectation. We expected the greatest performance with the lowest W_n . A factor contributing to this phenomenon is that there is some noise in the perception, even with the clear weather, and hence the controller with low W_n amplifies the effect of the noise. As a result, on average, the controllers with $W_n = 2.0$ and $W_n = 0.0625$ have similar steady-state error performance.



(a) Average of the trajectories



(b) Standard deviations of the trajectories

Figure 11. Performance of H_∞ controllers having different W_n in clear weather

However, in Figure 11b, it can be observed that the standard deviation with $W_n = 0.0625$ diverges in contrast to a low standard deviation with $W_n = 2.0$. The increase in standard deviation suggests that the controller with $W_n = 0.0625$ is not stabilizing. From the tests with controllers having different values of W_n, we selected the one with $W_n = 1.0$ as noise-robust controller and the one with $W_n = 0.25$ as high-performance controller. The high-performance controller has the best error performance in clear weather conditions, as in Figure 11a. Likewise, noise-robust controller has $W_n = 1.0$ that is four times greater than high-performance's W_n. The noise-robust controller had reasonable tracking performance, as shown in Figure 11a.

3.4.2 Performance of the Two Controllers in Varying Weather Conditions

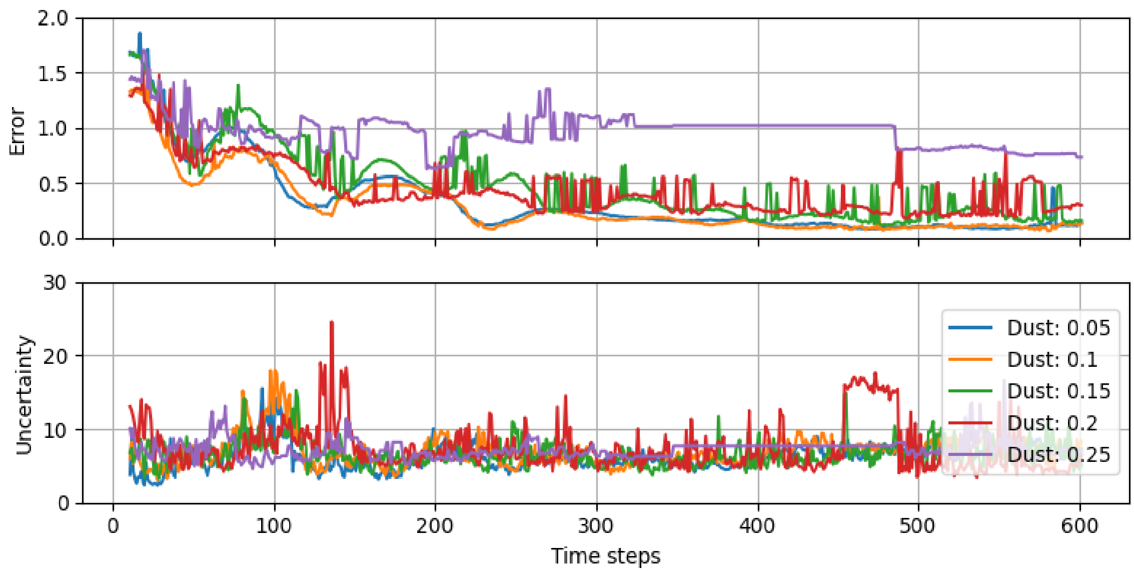
We tested the tracking performance of high-performance controller and noise-robust controller under different weather conditions. The AIRSIM [37] environment contains a weather setting function that has Rain, Snow, MapleLeaf, Dust, and Fog parameters. For our experiment, we varied the Dust level from 0 to 25%, as shown in Figure 12.

As Dust level increases the YOLO object detector has increasing difficulty in making correct predictions. In Figure 12 (c), (d), and (e), the detector could not classify the object correctly. Furthermore, in Figure 12 (d) and (e), it detects objects that do not exist in the camera view.

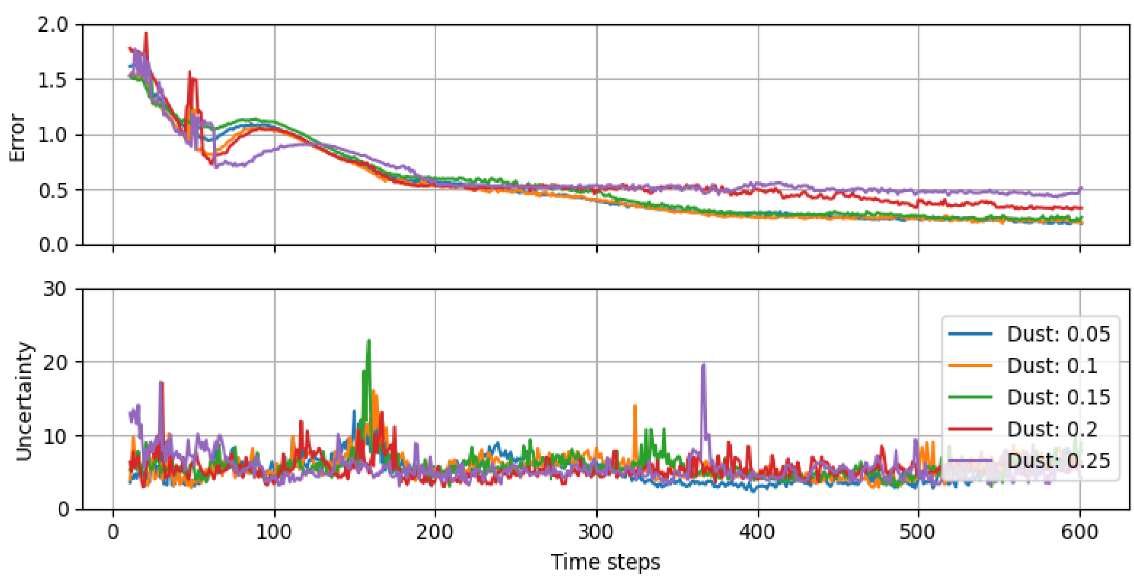
(a) *Dust*: 5% (b) *Dust*: 10% (c) *Dust*: 15% (d) *Dust*: 20% (e) *Dust*: 25%

