STOCHASTICALLY OPTIMIZED MONOCULAR VISION-BASED NAVIGATION AND GUIDANCE

A Thesis
Presented to
The Academic Faculty

by

Yoko Watanabe

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Aerospace Engineering

Georgia Institute of Technology
April 2008

STOCHASTICALLY OPTIMIZED MONOCULAR VISION-BASED NAVIGATION AND GUIDANCE

Approved by:

Professor Eric N. Johnson,
Committee Chair
School of Aerospace Engineering
Georgia Institute of Technology

Professor Anthony J. Calise, Advisor
School of Aerospace Engineering
Georgia Institute of Technology

Professor Allen R. Tannenbaum
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor J.V.R. Prasad
School of Aerospace Engineering
Georgia Institute of Technology

Professor Panagiotis Tsiotras
School of Aerospace Engineering
Georgia Institute of Technology

Date Approved: 28 November 2007

To my parents

ACKNOWLEDGEMENTS

I would like to thank my advisors Dr. Anthony J. Calise and Dr. Eric N. Johnson for providing me with the great opportunity to study and work on my research in the UAV Laboratory at Georgia Tech, where has one of the best UAV research facilities in the world. I deeply appreciate them for their insightful academic advices, continuous support and encouragement they gave me during my study. I also thank them for letting me choose my research topic according to my own interest, and for being patient with my research progress. It was my great honor to be their student and work with them. I would like to acknowledge AFOSR for funding the research project. I would like to thank Dr. Allen Tannenbaum, Dr. J.V.R. Prasad and Dr. Panagiotis Tsiotras for serving as my thesis committee, and for giving me beneficial suggestions and comments on my research. I also thank them and the other faculties at Georgia Tech who offered me courses, through which I have nourished my knowledge about the control theories and system dynamics.

I would like to express my gratitude to all the members at the UAV Laboratory for their contributions to my research. In particular, I would like to thank Drs. Venky Madyastha and Ramachandra Sattigeri for all the academic discussions we had on the nonlinear estimation theories, Dr. Suresh Kannan and Nimrod Rooz for helping me implement my algorithms to the flight system, Jincheol Ha for developing the image processing algorithms I used in my work, and the other members including Jeong Hur, Henrik Christophersen, Weyne Pickell, Allen Wu, and Claus Christmann for their assists in conducting the flight tests. I would also like to acknowledge Alison Proctor and Adrian Koller for their contributions to the UAV Lab.

I had met many great people during my study at Georgia Tech, and I am very

glad to have them as my friends. First, I would like to thank Dr. Fang Zhong with whom I shared an apartment for three years. I thank her for always being supportive and giving me many advices on a PhD student life. I really had a wonderful time living with her. Second, I would like to thank all the friends who made my first year here fulfilling. I had a great time having academic and nonacademic interactions with Kenneth, Ryan, Ousmane, Julien, Julide, Kyung-Jin and Arnaud. I specially thank Kenneth who always motivated me to pursue a PhD degree when I lost my confidence. I also thank Ryan and Ousmane for the best friendship among three of us. Many thanks to Marika, and Drs. Tadashi Sakai and Peter Nagy for their supports. Third, I appreciate Dr. Serguei Norine for being my friend and also my advisor for my Master's degree in Mathematics. I admire his intelligence and am very proud of being his first graduate student in his career. I would like to address my great thanks to Maxime, Ludvic, Stephanie, Shunya, Laszlo, Bernard, Rene and Eva who made my last year at Georgia Tech very enjoyable. I thank Ludvic and Stephanie for always caring for me and being such a nice friend. I gratefully appreciate Maxime for everything he has done to support me in both of my work and my personal life. Because of his continuous encouragement, I could have gone through all the difficulties in finishing my PhD study at Georgia Tech. I wish him my very best in his study towards his PhD degree and also in his future career.

Finally, I would like to give my appreciation to my parents Mr. Yoshinori and Mrs. Mariko Watanabe and my sisters Masami and Ryoko for their endless support and fully understanding throughout my study. In particular, I thank my parents for respecting my decision to come to this country to make my PhD degree apart from them. I also would like to thank my grandparents for always wishing for my happiness and success. Without the mental and physical backup of my family, I could not have accomplished my PhD degree. I would like to dedicate this thesis to my family with my love.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The objective of this thesis is to design a relative navigation and guidance law for unmanned aerial vehicles, or UAVs, for vision-based control applications. The autonomous operation of UAVs has progressively developed in recent years. In particular, vision-based navigation, guidance and control has been one of the most focused on research topics for the automation of UAVs. This is because in nature, birds and insects use vision as the exclusive sensor for object detection and navigation. Furthermore, it is efficient to use a vision sensor since it is compact, light-weight and low cost. Therefore, this thesis studies the monocular vision-based navigation and guidance of UAVs.

Since 2-D vision-based measurements are nonlinear with respect to the 3-D relative states, an extended Kalman filter (EKF) is applied in the navigation system design. The EKF-based navigation system is integrated with a real-time image processing algorithm and is tested in simulations and flight tests. The first closed-loop vision-based formation flight between two UAVs has been achieved, and the results are shown in this thesis to verify the estimation performance of the EKF. In addition, vision-based 3-D terrain recovery was performed in simulations to present a navigation design which has the capability of estimating states of multiple objects. In this problem, the statistical z-test is applied to solve the correspondence problem of relating measurements and estimation states.

As a practical example of vision-based control applications for UAVs, a vision-based obstacle avoidance problem is specially addressed in this thesis. A navigation and guidance system is designed for a UAV to achieve a mission of waypoint tracking while avoiding unforeseen stationary obstacles by using vision information. An EKF is applied to estimate each obstacles' position from the vision-based information. A collision criteria is established by using a collision-cone approach and a time-to-go criterion. A minimum-effort guidance (MEG) law for multiple target tracking is applied for a guidance design to achieve the mission. Through simulations, it is shown that the control effort can be reduced by using the MEG-based guidance design instead of a conventional proportional navigation-based one. The navigation and guidance designs are implemented and evaluated in a 6 DoF UAV flight simulation. Furthermore, the vision-based obstacle avoidance system is also tested in a flight test using a balloon as an obstacle.

For monocular vision-based control problems, it is well-known that the separation principle between estimation and control does not hold. In other words, that vision-based estimation performance highly depends on the relative motion of the vehicle with respect to the target. Therefore, this thesis aims to derive an optimal guidance law to achieve a given mission under the condition of using the EKF-based relative navigation. Unlike many other works on observer trajectory optimization, this thesis suggests a stochastically optimized guidance design that minimizes the expected value of a cost function of the guidance error and the control effort subject to the EKF prediction and update procedures. A suboptimal guidance law is derived based on an idea of the one-step-ahead (OSA) optimization, in which the optimization is performed under the assumption that there will be only one more final measurement at the one time step ahead. The OSA suboptimal guidance law is applied to problems of vision-based rendezvous and vision-based obstacle avoidance. Simulation results are presented to show that the suggested guidance law significantly improves the guidance performance. The OSA suboptimal optimization approach is generalized as the n-step-ahead (nSA) optimization for an arbitrary number of n. Furthermore, the nSA suboptimal guidance law is extended to the p %-ahead suboptimal guidance by changing the value of n at each time step depending on the current time. The nSA (including the OSA) and the p %-ahead suboptimal guidance laws are applied to a 2-D vision-based target tracking problem, and their optimality and computational cost are investigated through simulation results.

CHAPTER I

INTRODUCTION

The objective of this thesis is to develop a monocular vision-based navigation and guidance design for unmanned aerial vehicles (UAVs). The first challenge in this thesis is to realize closed-loop vision-based control in actual flights. This requires a vehicle's flight system to have the capabilities of real-time image processing, target and ownship state estimation, and flight guidance and control. Each of these are substantial research topics in their own right. This thesis, however, focuses on the problems of target state estimation and vehicle flight guidance. A second challenge is to establish an optimal guidance design while accounting for the dependence between estimation and guidance in vision-based control problems. By including this dependence in a guidance design, it is expected to improve the estimation accuracy and hence improve the overall guidance performance. This chapter outlines the problems that this thesis aims to solve and also discusses relevant literature.

1.1 UAV Autonomous Flight

UAVs are expected to play an important role in both military and commercial applications. Two main advantages of using UAVs instead of manned aircraft for these operations are the following: they are considered to be cost-effective and there is no risk in the loss of human pilot life[9]. Tremendous work in navigation and flight control systems have been done since the early 1990's, and many successful automated flights of UAVs have been achieved[10][71]. The development of new sensors, such as the Global Positioning System (GPS), and communication technologies have made large contributions to autonomous UAV flights. There already exist some cases in

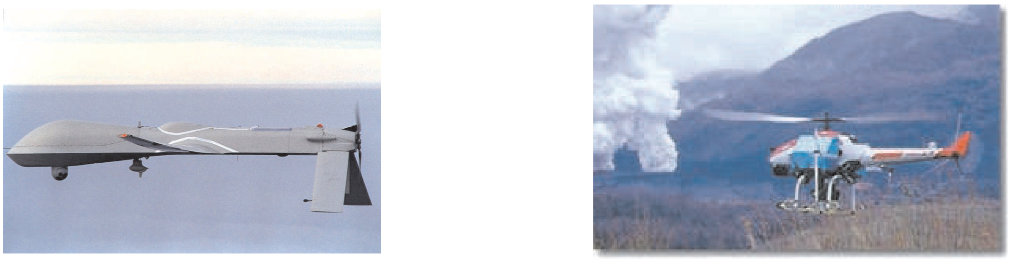


Figure 1: (Left) U.S. Airforce Predator used for military operations[80], (Right) Yamaha RMax used for volcano observations[65]

which UAVs have been utilized for practical missions by means of remotely commanded/controlled operation. The U.S. Air Force Predator (Figure 1, Left) is one of the most recognized examples of these UAVs. Since 1995, the Predator has been operational in actual battlefields for surveillance and reconnaissance[80]. As another example, Yamaha's unmanned helicopter RMax (Figure 1, Right) has operated in the proximity of active volcanos for observation purposes[65]. Moreover, UAVs are considered to have significant potential for mine detection and removal applications[69] because UAVs can detect mines without touching them.

The autonomous operation of UAVs has been progressively developed in recent years[16]. Autonomy is distinguished from automation as it requires the capability of making decisions as well as executing them. Autonomous control systems enable UAVs to accomplish high-level missions with less or no human operators. One of the most challenging problems for autonomous UAV flight is situational awareness. Since in most missions, UAVs are expected to operate in an uncertain and possibly adversarial environment, the system requires active or passive sensors to provide information about the environment, and also requires algorithms that extract objects of interest from that information. For example, laser rangefinders can provide highly accurate data about the surrounding environment[53]. 3-D terrain can also be accurately modelled by the sensor fusion of laser range readings with camera images[79][52]. However, a laser rangefinder takes a long time to scan and render the information, which is not preferable for some missions such as target tracking.

As seen in nature among birds and insects, a 2-D passive vision sensor can provide sufficient information as an exclusive sensor for object detection and navigation. Furthermore, it is efficient to use a vision sensor since it is compact, light-weight and low cost compared to a laser rangefinder. Therefore, this thesis considers the utilization of a single 2-D passive vision sensor to detect objects. The next section discusses the vision-based navigation and guidance problem.

1.2 Vision-Based Navigation and Control

As stated above, vision-based UAV operation has significant potential. Flying insects and birds highly rely on visual information to navigate themselves in a 3-D environment. Some have experimentally investigated the visual strategies of insects, and have consequently designed robot navigation and control system inspired by their results[76][55]. Srinivasan et al. studied the visual landing strategy of honeybees and applied it to UAVs[77]. These nature-inspired algorithms are very simple and efficient.

Monocular vision-based navigation and control is one of the most focused on research topics for the automation of UAVs. Vision-based navigation and control have been studied in many applications. Some studies focus on vision-based vehicle localization. Vision information is utilized to determine the vehicle states, such as position and attitude, in case of GPS failure[90][43][88]. Others have developed algorithms to recover the 3-D environment from 2-D vision measurements, assuming all the vehicle states are known through its own-ship navigation system. [18], [19], [20] and [85] address the terrain recovery problem. Terrain models can be created by visually detecting features on the ground while flying a UAV over a field. Collecting this environmental information is very important in order to operate the vehicle safely in unknown and possibly adversarial environments. The resulting terrain map can be used for path planning and obstacle avoidance. Vision-based target tracking has also been intensively investigated. In the target tracking problem, a given target (stationary or moving) is detected in an image, its relative state with respect to the vehicle is estimated and the vehicle is guided to track it by using the estimate. Vision-based autonomous landing[74] and ground object and vehicle tracking[63][56] are examples of air-to-ground tracking. Vision-based formation flight[34] and aerial refueling[81] are examples of air-to-air tracking. Furthermore, simultaneous localization and modeling (SLAM) problems have been investigated. This problem is even more difficult since there is neither knowledge about the vehicle states nor its environment. However, it has been proven that the problem has a solution and several applications have been presented[17][46][33].

To achieve autonomous vision-based UAV operations, a real-time image processing algorithm is required. In particular, the processing time is critical for guidance with the purpose of object tracking. Image processing itself is a very challenging topic. Even though so many different kinds of image processing algorithms have been developed, it is still difficult to process complex images in realtime. This thesis does not focus on image processing algorithms, but makes use of an active contour method developed in [24][25]. Active contours are processes that use image coherence to track features of interest over time[8]. They fit naturally into control frameworks and have often been employed in conjunction with Kalman filtering. Specifically, the fast marching level set method is used in our application. The level set method performs

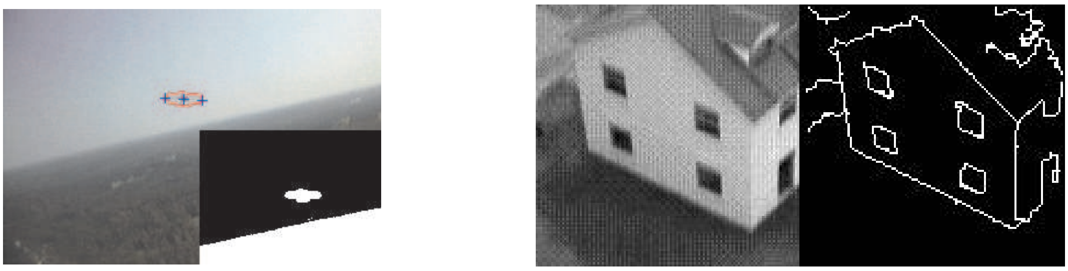


Figure 2: Examples of Image Processing: (Left) Detecting an airplane flying ahead, (Right) Detecting edges of a ground obstacle

processing calculations only in a limited region near the active contour[70]. Therefore, it can reduce the computational burden and make the process realtime applicable. Furthermore, adding a particle filtering to an object search enables the image processor to track multiple objects (possibly moving) in a cluttered background. Figure 2 shows examples of image processing.

Assuming the real-time image processor detects the target in each image frame, an estimator is designed to estimate the target relative state needed in the guidance law. Kalman suggested an optimal estimator for a linear system, known as the Kalman filter in [40]. The Kalman filtering process consists of two different steps: one is prediction, which is performed based on the state dynamics, and the other is an update, which is an estimate correction based on the measurement. The linear Kalman filter was extended so that it could be applied to nonlinear system by linearizing about the predicted estimate at each time step. This nonlinear filter is called the extended Kalman filter or the EKF[12][93]. In our applications, the EKF is utilized since the vision-based measurement is a nonlinear function of the relative state. Although convergence of the EKF estimation is not theoretically guaranteed, its performance has been demonstrated in many practical applications. However, when a system is highly nonlinear and cannot be well propagated by its first-order approximation, an EKF can cause poor estimation performance. To overcome this problem, other nonlinear filtering methods such as the unscented Kalman filter (UKF) and the particle filter (PF) have been suggested[39]. The UKF and the PF use Monte Carlo sampling to propagate the nonlinear system, and they capture the system's nonlinearity more accurately[11]. [57] and [58] compare estimation performance between the three different estimation frameworks for a vision-based target tracking problem and conclude a superiority of the UKF and the PF over the EKF. Moreover, adaptive estimation is also an efficient filtering method when there is uncertainty in the dynamics. Especially, it can be useful when the target's maneuver is unknown.

[49] and [67] suggest adding a neural-network based adaptation to an EKF. In their work, the adaptive estimator is applied to a vision-based maneuvering target tracking problem. The adaptation compensates for an unknown target acceleration and hence it improves the EKF estimation performance. Works showed that these advanced filtering methods can provide more accurate estimates for some cases. However, the EKF is applied in this thesis because it has already been successfully utilized for systems without large uncertainties. As an example, the estimation results from actual flight tests of vision-based formation flight are presented in this thesis to show that the EKF attains sufficiently accurate estimation in our application. The estimated states are fed back to a guidance law to achieve some given mission.

1.3 Vision-Based Obstacle Avoidance

As an important and practical example of vision-based control applications for UAVs, a vision-based obstacle avoidance problem is specially addressed in this thesis. For some missions UAVs have to operate in congested environments that include unknown obstacles. For such missions, obstacle avoidance is an anticipated requirement. This thesis presents a vision-based navigation and guidance design for UAVs to detect and avoid obstacles while executing a given mission such as preplanned path following or waypoint tracking.

Kumar and Ghose proposed a navigation and guidance law that achieves both waypoint tracking and collision avoidance[44]. This algorithm assumes range information is available from a radar. A method described in [45] also assumes a radar sensor system for collision avoidance. A laser scanner is suggested as an alternative sensor and successful flight test results of laser-based obstacle avoidance are presented in [71] and [23]. On the other hand, passive vision sensors are also commonly used for obstacle sensing especially for small scale UAVs. Stereo vision system has been designed for real-time obstacle detection[31][62]. However, stereo vision systems require precise camera calibration and also a feature matching algorithm for relating two images. Hence, this thesis focuses on monocular vision-based obstacle detection and avoidance.

Many studies have been done on a guidance strategy for collision avoidance. A method discussed in [68] uses a combination of circular paths as a vehicle's avoiding maneuver, so that the vehicle can efficiently sense its surroundings in every direction. Pollini et al. suggested giving a velocity command so that a vehicle keeps a certain distance from each obstacle[61]. In their work, a shadow cone is used in the guidance design to avoid collisions with a group of multiple obstacles. [27] and [87] proposed a guidance design based on a proportional navigation (PN) law along with the use of the collision cone approach suggested by [15]. However, all of these algorithms are limited to the case in which the vehicle and the obstacles stay in a 2-D plane. That is, they are effective only for operations on ground or in planar flight at constant altitude. This thesis establishes a guidance design for 3-D waypoint tracking and obstacle avoidance by extending the collision cone approach from 2-D to 3-D. Furthermore, a minimum-effort guidance (MEG) for multiple target tracking derived in [3] and [4] is applied instead of PN. It is always important to minimize the required control energy in UAV operations to save fuel. Simulation results show that the MEG-based guidance law achieves waypoint tracking and obstacle avoidance mission with less control effort than the PN-based guidance law[82][83]. The suggested MEG-based guidance design is integrated with a real-time image processor and an EKF-based navigation filter, and is implemented and evaluated in the image-in-the-loop 6 DoF UAV flight simulation. Finally, the vision-based obstacle avoidance system is implemented in the autonomous flight system of an actual UAV, and flight test results are presented in this thesis.

1.4 Observer Trajectory Optimization

The guidance law for the UAV in vision-based control applications should be designed to achieve given missions by using vision-based navigation. Generally, the guidance design is formulated as a vehicle trajectory optimization problem. There is a large body of research on this subject[64][6][2]. Most considered off-line trajectory generation and aimed to solve the optimization problems efficiently under the assumption of full state information. Even if the algorithm is realtime applicable, the state is not fully available in reality. In real applications, the optimal vehicle trajectory needs to be calculated by using the state estimate which is updated at each time step by available measurements. A common way to determine the vehicle guidance law for such a case is to replace the true states by their estimates in the optimal solution that is obtained by assuming full state information. However, this approach is not optimal and can even cause mission failure due to estimation errors. Since a vehicle guidance law uses these estimates, obtaining good guidance performance requires accurate estimates. Blackmore suggested a guidance design which considers estimation and modeling uncertainties[7]. This work designs a guidance law for obstacle avoidance by using the estimated vehicle states. The guidance design utilizes particles to measure the probabilistic estimation uncertainties and the vehicle is guided to a further distance from the obstacle when the estimate includes a larger uncertainty. Frew and Sangupta also developed a guidance design for the vision-based obstacle avoidance problem, including an error model of a stereo vision system[21].

For vision-based relative navigation, it is well known that estimation performance significantly depends on sensor motion relative to the target[51]. In other words, unlike linear systems, the separation principle does not hold between estimation and control. Observability of the vision-based estimator depends on the vehicle motion which is determined by the guidance law. At the same time, guidance performance directly depends on estimation accuracy since the estimate is fed back to the guidance law. Therefore, an objective of this thesis is to derive an optimal guidance design which can achieve given missions while minimizing the estimation uncertainty. Estimation enhancement by trajectory modification was first considered in a homing missile guidance problem in the 1980's[75][32][28]. This problem is called *Dual Control*. In dual control, a guidance law is designed to maximize the information content of a missile's intercept trajectory. The resulting trajectory ensures that the line-of-sight between the missile and the target is always in motion in order to increase observability. In [32], the maximization of the lateral motion is formulated in a quadratic form and an analytical optimal solution is derived. Observer trajectory optimization for estimation improvement was also treated by Hammel et al[26]. Similar studies have been performed in bearing-only localization and target tracking[59][60][47] and in vision-based estimation[22][84]. In these papers, an optimal observer trajectory is calculated so that the estimation error is minimized. Singh et al. formulated a more generalized problem called sensor scheduling[73]. They introduced the concept of sensor actions which could not only be a sensor trajectory but also a choice of sensor to be used, tunable parameters and so forth. The estimation error is minimized over available sensor actions. However, only sensor trajectory optimization is treated in their paper. Some studies listed above have a target tracking mission which appears as a terminal condition in the optimization problem. The main focus of these works was cost function selection so that the resulting optimization problem can be efficiently solved. For example, [26] and [59] choose the determinant of the Fisher information matrix and utilize a direct gradient numerical method to maximize it over the observer trajectory. Frew et al. minimize the determinant of the predicted estimation error covariance matrix over the discretized observer motion[22]. There are two main issues associated with these works. One is that there has not yet been established a systematic way to choose the cost function. The other is that most of their algorithms require iterative calculations to obtain the optimal solution and they are computationally expensive. Therefore, this thesis aims to establish a real-time applicable guidance design which can be systematically determined from the original vehicle trajectory optimization problem.

1.5 Stochastically Optimal Guidance Design

Since the estimation error is assumed to be white Gaussian noise with zero mean and with its estimated covariance matrix obtained in the EKF process, stochastic optimization can be performed for the original vehicle trajectory optimization problem which has been set up to achieve given missions. In this approach, the cost function will be systematically obtained. Kim and Rock also suggested a stochastic feedback controller design for bearing-only tracking[42]. Since the EKF update law is nonlinear with regard to the relative motion dynamics, a solution of the resulting stochastic optimization problem can only be obtained numerically by using iterative calculations such as dynamic programming. To reduce the computational cost, we will try to obtain an approximately optimal or suboptimal solution. In [42], a steady state solution is assumed and the optimal guidance law is derived by solving the algebraic Ricatti equation (ARE). This assumption is not appropriate when a finite terminal time is given. Logothetis et al. compared several different suboptimal techniques for observer trajectory optimization for the bearing-only tracking problem[48]. The idea of one-step-ahead (OSA) optimization was introduced as one of these techniques in [48]. This is a suboptimal optimization strategy in which the optimization problem is solved under the assumption that there will be only one more final measurement at the next time step. Under this assumption, the observer trajectory optimization needs to be considered only for a guidance input at the current time step since there will be no chance to improve the estimation accuracy after the final measurement is obtained. Therefore, the optimization will be performed over a single vector representing a guidance input at the current time step, and it can be solved by an algebraic equation.

This thesis combines the stochastic optimization formulation and the OSA optimization approach to establish a suboptimal guidance design for the monocular vision-based control problem. Through simulations of vision-based rendezvous and vision-based obstacle avoidance, it is observed that the suggested approach significantly improves the guidance performance without large computational load[86]. Furthermore, the OSA suboptimal guidance law is generalized to n-steps-ahead (nSA) optimization for an arbitrary number of n in this thesis. As the number of steps n increases, the suboptimal solution approaches the optimal solution, but at the same time it requires more computations. The optimal solution can be numerically derived by using a sufficiently large value for n. Then the OSA suboptimal guidance law is just a special case of $n = 1$. In addition, the p %-ahead suboptimal guidance is suggested by modifying the nSA suboptimal guidance. In that approach, at each time step, a value for n is determined by a number of steps including in the first p % of the time period from the current time to the terminal time. The nSA and the p %-ahead suboptimal guidance designs are applied to a vision-based 2-D target tracking problem. Simulation results are compared for different values of n (including the OSA suboptimal guidance, which corresponds to $n = 1$) and of p, and their optimality and computational costs are examined in this thesis.