Figure 12. Varying dust levels of the environment

Likewise, average performances (over five random trajectories) of high-performance and noise-robust controllers in different Dust levels are shown in Figure 13. With low dust levels, the high-performance has the best steady-state error performance. However, as the weather condition becomes more adverse, the error also increases with the high-performance controller. In contrast, noise-robust controller has better performance than high-performance controller in adverse weather conditions, as illustrated in Figure 13.



(a) High-performance controller (W_n = 0.25)



(b) Noise-robust controller (W_n = 1.0)

Figure 13. Average performance of high-performance controller and noise robust controller with varying Dust levels

Comparing Figure 13(a) and (b), it can be noted that the uncertainty level with noise-robust controller is slightly less than with the high-performance controller. Notwithstanding, the difference is not significant. In the next section, it will be analyzed whether the difference is statistically significant.

3.4.3 Switching Controller Given Uncertainty Estimate

It is not trivial to make the best decision on switching between the high-performance controller and noise-robust controller. For proof of concept, we employed a simple switching procedure. First, the initial controller is always the high-performance controller. Likewise, there is a threshold value of the uncertainty level. When the uncertainty level exceeds the threshold, the controller is switched to noise-robust controller. Once the controller is switched to noise-robust, switching back to high-performance controller is not allowed until the next take-off.

The uncertainty of the object detector depends not only on the weather conditions but also on the occlusion, speed, and acceleration of the vehicle. Therefore, the uncertainty level would fluctuate over time in the trajectories as the UAV approaches the target. As shown in Figure 13, the average uncertainty value still varies over time. Thus, we applied a first-order smoothing filter to the uncertainty value to use the threshold switch. The smoothing filter is as follows:

$$y(k+1) = 0.9\, y(k) + 0.1\, u(k) \tag{4}$$

We explored the different values to find a suitable threshold value for the switch. In Figure 13, the Uncertainty value ranges between 0 and 30. Therefore, we tried 5, 10, 15, and 20 threshold values. To evaluate the performance of the switching threshold, we ran 25 random flights for each threshold value. Each flight is initialized with a random initial position and a random dust level. The dust level ranges between 0 and 25%. The data exploration is summarized in Table 1.

Table 1. Performance of the controllers with different switching thresholds (n=25)

Threshold	Controller engaged		Terminal Error	Detection	Energy	Uncertainty	Dust Level
	Perform.	Robust	(avg ± stdev)	(avg)	(avg)	(avg ± stdev)	(avg ± stdev)
0	0%	100%	0.301 ± 0.138	82%	0.168	5.284 ± 0.528	12% ± 8.0%
5	2%	98%	0.366 ± 0.186	82%	0.200	5.523 ± 0.720	12% ± 9.2%
10	36%	64%	0.301 ± 0.190	85%	0.246	7.063 ± 1.479	11% ± 8.2%
15	74%	26%	0.220 ± 0.138	85%	0.325	7.222 ± 1.738	10% ± 8.4%
20	94%	6%	0.407 ± 0.518	78%	0.454	7.000 ± 1.617	12% ± 9.5%
∞	100%	0%	0.394 ± 0.487	77%	0.450	6.686 ± 1.061	13% ± 9.2%

As it can be noted in the table, as we increase the threshold value from 0 to infinity, noise-robust controller's engagement decreases from 100% to 0%. The control energy (denoted as Energy in the table) increases as high-performance controller engagement increases. Additionally, Detection of the target object (represented as Detection) decreased as high-performance controller engagement increased. This is expected because the high acceleration due to high-performance controller would increase the chance of losing the target object in the camera frame.

We performed Student's t-test [42] to see whether the uncertainty levels of noise-robust (the first row in Table 1) and high-performance (the last row in Table 1) are different. The null hypothesis is that their means are the same. On the other hand, the alternative hypothesis is that their means are not the same. We use the significance level at α = 0.05.