1.6 Thesis Objectives and Outline

Figure 3 is a diagram of a vision-based autonomous flight system for a UAV. The goals of this thesis are to design the relative navigation and guidance systems, to integrate them with the other subsystems (which are assumed to be available), and to evaluate the entire system in realistic simulations and also in actual flight tests.

The thesis is organized as follows: Chapter 2 presents a vision-based navigation design using an extended Kalman filter. A standard EKF formulation is provided, and the navigation design for vision-based formation flight between two UAVs is

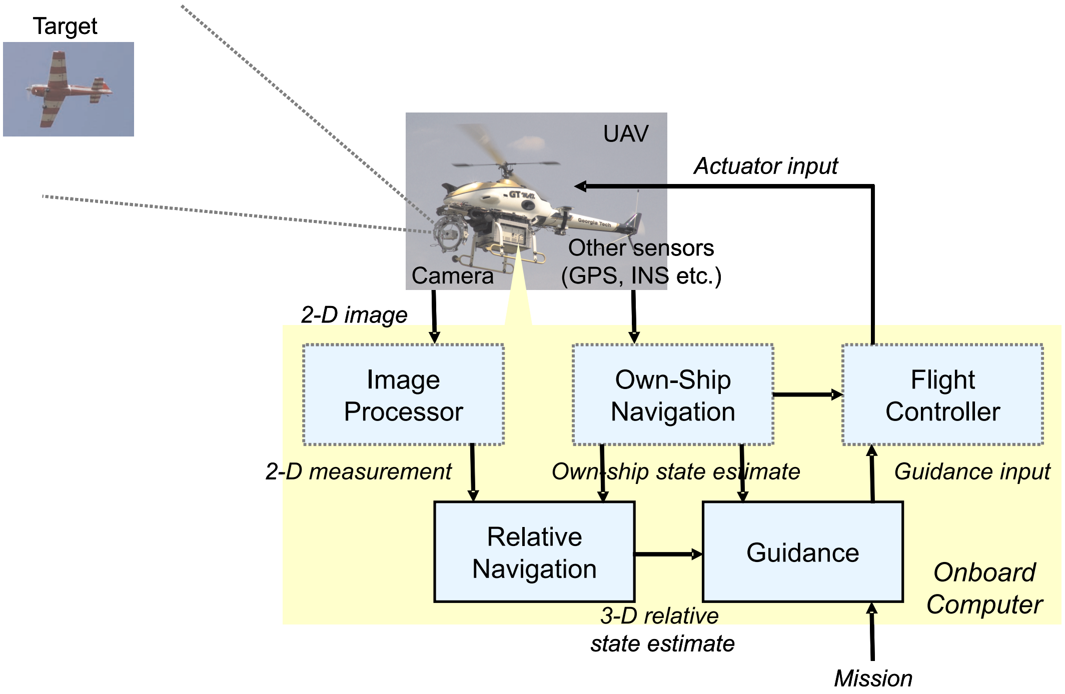


Figure 3: Vision-Based Autonomous Flight System

shown as an example. The navigation results of both simulation and flight test are presented to verify the EKF estimation performance. For another example, the EKF-based navigation design is applied to a vision-based 3-D obstacle modeling problem. In this example, multiple objects need to be tracked. The statistical z-test is used to solve the correspondence problem between measurements and estimates. Chapter 3 focuses on a 3-D vision-based obstacle avoidance problem for UAVs. A relative navigation and guidance system is developed and tested in simulations and flight tests. The results show that the control effort can be reduced by using the suggested minimum-effort guidance. Chapter 4 develops the stochastically optimized guidance design. For a given trajectory optimization problem, a stochastic optimization is formulated subject to the vehicle dynamics and the EKF estimation process. A real-time applicable suboptimal guidance law is designed by applying the one-step-ahead (OSA) optimization, and its guidance performance is evaluated in simulations with vision-based target tracking applications. In addition, the OSA suboptimal guidance law is extended to the n-step-ahead and the p %-ahead suboptimal guidance designs. Simulation results of these suboptimal guidance laws are compared with a numerically obtained optimal solution, and their optimality and computational loads are examined. Chapter 5 summarizes contributions of the thesis and addresses recommended future work related to the research in this thesis.

CHAPTER II

VISION-BASED NAVIGATION DESIGN

This chapter describes a vision-based navigation design. Since the 2-D vision-based measurement is nonlinear with respect to the 3-D relative state, an extended Kalman filter (EKF) is applied to estimate the relative state from the measurement. The EKF-based relative navigation design is applied to vision-based formation flight and to vision-based 3-D obstacle modeling problems, and its estimation performance is evaluated in simulations. Furthermore, successful flight test results of a closed-loop vision-based formation flight between two UAVs are presented.

2.1 Problem Formulation

Let $\boldsymbol{X}_v$ and $\boldsymbol{V}_v$ be a vehicle's position and velocity vectors expressed in an inertial frame (denoted by F_I). Suppose that the vehicle dynamics can be modeled as the following simple linear system.

$$\dot{\boldsymbol{X}}_v(t) = \boldsymbol{V}_v(t), \quad \dot{\boldsymbol{V}}_v(t) = \boldsymbol{a}_v(t) \tag{2.1}$$

where $\boldsymbol{a}_v$ is the vehicle's acceleration input. It is assumed that all the vehicle states are available through its own-ship navigation system. The target dynamics are similarly given by

$$\dot{\boldsymbol{X}}_t(t) = \boldsymbol{V}_t(t), \quad \dot{\boldsymbol{V}}_t(t) = \boldsymbol{a}_t(t) \tag{2.2}$$

where $\boldsymbol{X}_t$, $\boldsymbol{V}_t$ and $\boldsymbol{a}_t$ are the target's position, velocity and acceleration vectors, respectively. Relative position, velocity and acceleration of the target with respect to the vehicle are defined by

$$\begin{aligned}\boldsymbol{X}(t) &= \boldsymbol{X}_t(t) - \boldsymbol{X}_v(t) \\ \boldsymbol{V}(t) &= \boldsymbol{V}_t(t) - \boldsymbol{V}_v(t) \\ \boldsymbol{a}(t) &= \boldsymbol{a}_t(t) - \boldsymbol{a}_v(t)\end{aligned} \tag{2.3}$$

Then the relative motion dynamics are formulated as follows.

$$\dot{\boldsymbol{X}}(t) = \boldsymbol{V}(t), \quad \dot{\boldsymbol{V}}(t) = \boldsymbol{a}(t) \tag{2.4}$$

Since the target's maneuver is unknown in most cases, we need to apply some model for $\boldsymbol{a}_t$ based on its prior knowledge. For example, the following target model is called Singer model[72].

$$\dot{\boldsymbol{a}}_t(t) = -\alpha \boldsymbol{a}_t(t) + \boldsymbol{w}(t), \quad \alpha > 0$$

where $\boldsymbol{w}(t)$ is a zero mean Gaussian noise. A target's velocity is modeled as a random walk process in the following.

$$\boldsymbol{a}_t(t) = \mathbf{0} + \boldsymbol{w}(t)$$

In this problem, for simplicity, it is assumed that a 2-D passive vision sensor is mounted at the center of gravity of the vehicle. (This assumption can be easily removed by including extra rotational dynamics caused by an offset in the relative motion model.) We also assume that an image processor which is able to detect a target position in an image frame is available. Let L_c denote a known camera attitude represented by a rotation matrix from the inertial frame F_I to a camera frame which is denoted by F_C. A camera frame is taken so that the camera's optical axis aligned with its X_c axis. Then the relative position expressed in F_C will be

$$\boldsymbol{X}_c(t) = L_c(t)\boldsymbol{X}(t) = \begin{bmatrix} X_c(t) & Y_c(t) & Z_c(t) \end{bmatrix}^T \tag{2.5}$$

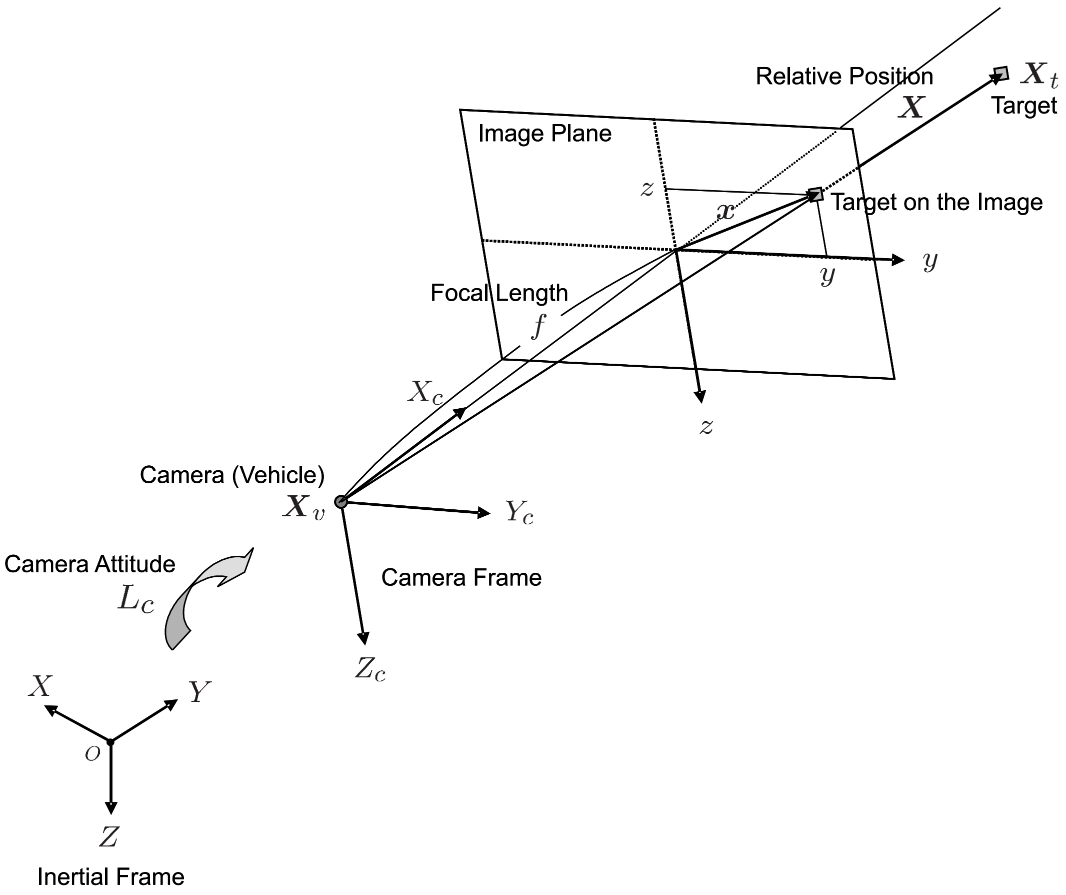


Figure 4: Pin-Hole Camera Model

Assuming a pin-hole camera model shown in Figure 4, the target position in the image at a k-th time step t_k is given by

$$\boldsymbol{x}_k = \begin{bmatrix} y_k \\ z_k \end{bmatrix} = \frac{f}{X_c(t_k)} \begin{bmatrix} Y_c(t_k) \\ Z_c(t_k) \end{bmatrix} \tag{2.6}$$

where f is the focal length of the camera. In this thesis, $f = 1$ is used without loss of generality. More target information will be available if the image processor can detect the target's shape, size, contours or other characteristics in addition to its center position $\boldsymbol{x}$. The vision-based navigation objective is to estimate the unknown target states from the image processor outputs and the known vehicle and camera states.

2.2 Extended Kalman Filter

Since the 2-D vision-based measurement (2.6) is a nonlinear function with respect to the 3-D relative state, an extended Kalman filter (EKF) is applied to estimate the relative states from the measurement. The Kalman filter is a recursive solution to the least-squares method for a linear filtering problem[40]. Since the filter was introduced by Kalman in 1960, it has been the subject of extensive research and application particularly in the area of autonomous navigation[89]. The EKF is an extension of the standard linear Kalman filter so that it can be applied to nonlinear systems by linearizing the system about the predicted estimate at each time step[11][12][93]. Even though the convergence of its estimate cannot be theoretically proven, good estimation performance of the EKF has been demonstrated in many practical applications.

A general formulation of the EKF is presented in this section. Consider the following nonlinear system.

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)) + \boldsymbol{w}(t) & (2.7) \\ \boldsymbol{z}_k &= \boldsymbol{h}(\boldsymbol{x}_k) + \boldsymbol{\nu}_k & (2.8) \end{aligned}$$

where $\boldsymbol{x}$ is a state vector, $\boldsymbol{u}$ is a system input, $\boldsymbol{w}$ is a random zero-mean process noise, $\boldsymbol{z}_k$ is a measurement at time step k and $\boldsymbol{\nu}_k$ is a discrete measurement noise. Let $\hat{\boldsymbol{x}}_k^-$ and $\hat{\boldsymbol{x}}_k$ be the predicted and updated estimates of $\boldsymbol{x}(t_k)$, and P_k^- and P_k be their estimated error covariance matrices. A state estimate at t_{k-1} is propagated to the next time step t_k through a first-order Euler integration of the original nonlinear system (2.7).

$$\hat{\boldsymbol{x}}_k^- \simeq \hat{\boldsymbol{x}}_{k-1} + \boldsymbol{f}(\hat{\boldsymbol{x}}_{k-1}, \boldsymbol{u}_{k-1})(t_k - t_{k-1}) \tag{2.9}$$

Then the system dynamics and the measurement model (2.7-2.8) are linearized about $\hat{\boldsymbol{x}}_k^-$ and discretized as follows.

$$\begin{aligned} \boldsymbol{x}_k &= \Phi_k \boldsymbol{x}_{k-1} + \Gamma_k \boldsymbol{u}_{k-1} + \boldsymbol{w}_{k-1} & (2.10) \\ \boldsymbol{z}_k &= H_k \boldsymbol{x}_k + \boldsymbol{\nu}_k & (2.11) \end{aligned}$$

where

$$
\begin{aligned}
\Phi_k &= e^{F_k(t_k - t_{k-1})} && (2.12)\\
\Gamma_k &= \int_{t_{k-1}}^{t_k} e^{F_k(t_k - s)} G_k ds && (2.13)\\
\boldsymbol{w}_{k-1} &= \int_{t_{k-1}}^{t_k} e^{F_k(t_k - s)} \boldsymbol{w}(s) ds && (2.14)
\end{aligned}
$$

and

$$F_k = \frac{\partial \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u})}{\partial \boldsymbol{x}}\Big|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_k^-,\boldsymbol{u}=\boldsymbol{u}_{k-1}},\ G_k = \frac{\partial \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u})}{\partial \boldsymbol{u}}\Big|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_k^-,\boldsymbol{u}=\boldsymbol{u}_{k-1}},\ H_k = \frac{\partial \boldsymbol{h}(\boldsymbol{x}_k)}{\partial \boldsymbol{x}_k}\Big|_{\boldsymbol{x}_k=\hat{\boldsymbol{x}}_k^-} \quad (2.15)$$

Now the linear discrete Kalman filtering algorithm can be applied to the linearized system (2.10-2.11). The predicted and updated error covariance matrices and the Kalman gain are calculated by

$$
\begin{aligned}
P_k^- &= \Phi_k P_{k-1} \Phi_k^T + Q_k && (2.16)\\
K_k &= P_k^- H_k^T \left(H_k P_k^- H_k^T + R_k\right)^{-1} && (2.17)\\
P_k &= \left(I - K_k H_k\right) P_k^- && (2.18)
\end{aligned}
$$

where Q_k and R_k are covariance matrices of the discrete process noise $\boldsymbol{w}_k$ and measurement noise $\boldsymbol{\nu}_k$. Finally, the updated state estimate is obtained by

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_k^- + K_k \left(\boldsymbol{z}_k - \boldsymbol{h}(\hat{\boldsymbol{x}}_k^-)\right) \quad (2.19)$$

Since a camera's field of view is limited and the image processor may sometimes fail to capture the target, the vision-based measurement is not always available. When this happens, only the EKF prediction procedure (2.9, 2.16) is performed. The absence of a measurement corresponds to having a measurement with an infinitely large noise. When $R_k = \infty$ in (2.17), the Kalman gain K_k becomes zero. It results in $\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_k^-$ and $P_k = P_k^-$, and nothing will be changed in the EKF update procedure (2.18, 2.19).

2.3 Application 1 : Vision-Based Formation Flight

In order to evaluate the EKF estimation performance for applications of vision-based navigation, the EKF-based relative navigation design is applied and tested in two different frameworks. For missions such as surveillance and reconnaissance, multiple UAVs are expected to execute the missions more efficiently by maintaining a formation. Moreover, it will be important to minimize the communication between vehicles. For those reasons, vision-based formation flight between two UAVs is addressed in this section. In this application, an EKF is designed to estimate the relative state of the leader aircraft with respect to the follower from the vision information. The EKF is implemented and tested in an image-in-the-loop 6 DoF multiple UAV flight simulation and in actual flights of UAVs. An image processor, an own-ship navigation filter and a flight guidance and control system have already been developed and implemented. The image processor used is developed for realtime target tracking by Ha et al. in [24] and [25]. In flight experiments, closed-loop vision-based formation flights of two UAVs have been successfully achieved[35]. Estimation results obtained in both simulations and flight tests are illustrated in this section.

2.3.1 Navigation Filter Design

(a) Process Model

In the framework of the vision-based formation flight of two UAVs (a leader and a follower), the EKF is designed to estimate the relative position of the leader aircraft with respect to the follower from the vision-based measurement. In the EKF design for this application, an estimation state vector is taken as follows.

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{u}^T & \dot{\boldsymbol{u}}^T & \frac{1}{r} & \frac{\dot{r}}{r} & b \end{bmatrix}^T \tag{2.20}$$

$\boldsymbol{u}$ is an unit vector pointing from the follower to the leader and r is a range between the two aircraft. They are defined by

$$\boldsymbol{u} = \frac{\boldsymbol{X}}{\|\boldsymbol{X}\|}, \quad r = \|\boldsymbol{X}\| \tag{2.21}$$

where $\boldsymbol{X}$ is the relative position vector. In bearing-only tracking problem, it is very common to use an inverse of range $1/r$ instead of the range itself to reduce the nonlinearity associated with its dynamics[1]. Moreover, the unit vector is chosen as an estimation state, instead of bearing and elevation angles which are commonly used[1], to avoid a singularity. b in (2.20) is a constant wingspan of the leader airplane. When using the random walk model for the target's velocity, the process model is written as follows.

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\boldsymbol{u}} \\ -\left(\|\dot{\boldsymbol{u}}\|^2 - \frac{1}{r}\boldsymbol{u}\cdot\boldsymbol{a}_v\right)\boldsymbol{u} - 2\frac{\dot{r}}{r}\dot{\boldsymbol{u}} - \frac{1}{r}\boldsymbol{a}_v \\ -\frac{1}{r}\frac{\dot{r}}{r} \\ \|\dot{\boldsymbol{u}}\|^2 - \frac{1}{r}\boldsymbol{u}\cdot\boldsymbol{a}_v - \left(\frac{\dot{r}}{r}\right)^2 \\ 0 \end{bmatrix} + \boldsymbol{w} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{a}_v) + \boldsymbol{w} \tag{2.22}$$

where $\boldsymbol{a}_v$ is the follower's acceleration input and $\boldsymbol{w}$ is the process noise.

(b) Measurement Model

It is well-known that the range information is not always observable when only a 2-D image position of a target center is measured from a single camera[51]. To guarantee range observability, the line-of-sight from the camera to the target needs to be in motion[84]. However, in the formation flight application, there should not be a relative motion between the two aircraft and this may result in poor range estimation. To overcome this observability issue, an angle that is subtended by the leader's size (e.g. wingspan) is introduced as an additional measurement. The subtended angle α is defined by

$$\alpha = 2\tan^{-1}\frac{b}{2r} \tag{2.23}$$

The image processor developed in [24] tracks the target's contour in images, and so it can extract the target size. Figure 5 shows an example of the image tracking result.

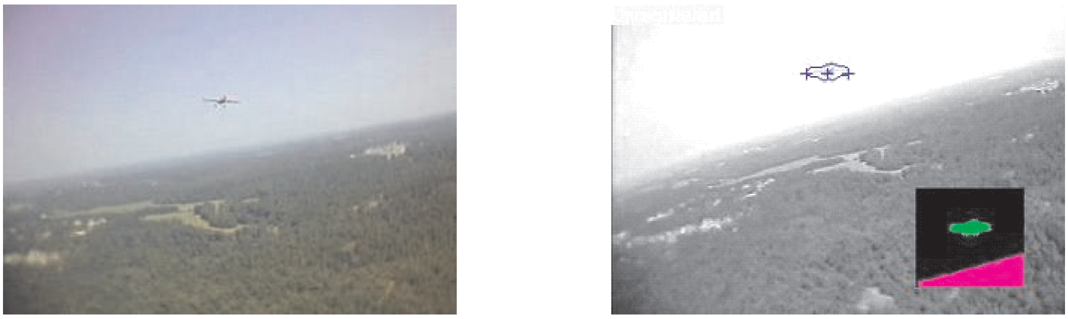


Figure 5: Image Tracking Result: (Left) Original image taken by the onboard camera, (Right) Center and two wing-tip positions of the leader airplane are detected

The image processor processes original images captured by the onboard camera, and then outputs image coordinates of the center and the two wing-tips of the leader airplane. Let $\boldsymbol{x}_c$ be an image coordinate of the center position, and $\boldsymbol{x}_l$ and $\boldsymbol{x}_r$ be those of the left and right wing-tip positions respectively. Since a pin-hole camera model is assumed, the subtended angle measurement can be calculated by using $\boldsymbol{x}_l$ and $\boldsymbol{x}_r$ as follows.

$$\alpha = 2\tan^{-1}\frac{\|\boldsymbol{x}_l - \boldsymbol{x}_r\|}{2} \tag{2.24}$$

The measurement vector in the EKF is chosen as

$$\boldsymbol{z} = \begin{bmatrix} \boldsymbol{u}_c^T & \alpha \end{bmatrix}^T \tag{2.25}$$

where $\boldsymbol{u}_c$ is the unit vector expressed in the camera frame F_C. The unit vector is chosen as a measurement instead of the two angles for the same reason as choosing it as an estimation state. The measurement vector $\boldsymbol{z}$ can be expressed as a nonlinear function of the estimation state $\boldsymbol{x}$. The measurement model is given by

$$\boldsymbol{z}_k = \begin{bmatrix} L_{c_k}\boldsymbol{u}_k \\ 2\tan^{-1}\frac{b_k}{2}\left(\frac{1}{r}\right)_k \end{bmatrix} + \boldsymbol{\nu}_k = \boldsymbol{h}(\boldsymbol{x}_k) + \boldsymbol{\nu}_k \tag{2.26}$$

where L_{c_k} is the known camera attitude at the time step t_k and $\boldsymbol{\nu}_k$ is a measurement error. The EKF prediction and update procedures discussed in the previous section are applied to the process and measurement models given in (2.22) and (2.26).

(c) State Constraint Enforcement

Since the unit vector $\boldsymbol{u}$ defined in (2.21) always satisfies $\|\boldsymbol{u}\| = 1$, the estimates of $\boldsymbol{u}$ and its changing rate $\dot{\boldsymbol{u}}$ in the state vector used in the EKF design have the following constraints.

$$\|\boldsymbol{u}\| = 1, \quad \boldsymbol{u}^T\dot{\boldsymbol{u}} = 0 \tag{2.27}$$

These constraints are naturally satisfied in the EKF prediction procedure (2.9), but not in the EKF update procedure (2.19). Therefore, the constraints (2.27) need to be enforced after the EKF update at each time step as follows.

$$\hat{\boldsymbol{u}}_k = \frac{\hat{\boldsymbol{u}}_k}{\|\hat{\boldsymbol{u}}_k\|} \tag{2.28}$$

$$\hat{\dot{\boldsymbol{u}}}_k = \hat{\dot{\boldsymbol{u}}}_k - \left(\hat{\boldsymbol{u}}_k^T\hat{\dot{\boldsymbol{u}}}_k\right)\hat{\boldsymbol{u}}_k \tag{2.29}$$

Calise proposed another method to enforce the state constraints in the EKF design[14]. In his work, the Kalman gain was computed by minimizing the augmented Lagrangian that includes the estimation error and the state constraint. This results in simply adding the correction term to the original EKF update laws (2.18, 2.19). However, this approach is not used in this work to avoid using a singular measurement error covariance matrix.