Noise-robust Controller:

$$\overline{x_1} = 5.284$$

$$s_1 = 0.528$$

$$n_1 = 25$$

High-performance Controller:

$$\overline{x_2} = 6.686$$

$$s_2 = 1.061$$

$$n_2 = 25$$

Then,

$$\text{Degrees of freedom: } df = n_1 + n_2 - 2 = 48$$

$$s_{pooled} = \sqrt{\frac{(n_1-1){s_1}^2+(n_2-1){s_2}^2}{df}} \tag{5}$$

$$s_{pooled} \approx 0.83874$$

T-statistic (t):

$$t = \frac{\overline{x_1}-\overline{x_2}}{s_{pooled}\sqrt{\frac{1}{n_1}+\frac{1}{n_2}}} \tag{6}$$

$$t \approx -5.918$$

$$\mathrm{p}_{-value} = 10^{-6}$$

The t-statistic should be negative which indicates a difference in the means. The p-value for the two-tailed test is less than 10^{-6} and hence, we reject the null hypothesis and accept the alternative hypothesis. Consequently, it can be concluded that their means are not the same.

Therefore, noise-robust and high-performance have statistically significant differences in the uncertainty level while operating in the same environment. This suggests that control design can affect the performance of object detection.

Based on the data in Table 1, our choice of the suitable threshold value is 15 since it provides the lowest terminal error and a high detection percentage. It is important to note that the dust levels are equally randomly selected from the set of 0%, 5%, 10%, 15%, 20%, and 25% in each random flight. This decision is motivated by the intention to simulate real-world conditions in our experiments, reflecting the stochastic nature observed in the act of rolling a die. On average, the dust levels for these simulations ranged from 10% to 13%. Subsequently, we test whether the proposed switching controller is better than noise-robust or high-performance in terms of terminal error.

In a similar fashion, we ran two Student's t-test to compare noise-robust versus switching and high-performance versus switching. Likewise, we use the significance level at $\alpha = 0.05$.

Noise-robust Controller:

$$\overline{x_1} = 0.301$$

$$s_1 = 0.138$$

$$n_1 = 25$$

High-performance Controller:

$$\overline{x_2} = 0.394$$

$$s_2 = 0.487$$

$$n_2 = 25$$

Switching Controller:

$$\overline{x_3} = 0.220$$

$$s_3 = 0.138$$

$$n_3 = 25$$

Therefore,

$$\text{Degrees of freedom: } df = n_1 + n_2 - 2 = 48$$

Comparing the noise-robust controller vs the switching controller,

$$s_{pooled} = \sqrt{\frac{(n_1-1){s_1}^2+(n_3-1){s_3}^2}{df}} = 0.1376$$

T-statistic (t):

$$t = \frac{\overline{x_1} - \overline{x_3}}{s_{pooled}\sqrt{\frac{1}{n_1} + \frac{1}{n_3}}} \approx 2.084$$

$$\mathrm{p}_{-value} = 0.0213$$

Likewise, comparing the high-performance controller vs the switching controller,

$$s_{pooled} = \sqrt{\frac{(n_2-1){s_1}^2+(n_3-1){s_3}^2}{df}} = 0.3582$$

T-statistic (t):

$$t = \frac{\overline{x_2} - \overline{x_3}}{s_{pooled}\sqrt{\frac{1}{n_2}+\frac{1}{n_3}}} \approx 1.716$$

$$\mathrm{p}_{-value} = 0.0463$$

The p-values of the one-tailed test for the two cases are 2.13% and 4.63%, respectively. Therefore, we can conclude that they are statistically significant (α = 5%). This suggests that the proposed switching controller performs better than the noise-robust controller and the high-performance controller.

Chapter 4

CONCLUSION AND FUTURE WORK

4.1 Conclusion of Current Work

The overall goal of this work is to present a framework to deal with the uncertainty of the object detector for unmanned aerial vehicles in an adverse environment. To address this challenge, we proposed a switching controller framework that selects noise-robust or high-performance controllers during the initial phase of flight. The two controllers were designed using H infinity synthesis with different weights on the noise channel. Using the random simulations with the virtual reality environment, we show that the noise-robust controller performed better than the high-performance controller in adverse weather conditions. In turn, the high-performance controller performed better in clear weather conditions. This test result motivates us to test the proposed switching control framework. The switching depends on the uncertainty estimate from the object detector. This uncertainty estimate was quantified using the ensemble method. In the random experiment with the virtual reality simulator, we saw that the proposed switching framework performed better than the high-performance and noise-robust controller. Furthermore, the noise-robust controller and the high-performance controller have statistically significant differences in the uncertainty level while operating in the same environment. Thus, this data suggests that the uncertainty level of the object detector can be improved by the controller design. Ultimately, our results demonstrate that the proposed control framework can mitigate the uncertainties of the machine learning perception by decreasing the system gain from the sensor noise to the output of the vehicle dynamics.

4.2 Future Work

Future efforts will involve enhancing the switching rule through the utilization of optimal sequential decision-making tools like dynamic programming and reinforcement learning. For instance, DQN (Deep Q-Network) and DDPG (Deep Deterministic Policy Gradient) are two popular algorithms in the field of reinforcement learning that can be implemented for this future work. One possible preliminary concept of the reward function that can be outlined as follows: running reward can be calculated as the negative of the tracking error and terminal reward can be computed as the negative sum of the terminal tracking error and the count of switches that occurred during a flight. Likewise, unnecessary switching must be avoided since it can make the system unstable.