(d) Leader's State Estimation

Once the estimation state is updated by using a measurement at t_k in the EKF, the leader's estimated position and velocity can be calculated from the updated estimate $\hat{\boldsymbol{x}}_k$ and from the known follower's state as shown below.

$$\hat{\boldsymbol{X}}_{t_k} = \frac{\hat{\boldsymbol{u}}_k}{\widehat{\left(\frac{1}{r}\right)}_k} + \boldsymbol{X}_{v_k} \tag{2.30}$$

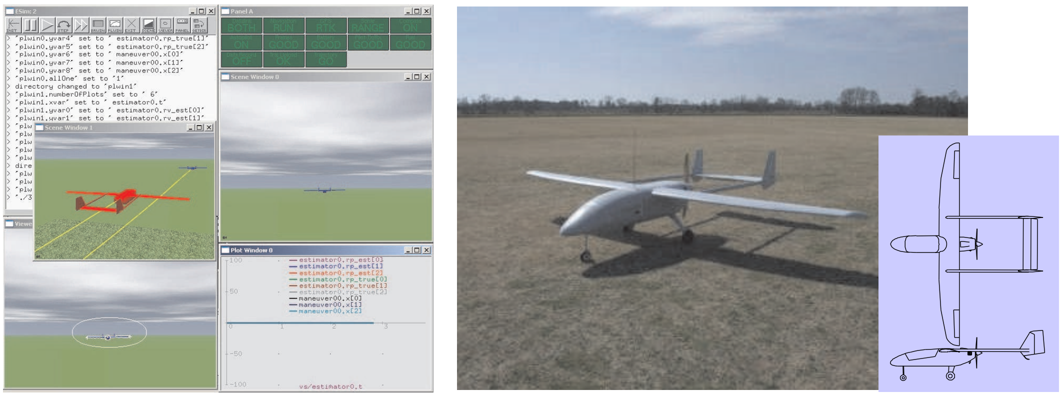
$$\hat{\boldsymbol{V}}_{t_k} = \frac{\hat{\dot{\boldsymbol{u}}}_k}{\widehat{\left(\frac{1}{r}\right)}_k} + \frac{\hat{\boldsymbol{u}}_k}{\widehat{\left(\frac{1}{r}\right)}_k} \cdot \widehat{\left(\frac{\dot{r}}{r}\right)}_k + \boldsymbol{V}_{v_k} \tag{2.31}$$

Since a non-accelerating target is assumed in the EKF design, the leader's acceleration is estimated by $\hat{\boldsymbol{a}}_{t_k} = \boldsymbol{0}$. These estimated leader states are fed back to the guidance system to maintain the formation.

2.3.2 Simulation Results

Figure 6 shows a 6 DoF multi-airplane flight simulation in a formation flight configuration (left) and the airplane model used in this simulation (right). The airplane is the MURI research airplane whose wingspan is 15.7 (ft) and overall length is 11 (ft). A basic controller and guidance system have already been implemented in this simulation. The controller is a neural-network based adaptive flight controller and it determines actuator commands based on the navigation system outputs and a position/velocity/attitude command[36]. In addition to that, the follower airplane has a camera and its images are also simulated. The synthetic images are processed and provide the locations of the leader's center and the two wing-tips. The synthetic image and the image processor outputs are shown in the left bottom window of the simulation interface in Figure 6.

The estimation performance of the suggested EKF design has been evaluated in this flight simulation. In the simulation, the leader airplane flies straight with a constant speed of 100 (ft/sec). The follower aircraft is commanded to maintain a given position relative to the leader. The formation is maintained by using the estimation result. Figure 7 compares the estimated (in blue) and the true (in green) relative positions and velocities. The vision-based estimation results match with their corresponding true state very well in the simulation. There is a bias in the range (X position) estimation. Even though the range estimation accuracy is improved after the vehicle's lateral motion due to the position command changes, a small bias remains. This is because of a measurement bias in the leader's size. However, the estimation is sufficiently accurate to achieve the closed-loop vision-based formation flight. Since the EKF-based estimation performance is validated in the simulation, it is implemented and tested in actual flights of UAVs. The next subsection discusses the flight experiment.



```
> 'plwin0.yvar4' set to ' estimator0.rp_true[1]'
> 'plwin0.yvar5' set to ' estimator0.rp_true[2]'
> 'plwin0.yvar6' set to ' maneuver00.x[0]'
> 'plwin0.yvar7' set to ' maneuver00.x[1]'
> 'plwin0.yvar8' set to ' maneuver00.x[2]'
> 'plwin0.allOne' set to '1'
> directory changed to 'plwin1'
> 'plwin1.numberOfPlots' set to ' 6'
> 'plwin1.xvar' set to ' estimator0.t'
> 'plwin1.yvar0' set to ' estimator0.rv_est[0]'
> 'plwin1.yvar1' set to ' estimator0.rv_est[1]'
> 'plw
> 'plw
> 'plw
> 'plw
> 'plw
> dire
> 'plw
> 'plw
> 'plw
> './3
```

```
estimator0.rp_est[0]
estimator0.rp_est[1]
estimator0.rp_est[2]
estimator0.rp_true[0]
estimator0.rp_true[1]
estimator0.rp_true[2]
maneuver00.x[0]
maneuver00.x[1]
maneuver00.x[2]
vs/estimator0.t
```

Figure 6: (Left) 6 DoF Multi-Airplane Flight Simulation Interface, (Right) MURI Research Airplane

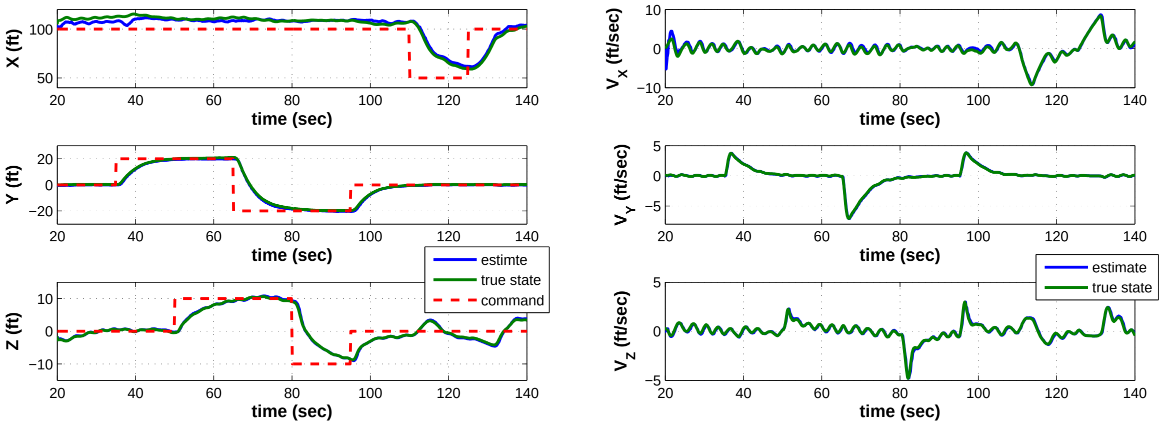


Figure 7: Estimated vs. True Relative Position and Velocity

2.3.3 Flight Test Results

(a) Platform Aircraft

The same image processing algorithm and the same relative navigation system used in the 6 DoF flight simulation have been implemented in the autonomous flight system of two different types of UAVs. Georgia Tech's unmanned helicopter, the GTMax, is used as a follower aircraft. The GTMax is based on the Yamaha RMax industrial helicopter. As an alternative follower aircraft, the GTYak, which is a model airplane of Yak-54, is used. The basic flight controller, own-ship navigation and guidance system of the GTMax and GTYak have already been developed[36][37]. For the vision-based guidance and control, the UAVs are equipped with cameras fixed to their bodies. The real-time image processor has also been implemented in the flight system of the GTMax and the GTYak. As a leader airplane, another Georgia Tech fixed wing airplane, the GTEdge, is used. The GTEdge is a 33% scale model of the aerobatic Edge 540T airplane. The GTEdge is a highly maneuverable airplane having the capability of vertical hovering flight[91]. The same adaptive flight controller used on the GTMax and the GTYak is used to fly the GTEdge. Table 1 summarizes specifications of the three UAVs used in the flight tests.

Table 1: Specifications of Georgia Tech UAVs

Name	**GTMax**	**GTYak**	**GTEdge**
Model	Yamaha RMax 	Yak-54 	Edge540T
Wingspan (ft)	-	8.50	8.75
Rotor Diameter (ft)	10.2	-	-
Overall Length (ft)	11.9	7.90	7.80
Empty Weight (lbs.)	157	40	35
Engine (cc)	246	100	100

(b) EKF Modification

In the flight test of the vision-based formation flight, the GTEdge is commanded to fly in a circle with a constant ground speed V_t and a constant angular rate ω_t. Unlike the straight level flight used in the simulations, a circling motion is chosen because of a limited flight test field size. Since the leader's motion does not satisfy the zero acceleration assumption, the EKF design discussed in Subsection 2.3.1 is slightly modified. Suppose that it is known that the leader is flying in a circle at a constant altitude. In the modified EKF design, the leader's lateral acceleration a_{lat} is added as an extra estimation state of the EKF. Since a_{lat} should be constant in a circling motion, the dynamics can be modeled by

$$\dot{a}_{lat} = 0 + w_{a_{lat}} \tag{2.32}$$

where $w_{a_{lat}}$ is a zero mean Gaussian noise. By assuming that the leader's lateral acceleration is perpendicular to the velocity vector in the horizontal X-Y plane, the leader's acceleration is estimated by

$$\boldsymbol{a}_t = \frac{a_{lat}}{\sqrt{V_{X_t}^2 + V_{Y_t}^2}} \begin{bmatrix} -V_{Y_t} \\ V_{X_t} \\ 0 \end{bmatrix} \tag{2.33}$$

where the leader's velocity vector $\boldsymbol{V}_t = \begin{bmatrix} V_{X_t} & V_{Y_t} & V_{Z_t} \end{bmatrix}^T$ is given in (2.31). In the process model (2.22), $\boldsymbol{a}_v$ should be replaced by $\boldsymbol{a}_v - \boldsymbol{a}_t$.

(c) Results

The first sustained closed-loop vision-based formation flight between the GTMax and the GTEdge was achieved in June, 2006 [35][38]. This may have been the first time automated formation flight based on vision only has ever been done. Figure 8 shows the GTMax and the GTEdge in a formation configuration. For the leader's circling motion, $V_t = 65$ (ft/sec) and $\omega_t = 0.1$ (rad/sec) were used. The GTMax is commanded to maintain a relative position $\boldsymbol{X}_{com} = \begin{bmatrix} 100 & -15 & -20 \end{bmatrix}^T$ (ft) in its body frame. This means that the GTMax follows the GTEdge from 100 (ft) behind, 15 (ft) inside the circle and 20 (ft) below. Formation flight performance was improved by feeding back the vision-based navigation output instead of the transmitted navigation output computed on the GTEdge. This is because the time delay due to the communication was removed. Figure 9 illustrates the image processing performance during the flight. The images from the onboard camera were processed at about 10 (Hz). Although the image processor sometimes failed to capture the leader airplane, it could still detect the leader even after several frames. Figure 10 compares the vision-based estimation results with the GPS-based estimates which are supposed to include very small errors. It can be concluded that the performance of the vision-based estimator is sufficiently accurate as long as the image processor's performance is good.

The closed-loop vision-based formation flight between two fixed-wing airplanes, the GTYak and the GTEdge, was achieved in July, 2007. In this flight test, the GTEdge flew in a circle with $V_t = 75$ (ft/sec) and $\omega_t = 0.1$ (rad/sec). Since the GTYak has more maneuverability than the GTMax, a higher speed was used. The relative position command was $\boldsymbol{X}_{com} = \begin{bmatrix} 80 & 0 & -20 \end{bmatrix}^T$ (ft). Figure 11 shows the GTYak and the GTEdge flying in formation. A left plot of Figure 12 shows the image processor outputs. The image processor was able to track the GTEdge with much less failures in this flight. A right plot of Figure 12 and Figure 13 show the vision-based estimates of the relative position, velocity and acceleration. The estimates are compared with the GPS-based estimates (shown in dashed green lines). As seen in the figures, the vision-based estimation performs accurately. The achievement of the closed-loop formation flight verified the estimation performance of the vision-based relative navigation filter design by using an EKF.

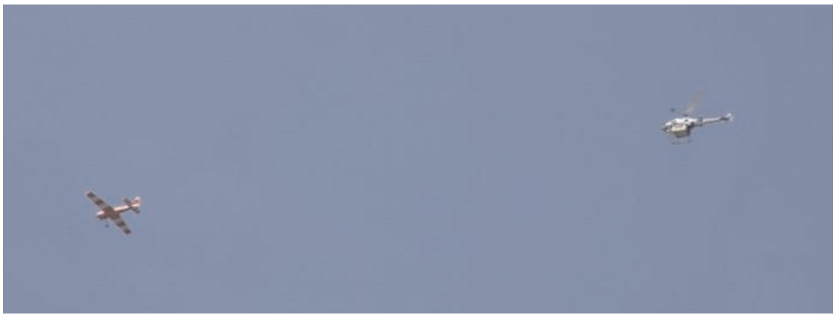


Figure 8: GTMax and GTEdge in Formation

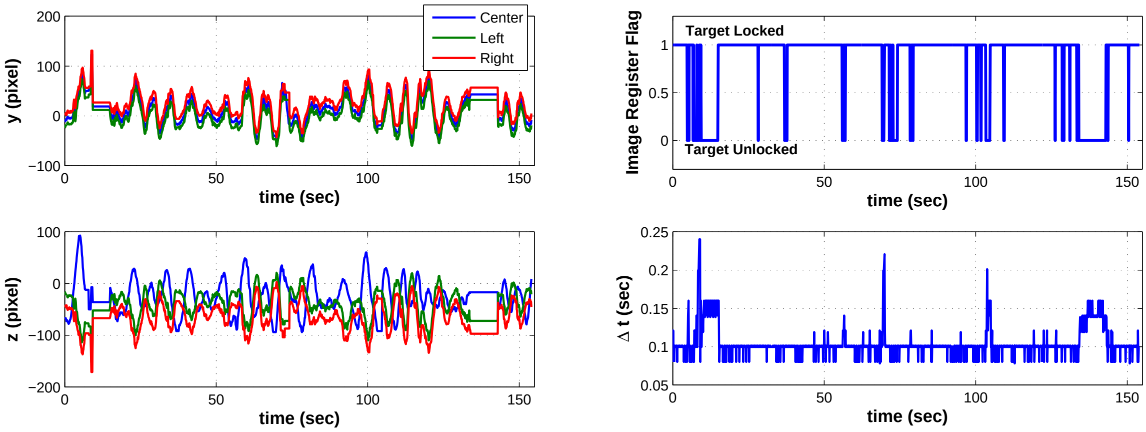


Figure 9: Image Processor Outputs and Processing Time

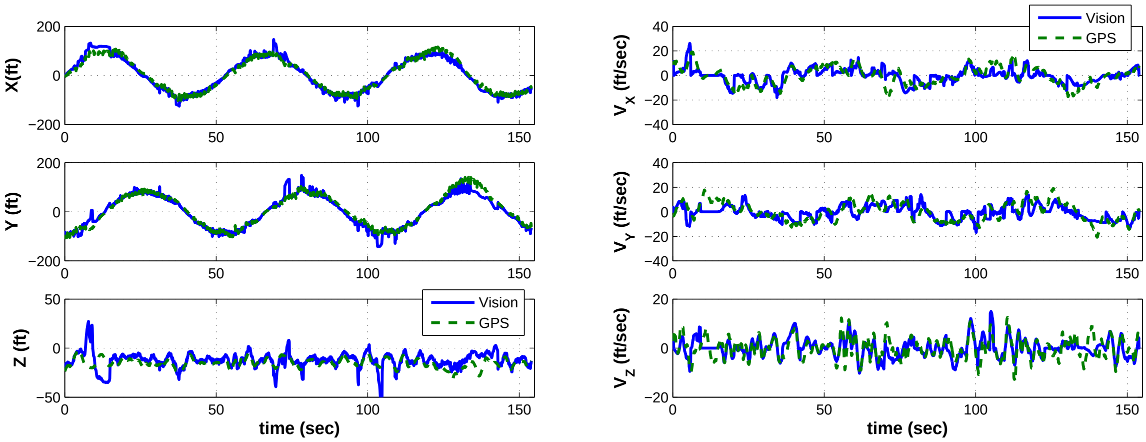


Figure 10: Vision-based vs. GPS-based Estimated Relative Position (Left) and Relative Velocity (Right)

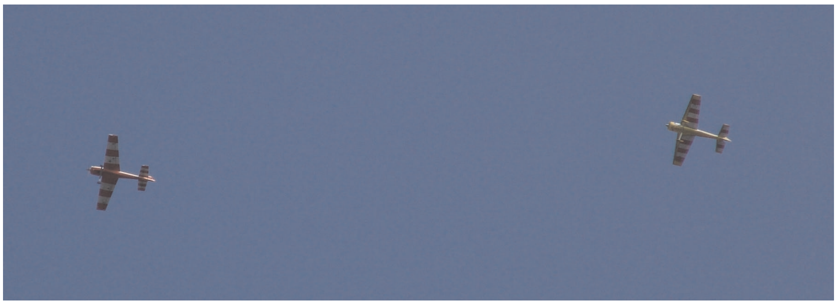


Figure 11: GTYak and GTEdge in Formation

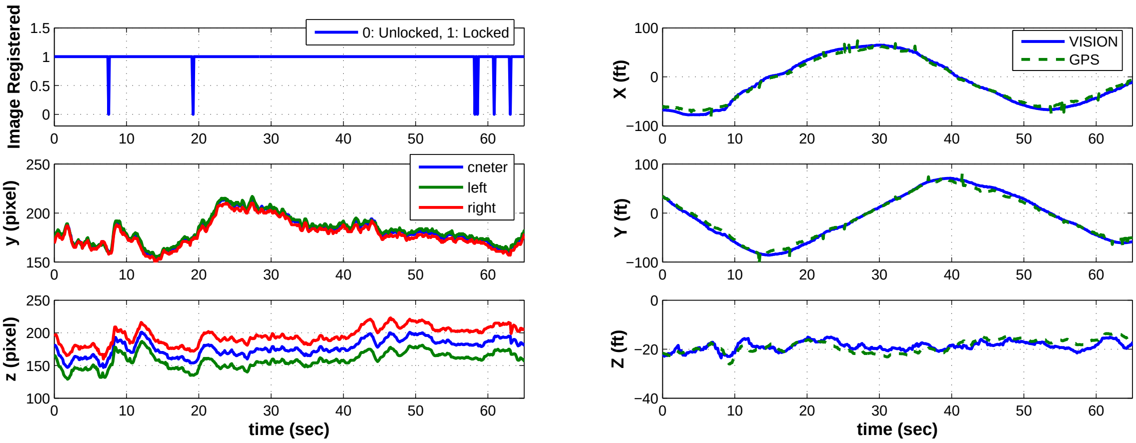


Figure 12: (Left) Image Processor Outputs, (Right) Vision-based vs. GPS-based Estimated Relative Position

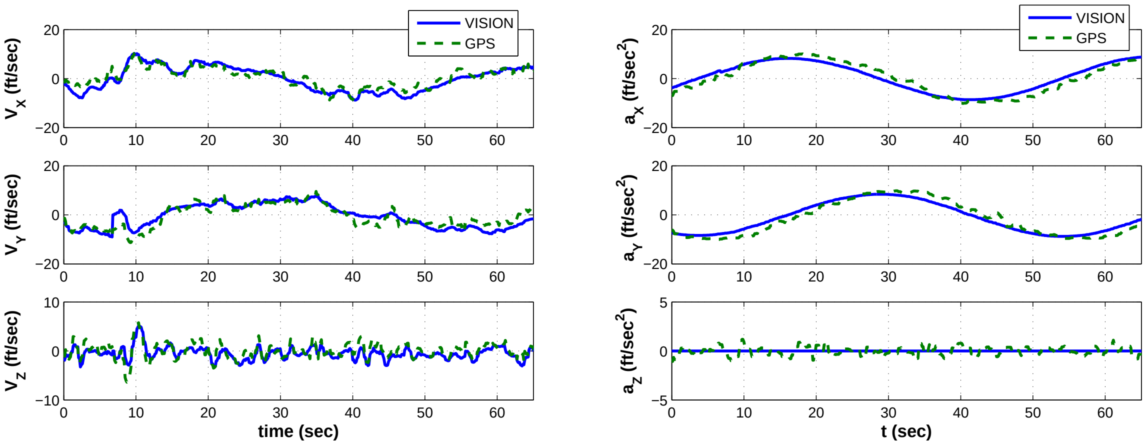


Figure 13: Vision-based vs. GPS-based Estimated Relative Velocity (Left), and Acceleration (Right)

2.4 Application 2 : Vision-Based 3-D Obstacle Modeling

The second application of vision-based navigation is 3-D obstacle modeling. This work is for automatic obstacle detection when a vehicle operates in the vicinity of unknown terrain or structures. Unlike the formation flight discussed in the previous section, there can be multiple targets to track, and the data correspondence problem needs to be solved in this application. In some studies, vision-based terrain modeling is achieved by tracking feature points in a sequence of images and updating estimates of their actual 3-D positions[54][50]. Unlike the feature point-based estimator, a line-based estimator design is developed in this thesis[85]. In urban areas, obstacles are mostly artificial structures (e.g.buildings) with sharp edges. By an image segmentation technique, such obstacle edges are detected as a set of straight line segments in an image. Therefore, the vision-based obstacle modeling can be performed by recovering the actual 3-D obstacle edges from the detected line segments.

2.4.1 Correspondence Problem

As stated above, the image processor detects line segments in an image, which correspond to obstacle edges. Each measured line segment (ℓ) is expressed by two endpoint positions in image coordinates as $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. Once the line segments are measured, they are associated with 3-D estimated lines in a database. Each estimated line ($\hat{L}$) is stored in the form of its two endpoint positions $\hat{\boldsymbol{X}}_1$ and $\hat{\boldsymbol{X}}_2$ in the inertial frame. Before examining the correspondence between the estimates and the measurements, all the 3-D estimated lines are predicted and projected onto the current image plane by using a known own-ship position $\boldsymbol{X}_v$ and a known camera attitude L_c. Hence, the projected line estimate ($\hat{\ell}^-$) is expressed by its two endpoints $\hat{\boldsymbol{x}}_1^-$ and $\hat{\boldsymbol{x}}_2^-$, which are given by

$$\hat{\boldsymbol{x}}_i^- = \frac{1}{\hat{X}_{c_i}^-}\begin{bmatrix}\hat{Y}_{c_i}^- \\ \hat{Z}_{c_i}^-\end{bmatrix}, \quad \hat{\boldsymbol{X}}_{c_i}^- = L_c\left(\hat{\boldsymbol{X}}_i^- - \boldsymbol{X}_v\right), \quad i = 1,2 \tag{2.34}$$

The statistical z-test is utilized to associate the measurement ℓ with the estimate $\hat{\ell}^{-}$. The z-test value is taken for a given error index J. It is defined by the square of the error J divided by its variance[29]. That is, the z-test value is inversely related to the likelihood of the correspondence between ℓ and $\hat{\ell}^{-}$. If there is a large error between them but the measurement has a large uncertainty, then the probability of their correspondence should be higher than the case in which the measurement has a small uncertainty. Therefore, each measured line segment should be assigned to the estimated line data which attains the least z-test value or the highest likelihood. Define the error index J as follows.

$$J = d_1^2 + d_2^2 + c(e_1^2 + e_2^2) \tag{2.35}$$

where d_1 and d_2 are signed distances perpendicular to the projected line estimate $\hat{\ell}^{-}$, and e_1 and e_2 are signed distances parallel to $\hat{\ell}^{-}$ (as shown in Figure 14), and c is a constant weight. Those distances are determined by $\ell = (\boldsymbol{x}_1, \boldsymbol{x}_2)$ and $\hat{\ell}^{-} = \left(\hat{\boldsymbol{x}}_1^{-}, \hat{\boldsymbol{x}}_2^{-}\right)$ as follows.

$$d_1 = \frac{\left(\boldsymbol{x}_1 - \hat{\boldsymbol{x}}_1^{-}\right) \times \left(\hat{\boldsymbol{x}}_1^{-} - \hat{\boldsymbol{x}}_2^{-}\right)}{\|\hat{\boldsymbol{x}}_1^{-} - \hat{\boldsymbol{x}}_2^{-}\|}, \qquad d_2 = \frac{\left(\boldsymbol{x}_2 - \hat{\boldsymbol{x}}_1^{-}\right) \times \left(\hat{\boldsymbol{x}}_1^{-} - \hat{\boldsymbol{x}}_2^{-}\right)}{\|\hat{\boldsymbol{x}}_1^{-} - \hat{\boldsymbol{x}}_2^{-}\|} \tag{2.36}$$

$$e_1 = \frac{\left(\boldsymbol{x}_1 - \hat{\boldsymbol{x}}_1^{-}\right) \cdot \left(\hat{\boldsymbol{x}}_1^{-} - \hat{\boldsymbol{x}}_2^{-}\right)}{\|\hat{\boldsymbol{x}}_1^{-} - \hat{\boldsymbol{x}}_2^{-}\|}, \qquad e_2 = \frac{\left(\boldsymbol{x}_2 - \hat{\boldsymbol{x}}_1^{-}\right) \cdot \left(\hat{\boldsymbol{x}}_1^{-} - \hat{\boldsymbol{x}}_2^{-}\right)}{\|\hat{\boldsymbol{x}}_1^{-} - \hat{\boldsymbol{x}}_2^{-}\|} \tag{2.37}$$

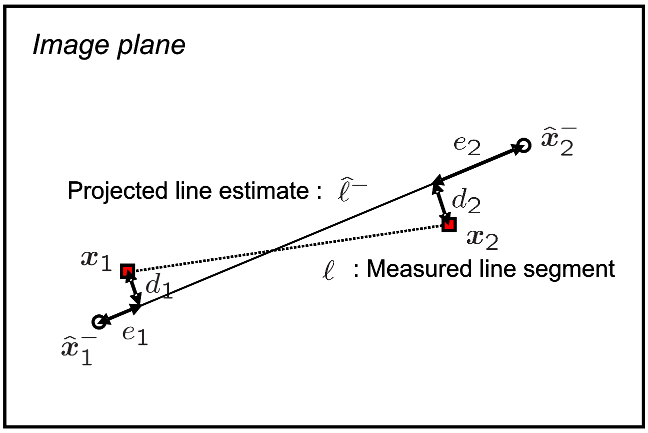


Image plane
e_2
$\hat{\boldsymbol{x}}_2^{-}$
Projected line estimate : $\hat{\ell}^{-}$
d_2
$\boldsymbol{x}_2$
$\boldsymbol{x}_1$
ℓ : Measured line segment
d_1
e_1
$\hat{\boldsymbol{x}}_1^{-}$

Figure 14: Projected Line Data and Measured Line Segment

Then, the z-test value of J is defined by

$$ztest = \frac{J^2}{\sigma_J^2} \tag{2.38}$$

where σ_J is the variance of the error index computed using the predicted estimation error covariance matrix P^- and the measurement error covariance matrix R.