Through our simulation experiences, we have discerned that uncertainty is influenced not solely by weather conditions but also by flight trajectories. Therefore, there is potential for improvement by extending the decision-making process for switching to consider both the uncertainty level of the object detector and the current state of the vehicle.

APPENDIX A: Stability Analysis

This section explores and analyzes the stability of the noise-robust controller and high-performance controller designs.

Table 2 shows the eigenvalues of the noise-robust controller (W_n = 1) and high-performance controller (W_n = 0.25) respectively.

Table 2. Eigenvalues of the Two Controller Designs

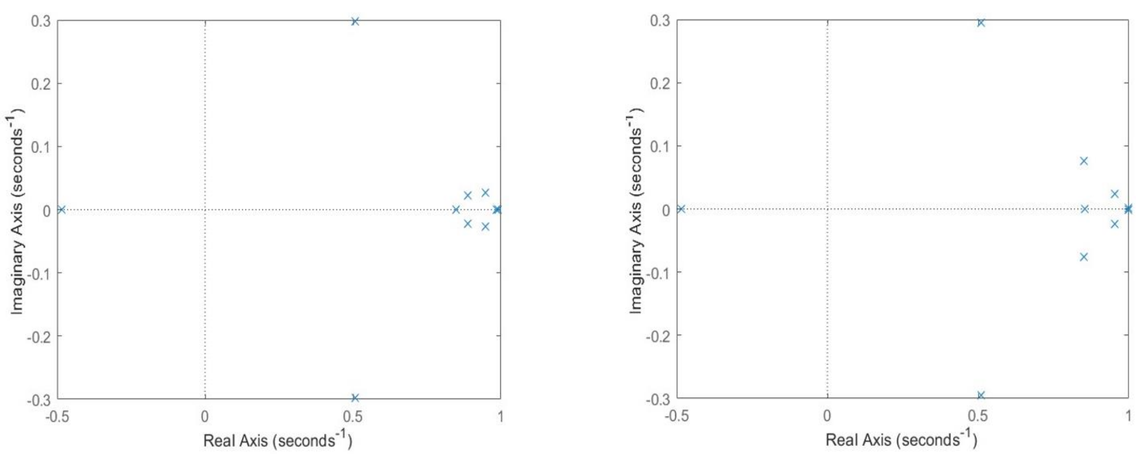
Controller					
Noise-robust			Performance		
Re(z)	Im(z)	Magnitude	Re(z)	Im(z)	Magnitude
0.508	+ 0.2972i	0.5886	0.5106	+ 0.2947i	0.5895
0.508	- 0.2972i	0.5886	0.5106	- 0.2947i	0.5895
0.8473	0.0000i	0.8473	0.8508	+ 0.0759i	0.8542
0.8889	+ 0.0217i	0.8892	0.8508	- 0.0759i	0.8542
0.8889	- 0.0217i	0.8892	0.8528	+ 0.0000i	0.8562
0.9487	+ 0.0275i	0.9491	0.9537	+ 0.0245i	0.9540
0.9487	- 0.0275i	0.9491	0.9537	- 0.0245i	0.9540
0.9902	+ 0.0000i	0.9902	0.9977	+ 0.0014i	0.9977
0.9849	+ 0.0000i	0.9849	0.9977	- 0.0014i	0.9977
-0.4839	+ 0.0000i	0.4839	-0.4839	+ 0.0000i	0.4839
-0.4839	+ 0.0000i	0.4839	-0.4839	+ 0.0000i	0.4839
-0.4839	+ 0.0000i	0.4839	-0.4839	+ 0.0000i	0.4839

It is important to note that the system is modeled in discrete time since the simulations were conducted on digital computers. Therefore, when referring to the stability of a discrete-time system, the eigenvalues should be inside the unit circle in the complex plane.

As it can be observed in Table 2, all the eigenvalues have magnitudes less than 1meaning that they lie inside the unit circle in the complex plane and hence, the system is stable.

This also demonstrates that the controller did not change the stability condition of the system since the system without the controller exhibits complex-conjugate poles at approximately 0.5367 ± 0.2983i, a real pole at 0.8037, and another real pole at 0.9649. Likewise, there is a pair of poles at 0.9990 ± 0.0010i. The system without the H infinity synthesis controller is previously stabilized by the cascaded PID controller that is embedded into the AIRSIM dynamics.

Additionally, Figure 14 shows a graphical representation of the poles of the system's closed loop system for the two respective controller designs.



a) Noise Robust

b) High-Performance

Figure 14. Poles and Zeros of the Closed_Loop System

Figures 15 and 16 illustrated the sigma values and step response plots of the noise-robust controller.