For each measurement, the z-test value is calculated for all the estimates in the database. As stated above, the measurement is assigned to the line estimate which attains the least z-test value. However, when the least z-test value is larger than a given threshold value $ztest_{max}$, the measurement does not match well with the assigned line estimate. In such a case, the measured line segment is considered to be a newly detected obstacle edge and new line data is added to the database. On the other hand, if an existing line in the database is supposed to be visible but is not detected by the vision sensor, then the line may no longer exist and is deleted from the database. The line addition and deletion procedure will be discussed in detail in Subsection 2.4.3.

2.4.2 EKF Formulation

After all measurements are assigned, an EKF is applied to estimate the 3-D obstacle edge line from the assigned measurements. In this application, more than one measured line segment may be associated with a single obstacle edge. Suppose that the estimated edge $\hat{L} = \left(\hat{\boldsymbol{X}}_1, \hat{\boldsymbol{X}}_2\right)$ is updated by multiple measurements $\ell_1 = (\boldsymbol{x}_{1_1}, \boldsymbol{x}_{2_1})$, $\ell_2 = (\boldsymbol{x}_{1_2}, \boldsymbol{x}_{2_2})$, $\cdots$, then the EKF is designed to estimate the 3-D endpoint positions $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ from the residuals d_1 and d_2 defined in (2.36). That is, the estimation state vector and the measurement vector are taken as follows.

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{X}_1 \\ \boldsymbol{X}_2 \end{bmatrix}, \quad \boldsymbol{z} = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \tag{2.39}$$

Assuming static obstacles, the process model is given by $\dot{\boldsymbol{x}} = \mathbf{0}$. Without a process noise, this results in no change in the estimation state nor in the error covariance matrix by the EKF prediction procedure. When there is more than one measurement, the Kalman gain and the corresponding correction terms are computed individually for each assigned measurement. Then, the predicted estimate $\hat{\boldsymbol{x}}^-$ and its error covariance matrix P^- are updated by the sum of them. This is known as a sequential EKF update[57]. Let K_{k_i} denote the Kalman gain calculated by using the i-th measurement $\boldsymbol{z}_i$ that corresponds to the i-th measured line segment ℓ_i. The EKF update law is formulated by

$$P_k = P_k^- - \sum_i K_{k_i} H_k P_k^- \tag{2.40}$$

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_k^- + \sum_i K_{k_i} \left(\boldsymbol{z}_{k_i} - \boldsymbol{h}(\hat{\boldsymbol{x}}_k^-) \right) = \hat{\boldsymbol{x}}_k^- + \sum_i K_{k_i} \boldsymbol{z}_{k_i} \tag{2.41}$$

where the summations are taken over all of the assigned measurements ℓ_1, ℓ_2, $\cdots$. Since the measurement vector includes residuals from the projected line estimate, the estimated measurement $\boldsymbol{h}(\hat{\boldsymbol{x}}_k^-)$ becomes zero in (2.41).

2.4.3 Line Addition and Deletion

Since a camera's field of view changes due to the vehicle's motion, algorithms have been included for adding and deleting line data as obstacles enter and leave the field of view. As mentioned in Subsection 2.4.1, unassigned line segments are treated as newly detected obstacle edge lines and new estimated line data should be added to the database. In order to create a new line estimate, the initial estimate of its endpoint positions in a 3-D local frame F_I has to be determined from the 2-D information. When other information about the line is not available, a new line in the database is created by assuming that the line is on the zero altitude surface. With this assumption, the initial 3-D line estimate $\hat{L}_0 = \left(\hat{\boldsymbol{X}}_{1_0}, \hat{\boldsymbol{X}}_{2_0} \right)$ can be obtained from the measurement $\ell = (\boldsymbol{x}_1, \boldsymbol{x}_2)$ and the current camera attitude and the vehicle's position as follows. Suppose that the local frame F_I is chosen as the North-East-Down (NED) frame fixed on the zero altitude surface. Then, the initial guess for each of the two endpoints can be written by $\hat{\boldsymbol{X}}_i = \begin{bmatrix} \hat{X}_{i_0} & \hat{Y}_{i_0} & 0 \end{bmatrix}^T$ for $i = 1, 2$. By assuming zero measurement error, the following relationship has to be satisfied.

$$\hat{\boldsymbol{X}}_i = \boldsymbol{X}_v + L_c^T \boldsymbol{X}_c = \boldsymbol{X}_v + L_c^T \begin{bmatrix} 1 \\ \boldsymbol{x}_i \end{bmatrix} \hat{X}_{c_i} = \boldsymbol{X}_v + \bar{\boldsymbol{X}}_i \hat{X}_{c_i}, \quad i = 1, 2 \tag{2.42}$$

where $\hat{X}_{c_i}$ is the depth information and it is the only unknown parameter in the right hand side. Since $\hat{Z}_i = 0$ is assumed, the depth is given by $\hat{X}_{c_i} = -Z_v / \bar{Z}_i$, and finally the initial estimate $\hat{\boldsymbol{X}}_i$ is calculated by substituting it into (2.42).

On the other hand, if a line estimate in the database is supposed to be visible, but is not detected by the image processor, then the line estimate may not exist and it is deleted from the database. To ensure that only lines that do not exist are deleted, only lines with no measurement assigned for more than $N (\geq 1)$ consecutive time steps are removed. In addition to the algorithms for line addition and deletion, procedures for line extension and shrinkage are included[85].

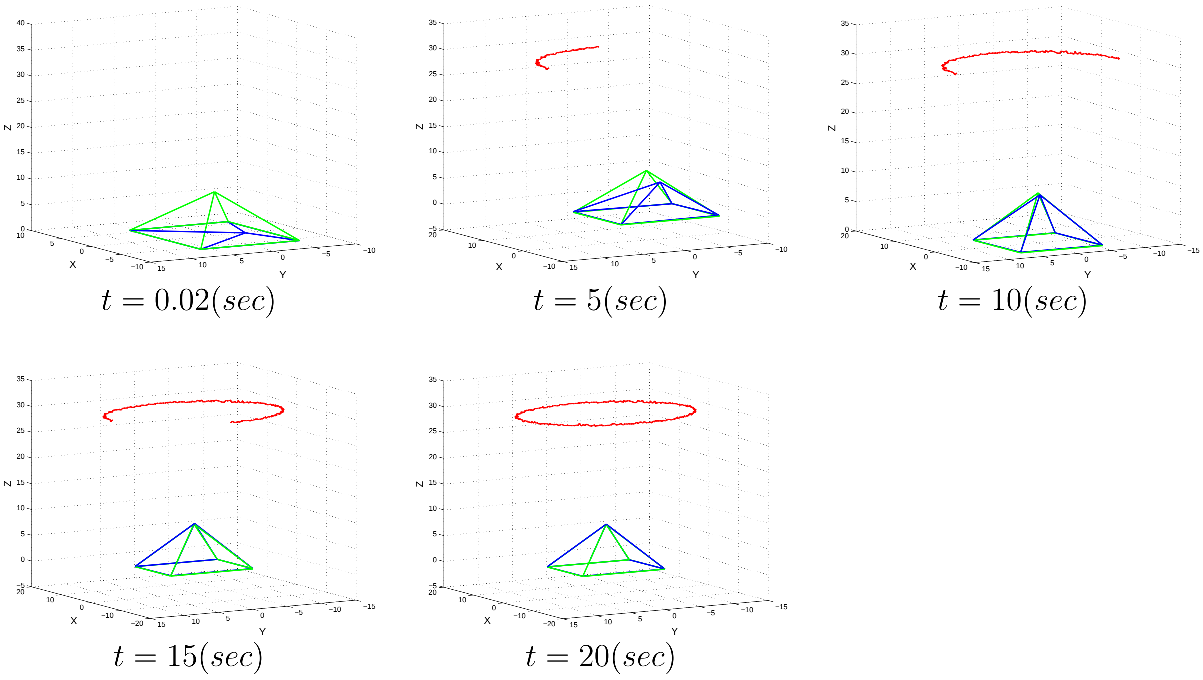
2.4.4 Simulation Results

(a) Preliminary Simulation : 3-D Obstacle Modeling

The line-based estimator developed in the previous subsections is applied and evaluated in simulations of 3-D obstacle modeling. In the simulations, image processor outputs are created by adding a random noise to the true image coordinate of each endpoint of the object edge line. $\Delta t = 0.02$ (sec) is used as the sampling time. Figure 15 and Figure 16 show simulation results for two different cases; Case 1 and Case 2. In Case 1, a vehicle flies over a pyramid shaped obstacle. The vehicle has a circling motion with a constant angular velocity $\omega = \pi/10$ (rad/sec), as shown in red in Figure 15. A camera is fixed on the vehicle's body with a mount angle of $\pi/6$ (rad) from the vertical downward axis (i.e., Z-axis in a body frame). In Figure 15, the modeling result (in blue) is compared with the actual obstacle edges (in green). From the result at the initial time step $t = 0.02$ (sec), it is seen that the initial estimates of the detected edge lines are created on the zero altitude surface. Then, the estimated edges come closer to the actual ones as the vehicle flies over the obstacle. After one revolution of the circle, all the eight edges of the pyramid are modeled very accurately in 3-D space. In this simulation, the whole obstacle is always within the camera's field of view. This means that the line addition and deletion algorithms introduced in Subsection 2.4.3 are not necessary. Figure 16 shows the modeling results of Case 2, in which the vehicle flies closer to a square-frame obstacle, and so the camera can only see part of the obstacle at each time instant. In this simulation, the vehicle has a zig-zag trajectory with the camera looking straight downward. At the initial time step $t = 0.02$ (sec), only a small part of the two edges were visible. As the vehicle moves, the camera detected more of the obstacle and the estimated edge lines in the database were properly extended or added. Even though the resulting obstacle model has some missing parts of one edge, the four edge lines are modeled accurately after the vehicle flies over the whole obstacle. These results verify the performance of the line addition and extension algorithms.

(b) Image-in-the-Loop Simulation : 3-D Grid Terrain Modeling

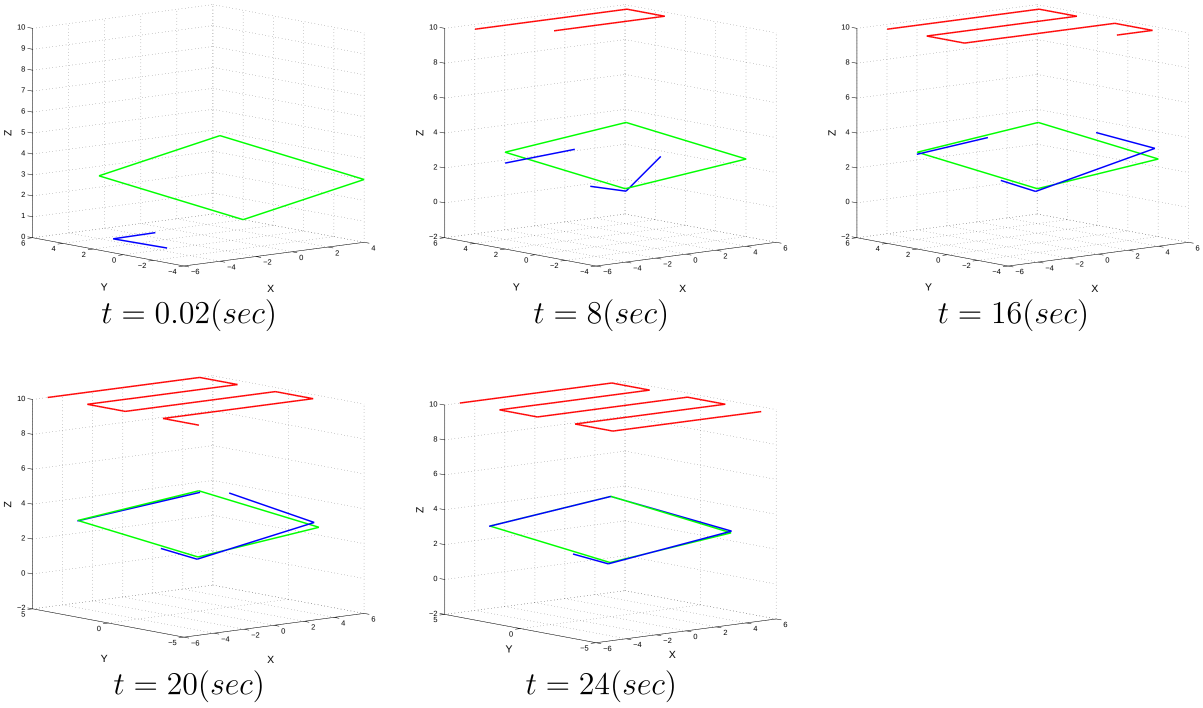
The line-based estimator can be applied to 3-D terrain mapping by partitioning the terrain into many grids. Figure 17 shows a vision-based 3-D terrain mapping simulation interface. The top right window illustrates current states including camera position and attitude, the estimated terrain model, and the actual grid terrain. The bottom right window shows a simulated camera image and the image segmentation outputs. A navigation system for the camera motion and an image processor have been implemented in this simulation. From the image segmentation technique developed in [78], the image processor computes each grid-point position in the image and the connectivity between all points. The connectivity corresponds to line segments which are measurements for the line-based estimator.



$t = 0.02(sec)$ $t = 5(sec)$ $t = 10(sec)$

$t = 15(sec)$ $t = 20(sec)$

Figure 15: 3-D Object Modeling Result (Case 1 : Pyramid, Circling Motion)

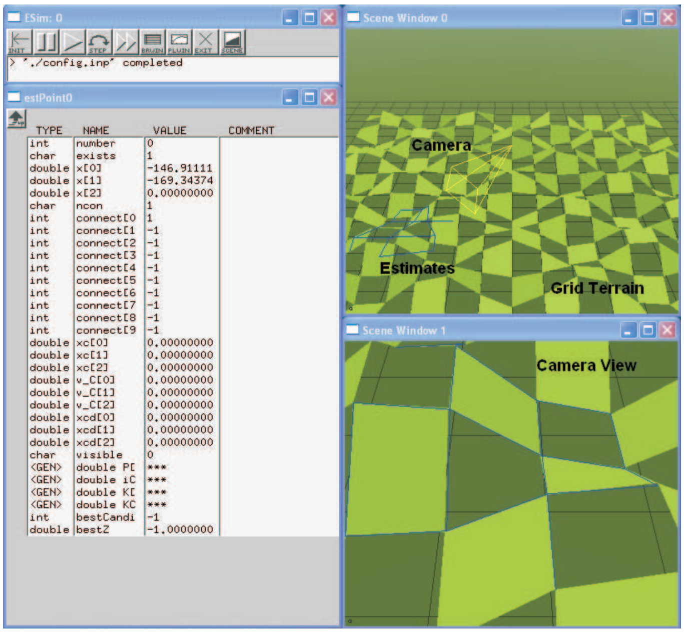


$t = 0.02(sec)$ $t = 8(sec)$ $t = 16(sec)$

$t = 20(sec)$ $t = 24(sec)$

Figure 16: 3-D Object Modeling Result (Case 2 : Square-Frame, Zig-Zag Motion)

In this simulation, the camera is moving over the terrain in a circle of radius 100 (ft) with a constant speed of 20 (ft/sec). The simulation runs for 200 seconds, and the camera flies through approximately 6 full revolutions. Figure 18 shows position estimation errors for three different sample points (Point 73, Point 139 and Point 227) that were randomly chosen. Figure 19 illustrates the 3-D terrain model constructed by the line-based estimator after times $t = 0, 50, 100, 150$ and 200 (sec). Since the initial estimate for each terrain edge is chosen on the zero altitude surface, the terrain model at the beginning is flat. As the camera flies over the terrain, the estimation errors converge to zero as shown in Figure 18. The last picture in Figure 19 demonstrates that a sufficiently accurate 3-D terrain model is obtained after 200 seconds. The estimation convergence is slow in this simulation because the camera is moving fast and each terrain edge is in the field of view for only a short time interval per each revolution. The rate of convergence can be improved by using a faster image processor and also by taking a larger initial covariance matrix in the EKF design.



```
> './config.inp' completed
```

TYPE	NAME	VALUE	COMMENT
int	number	0	
char	exists	1	
double	x[0]	-146.91111	
double	x[1]	-169.34374	
double	x[2]	0.00000000	
char	ncon	1	
int	connect[0	1	
int	connect[1	-1	
int	connect[2	-1	
int	connect[3	-1	
int	connect[4	-1	
int	connect[5	-1	
int	connect[6	-1	
int	connect[7	-1	
int	connect[8	-1	
int	connect[9	-1	
double	xc[0]	0.00000000	
double	xc[1]	0.00000000	
double	xc[2]	0.00000000	
double	v_C[0]	0.00000000	
double	v_C[1]	0.00000000	
double	v_C[2]	0.00000000	
double	xcd[0]	0.00000000	
double	xcd[1]	0.00000000	
double	xcd[2]	0.00000000	
char	visible	0	
<GEN>	double P[	***	
<GEN>	double iC	***	
<GEN>	double K[	***	
<GEN>	double KC	***	
int	bestCandi	-1	
double	bestZ	-1.0000000	

Figure 17: 3-D Grid Terrain Mapping Simulation

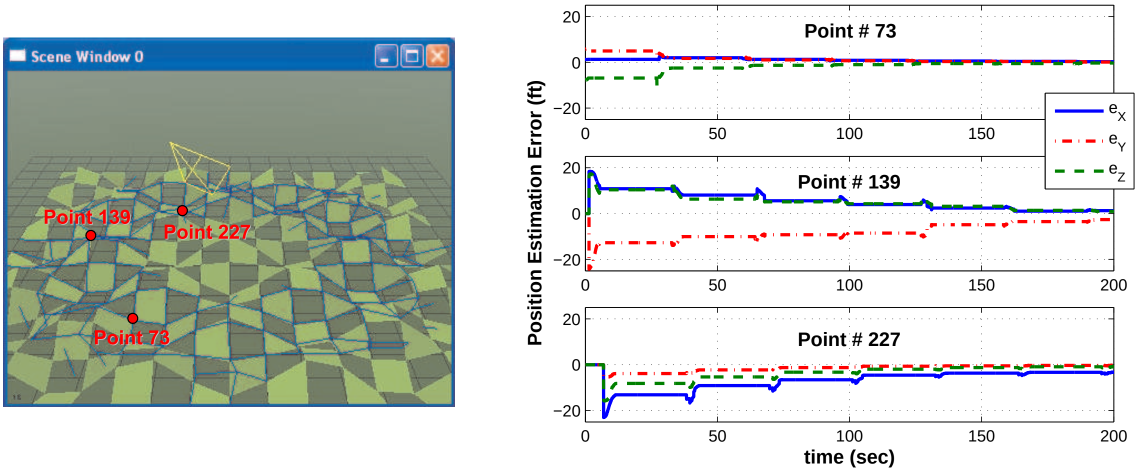
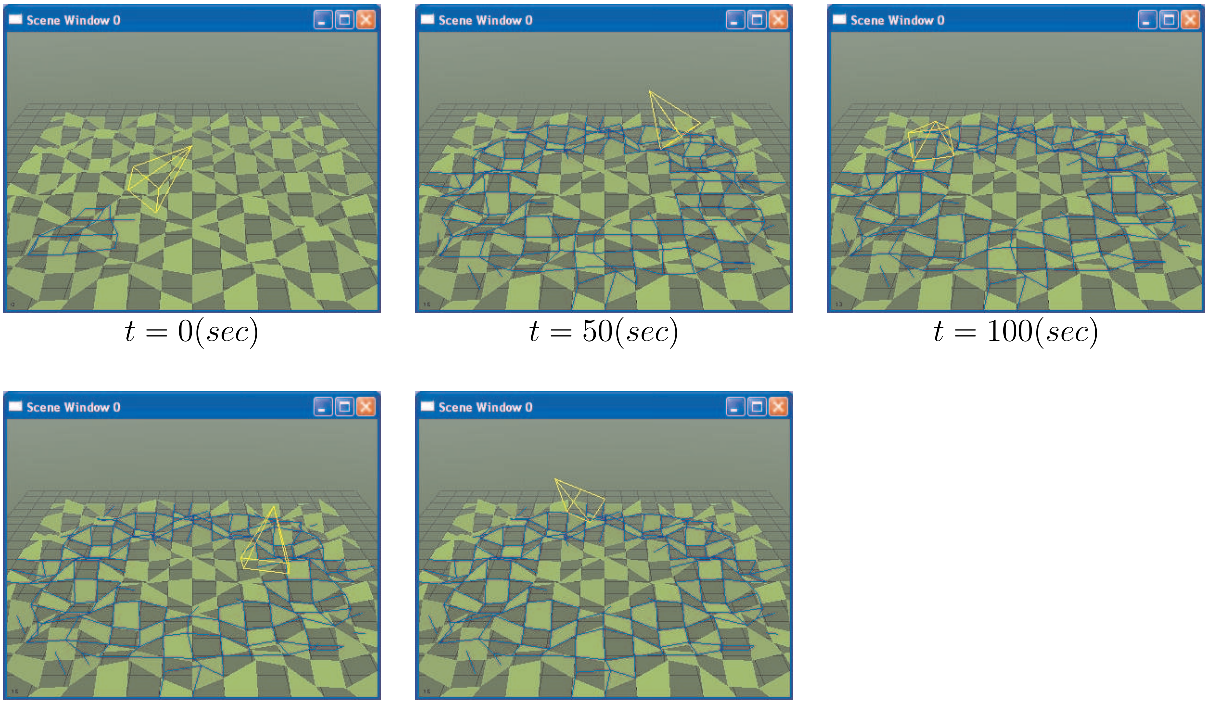


Figure 18: (Left) Locations of Three Sample Points: Point 73, Point 139 and Point 227, (Right) Position Estimation Errors



$t = 0(sec)$

$t = 50(sec)$

$t = 100(sec)$

$t = 150(sec)$

$t = 200(sec)$

Figure 19: Estimated vs. True Grid Terrain

2.5 Conclusion

This chapter designed a relative navigation system which estimates the relative state of a target with respect to a vehicle from 2-D information from a single camera by applying an extended Kalman filter. The EKF-based relative navigation design has been applied to vision-based formation flight and vision-based obstacle modeling problems. In the formation flight application, the target size was used as additional information so that range observability could be assured. The navigation filter has been tested in a 6 DoF multiple aircraft flight simulation and also in actual flights of UAVs. The achievement of closed-loop vision-based formation flight in flight tests proved its accurate estimation performance. In the vision-based obstacle modeling, a line-based estimator was suggested to recover 3-D obstacle edge lines from their 2-D projection onto each image plane. In addition to the EKF, a statistical z-test was used to address the correspondence problem between measurements and estimates. 3-D obstacle modeling accuracy of the line-based estimator has been verified in simulations with different shaped obstacles and with different camera motions. Then, the estimator was implemented and evaluated in the image-in-the-loop 3-D grid terrain mapping simulation.

CHAPTER III

VISION-BASED OBSTACLE AVOIDANCE

Vision-based obstacle avoidance is one of the most important and practical application of vision-based control of UAVs. UAVs are sometimes expected to operate in an uncertain environment that includes both fixed obstacles such as trees and buildings and moving obstacles such as other vehicles flying in proximity to them. Then, autonomous obstacle detection and avoidance is a requirement. Therefore, this chapter focuses on developing a vision-based navigation system for obstacle detection and a guidance law for obstacle avoidance. The EKF-based navigation discussed in Chapter 2 is applied. The guidance law is derived based on minimum-effort guidance for multiple target tracking[3][4]. The navigation and guidance systems are integrated with a real-time image processor, and autonomous vision-based obstacle avoidance with a static obstacle has been achieved in flight test.