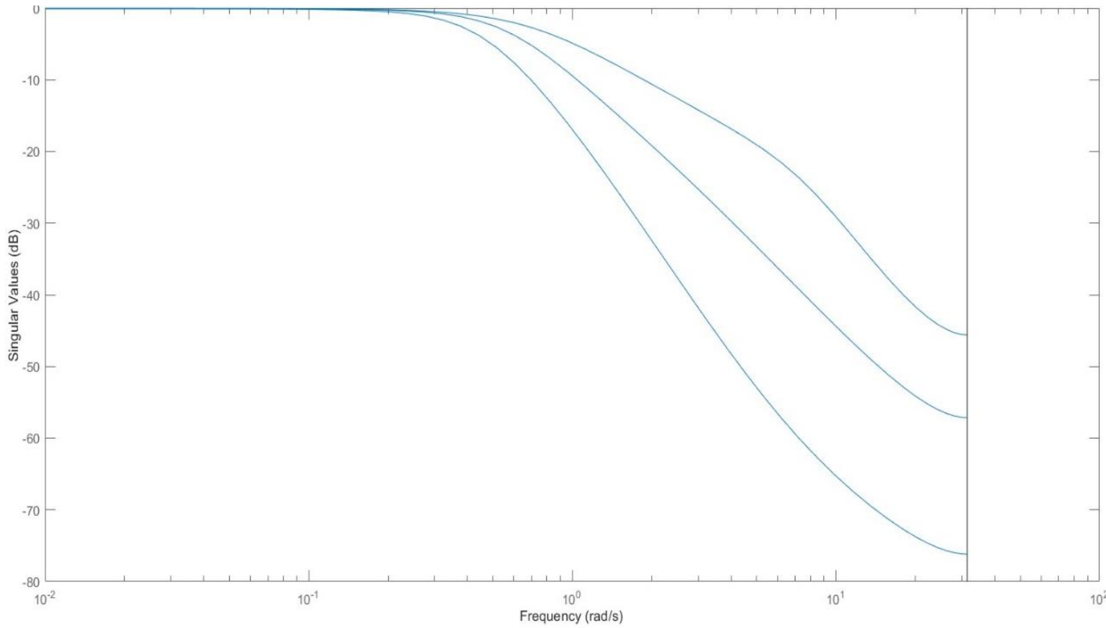


Figure 15. Sigma Values, noise-robust controller

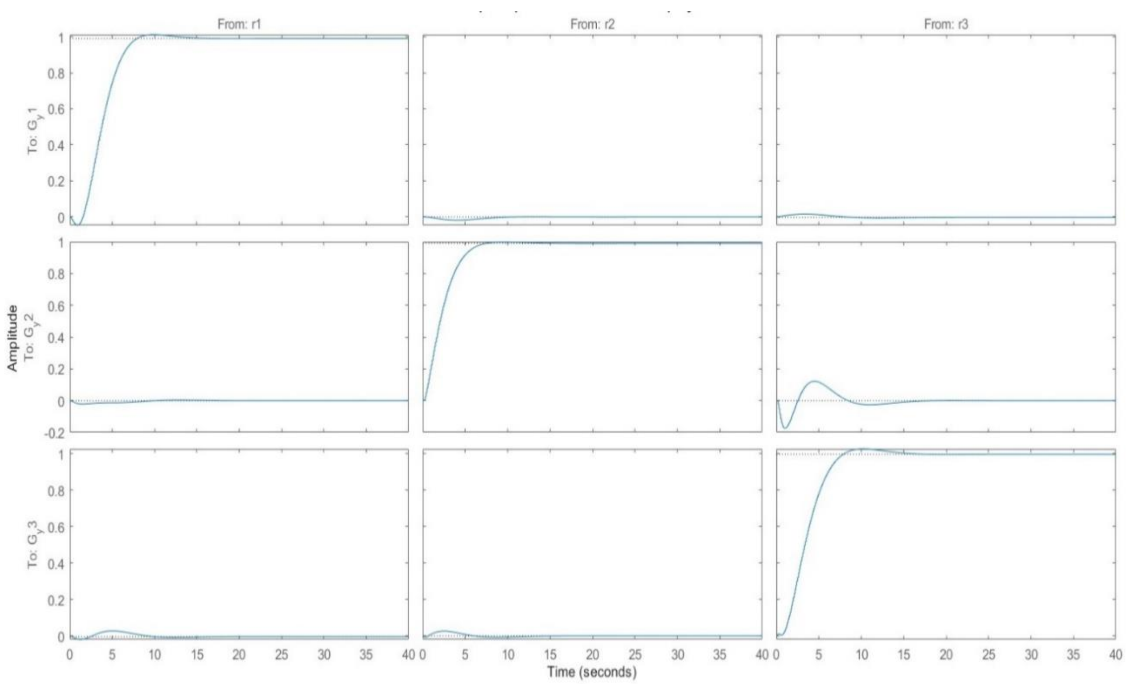


Figure 16. Step Response, noise-robust controller

Similarly, figures 17 and 18 illustrated the sigma values and step response plots of the high-performance controller.

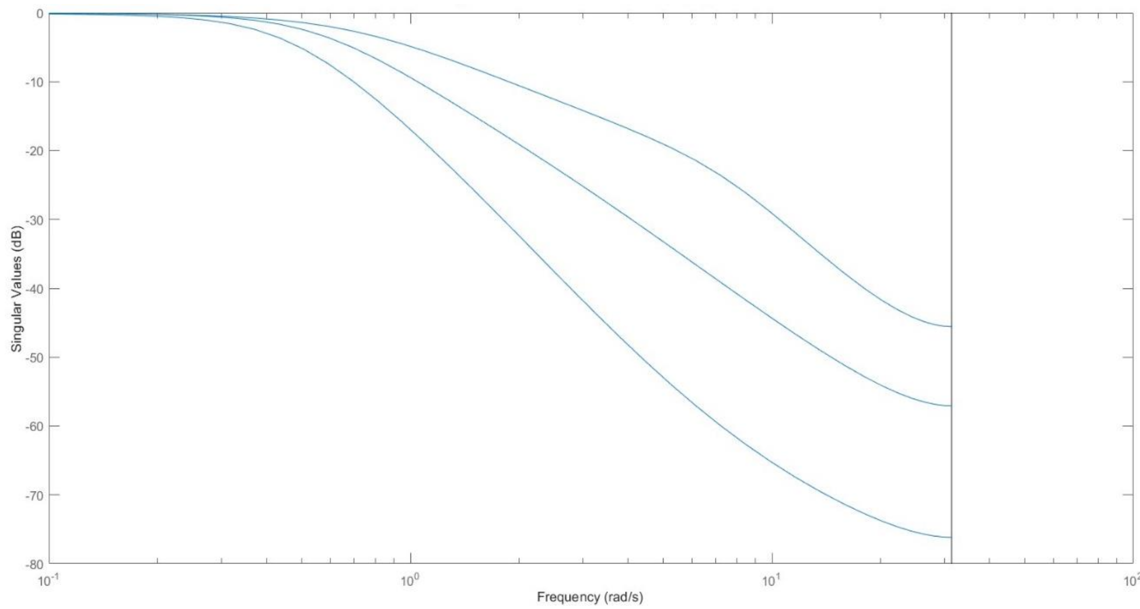


Figure 17. Sigma Values, high-performance controller

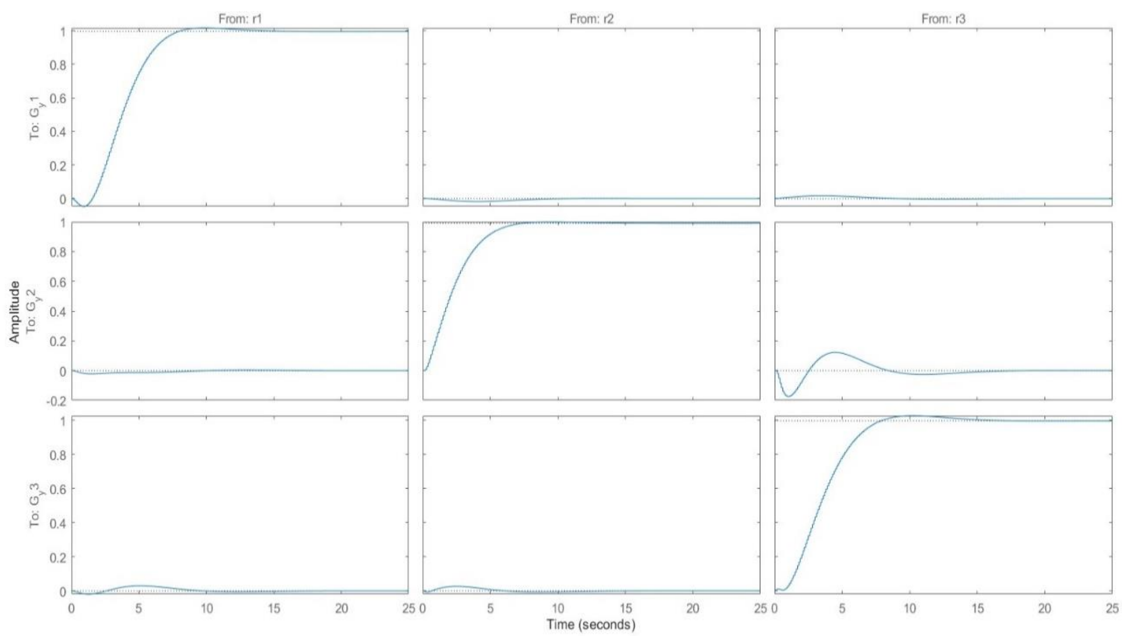


Figure 18. Step Response, high-performance controller

The examination of the sigma values and step response plots for the two controller designs provides valuable insights into the system's behavior. As illustrated in the figures above, both controller configurations demonstrated stable responses to a step input as well as a faster response from the high-performance controller in terms of frequency as the sigma values plots illustrate. Thus, these results corroborate the stability of the system when employing both controller designs.

REFERENCES

[1] Mohsan, S. A. H., Othman, N. Q. H., Li, Y., Alsharif, M. H., and Khan, M. A., "Unmanned aerial vehicles (UAVs): practical aspects, applications, open challenges, security issues, and future trends," Intelligent Service Robotics, 2023, pp. 1–29.

[2] Paulet, Alita. (2023). Unmanned Aerial Vehicle (UAV) Market Growth Dynamic Report 2023-2030.

[3] "Aerospace Students Revive Leonardo Da Vinci's Aerial Screw (and Turn It into an Electric Personal Aerial Vehicle for Today)." TU Delft, www.tudelft.nl/en/stories/articles/aerospace-students-revive-leonardo-da-vincis-aerial-screw-and-turn-it-into-an-electric-personal-aerial-vehicle-for-today. Accessed 16 Jan. 2024.