3.1 Problem Formulation

Figure 20 summarizes the problem geometry of vision-based obstacle avoidance considered in this chapter. The linear vehicle dynamics model (2.1) is assumed. Let $\boldsymbol{a}_v = \begin{bmatrix} a_x & a_y & a_z \end{bmatrix}^T$ be the vehicle's acceleration input. In this problem, the vehicle is assumed to have a constant speed in the inertial X direction. It means that $a_x = 0$ is always applied and the vehicle is controlled by commanding a lateral acceleration a_y and a vertical acceleration a_z.

Consider that the vehicle is required to reach a given waypoint location $\boldsymbol{X}_{wp} = \begin{bmatrix} X_{wp} & Y_{wp} & Z_{wp} \end{bmatrix}^T$ that is expressed in the inertial frame F_I. Then, the waypoint

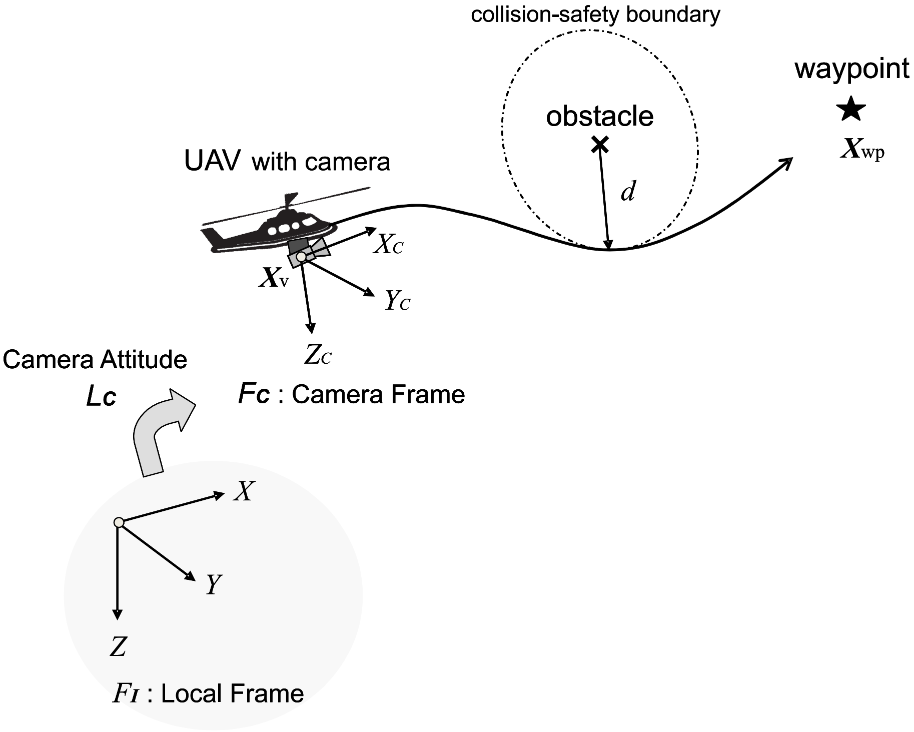


Figure 20: Problem Geometry of Vision-Based Obstacle Avoidance

tracking mission is achieved if

$$Y_v(t_f) = Y_{wp}, \quad Z_v(t_f) = Z_{wp} \tag{3.1}$$

where t_f is a time at which $X_v(t_f) = X_{wp}$ is satisfied. Since the waypoint position is known to the vehicle, this mission itself is not very difficult to accomplish. However, there may exist unforeseen obstacles on the vehicle's path to the waypoint. Let $\boldsymbol{X}_{obs}$ be an obstacle's position in F_I and assume $\dot{\boldsymbol{X}}_{obs} = \boldsymbol{0}$, i.e., stationary obstacles. In order to avoid obstacles, the vehicle is required to keep a minimum separation distance d from every obstacle's center position. Hence, as shown in Figure 20, a collision-safety boundary can be defined for each obstacle by a spherical surface with radius d and center at $\boldsymbol{X}_{obs}$. To achieve waypoint tracking without colliding with the obstacles, the vehicle needs to satisfy (3.1) while always maintaining

$$\|\boldsymbol{X}_{obs} - \boldsymbol{X}_v\| > d \tag{3.2}$$

for all obstacles. However, the obstacle's location $\boldsymbol{X}_{obs}$ is unknown to the vehicle, and so the guidance law can only access its estimate. The next section presents the navigation filter design that estimates $\boldsymbol{X}_{obs}$ by using 2-D vision-based information from an onboard camera.

3.2 Estimator Design

This section designs a vision-based estimator which estimates each obstacle's position from a 2-D vision-based measurement. First, the z-test algorithm used in Section 2.4 is applied to make a correspondence between measurements and estimated obstacles in a database. Then, an EKF is formulated to calculate the estimated 3-D obstacle positions.

3.2.1 Correspondence Problem

Since there can be multiple obstacles in the vehicle's surroundings, the image processor may detect more than one obstacle in the same image frame. Suppose that the image processor calculates the center position of the detected obstacles on each image frame. Let $\boldsymbol{X} = \boldsymbol{X}_{obs} - \boldsymbol{X}_v$ and $\boldsymbol{X}_c = L_c\boldsymbol{X}$ be a relative position vector of the i-th obstacle expressed in the inertial frame F_I and in the camera frame F_C, respectively. Then, a 2-D position of the obstacle measured at the k-th time step t_k is modeled by adding a measurement noise ν_k to (2.6).

$$\boldsymbol{z}_k = \frac{1}{X_{c_k}} \begin{bmatrix} Y_{c_k} \\ Z_{c_k} \end{bmatrix} + \boldsymbol{\nu}_k = \boldsymbol{h}(\boldsymbol{X}_k) + \boldsymbol{\nu}_k \tag{3.3}$$

$\boldsymbol{\nu}_k$ is a zero mean Gaussian discrete white noise process with its covariance matrix $R_k = \boldsymbol{E}\left[\boldsymbol{\nu}_k\boldsymbol{\nu}_k^T\right] = \sigma^2 I$. Suppose that n different obstacles (denoted by $\boldsymbol{z}_{k_1}$, $\boldsymbol{z}_{k_2}$, $\cdots$, $\boldsymbol{z}_{k_n}$) are detected on an image at t_k. Also suppose that the predicted estimate of the relative position of m obstacles (denoted by $\hat{\boldsymbol{X}}_{k_1}^-$, $\hat{\boldsymbol{X}}_{k_2}^-$, $\cdots$, $\hat{\boldsymbol{X}}_{k_m}^-$) have been obtained and stored in the database. In order to update each estimate correctly, it is very important to create the right correspondence between the measurements and the estimates before applying the EKF routine. As in Section2.4, the z-test is applied for this purpose. In this problem, the z-test value of the correspondence between the i-th measurement and the j-th estimate is calculated for the residual

$$\boldsymbol{r}_{ij} = \boldsymbol{z}_{k_i} - \hat{\boldsymbol{z}}_{k_j}^- = \boldsymbol{z}_{k_i} - \boldsymbol{h}(\hat{\boldsymbol{X}}_{k_j}^-) \tag{3.4}$$

Then the z-test value is defined by

$$ztest_{ij} = \boldsymbol{r}_{ij}^T \left(\boldsymbol{E}\left[\boldsymbol{r}_{ij}\boldsymbol{r}_{ij}^T\right]\right)^{-1} \boldsymbol{r}_{ij} = \boldsymbol{r}_{ij}^T \left(H_{k_j} P_{k_j}^- H_{k_j}^T + R_k\right)^{-1} \boldsymbol{r}_{ij} \tag{3.5}$$

where H_{k_j} is a measurement matrix defined by (2.15) and $P_{k_j}^-$ is a predicted estimation error covariance matrix associated with the j-th predicted estimate $\hat{\boldsymbol{X}}_{k_j}^-$. As stated in Subsection 2.4.1, a small z-test value $ztest_{ij}$ indicates a high correspondence of a chosen pair $(\boldsymbol{z}_{k_i}, \hat{\boldsymbol{X}}_{k_j}^-)$. For each measurement, the z-test value is calculated for every predicted estimate. Then the i-th measurement $\boldsymbol{z}_{k_i}$ updates the predicted estimate $\hat{\boldsymbol{X}}_{k_j}^-$ if

$$ztest_{ij} = \min_{j=1,2,\cdots,m} ztest_{ij} \quad AND \quad ztest_{ij} \leq ztest_{max} \tag{3.6}$$

is satisfied. $ztest_{max}$ is a given threshold value of the z-test value.

When the least value of $ztest_{ij}$ is still larger than the threshold, the measurement is considered to come from a newly detected obstacle and a new estimated obstacle data $\hat{\boldsymbol{X}}_{k_{m+1}}$ is added to the existing data set. Assume that some knowledge about a range (r_0) and its error standard deviation (σ_{r_0}) is available. Then the $(m+1)$-th new estimate and its error covariance matrix are created as follows.

$$\hat{\boldsymbol{X}}_{k_{m+1}} = r_0 \begin{bmatrix} 1 \\ \boldsymbol{z}_{k_i} \end{bmatrix}, \quad P_{k_{m+1}} = \sigma_{r_0}^2 \begin{bmatrix} 1 & \boldsymbol{0} \\ \boldsymbol{0} & R_k \end{bmatrix} \tag{3.7}$$

After all the n measurements' correspondences are made, there may remain a predicted estimate which was not updated by any of the measurements. This happens when the corresponding obstacle lies outside of the camera's field of view or when the image processor fails to detect it. For such an estimate, only the EKF prediction procedure is executed. The next subsection formulates the EKF update and prediction procedures.

3.2.2 EKF Formulation

An EKF is formulated to estimate the relative position of an obstacle from its 2-D image position measurement. Since static obstacles are assumed, obstacle motion dynamics are given by

$$\dot{\boldsymbol{X}}_{obs} = \boldsymbol{0} \tag{3.8}$$

Therefore, the EKF prediction process (2.9,2.16) is written by:

$$\begin{aligned} \hat{\boldsymbol{X}}_k^- &= \hat{\boldsymbol{X}}_{obs_{k-1}} - \boldsymbol{X}_{v_k} \qquad (3.9) \\ P_k^- &= P_{k-1} + Q_k \qquad (3.10) \end{aligned}$$

where $\boldsymbol{X}_{v_k}$ is the known vehicle position and Q_k is the covariance matrix of the process noise at t_k. The form $Q_k = \sigma_X^2 I \cdot \Delta t_k$ is used in this filter design. $\Delta t_k = t_k - t_{k-1}$ is the sampling time. Now recall the EKF update procedures (2.17-2.19).

$$\begin{aligned} \hat{\boldsymbol{X}}_k &= \hat{\boldsymbol{X}}_k^- + K_k \left(\boldsymbol{z}_k - \boldsymbol{h}(\hat{\boldsymbol{X}}_k^-) \right) \qquad (3.11) \\ P_k &= P_k^- - K_k H_k P_k^- \qquad (3.12) \\ K_k &= P_k^- H_k^T \left(H_k P_k^- H_k^T + R_k \right)^{-1} \qquad (3.13) \end{aligned}$$

where a measurement matrix H_k is derived as follows.

$$H_k = \frac{\partial \boldsymbol{h}(\hat{\boldsymbol{X}}_k^-)}{\partial \hat{\boldsymbol{X}}_k^-} = \frac{1}{\hat{X}_{c_k}^-} \begin{bmatrix} -\frac{\hat{Y}_{c_k}^-}{\hat{X}_{c_k}^-} & 1 & 0 \\ -\frac{\hat{Z}_{c_k}^-}{\hat{X}_{c_k}^-} & 0 & 1 \end{bmatrix} L_{c_k} = \frac{1}{\hat{X}_{c_k}^-} \begin{bmatrix} -\boldsymbol{h}(\hat{\boldsymbol{X}}_k^-) & I \end{bmatrix} L_{c_k} \tag{3.14}$$

After the EKF update, an absolute position estimate of the obstacle is computed by adding the known vehicle position to the relative position estimate.

$$\hat{\boldsymbol{X}}_{obs_k} = \hat{\boldsymbol{X}}_k + \boldsymbol{X}_{v_k} \tag{3.15}$$

3.3 Collision Criteria

For the purpose of obstacle avoidance, each obstacle in the estimate set is examined to determine if it is critical to the vehicle using the latest updated estimate of the obstacle positions. Chakravarthy and Ghose suggested a 2-D collision cone approach to establish a collision criteria[15]. In the collision cone approach, a collision cone is defined for each obstacle and an obstacle is considered to be critical if the vehicle's velocity vector relative to the obstacle lies within its collision cone.

This thesis applies their 2-D collision cone criteria to a 3-D obstacle avoidance problem by only considering a 2-D plane which includes the relative position vector $\boldsymbol{X}$ and the relative velocity vector $\boldsymbol{V} = -\boldsymbol{V}_v$. As discussed in Section 3.1, the vehicle is required to maintain a minimum separation distance d from every obstacle, and a collision-safety boundary of each obstacle is taken as a spherical surface with radius d and center at the obstacle position $\boldsymbol{X}_{obs}$. Then a collision cone is defined by a set of tangential lines from the vehicle to the obstacle's collision-safety boundary. As shown in Figure 21, the spherical collision-safety boundary appears as a circle in the 2-D plane formed by $\boldsymbol{X}$ and $\boldsymbol{V}$, and hence the collision cone is specified by two vectors (denoted by $\boldsymbol{p}_1$, $\boldsymbol{p}_2$) originating at the vehicle position $\boldsymbol{X}_v$ and tangential to the boundary circle. The vectors $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ can be expressed as follows.

$$\boldsymbol{p}_i = \boldsymbol{X} + d\boldsymbol{u}_i, \quad i = 1, 2 \tag{3.16}$$

where $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ are unit vectors from the obstacle position to the two tangential points:

$$\begin{cases} \boldsymbol{u}_1 = -\frac{1}{\|\boldsymbol{X}\|^2}\left(c(\boldsymbol{X} \cdot \boldsymbol{V}_v) + d\right) + c\boldsymbol{V}_v \\ \boldsymbol{u}_2 = \frac{1}{\|\boldsymbol{X}\|^2}\left(c(\boldsymbol{X} \cdot \boldsymbol{V}_v) - d\right) - c\boldsymbol{V}_v, \qquad c = \sqrt{\frac{\|\boldsymbol{X}\|^2 - d^2}{\|\boldsymbol{X}\|^2\|\boldsymbol{V}_v\|^2 - (\boldsymbol{X} \cdot \boldsymbol{V}_v)^2}} \end{cases} \tag{3.17}$$

The vehicle velocity is decomposed into two components; one parallel to $\boldsymbol{p}_1$ and the other parallel to $\boldsymbol{p}_2$.

$$\boldsymbol{V}_v = a\boldsymbol{p}_1 + b\boldsymbol{p}_2 \tag{3.18}$$

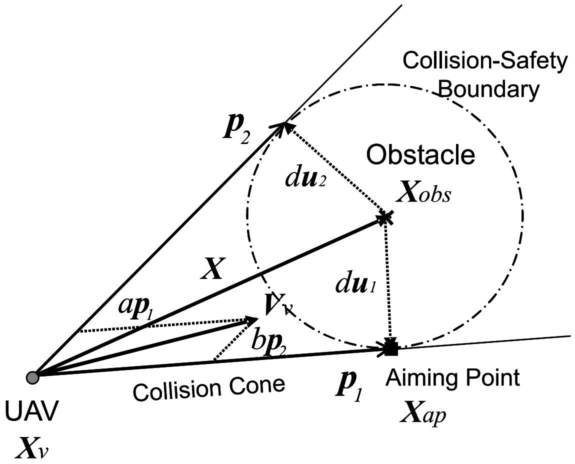


Figure 21: Collision Cone and Aiming Point

where the coefficients a and b are calculated as follows.

$$a = \frac{1}{2}\left(\frac{\boldsymbol{X}\cdot\boldsymbol{V}_v}{\|\boldsymbol{X}\|^2 - d^2} + \frac{1}{cd}\right), \quad b = \frac{1}{2}\left(\frac{\boldsymbol{X}\cdot\boldsymbol{V}_v}{\|\boldsymbol{X}\|^2 - d^2} - \frac{1}{cd}\right) \tag{3.19}$$

Since the approach is to examine if the vehicle's velocity vector $\boldsymbol{V}_v$ lies within the collision cone, the collision cone criterion is given by

$$a > 0 \quad AND \quad b > 0 \tag{3.20}$$

When the collision cone criterion (3.20) is satisfied, the vehicle is considered to be in danger of colliding with the obstacle and should execute some avoidance maneuver. The aiming point $\boldsymbol{X}_{ap}$ to be used for obstacle avoidance is given by

$$\boldsymbol{X}_{ap} = \begin{bmatrix} X_{ap} \\ Y_{ap} \\ Z_{ap} \end{bmatrix} = \begin{cases} \boldsymbol{p}_1 + \boldsymbol{X}_v, & 0 < b \leq a \\ \boldsymbol{p}_2 + \boldsymbol{X}_v, & 0 < a < b \end{cases} \tag{3.21}$$

Since the vehicle has constant speed in the X-direction, a time-to-go to the aiming point is derived as

$$t_{go} = t_k + \frac{X_{ap} - X_{v_k}}{U_{v_k}} \tag{3.22}$$

where U_{v_k} is the vehicle's speed in the X direction at the time step t_k. When $(t_{go} - t_k)$ is larger than a given threshold T, there is no urgency for the vehicle to execute an avoidance maneuver. Also, if it is negative or t_{go} is larger than the terminal time t_f, there is no chance of collision. Therefore, in addition to the collision cone criterion (3.20), we impose the following time-to-go criterion.

$$t_{go} - t_k < T \quad AND \quad 0 < t_{go} < t_f \tag{3.23}$$

An obstacle is considered to be critical only if both (3.20) and (3.23) are satisfied. If there is more than one critical obstacles, the one having the smallest time-to-go is chosen as the most critical obstacle. An avoiding maneuver is taken for the most critical obstacle.

3.4 Guidance Design

In this section, a guidance design to achieve waypoint tracking (3.1) with obstacle avoidance is presented. If there is a critical obstacle, the vehicle is guided to the aiming point specified in (3.21) so that it can avoid the obstacle. Refs. [27] and [87] design guidance laws by applying proportional navigation (PN) sequentially. On the other hand, this thesis proposes a guidance design based on the minimum-effort guidance (MEG) derived by Ben-Asher[3][4]. It is shown that the control cost can be reduced by using the suggested MEG-based guidance instead of the sequential PN approach.

3.4.1 Proportional Navigation for Waypoint Tracking

When there is no critical obstacle, the vehicle is guided directly to the waypoint. Then the vehicle's acceleration input for waypoint tracking can be derived by solving the following minimization problem.

$$\min_{\boldsymbol{a}_v} J = \frac{1}{2}\int_{t_k}^{t_f} \boldsymbol{a}_v^T(t)\boldsymbol{a}_v(t)dt = \frac{1}{2}\int_{t_k}^{t_f} \left(a_y(t)^2 + a_z(t)^2\right) dt \tag{3.24}$$

subject to the vehicle dynamics (2.1), with a terminal constraint (3.1). The terminal time t_f is given by

$$t_f = t_k + \frac{X_{wp} - X_{v_k}}{U_{v_k}} \tag{3.25}$$

Since the waypoint location and the vehicle's own-ship state are known, the optimal guidance is able to be realized. The solution of (3.24) can be analytically obtained by solving the Euler-Lagrange equations[13].

$$\boldsymbol{a}_{v_k}^* = 3\left(\frac{1}{(t_f - t_k)^2}\begin{bmatrix} 0 \\ Y_{wp} - Y_{v_k} \\ Z_{wp} - Z_{v_k} \end{bmatrix} - \frac{1}{(t_f - t_k)}\begin{bmatrix} 0 \\ V_{v_k} \\ W_{v_k} \end{bmatrix}\right) \tag{3.26}$$

This solution is the well-known PN guidance, which is considered to be a simple and very effective strategy in target interception[92].

3.4.2 Minimum-Effort Guidance for Obstacle Avoidance

When there is a critical obstacle, a corresponding aiming point $\boldsymbol{X}_{ap}$ and time-to-go t_{go} are provided from the collision criteria. However, since the obstacles' true positions are unknown, the guidance system can only access their estimated values $\hat{\boldsymbol{X}}_{ap}$ and $\hat{t}_{go}$ which are calculated from the estimated obstacle position $\hat{\boldsymbol{X}}_{obs}$. In order to avoid collision with the most critical obstacle, the vehicle should fly towards the aiming point.

(a) Sequential Proportional Navigation

A sequential PN-based approach was suggested by Han and Bang[27]. This method is derived by minimizing the control cost over a path from the vehicle's current position to the aiming point, and then over a path from the aiming point to the waypoint separately. Therefore, the vehicle's acceleration input at the current time t_k is obtained by solving

$$\min_{\boldsymbol{a}_v} J = \frac{1}{2}\int_{t_k}^{\hat{t}_{go}} \boldsymbol{a}_v^T(t)\boldsymbol{a}_v(t)dt = \frac{1}{2}\int_{t_k}^{t_f} \left(a_y(t)^2 + a_z(t)^2\right) dt \tag{3.27}$$

with the terminal constraint

$$Y_v(\hat{t}_{go}) = \hat{Y}_{ap}, \quad Z_v(\hat{t}_{go}) = \hat{Z}_{ap} \tag{3.28}$$

This is the same minimization problem as formulated in Subsection 3.4.1 and its solution is given by

$$\hat{\boldsymbol{a}}_{v_k}^{PN} = 3 \left(\frac{1}{(\hat{t}_{go} - t_k)^2} \begin{bmatrix} 0 \\ \hat{Y}_{ap} - Y_{v_k} \\ \hat{Z}_{ap} - Z_{v_k} \end{bmatrix} - \frac{1}{(\hat{t}_{go} - t_k)} \begin{bmatrix} 0 \\ V_{v_k} \\ W_{v_k} \end{bmatrix} \right) \tag{3.29}$$

(b) Minimum-Effort Guidance

As an alternative, the minimum-effort guidance (MEG) for multiple targets tracking is applied in this thesis. In this problem, the aiming point $\hat{\boldsymbol{X}}_{ap}$ and the waypoint $\boldsymbol{X}_{wp}$ are considered as two targets to be tracked. The minimum-effort guidance law is derived by solving

$$\min_{\boldsymbol{a}_v} J = \frac{1}{2} \int_{t_k}^{t_f} \boldsymbol{a}_v^T(t) \boldsymbol{a}_v(t) dt = \frac{1}{2} \int_{t_k}^{t_f} \left(a_y(t)^2 + a_z(t)^2 \right) dt \tag{3.30}$$

subject to the vehicle's dynamics (2.1) with the following interior and terminal point constraints.

$$\begin{cases} Y_v(\hat{t}_{go}) = \hat{Y}_{ap}, \quad Z_v(\hat{t}_{go}) = \hat{Z}_{ap} \\ Y_v(t_f) = Y_{wp}, \quad Z_v(t_f) = Z_{wp} \end{cases} \tag{3.31}$$

Since the MEG-based guidance law minimizes the control effort over the vehicle's entire flight path while the sequential PN guidance minimizes it only for the vehicle's path to reach the aiming point, the MEG-based guidance law is expected to make the vehicle achieve the mission with less control effort than the sequential PN guidance law does. From the Euler-Lagrange equation, this minimization problem can be solved analytically[13]. The resulting optimal solution is given as follows[82][83].

$$\hat{\boldsymbol{a}}_{v_k}^{MEG} = \hat{\boldsymbol{a}}_{v_k}^{PN} + \Delta \boldsymbol{a}_{v_k}^{MEG} \tag{3.32}$$

where $\hat{\boldsymbol{a}}_{v_k}^{PN}$ is the PN guidance input for the aiming point tracking given in (3.29) and $\Delta\boldsymbol{a}_{v_k}^{MEG}$ is an additional guidance input given by

$$\Delta\boldsymbol{a}_{v_k}^{MEG} = \frac{3}{3(\hat{t}_{go}-t_k)+4(t_f-\hat{t}_{go})}\left(\frac{3}{\hat{t}_{go}-t_k}\begin{bmatrix}0\\ \hat{Y}_{ap}-Y_{v_k}\\ \hat{Z}_{ap}-Z_{v_k}\end{bmatrix}-\frac{2}{t_f-\hat{t}_{go}}\begin{bmatrix}0\\ Y_{wp}-\hat{Y}_{ap}\\ Z_{wp}-\hat{Z}_{ap}\end{bmatrix}-\begin{bmatrix}0\\ V_{v_k}\\ W_{v_k}\end{bmatrix}\right) \tag{3.33}$$

A derivation of the MEG-based guidance (3.32,3.33) is presented in Appendix A. In this thesis, only a case of single waypoint tracking is considered. However, the MEG-based guidance law can be easily derived for the case of multiple waypoint tracking[3][4].