[4] Cureton, Paul. "How Drones and Aerial Vehicles Could Change Cities." Singularity Hub, 2 Mar. 2021, singularityhub.com/2020/07/08/how-drones-and-aerial-vehicles-could-change-cities/. Accessed 18 Jan. 2024.

[5] Yi, S., Lin, D., Xu, X., Watanabe, K., and Nagai, I., "Robust Control for a Tandem Rotor UAV Under Wind Disturbances," 2022 IEEE International Conference on Mechatronics and Automation (ICMA), 2022, pp. 1744–1749. https://doi.org/10.1109/ICMA54519.2022.9856251.

[6] Abdelmoeti, S., and Carloni, R., "Robust control of UAVs using the parameter space approach," 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2016, pp. 5632–5637. https://doi.org/10.1109/IROS.2016.7759828.

[7] Jasim, O. A., and Veres, S. M., “A robust controller for multi rotor UAVs,” Aerospace Science and Technology, Vol. 105, 2020, p. 106010. https://doi.org/https://doi.org/10.1016/j.ast.2020.106010, URL https://www.sciencedirect.com/science/article/pii/S1270963820306921.

[8] Bianchi, D., Di Gennaro, S., Di Ferdinando, M., and Acosta Lùa, C., “Robust Control of UAV with Disturbances and Uncertainty Estimation,” Machines, Vol. 11, No. 3, 2023, p. 352.

[9] Mokhtari, M. R., and Cherki, B., “A new robust control for minirotorcraft unmanned aerial vehicles,” ISA Transactions, Vol. 56, 2015, pp. 86–101. https://doi.org/https://doi.org/10.1016/j.isatra.2014.12.003, URL https://www.sciencedirect.com/science/article/pii/S0019057814003048.

[10] Raffo, G. V., Ortega, M. G., and Rubio, F. R., “An integral predictive/nonlinear H infinity control structure for a quadrotor helicopter,” Automatica, Vol. 46, No. 1, 2010, pp. 29–39.https://doi.org/https://doi.org/10.1016/j.automatica.2009.10.018, URL https://www.sciencedirect.com /science/article/pii/S0005109809004798.

[11] Adlakha, R., and Zheng, M., “A Two-Step Optimization-Based Iterative Learning Control for Quadrotor Unmanned Aerial Vehicles,” Journal of Dynamic Systems, Measurement, and Control, Vol. 143, No. 7, 2021. https://doi.org/10.1115/1.4049566, URL https://doi.org/10.1115/1.4049566, 071006.

[12] Zhao, X., Yuan, M., Cheng, P., Xin, L., and Yao, L., “Robust H_∞ /S-plane Controller of Longitudinal Control for UAVs,” IEEE Access, Vol. 7, 2019, pp. 91367–91374. https://doi.org/10.1109/ACCESS.2019.2927000.

[13] Chen, M., and Huzmezan, M., “A combined mbpc/2 dof h infinity controller for a quad rotor uav,” AIAA guidance, navigation, and control conference and exhibit, 2003, p. 5520.

[14] Schioler, H., Leth, J., Leth, T., and Totu, L., “Stochastic design of switching controller for quadrotor UAV under intermittent localization,” 2016 Australian Control Conference (AuCC), 2016, pp. 243–248. https://doi.org/10.1109/AUCC.2016.7868196.

[15] Cheng, N., and Wang, C., “A Switching Mode Control Scheme for the Hovering Control of Quadrotor Unmanned Aerial Vehicles,” Mathematics, Vol. 11, No. 4, 2023. https://doi.org/10.3390/math11040994, URL https://www.mdpi.com/2227-7390/11/4/994.

[16] Jansch-Porto, J. P., and Dullerud, G. E., “Robust Decentralized Switching Control of UAVs using UWB-based Localization and Cooperation∗∗The authors were partially supported by NSF Grant 1629949.” IFAC-PapersOnLine, Vol. 53, No. 2, 2020, pp.7418–7423. https://doi.org/https://doi.org/10.1016/j.ifacol.2020.12.1281, URL https://www.sci ence direct.com/science/article/pii/S2405896320316839, 21st IFAC World Congress.

[17] Wu, X., Li, W., Hong, D., Tao, R., and Du, Q., “Deep Learning for Unmanned Aerial Vehicle-Based Object Detection and Tracking: A survey,” IEEE Geoscience and Remote Sensing Magazine, Vol. 10, No. 1, 2022, pp. 91–124. URL https://doi.org/ 10.1109%2Fmgrs.2021.3115137.

[18] Sahay, R., Birch, G. C., Stubbs, J. J., and Brinton, C. G., "Uncertainty Quantification-Based Unmanned Aircraft System Detection using Deep Ensembles," 2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring), 2022, pp. 1–5. https://doi.org/10.1109/VTC2022-Spring54318.2022.9860853.

[19] Ramachandran, A., and Sangaiah, A. K., "A review on object detection in unmanned aerial vehicle surveillance," International Journal of Cognitive Computing in Engineering, Vol. 2, 2021, pp. 215–228. https://doi.org/https://doi.org/10.1016/j.ijcce.2021. 11.005, URL https://www.sciencedirect.com/science/article/pii/S2666307421000267.

[20] Zhou, H., Ma, A., Niu, Y., and Ma, Z., "Small-Object Detection for UAV-Based Images Using a Distance Metric Method," Drones, Vol. 6, No. 10, 2022. https://doi.org/10.3390/drones6100308, URL https://www.mdpi.com/2504-446X/6/10/308 .

[21] Liu, M., Wang, X., Zhou, A., Fu, X., Ma, Y., and Piao, C., "UAV-YOLO: Small Object Detection on Unmanned Aerial Vehicle Perspective," Sensors, Vol. 20, No. 8, 2020. https://doi.org/10.3390/s20082238, URL https://www.mdpi.com/1424-8220/20/8/2238.

[22] Arnez, F., Radermacher, A., and Espinoza, H., "Quantifying and Using System Uncertainty in UAV Navigation,", 2022.

[23] Kraus, F., and Dietmayer, K., "Uncertainty Estimation in One-Stage Object Detection," 2019 IEEE Intelligent Transportation Systems Conference (ITSC), 2019, pp. 53–60. https://doi.org/10.1109/ITSC.2019.8917494.

[24] Pollet, F., Budinger, M., Delbecq, S., Moschetta, J.-M., and Liscouët, J., "Quantifying and mitigating uncertainties in design optimization including off-the-shelf components: Application to an electric multirotor UAV," Aerospace Science and Technology, Vol. 136, 2023, p. 108179. https://doi.org/https://doi.org/10.1016/j.ast.2023.108179, URL https://www.science direct.com/science/article/pii/S1270963823000767.

[25] Fernández Castaño, A., Patton, C., Yoon, H. J., & Voulgaris, P. (2024). "Uncertainty Quantification-Based Switching Control Method for Vision-Based Object Tracking in Unmanned Aerial Vehicles". AIAA SCITECH 2024 Forum. doi: 10.2514/6.2024-0516

[26] MATLAB, version 9.9.0 (R2020b), The MathWorks Inc., Natick, Massachusetts, 2020.

[27] "System Identification Toolbox." System Identification Toolbox Documentation, www.mathworks.com/help/ident/. Accessed 22 Jan. 2024.

[28] McKelvey, T., Akçay, H., and Ljung, L., "Subspace-based multivariable system identification from frequency response data," IEEE Transactions on Automatic control, Vol. 41, No. 7, 1996, pp. 960–979.

[29] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A., "You only look once: Unified, real-time object detection," Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779–788.

[30] "YOLOv5|Pytorch,", 2021. URL https://pytorch.org/hub/ultralytics_yolov5, accessed: 01/23/2024.

[31] Redmon, J., and Farhadi, A., “YOLO9000: better, faster, stronger,” Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 7263–7271.

[32] Sun, K., Mohta, K., Pfrommer, B., Watterson, M., Liu, S., Mulgaonkar, Y., Taylor, C. J., and Kumar, V., “Robust Stereo Visual Inertial Odometry for Fast Autonomous Flight”, 2018.

[33] Qin, T., Li, P., and Shen, S., “VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator,” IEEE Transactions on Robotics, Vol. 34, No. 4, 2018, pp. 1004–1020. https://doi.org/10.1109/TRO.2018.2853729.

[34] Liu, K., Ok, K., Vega-Brown, W., and Roy, N., “Deep Inference for Covariance Estimation: Learning Gaussian Noise Models for State Estimation,” 2018, pp. 1436–1443. https://doi.org/10.1109/ICRA.2018.8461047.

[35] Tsuei, S., Soatto, S., Tabuada, P., and Milam, M. B., “LearnedUncertainty Calibration forVisual Inertial Localization,” 2021 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2021. https://doi.org/10.1109/icra48506.2021.9561179, URL https://doi.org/10.1109%2Ficra48506.2021.9561179.

[36] Albaba, B. M., and Ozer, S., “SyNet: An Ensemble Network for Object Detection in UAV Images,” 2020.

[37] Shah, S., Dey, D., Lovett, C., and Kapoor, A., “AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles,” Field and Service Robotics, 2017. URL https://arxiv.org/abs/1705.05065.

[38] Stanford Artificial Intelligence Laboratory et al., “Robotic Operating System,”, 2018. URL https://www.ros.org.

[39] Glover, K., and Doyle, J. C., “State-space formulae for all stabilizing controllers that satisfy an H infinity-norm bound and relations to relations to risk sensitivity,” Systems & control letters, Vol. 11, No. 3, 1988, pp. 167–172.

[40] Doyle, J., Glover, K., Khargonekar, P., and Francis, B., “State-space solutions to standard H 2 and H infinity control problems,” 1988 American Control Conference, IEEE, 1988, pp. 1691–1696.

[41] Safonov, M. G., Limebeer, D., and Chiang, R., “Simplifying the H infinity theory via loop-shifting, matrix-pencil and descriptor concepts,” International Journal of Control, Vol. 50, No. 6, 1989, pp. 2467–2488.

[42] Mendenhall, W. M., and Sincich, T. L., Statistics for Engineering and the Sciences, CRC Press, 2016. 12