3.5 Simulation Results

The EKF-based estimator, the collision criteria and the guidance system for vision-based obstacle avoidance designed above have been implemented and evaluated in simulations. First, the algorithms are evaluated in simulations using simulated image measurements. Simulation results are presented comparing the sequential PN guidance law given in (3.29) and the suggested MEG-based guidance law given in (3.32). Then, the navigation and guidance algorithms are implemented and tested in a 6 DoF image-in-the-loop UAV flight simulation along with the real time image processor.

3.5.1 Preliminary Simulation Results

Simulation of vision-based obstacle avoidance is performed with simulated image measurements for the case of two obstacles. In this simulation, measurements are provided by a sum of a true obstacle's position on each image frame and a random measurement noise. The main purpose of this simulation is to show that the control effort required to achieve the waypoint tracking and obstacle avoidance mission is reduced by using the suggested MEG-based guidance law instead of the sequential PN guidance.

(a) Simulation Settings

A mission given to the vehicle is to reach a waypoint at $\boldsymbol{X}_{wp} = \begin{bmatrix} 80 & 0 & 0 \end{bmatrix}^T$ (ft) from the origin $\boldsymbol{X}_0 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$ (ft). The vehicle has a constant speed $U_v = 10$ (ft/sec) in the X direction. There are two obstacles, called Obstacle 1 and Obstacle 2, on the vehicle's way from $\boldsymbol{X}_0$ to $\boldsymbol{X}_{wp}$. The obstacles are located at

$$\boldsymbol{X}_{obs_1} = \begin{bmatrix} 20 \\ -5 \\ 5 \end{bmatrix} (ft), \quad \boldsymbol{X}_{obs_2} = \begin{bmatrix} 50 \\ 5 \\ -5 \end{bmatrix} (ft)$$

and are unknown to the vehicle. A minimum safety distance is given by $d = 10$ (ft), and $T = 5$ (sec) is used for the time-to-go threshold in this simulation. It is assumed that the camera's attitude always coincides with the inertial frame, i.e., $L_c = I$. Let $\boldsymbol{X}_i = \boldsymbol{X}_{obs_i} - \boldsymbol{X}_v(t_k) = \begin{bmatrix} X_{i_k} & Y_{i_k} & Z_{i_k} \end{bmatrix}^T$ be the relative position of the i-th obstacle ($i = 1, 2$) with respect to the vehicle at time step t_k. Assuming a 180-degree field of view for the camera, image processor outputs are created as follows when $X_{i_k} > f = 1$.

$$\boldsymbol{z}_{i_k} = \frac{1}{X_{i_k}} \begin{bmatrix} Y_{i_k} \\ Z_{i_k} \end{bmatrix} + \boldsymbol{\nu}_{i_k} = \boldsymbol{h}(\boldsymbol{X}_{i_k}) + \boldsymbol{\nu}_{i_k} \tag{3.34}$$

where $\boldsymbol{\nu}_{i_k}$ is a zero mean Gaussian white noise with its error covariance matrix of $R_k = 0.1^2 I$. An EKF is used to estimate the relative position $\boldsymbol{X}_i$ from the simulated measurement $\boldsymbol{z}_{i_k}$. The initial guess of the estimated relative position $\hat{\boldsymbol{X}}_{i_0}$ is 5 (ft) underestimated for Obstacle 1 and 5 (ft) overestimated for Obstacle 2 in every axes. The initial estimation error covariance matrix is $P_{i_0} = I$. True correspondence between the measurements and the estimation data is used in this simulation.

(b) Results

Simulation results are compared between the sequential PN guidance law given in (3.29) and the MEG-based guidance law given in (3.32). The results of the MEG-based guidance are drawn in blue solid lines, and the results of the sequential PN guidance law are drawn in green dashed lines. Figure 22 presents the vehicle's trajectories due to the PN-based and the MEG-based guidance laws. The starting point, waypoint, the two obstacles and their collision-safety boundaries are also shown. Both trajectories successfully reach the waypoint $\boldsymbol{X}_{wp}$. The vehicle's position along each axis is plotted in Figure 23. From this figure, it is clearer that the guidance laws created an avoiding maneuver in both lateral and vertical directions. Figure 23 also shows distances from the vehicle to Obstacle 1 and Obstacle 2. The minimum safety distance $d = 10$ (ft) is shown in red. When using the sequential PN guidance law, there is a violation of Obstacle 2's collision-safety boundary at about $t = 5$ (sec). This violation is due to an estimation error in its position. Figure 24 presents position estimation errors of Obstacle 1 and Obstacle 2, respectively. The difference in the estimation performance can be seen in estimating the depth (along the X-axis) to Obstacle 2. The estimation error converges faster when using the MEG-based guidance. This difference is due to the property of the MEG-based guidance law that creates larger lateral and vertical motions earlier in the maneuver in comparison to the sequential PN guidance law. Figure 25 is the resulting acceleration input calculated in (3.29) or (3.32) and the control effort. The total control cost required to make the vehicle achieve the waypoint tracking and obstacle avoidance mission is $J_c = 34.338$ when using the MEG and $J_c = 40.409$ when using the sequential PN. The control effort was reduced by 15.5% by using the suggested MEG-based guidance instead of the PN. In conclusion, the simulation results verified that the obstacle avoidance mission can be achieved with less control effort by the MEG-based guidance. In addition, the MEG approach also improves the vision-based estimation accuracy and hence improves the overall obstacle avoidance performance.

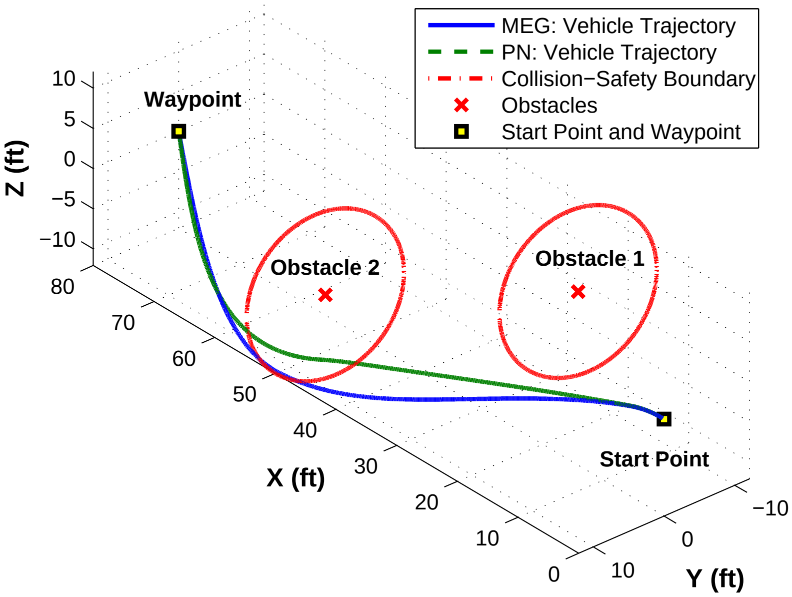


Figure 22: Vehicle Trajectory, and Locations of Two Obstacles, Starting Point and Waypoint

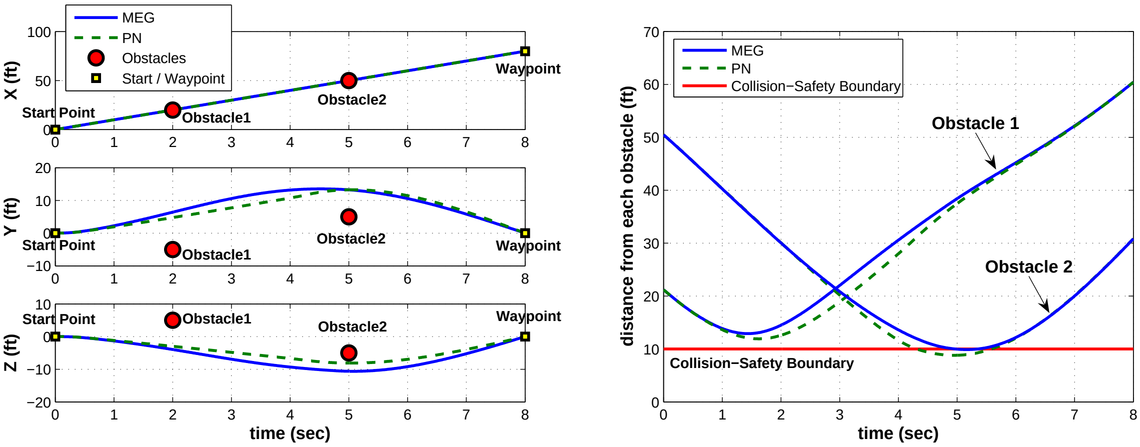


Figure 23: (Left) Vehicle's Position, (Right) Distance from Each Obstacle

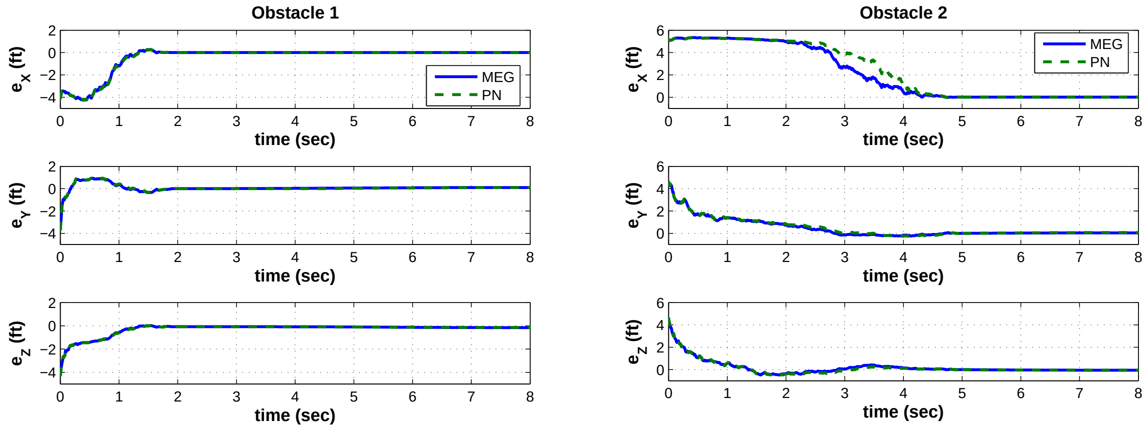


Figure 24: Position Estimation Errors : (Left) Obstacle 1, (Right) Obstacle 2

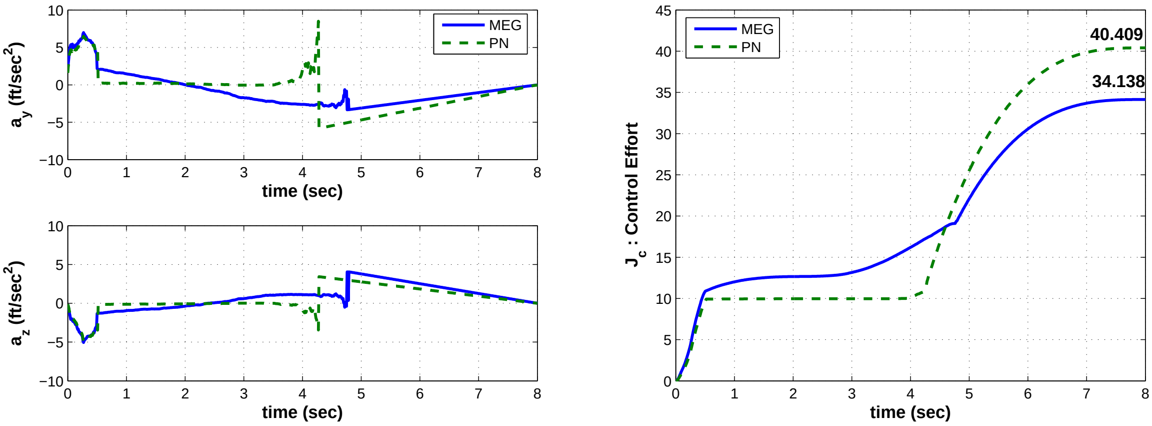


Figure 25: (Left) Vehicle's Acceleration Input, (Right) Control Effort

3.5.2 6 DoF Image-in-the-Loop Simulation

The entire vision-based navigation and guidance system for obstacle avoidance, including the EKF, the z-test algorithm, the collision criteria and the MEG-based guidance law, has been integrated with a real-time image processor and implemented in a 6 DoF UAV flight simulation. This is in direct preparation for flight tests using an actual vehicle. The image processing results and the estimation and guidance performances are presented in this subsection.

(a) 6 DoF UAV Flight Simulation

The vehicle modeled in the simulation is the GTMax whose specification is shown in Table 1. As stated in Subsection 2.3.3, the basic flight controller, own-ship navigation and guidance system of the vehicle have already been developed and implemented. A real-time image processor which is able to detect multiple obstacles in each simulated image frame has also been implemented[25]. Figure 26 is a display of the flight simulation in an obstacle avoidance configuration. Red spheres are obstacles which the vehicle needs to avoid. The window on the left is a map view from the top and the yellow line is the vehicle trajectory. Each grid has a length of 50 (ft). However, the GTMax helicopter does not appear in true scale in the simulation interface. The window at the top right shows a synthetic camera image in the simulation. The image processor outputs are represented by small green crosses in this window. The image processor is detecting the center positions of two obstacles in this picture. The right bottom window displays a chase view from behind of the vehicle. The estimated obstacle positions are indicated in the map view and the chase view windows.

(b) Simulation Settings

Before starting a mission, the vehicle is commanded to fly upward 400 (ft) and then forward 200 (ft) to reach a starting point $\boldsymbol{X}_0 = \begin{bmatrix} 200 & 0 & -400 \end{bmatrix}^T$ (ft) with velocity $\boldsymbol{V}_0 = \begin{bmatrix} 50 & 0 & 0 \end{bmatrix}^T$ (ft/sec) by using the basic guidance system. As soon as the vehicle

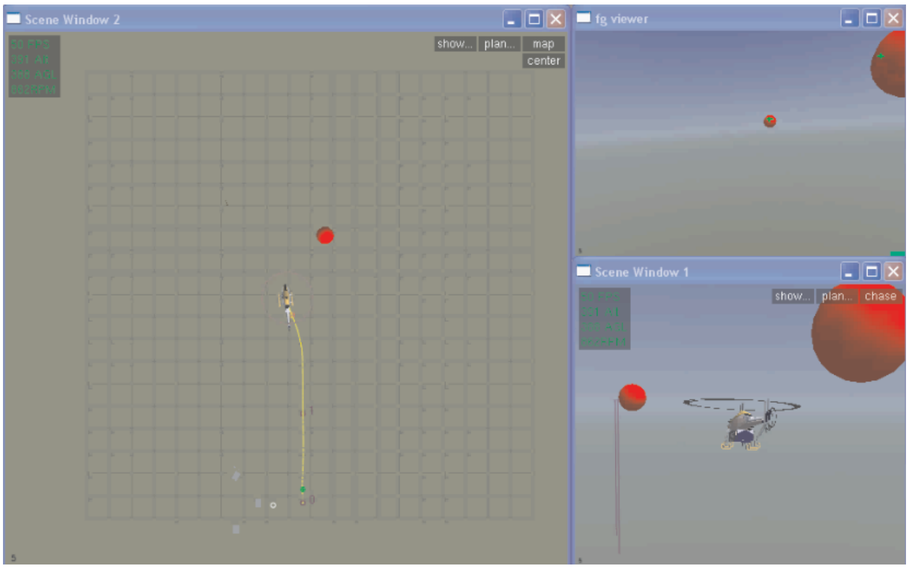


Figure 26: 6 DoF UAV Flight Simulation Interface in Vision-Based Obstacle Avoidance Configuration

passes the starting point, the vision-based obstacle avoidance system is turned on and the guidance system is switched to the one described in Section3.4. The vehicle is required to fly 1600 (ft) forward from the starting point, which means that a waypoint is given at $\boldsymbol{X}_{wp} = \begin{bmatrix} 1800 & 0 & -400 \end{bmatrix}^T$ (ft). On the way to the waypoint, there exist two unforeseen stationary obstacles at

$$\boldsymbol{X}_{obs_1} = \begin{bmatrix} 600 \\ 50 \\ -420 \end{bmatrix} (ft), \quad \boldsymbol{X}_{obs_2} = \begin{bmatrix} 1200 \\ 0 \\ -400 \end{bmatrix} (ft)$$

Both obstacles are given as spheres with radius 20 (ft). To avoid collision, the vehicle needs to maintain a minimum separation distance $d = 100$ (ft) from the obstacles during the entire flight. After reaching the waypoint, the guidance system is switched back to the basic one and it guides the vehicle to reach and stop at the terminal point $\boldsymbol{X}_f = \begin{bmatrix} 2000 & 0 & -400 \end{bmatrix}^T$ (ft).

For the navigation filter design, $\sigma = 0.1$ and $\sigma_X = 0.1$ were used for the measurement noise covariance matrix $R_k = \sigma^2 I$ and for the process noise covariance matrix $Q_k = \sigma_X^2 I \cdot \Delta t_k$, respectively. The EKF is initialized by using the first measurement

$\boldsymbol{z}_0$ obtained for each obstacle. It is assumed that we have some knowledge about range r_0 (only for the initialization). Then the initial estimate of a relative position and its error covariance matrix are set as

$$\hat{\boldsymbol{X}}_0 = r_0 \begin{bmatrix} 1 \\ \boldsymbol{z}_0 \end{bmatrix}, \quad P_0 = L_{c_0} \begin{bmatrix} \sigma_r^2 & 0 \\ \mathbf{0} & r_0 R_k \end{bmatrix} L_{c_0}^T \tag{3.35}$$

where L_{c_0} is a known camera attitude at that time step. In the simulation, $r_0 = 300$ (ft) was used for the first obstacle and $r_0 = 800$ (ft) was used for the second one, and $\sigma_r = 50$ (ft) was used for the both. If the image processor detects both obstacles immediately after starting the mission, the first and second obstacles are 400 (ft) and 1000 (ft) ahead of the vehicle at that time. Therefore, initially, a range to the first obstacle is underestimated by 100 (ft) and that to the second one is underestimated by 200 (ft). For the correspondence problem, $ztest_{max} = 3$ was set as the threshold value. By looking at the z-table[30], this threshold value implies that a hypothesis of the correspondence is rejected when its likelihood is less than 9.364 %. In the collision criteria, a threshold value for the time-to-go used in the simulation was $T = 4$ (sec). Since the vehicle maintains approximately 50 (ft/sec) speed in the X-direction, $T = 4$ (sec) means that an obstacle is not considered to be critical if it has a range more than double that of the minimum separation d from the vehicle.

(c) Results 1: Image Processing and Estimation

A left figure of Figure 27 plots the number of obstacles which are detected by the image processor and the measurement assignment result from z-test. In this simulation, the vehicle reached the starting point $\boldsymbol{X}_0$ at $t_0 = 69.9$ (sec), passed by the first obstacle $\boldsymbol{X}_{obs_1}$ at $t_1 = 78.1$ (sec) and the second obstacle $\boldsymbol{X}_{obs_2}$ at $t_2 = 91.5$ (sec), and finally reached the waypoint $\boldsymbol{X}_{wp}$ at $t_f = 106.1$ (sec). From Figure 27, the image processor detected only Obstacle 1 for the first 5 (sec). After that, Obstacle 1 went out the camera's field of view and the image processor detected Obstacle 2 until $t = 87.5$ (sec). Even though the image processor is capable of detecting multiple objects, it

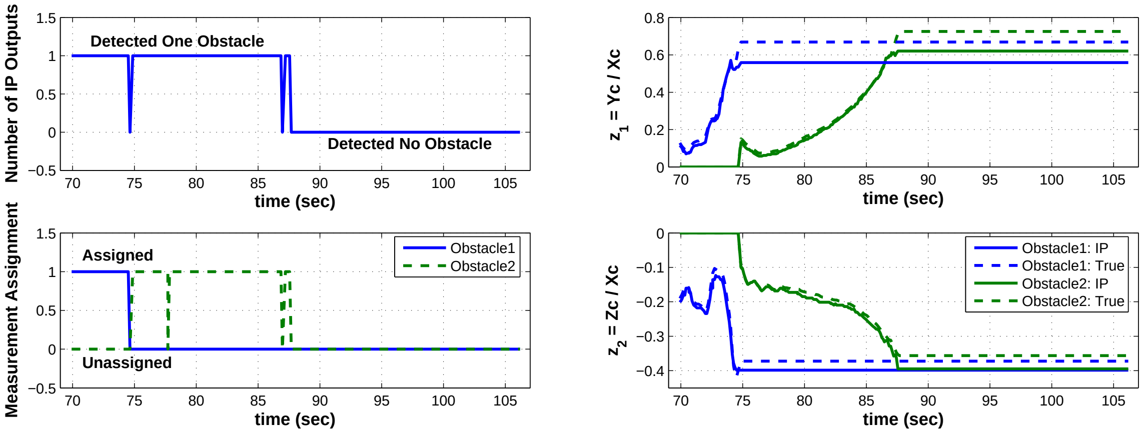


Figure 27: (Left) Image Registered Flag and Measurement Assignment Flag, (Right) Image Coordinates of Obstacles Position (IP Outputs vs. True)

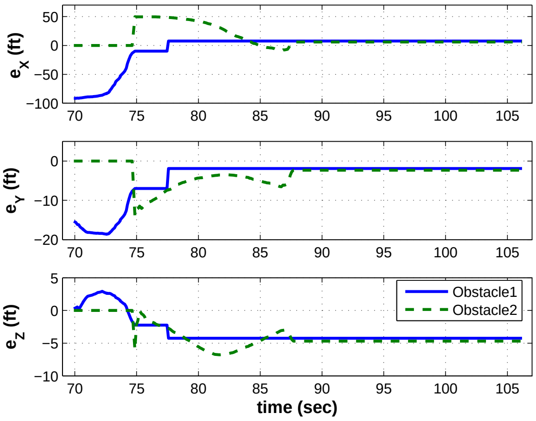


Figure 28: Obstacle Position Estimation Error

does not capture both the obstacles in the same image frame in this example. The second plot in Figure 27 shows the z-test results. At the initial time t_0, estimated obstacle data $\hat{\boldsymbol{X}}_1$ which corresponds to Obstacle 1 was created based on the first measurement. After that, the z-test value is calculated to check the correspondence between the measurement and the updated estimate at each time step. At $t = 74.5$ (sec), the z-test value became larger than its threshold value $ztest_{max} = 3$ and a new estimated obstacle date $\hat{\boldsymbol{X}}_2$ which corresponds to Obstacle 2 was created. From that point onwards, the z-test values are calculated to check the correspondence between the measurement and the estimates $\hat{\boldsymbol{X}}_1$ and $\hat{\boldsymbol{X}}_2$. The z-test value for $\hat{\boldsymbol{X}}_1$ was much larger than that for $\hat{\boldsymbol{X}}_2$ and hence the measurement was correctly assigned to the estimate corresponding to Obstacle 2 by using the z-test algorithm. In the right plot of Figure 27, the image coordinates of each obstacle's position detected by the image processor are compared with those calculated by using true states of the vehicle and the obstacles. These are perfectly matched at the beginning. However, since the obstacle's size on the image becomes larger as the vehicle (or camera) comes closer to the obstacle, the measurement error becomes larger. An average processing time of the image processor was $\Delta t = 0.1213$ (sec). Figure 28 presents the position estimation error for each obstacles. When the estimate is initialized for each obstacle, there is a very large range estimation error e_X (ft), which is 100 (ft) underestimated for Obstacle 1 and 50 (ft) overestimated for Obstacle 2. Those estimation errors are reduced to less than 10 (ft) through the EKF updates by using the image processor outputs. Even though there remains a small bias in the estimates (which is due to a bias in the measurement error), vision-based estimation performance is sufficiently accurate to be used in the collision criteria and the guidance system.

(d) Results 2: Guidance Performance

Figure 29 shows the vehicle trajectory and locations of the start point, waypoint and the two obstacles. Figure 30 shows a time profile of the vehicle's position in each direction and the distance from the vehicle to each obstacle. From these results, we can see that the suggested guidance law successfully guided the vehicle to a given waypoint while not violating the minimum separation distance $d = 100$ (ft) from the two obstacles. Figure 31 shows the acceleration input determined by the minimum-effort guidance (3.33). The flight controller implemented in the simulation calculates actuator inputs by using this acceleration command. The actual vehicle's acceleration is also shown in Figure 31. The lateral acceleration command is very large at around $t = 78$ (sec). This is because the denominator $(\hat{t}_{go}^{-} - t_k)$ went close to zero. Figure 31 also shows the critical obstacle flag, which is 1 when an obstacle is critical and 0 when it is not, for each obstacle. From the figure, it is verified that the collision criteria established in Section 3.3 can appropriately determine the critical obstacle.

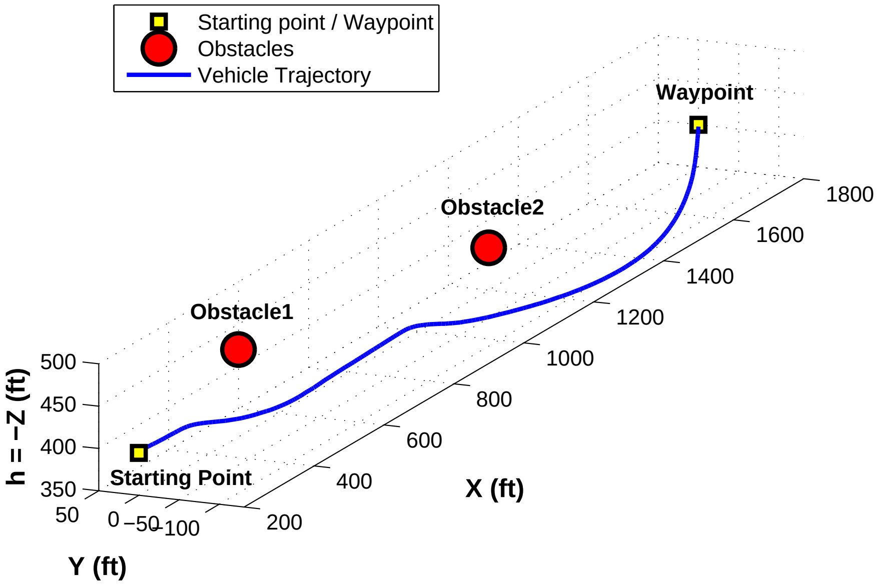


Figure 29: Vehicle Trajectory, and Locations of Two Obstacles, Starting Point and Waypoint

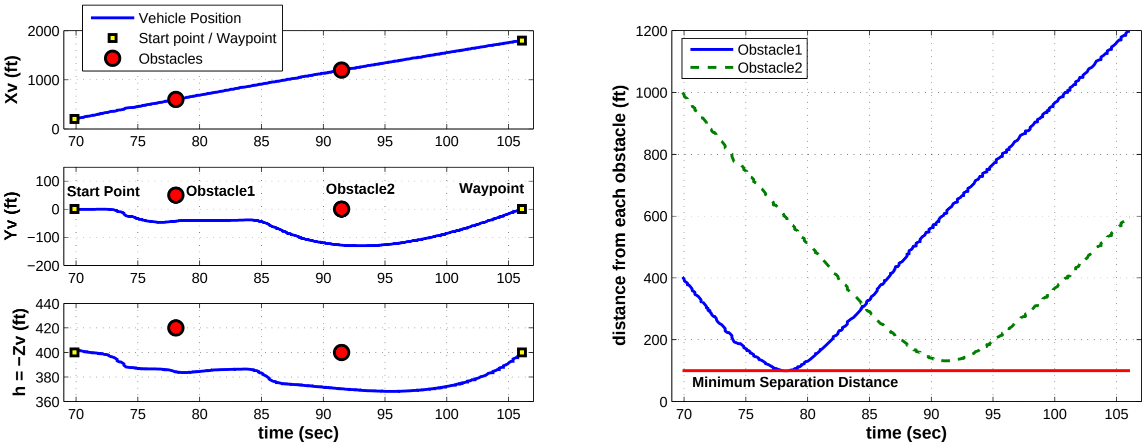


Figure 30: (Left) Vehicle's Position, (Right) Distance from Each Obstacle

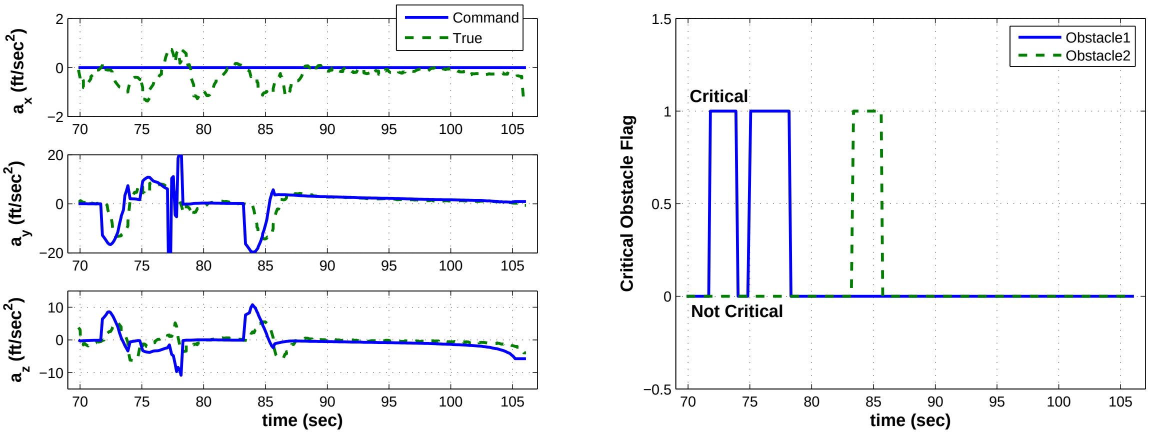


Figure 31: (Left) Vehicle's Acceleration, (Right) Flag for Critical Obstacle

3.6 Flight Test Results

The autonomous flight system for vision-based obstacle avoidance has been evaluated in realistic simulations in the previous section. Now the exact same system is implemented in an onboard flight computer of the GTMax helicopter (See Table 1) to realize closed-loop vision-based obstacle avoidance in flight. The flight test is conducted with a large balloon as an obstacle and successful test results are shown in this section.

3.6.1 Flight Test Settings

Flight tests of vision-based obstacle avoidance are performed with a single obstacle and waypoint. A red balloon 7 (ft) in diameter is used as the obstacle. Figure 32 shows a picture map of the flight test field. Each edge of a grid on the map is 50 (ft). The origin of the inertial (NED) frame is taken at a center of the runway. The balloon is launched at a corner of the runway to an altitude of about 50 (ft) and it is tethered to the ground by three strings. The obstacle position is approximately $\boldsymbol{X}_{obs} = \begin{bmatrix} 100 & -100 & -50 \end{bmatrix}^T$ (ft). The GTMax is guided to fly along the runway edge from a start point to a waypoint, whose locations are given at

$$\boldsymbol{X}_0 = \begin{bmatrix} 150 \\ 200 \\ -100 \end{bmatrix} (ft), \quad \boldsymbol{X}_{wp} = \begin{bmatrix} -50 \\ -400 \\ -100 \end{bmatrix} (ft) \tag{3.36}$$

The vehicle speed is maintained constant at 20 (ft/sec^2) along the direction of the relative vector $\boldsymbol{X}_{wp} - \boldsymbol{X}_0$. In other words, the acceleration command is always zero in that direction. In addition to the waypoint tracking, the vehicle is required to avoid the balloon obstacle with a 100 (ft) separation distance by using information from the image processor. $T = 10$ (sec) is used for the time-to-go threshold in the collision criteria. The EKF is initialized by using an initial estimated range of 400 (ft).

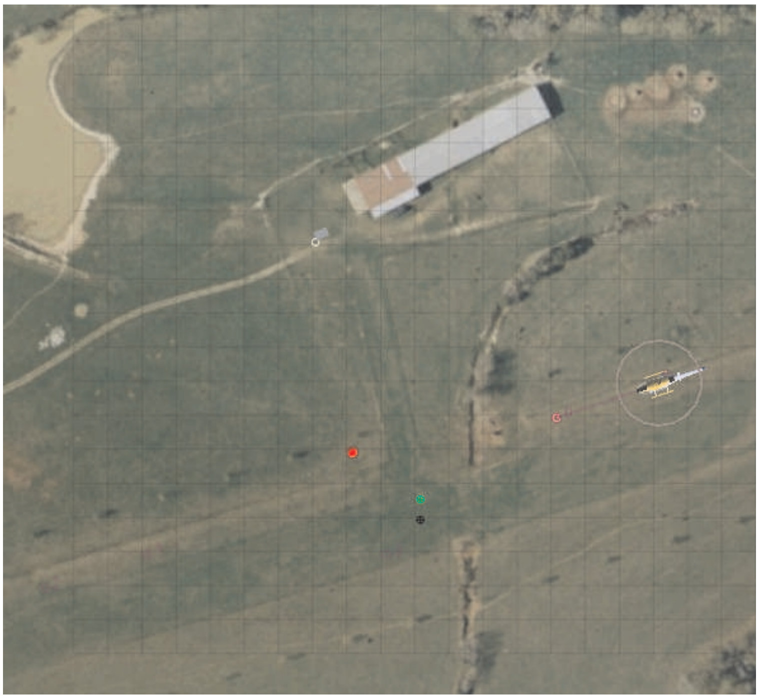


Figure 32: Vision-Based Obstacle Avoidance Flight Test Configuration

Figure 33: The GTMax and a Balloon Obstacle

3.6.2 Results

A first flight of closed-loop vision-based obstacle avoidance and waypoint tracking was achieved in October, 2007. Figure 33 shows the GTMax flying by the balloon obstacle. Figure 34 shows a trajectory of the GTMax with an approximate position of the true obstacle. An estimated obstacle position is also shown on this figure. The obstacle position was 100 (ft) underestimated at the beginning and the estimated position moved closer to its actual position. Figure 35 are time profiles of the vehicle position, estimated obstacle position and the aiming point position. This result shows that the GTMax achieved accurate waypoint tracking and took a left-upward avoiding maneuver to stay away from the obstacle. The first plot in Figure 36 illustrates the collision flag and estimated and true distances from the vehicle to the obstacle. Because of the underestimated range, the obstacle was considered to be critical after passing the starting point. Then the vehicle stayed on the collision-safety boundary ($d = 100$ (ft)) with its velocity being kept outside of the obstacle's collision cone. At $t = 27$ (sec), due to the estimation update, the estimated range violated the minimum separation distance and the avoiding maneuver was taken again. The green dashed line in the left plot of Figure 36 is the true distance between the vehicle and the obstacle, and the result shows that there was no violation of the minimum collision-safety distance. The right plot of Figure 36 compares the resulting acceleration input command with the vehicle's true acceleration. Even though the commanded acceleration becomes very large when the vehicle came close to the obstacle or the waypoint, the flight controller takes into account the vehicle's maneuver limits and prevents large actuator inputs. Figure 37 plots the image processor outputs. The image processor sometimes detected two objects in the same image frame. However, the z-test algorithm assigns the right one to update the estimate. The image processor processed images at about 5 (Hz) in the flight. Figure 37 also shows the standard deviations of the obstacle's position estimation errors.

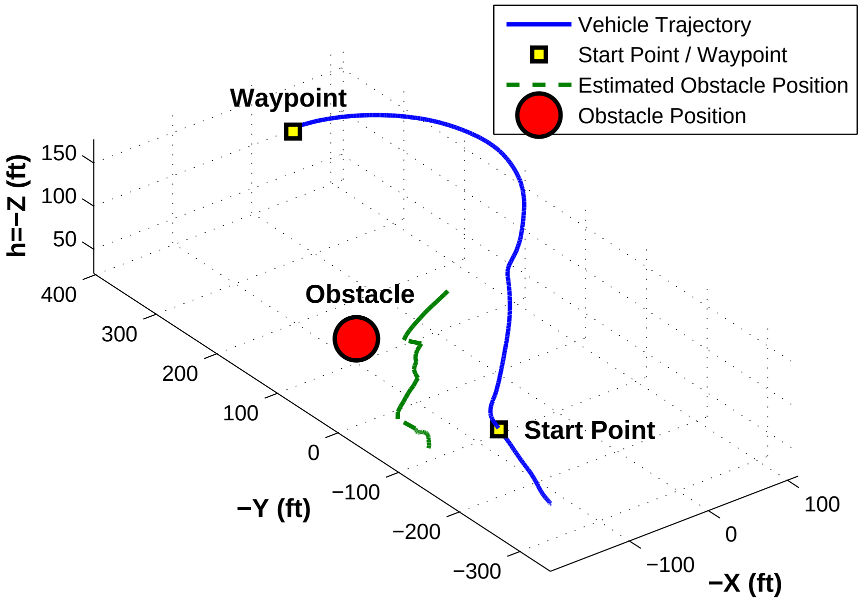


Figure 34: Vehicle Trajectory, Estimated Obstacle Position and True Locations of Obstacle, Starting Point and Waypoint

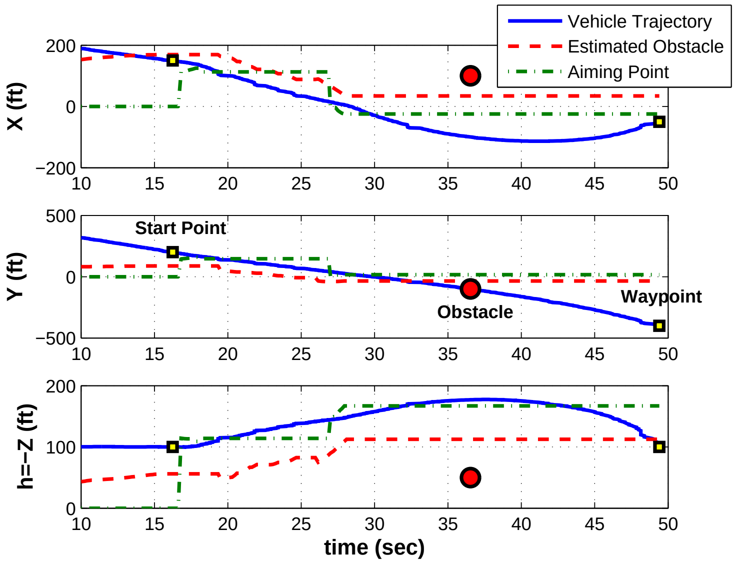


Figure 35: Vehicle Position, Estimated Obstacle Position and Aiming Point Position

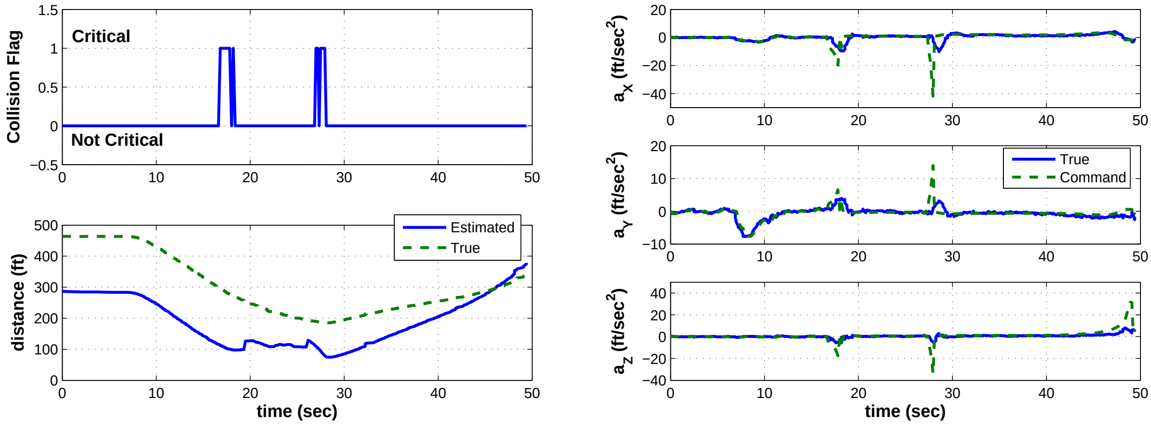


Figure 36: (Left) Collision Flag, and Estimated and True Distance, (Right) Vehicle Acceleration (True vs. Command)

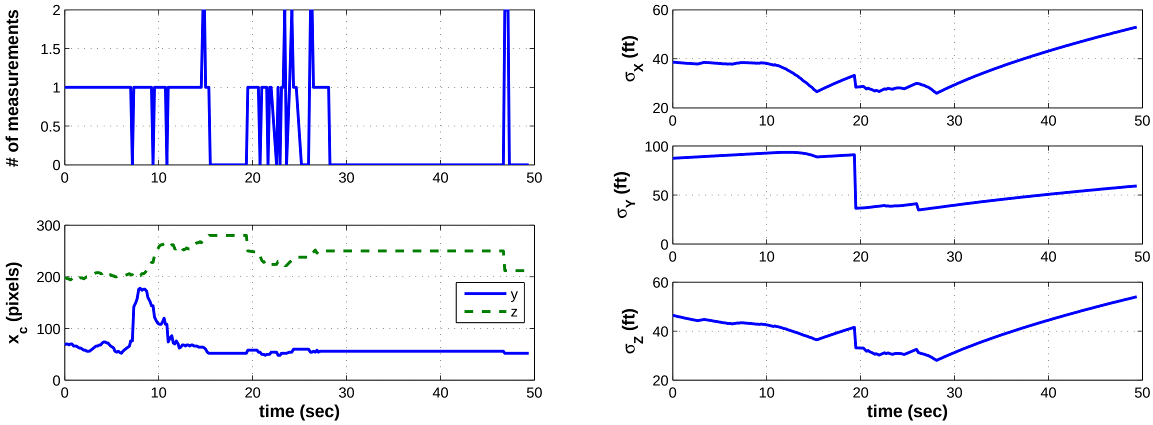


Figure 37: (Left) Image Processor Outputs, (Right) Standard Deviation of Estimation Error

3.7 Conclusion

This chapter developed a vision-based relative navigation and guidance system for a UAV to achieve 3-D waypoint tracking with vision-based obstacle avoidance. It also suggested applying a minimum-effort guidance law for multiple targets tracking as the guidance law for obstacle avoidance. Simulation results were compared with those due to the sequential PN guidance approach, which has been commonly used, and proved that the control effort required to achieve the mission was significantly reduced by using the suggested MEG-based guidance. All the algorithms designed in this chapter have been integrated with the real-time image processor and evaluated in realistic simulations and in flight tests using a UAV helicopter and a balloon as an obstacle.

CHAPTER IV

STOCHASTICALLY OPTIMIZED GUIDANCE DESIGN

This chapter discusses a stochastically optimized guidance design for monocular vision-based control applications of UAVs. The EKF-based navigation filter designed in Section 2 is used for relative navigation, and the estimates are fed back to the guidance system. For monocular vision-based relative navigation, it is well known that the estimation performance significantly depends on sensor motion relative to a target[51]. We have already seen an example in simulation results of vision-based obstacle avoidance when comparing the two different guidance laws (MEG vs. PN) in Subsection 3.5.1 in the previous chapter. In that example, the MEG created larger lateral motion which is preferable for the estimation and resulted in better performance in comparison with the PN. Therefore, this thesis suggests a guidance design which includes a sensor trajectory optimization to improve estimation accuracy, and hence improve overall guidance accuracy. Furthermore, a suboptimal optimization technique is applied to solve the optimization problem for real-time applicability of the guidance strategy.

4.1 Stochastic Optimization

The goal of the guidance design addressed in this chapter is to guide a vehicle to achieve a given mission such as waypoint tracking and path following using a single 2-D vision sensor. As discussed in Chapter 2, an EKF is applied to estimate the 3-D relative state assuming the linear relative dynamics given in (2.4). Let $\boldsymbol{x} = \begin{bmatrix} \boldsymbol{X}^T & \boldsymbol{V}^T \end{bmatrix}^T$ be the relative state vector where $\boldsymbol{X}$ and $\boldsymbol{V}$ are the relative position and velocity respectively. For simplicity, a non-accelerating target is assumed in this problem. Then the linear dynamics (2.4) can be rewritten as

$$\dot{\boldsymbol{x}}(t) = \begin{bmatrix} O & I \\ O & O \end{bmatrix} \boldsymbol{x}(t) + \begin{bmatrix} O \\ -I \end{bmatrix} \boldsymbol{a}_v(t) = F\boldsymbol{x}(t) + G\boldsymbol{a}_v(t) \tag{4.1}$$

Suppose that a mission given to the vehicle is represented as the following quadratic minimization problem

$$\begin{aligned} \min_{\boldsymbol{a}_v} J &= \frac{1}{2}\left(\boldsymbol{x}(t_f) - \boldsymbol{x}_f\right)^T S_f \left(\boldsymbol{x}(t_f) - \boldsymbol{x}_f\right) \\ &\quad + \frac{1}{2}\int_{t_0}^{t_f} \left[\left(\boldsymbol{x}(t) - \boldsymbol{x}_c(t)\right)^T A \left(\boldsymbol{x}(t) - \boldsymbol{x}_c(t)\right) + \boldsymbol{a}_v^T(t) B \boldsymbol{a}_v(t)\right] dt \end{aligned} \tag{4.2}$$

subject to the linear relative dynamics (4.1) with a given initial state $\boldsymbol{x}(t_0) = \boldsymbol{x}_0$. $\boldsymbol{x}_f$ is the desired terminal relative state and $\boldsymbol{x}_c(t)$ for $t_0 \leq t \leq t_f$ is the commanded relative state path. S_f, A and B are constant weight matrices satisfying

$$S_f \geq O, \quad A \geq O, \quad B > O \tag{4.3}$$

Then, the goal of the guidance design in this problem is to realize the optimal solution $\boldsymbol{a}_v^*(t)$ of the minimization problem (4.2).

4.1.1 Optimal Guidance

Consider the problem of trying to determine the acceleration input at a current time step t_k, $\boldsymbol{a}_v(t_k)$, given the current state $\boldsymbol{x}(t_k) = \boldsymbol{x}_k$. When the true state $\boldsymbol{x}$ is available, the optimal guidance input at t_k can be obtained by solving (4.2) with the initial time at t_k. An analytical solution for this minimization problem can be obtained by solving the Hamilton-Jacobi-Bellman (HJB) equation. The HJB equation is based on dynamic programming theory established by Bellman, and is given by a set of first-order nonlinear partial differential equations that must be solved with boundary conditions[13]. The optimal acceleration input at a current time step t_k and the optimal cost are given as follows

$$\boldsymbol{a}_v^*(t_k) = -B^{-1}G^T\left(S(t_k)\boldsymbol{x}(t_k) + \boldsymbol{v}(t_k)\right) \tag{4.4}$$

$$J^*(t_k) = \frac{1}{2}\boldsymbol{x}^T(t_k)S(t_k)\boldsymbol{x}(t_k) + \boldsymbol{x}^T(t_k)\boldsymbol{v}(t_k) + w(t_k) \tag{4.5}$$

where the matrix S, the vector $\boldsymbol{v}$ and the scalar w satisfy the following differential equations.

$$\begin{aligned} \dot{S} + SF + F^T S + A - SGB^{-1}G^T S &= O, \quad S(t_f) = S_f && (4.6) \\ \dot{\boldsymbol{v}} + \left(F^T - SGB^{-1}G^T\right)\boldsymbol{v} - A\boldsymbol{x}_c &= \mathbf{0}, \quad \boldsymbol{v}(t_f) = -S_f \boldsymbol{x}_f && (4.7) \\ \dot{w} - \frac{1}{2}\boldsymbol{v}^T GB^{-1}G^T \boldsymbol{v} + \frac{1}{2}\boldsymbol{x}_c^T A \boldsymbol{x}_c &= 0, \quad w(t_f) = \frac{1}{2}\boldsymbol{x}_f^T S_f \boldsymbol{x}_f && (4.8) \end{aligned}$$

A derivation of the optimal solution (4.4,4.5) using the HJB equations are shown in Appendix B. The first term of the guidance input (4.4) is feedback and the second term is feed-forward control. In the special case of terminal tracking (i.e., $A = O$), a closed form of the optimal solution can be derived as follows

$$\begin{aligned} \boldsymbol{a}_v^*(t_k) &= -B^{-1}G^T e^{F^T(t_f - t_k)} S_f \left(I + \mathcal{G}_k S_f\right)^{-1} \left(e^{F(t_f - t_k)}\boldsymbol{x}(t_k) - \boldsymbol{x}_f\right) && (4.9) \\ J^*(t_k) &= \frac{1}{2}\left(e^{F(t_f - t_k)}\boldsymbol{x}(t_k) - \boldsymbol{x}_f\right)^T \left(I + \mathcal{G}_k S_f\right)^{-T} S_f \left(e^{F(t_f - t_k)}\boldsymbol{x}(t_k) - \boldsymbol{x}_f\right) && (4.10) \end{aligned}$$

where

$$\mathcal{G}_k = \int_{t_k}^{t_f} e^{F(t_f - s)} GB^{-1}G^T e^{F^T(t_f - s)} ds$$

Note that $\mathcal{G}_k > 0$ when (F, G) is controllable and $B > 0$. The derivation of (4.9,4.10) is also shown in Appendix B.

4.1.2 Estimated Optimal Guidance

When the measurement model is linear, an optimal estimator and controller can be designed separately and it results in a standard linear Kalman filter and an LQG controller[5][94]. However, the 2-D vision-based measurement is a nonlinear function of the 3-D relative state, and for such a case, the separation principle does not hold between estimation and control, and closed form solutions are not available. A common way to determine a guidance law is to replace the true state by its estimate in (4.4).

$$\begin{aligned} \hat{\boldsymbol{a}}_v^*(t_k) &= -B^{-1}G^T \left(S(t_k)\hat{\boldsymbol{x}}(t_k) + \boldsymbol{v}(t_k)\right) && (4.11) \\ &= \boldsymbol{a}_v^*(t_k) + B^{-1}G^T S(t_k)\tilde{\boldsymbol{x}}(t_k) = \boldsymbol{a}_v^*(t_k) - \tilde{\boldsymbol{a}}_v^*(t_k) \end{aligned}$$

where $\tilde{\boldsymbol{x}}(t_k) = \boldsymbol{x}(t_k) - \hat{\boldsymbol{x}}(t_k)$ is the estimation error. In this thesis, we will refer to this guidance design as the *estimated optimal guidance* or the conventional guidance. Consider the case of $A = O$, then the estimated optimal guidance and the expected cost are

$$\begin{aligned} \hat{\boldsymbol{a}}_v^*(t_k) &= -B^{-1}G^T e^{F^T(t_f-t_k)} S_f \left(I + \mathcal{G}_k S_f\right)^{-1} \left(e^{F(t_f-t_k)}\hat{\boldsymbol{x}}(t_k) - \boldsymbol{x}_f\right) && (4.12) \\ \hat{J}^*(t_k) &= \boldsymbol{E}\left[J(t_k)\right] \\ &= J^*(t_k) + \frac{1}{2} tr P_k e^{F^T(t_f-t_k)} \left(I + \mathcal{G}_k S_f\right)^{-T} S_f \left(I + \mathcal{G}_k S_f\right)^{-1} e^{F(t_f-t_k)} && (4.13) \end{aligned}$$

assuming $\boldsymbol{E}\left[\tilde{\boldsymbol{x}}(t_k)\right] = \boldsymbol{0}$ and $\boldsymbol{E}\left[\tilde{\boldsymbol{x}}(t_k)\tilde{\boldsymbol{x}}^T(t_k)\right] = P_k$. The second term of the right hand side in the equation (4.13) corresponds to an increase in cost due to the estimation error $\tilde{\boldsymbol{x}}(t_k)$. Since the optimization is performed by assuming zero estimation error in the previous subsection, this guidance policy can cause poor guidance performance when the estimation error is large.

For vision-based measurements, estimation performance significantly depends on the relative motion with respect to the target[51]. For example, range information becomes unobservable when a camera moves straight towards the target without any change in the line-of-sight. Hence, the estimation performance can be improved by a guidance design which creates some relative motion lateral to the line-of-sight, and hence the overall guidance performance is also improved by it. This thesis suggests designing a guidance law by stochastically minimizing J in (4.2) under the condition of using the EKF. In this approach, the guidance design is separated from the estimator design, but the resulting guidance design takes into account the dependence of the guidance on the estimation performance. The resulting guidance is conditionally optimal instead of globally optimal. A detail of the stochastic optimization approach will be discussed in the next subsection.

4.1.3 Stochastic Optimization

The guidance design suggested in this thesis minimizes the expected value of the cost function defined by (4.2) under the condition of using the EKF-based relative navigation. The expected value of J is taken as follows.

$$\begin{aligned}
\boldsymbol{E}[J] &= \frac{1}{2}\boldsymbol{E}\left[(\boldsymbol{x}(t_f)-\boldsymbol{x}_c(t_f))^T S_f (\boldsymbol{x}(t_f)-\boldsymbol{x}_c(t_f))\right] \\
&\quad +\frac{1}{2}\int_{t_0}^{t_f}\left(\boldsymbol{E}\left[(\boldsymbol{x}(t)-\boldsymbol{x}_c(t))^T A (\boldsymbol{x}(t)-\boldsymbol{x}_c(t))\right]+\boldsymbol{a}_v^T(t)B\boldsymbol{a}_v(t)\right)dt \\
&= \frac{1}{2}\boldsymbol{E}\left[(\hat{\boldsymbol{x}}(t_f)-\boldsymbol{x}_c(t_f)+\tilde{\boldsymbol{x}}(t_f))^T S_f (\hat{\boldsymbol{x}}(t_f)-\boldsymbol{x}_c(t_f)+\tilde{\boldsymbol{x}}(t_f))\right] \\
&\quad +\frac{1}{2}\int_{t_0}^{t_f}\left(\boldsymbol{E}\left[(\hat{\boldsymbol{x}}(t)-\boldsymbol{x}_c(t)+\tilde{\boldsymbol{x}}(t))^T A (\hat{\boldsymbol{x}}(t)-\boldsymbol{x}_c(t)+\tilde{\boldsymbol{x}}(t))\right]+\boldsymbol{a}_v^T(t)B\boldsymbol{a}_v(t)\right)dt \\
&= \frac{1}{2}(\hat{\boldsymbol{x}}(t_f)-\boldsymbol{x}_c(t_f))^T S_f (\hat{\boldsymbol{x}}(t_f)-\boldsymbol{x}_c(t_f)) \\
&\quad +\frac{1}{2}\int_{t_0}^{t_f}\left((\hat{\boldsymbol{x}}(t)-\boldsymbol{x}_c(t))^T A (\hat{\boldsymbol{x}}(t)-\boldsymbol{x}_c(t))+\boldsymbol{a}_v^T(t)B\boldsymbol{a}_v(t)\right)dt \\
&\quad +(\hat{\boldsymbol{x}}(t_f)-\boldsymbol{x}_c(t_f))^T S_f \boldsymbol{E}[\tilde{\boldsymbol{x}}(t_f)]+\int_{t_0}^{t_f}(\hat{\boldsymbol{x}}(t)-\boldsymbol{x}_c(t))^T A\boldsymbol{E}[\tilde{\boldsymbol{x}}(t)]\,dt \\
&\quad +\frac{1}{2}tr\boldsymbol{E}\left[\tilde{\boldsymbol{x}}(t_f)\tilde{\boldsymbol{x}}^T(t_f)\right]S_f+\frac{1}{2}\int_{t_0}^{t_f}tr\boldsymbol{E}\left[\tilde{\boldsymbol{x}}(t)\tilde{\boldsymbol{x}}^T(t)\right]A\,dt
\end{aligned}$$

Assuming $\boldsymbol{E}[\tilde{\boldsymbol{x}}(t)]=\boldsymbol{0}$ and $\boldsymbol{E}\left[\tilde{\boldsymbol{x}}(t)\tilde{\boldsymbol{x}}^T(t)\right]=P(t)$, the stochastically optimized guidance design can be formulated as follows

$$\begin{aligned}
\min_{\boldsymbol{a}_v}\boldsymbol{E}[J] &= \frac{1}{2}(\hat{\boldsymbol{x}}(t_f)-\boldsymbol{x}_f)^T S_f (\hat{\boldsymbol{x}}(t_f)-\boldsymbol{x}_f) \\
&\quad +\frac{1}{2}\int_{t_0}^{t_f}\left((\hat{\boldsymbol{x}}(t)-\boldsymbol{x}_c(t))^T A (\hat{\boldsymbol{x}}(t)-\boldsymbol{x}_c(t))+\boldsymbol{a}_v^T(t)B\boldsymbol{a}_v(t)\right)dt \\
&\quad +\frac{1}{2}trP(t_f)S_f+\frac{1}{2}\int_{t_0}^{t_f}trP(t)A\,dt \qquad (4.14)
\end{aligned}$$

subject to the EKF prediction and update laws (2.9, 2.16-2.19). The EKF process is recalled here for the case of a linear process model:

Prediction :

$$\hat{\boldsymbol{x}}_{k+1}^- = \Phi_{k+1}\hat{\boldsymbol{x}}_k+\Gamma_{k+1}\boldsymbol{a}_v(t_k) \qquad (4.15)$$

$$P_{k+1}^- = \Phi_{k+1}P_k\Phi_{k+1}^T+Q_{k+1} \qquad (4.16)$$

where

$$\Phi_{k+1} = I + F(t_{k+1} - t_k), \quad \Gamma_{k+1} = G(t_{k+1} - t_k) + \frac{1}{2}FG(t_{k+1} - t_k)^2 \tag{4.17}$$

Update :

$$\hat{\boldsymbol{x}}_{k+1} = \hat{\boldsymbol{x}}_{k+1}^{-} + K_{k+1}\left(\boldsymbol{z}_{k+1} - \boldsymbol{h}(\hat{\boldsymbol{x}}_{k+1}^{-})\right) \tag{4.18}$$

$$P_{k+1} = \left(I - K_{k+1}H_{k+1}\right)P_{k+1}^{-} \tag{4.19}$$

where

$$K_{k+1} = P_{k+1}^{-}H_{k+1}^{T}\left(H_{k+1}P_{k+1}^{-}H_{k+1}^{T} + R_{k+1}\right)^{-1} \tag{4.20}$$

The optimization problem (4.14) is nonlinear due to the nonlinear vision-based measurement, and the analytical optimal solution cannot be obtained in most cases. There are numerous numerical optimization algorithms, such as dynamic programming, which can be used to solve this problem. However, these numerical optimizations require iterative computations, and thus they are not suitable for realtime application. Therefore, this thesis aims to establish suboptimal optimization strategies which reduce the computational burden in performing the optimization.

4.2 One-Step-Ahead Optimization Approach

In this section, the one-step-ahead (OSA) suboptimal optimization technique is applied to solve the stochastic optimization problem established in Subsection 4.1.3. The idea of the OSA optimization is introduced by Logothetis et al. in [48]. In this approach, at the current time step t_k, the optimization is performed under the assumption that the observer anticipates only one more final measurement at one time step ahead t_{k+1}. In [48], the OSA optimization is applied to minimize several different estimation performance costs for bearings-only tracking problem. This section applies the OSA optimization approach to perform the stochastic optimization[86], and presented simulation results of the guidance performance with applications of vision-based target tracking and vision-based obstacle avoidance problems.

4.2.1 Approach

Consider to determine the guidance input at a current time step t_k, given the current updated estimate $\hat{\boldsymbol{x}}_k$ and its error covariance matrix P_k. Let

$$\boldsymbol{a}_v^1(t) = \hat{\boldsymbol{a}}_{v_k}^*(t) + \Delta\boldsymbol{a}_k^1 \tag{4.21}$$

be the vehicle acceleration input for a time interval $t_k \leq t \leq t_{k+1}$. The input $\hat{\boldsymbol{a}}_{v_k}^*(t)$ denotes the estimated optimal guidance given in (4.11), which can be considered as a zero-step-ahead suboptimal guidance law ($\boldsymbol{a}_v^0(t) = \hat{\boldsymbol{a}}_{v_k}^*(t)$ for $t_k \leq t \leq t_f$). $\Delta\boldsymbol{a}_k^1$ in (4.21) is an additional input which is for the estimation improvement. Under the one-step-ahead optimization assumption, there will be only one more measurement available at the next time step. Therefore, the estimation accuracy will not be improved after t_{k+1}. This means that what can be done after t_{k+1} is only to apply the estimated optimal guidance law which was recalculated at t_{k+1} using the updated estimate $\hat{\boldsymbol{x}}_{k+1}$.

$$\boldsymbol{a}_v^1(t) = \hat{\boldsymbol{a}}_{v_{k+1}}^*(t), \quad t_{k+1} \leq t \leq t_f \tag{4.22}$$

Then, the expected cost $\boldsymbol{E}\left[J\right]$ at t_k can be derived by assuming $\boldsymbol{E}\left[\tilde{\boldsymbol{x}}(t_{k+i})\right] = \boldsymbol{0}$ and $\boldsymbol{E}\left[\tilde{\boldsymbol{x}}(t_{k+i})\tilde{\boldsymbol{x}}^T(t_{k+i})\right] = P_{k+i}$ for $i = 0,\ 1$. In addition, a zero process noise covariance $Q_k = O$ is assumed in the calculation of $\boldsymbol{E}\left[J\right]$ for simplicity. The expected cost results in a function of the current estimate $\hat{\boldsymbol{x}}_k$, its error covariance P_k and the additional acceleration input $\Delta\boldsymbol{a}_k^1$. Since $\hat{\boldsymbol{x}}_k$ and P_k are given, the minimization problem (4.14) now becomes a minimization over a single vector $\Delta\boldsymbol{a}_k^1$, and the optimal $\Delta\boldsymbol{a}_k^1$ which minimizes $\boldsymbol{E}\left[J(t_k)\right]$ can be derived by solving the following algebraic equation.

$$\frac{\partial \boldsymbol{E}\left[J(t_k)\right]}{\partial \Delta\boldsymbol{a}_k^1} = \boldsymbol{0} \tag{4.23}$$

The resulting $\Delta\boldsymbol{a}_k^1$ can be expressed in terms of the given estimate $\hat{\boldsymbol{x}}_k$ and its error covariance matrix P_k.

4.2.2 Special Case of Terminal Tracking

The OSA suboptimal guidance law is derived for the special case of terminal tracking, i.e., a case of $A = O$ in this subsection. As derived in (4.12), the estimated optimal guidance calculated at t_k or the zero-step-ahead suboptimal guidance is

$$\boldsymbol{a}_v^0(t) = \hat{\boldsymbol{a}}_{v_k}^*(t) = -B^{-1}G^T e^{F^T(t_f-t)} S_f \left(I + \mathcal{G}_k S_f\right)^{-1} \left(e^{F(t_f-t_k)}\hat{\boldsymbol{x}}_k - \boldsymbol{x}_f\right) \tag{4.24}$$

When applying $\boldsymbol{a}_v^0(t)$ for $t_k \le t \le t_{k+1}$, the true relative state at t_{k+1} will be

$$\begin{aligned}
\boldsymbol{x}^0(t_{k+1}) &= e^{F(t_{k+1}-t_k)}\boldsymbol{x}(t_k) + \int_{t_k}^{t_{k+1}} e^{F(t_{k+1}-s)} G \boldsymbol{a}_{v_k}^0(s) ds \\
&= e^{F(t_{k+1}-t_k)} \left(\hat{\boldsymbol{x}}_k + \tilde{\boldsymbol{x}}_k\right) \\
&\quad - e^{-F(t_f-t_{k+1})} \left(\mathcal{G}_k - \mathcal{G}_{k+1}\right) S_f \left(I + \mathcal{G}_k S_f\right)^{-1} \left(e^{F(t_f-t_k)}\hat{\boldsymbol{x}}_k - \boldsymbol{x}_f\right)
\end{aligned} \tag{4.25}$$

The OSA suboptimal guidance for $t_k \le t \le t_{k+1}$ is given by (4.21), and the true relative state at t_{k+1} becomes

$$\begin{aligned}
\boldsymbol{x}^1(t_{k+1}) &= e^{F(t_{k+1}-t_k)}\boldsymbol{x}(t_k) + \int_{t_k}^{t_{k+1}} e^{F(t_{k+1}-s)} G \left(\boldsymbol{a}_{v_k}^0(s) + \Delta\boldsymbol{a}_k^1\right) ds \\
&= \boldsymbol{x}^0(t_{k+1}) + \Gamma_{k+1}\Delta\boldsymbol{a}_k^1 = \boldsymbol{x}^0(t_{k+1}) + \Delta\boldsymbol{x}_{k+1}^1
\end{aligned} \tag{4.26}$$

where $\Gamma_{k=1}$ is defined in (4.17) and $\Delta\boldsymbol{x}_{k+1}^1 = \Gamma_{k+1}\Delta\boldsymbol{a}_k^1$. Then, the estimated optimal guidance is recalculated at t_{k+1} as follows.

$$\begin{aligned}
\hat{\boldsymbol{a}}_{v_{k+1}}^*(t) &= -B^{-1}G^T e^{F^T(t_f-t)} S_f \left(I + \mathcal{G}_{k+1} S_f\right)^{-1} \left(e^{F(t_f-t_{k+1})}\hat{\boldsymbol{x}}_{k+1} - \boldsymbol{x}_f\right) \\
&= \boldsymbol{a}_v^0(t) - B^{-1}G^T e^{F^T(t_f-t)} S_f \left(I + \mathcal{G}_{k+1} S_f\right)^{-1} e^{F(t_f-t_{k+1})} \left(\tilde{\boldsymbol{x}}_{k+1}^- - \tilde{\boldsymbol{x}}_{k+1}\right) \\
&\quad - B^{-1}G^T e^{F^T(t_f-t)} S_f \left(I + \mathcal{G}_{k+1} S_f\right)^{-1} e^{F(t_f-t_{k+1})} \Delta\boldsymbol{x}_{k+1}^1
\end{aligned} \tag{4.27}$$

where $\tilde{\boldsymbol{x}}_{k+1}^- = e^{F(t_{k+1}-t_k)}\tilde{\boldsymbol{x}}_k$ is the predicted estimation error at t_{k+1}. In the OSA suboptimal guidance law, $\hat{\boldsymbol{a}}_{v_{k+1}}^*(t)$ is applied for $t_{k+1} \le t \le t_f$. Then the terminal tracking error is given by

$$\begin{aligned}
\boldsymbol{x}^1(t_f) - \boldsymbol{x}_f &= e^{F(t_f-t_{k+1})}\boldsymbol{x}^1(t_{k+1}) + \int_{t_{k+1}}^{t_f} e^{F(t_f-s)} G \boldsymbol{a}_{v_{k+1}}^0(s) ds \\
&= \left(\boldsymbol{x}^0(t_f) - \boldsymbol{x}_f\right) - \mathcal{G}_{k+1} S_f \left(I + \mathcal{G}_{k+1} S_f\right)^{-1} e^{F(t_f-t_{k+1})} \left(\tilde{\boldsymbol{x}}_{k+1}^- - \tilde{\boldsymbol{x}}_{k+1}\right) \\
&\quad + \left(I + \mathcal{G}_{k+1} S_f\right)^{-1} e^{F(t_f-t_{k+1})} \Delta\boldsymbol{x}_{k+1}^1
\end{aligned} \tag{4.28}$$

The second term in (4.28) represents the deviation from the estimated optimal guidance due to the estimation update at t_{k+1}, and the third term represents the deviation due to the additional input $\Delta\boldsymbol{a}_k^1$. The expected cost $\hat{J}^1(t_k) = \boldsymbol{E}\left[J(t_k)\right]$ is calculated by assuming ideal estimates having $\boldsymbol{E}\left[\tilde{\boldsymbol{x}}_{k+i}\right] = \boldsymbol{0}$ and $\boldsymbol{E}\left[\tilde{\boldsymbol{x}}_{k+i}\tilde{\boldsymbol{x}}_{k+i}^T\right] = P_{k+i}$ for $i = 0, 1$.

$$\begin{aligned}
J^1(t_k) &= \frac{1}{2}\boldsymbol{E}\left[\left(\boldsymbol{x}^1(t_f) - \boldsymbol{x}_f\right)^T S_f\left(\boldsymbol{x}^1(t_f) - \boldsymbol{x}_f\right) + \frac{1}{2}\int_{t_k}^{t_f} \boldsymbol{a}_v^{1^T}(t)B\boldsymbol{a}_v^1(t)dt\right] \\
&\simeq \hat{J}^*(t_k) - \frac{1}{2}tr\left(P_{k+1}^- - P_{k+1}\right)e^{F^T(t_f-t_{k+1})}S_f\mathcal{G}_{k+1}S_f\left(I + \mathcal{G}_{k+1}S_f\right)^{-1}e^{F(t_f-t_{k+1})} \\
&\quad + \frac{1}{2}\Delta\boldsymbol{a}_k^{1^T}\left(\Gamma_{k+1}^T e^{F^T(t_f-t_{k+1})}S_f\left(I + \mathcal{G}_{k+1}S_f\right)^{-1}e^{F(t_f-t_{k+1})}\Gamma_{k+1} + B_{k+1}\right)\Delta\boldsymbol{a}_k^1 \\
&= J^0(t_k) - \frac{1}{2}tr\left(P_{k+1}^- - P_{k+1}\right)\mathcal{S}_{k+1} + \frac{1}{2}\Delta\boldsymbol{a}_k^{1^T}\mathcal{B}_{k+1}\Delta\boldsymbol{a}_k^1 \qquad (4.29)
\end{aligned}$$

where $B_{k+1} = B(t_{k+1} - t_k)$ and $\mathcal{S}_{k+1}$ and $\mathcal{B}_{k+1}$ are constant matrices defined by

$$\mathcal{S}_{k+1} = e^{F^T(t_f-t_{k+1})}S_f\mathcal{G}_{k+1}S_f\left(I + \mathcal{G}_{k+1}S_f\right)^{-1}e^{F(t_f-t_{k+1})} \qquad (4.30)$$

$$\mathcal{B}_{k+1} = \Gamma_{k+1}^T e^{F^T(t_f-t_{k+1})}S_f\left(I + \mathcal{G}_{k+1}S_f\right)^{-1}e^{F(t_f-t_{k+1})}\Gamma_{k+1} + B_{k+1} \qquad (4.31)$$

$\hat{J}^*(t_k) = J^0(t_k)$ in (4.29) is the expected cost when applying the estimation optimal guidance for $t_k \le t \le t_f$. For the case of $A = O$, it is given by (4.13). The second term in (4.29) represents a decrease in terminal tracking error due to the estimation improvement by the measurement update at t_{k+1}, and the third term represents an increase in control cost due to the additional input $\Delta\boldsymbol{a}_k^1$. From the EKF update law (4.19, 4.20), the second term in (4.29) can be rewritten as follows.

$$\begin{aligned}
\frac{1}{2}tr\left(P_{k+1}^- - P_{k+1}\right)\mathcal{S}_{k+1} &= \frac{1}{2}trK_{k+1}H_{k+1}P_{k+1}^-\mathcal{S}_{k+1} \\
&= \frac{1}{2}trH_{k+1}^T\left(H_{k+1}P_{k+1}^-H_{k+1}^T + R_{k+1}\right)^{-1}H_{k+1}\bar{\mathcal{S}}_{k+1} \qquad (4.32)
\end{aligned}$$

where $\bar{\mathcal{S}}_{k+1} = P_{k+1}^-\mathcal{S}_{k+1}P_{k+1}^-$ is a constant matrix which is determined by given P_k.

Now the OSA guidance policy is to find $\Delta\boldsymbol{a}_k^1$ which maximizes the decrease in cost. Hence the optimal solution for $\Delta\boldsymbol{a}_k^1$ can be obtained by solving the following algebraic equation.

$$\frac{\partial}{\partial\Delta\boldsymbol{a}_k^1}\left(J^0(t_k) - J^1(t_k)\right) = \boldsymbol{0} \qquad (4.33)$$

From (4.29) and (4.32), it can be expanded as

$$\frac{\partial}{\partial \Delta \boldsymbol{a}_k^1}\left(tr H_{k+1}^T \left(H_{k+1} P_{k+1}^- H_{k+1}^T + R_{k+1} \right)^{-1} H_{k+1} \bar{\mathcal{S}}_{k+1} - \Delta \boldsymbol{a}_k^{1^T} \mathcal{B}_{k+1} \Delta \boldsymbol{a}_k^1 \right) = \mathbf{0}$$

$$\therefore \quad \Delta \boldsymbol{a}_k^1 = \frac{1}{2} \mathcal{B}_{k+1}^{-T} \left[\frac{\partial}{\partial \Delta \boldsymbol{a}_k^1} tr \left(H_{k+1} P_{k+1}^- H_{k+1}^T + R_{k+1} \right)^{-1} H_{k+1} \bar{\mathcal{S}}_{k+1} \right]^T \tag{4.34}$$

H_{k+1} on the right hand side of (4.34) is a measurement matrix defined by a Jacobian of the measurement model $\boldsymbol{h}(\boldsymbol{x})$ evaluated at $\boldsymbol{x} = \hat{\boldsymbol{x}}_{k+1}^-$, and it becomes a function of $\Delta \boldsymbol{a}_k^1$. To simplify the calculation, we will approximate the solution for (4.34) by

$$\therefore \quad \Delta \boldsymbol{a}_k^1 = \frac{1}{2} \mathcal{B}_{k+1}^{-T} \left[\frac{\partial}{\partial \Delta \boldsymbol{a}_k^1} tr \left(H_{k+1} P_{k+1}^- H_{k+1}^T + R_{k+1} \right)^{-1} H_{k+1} \bar{\mathcal{S}}_{k+1} \right]^T \Bigg|_{\hat{\boldsymbol{x}}_{k+1}^- = \hat{\boldsymbol{x}}_{k+1}^{*-}} \tag{4.35}$$

where $\hat{\boldsymbol{x}}_{k+1}^{*-}$ is the predicted estimate at t_{k+1} when applying only the estimated optimal guidance $\boldsymbol{a}_v(t) = \hat{\boldsymbol{a}}_{v_k}^*(t) = \boldsymbol{a}_v^0(t)$ for $t_k \leq t \leq t_{k+1}$. From the form of (4.35), it can be said that the $\bar{\mathcal{S}}_{k+1}$ matrix plays the role of weighing the additional input $\Delta \boldsymbol{a}_k^1$. Since $\bar{\mathcal{S}}_{k+1}$ is a quadratic function of the current estimation error covariance P_k, the resulting $\Delta \boldsymbol{a}_k^1$ is small when having accurate estimation and it is large when having poor estimation. This is reasonable because the vehicle does not need to create the extra maneuver to improve the estimation when the estimate is already sufficiently accurate.

4.2.3 Application 1 : Vision-Based Rendezvous

The EKF-based navigation system and the OSA suboptimal guidance law designed in Subsection 4.2.2 have been applied to two different applications. The first application is a vision-based rendezvous problem with a stationary target in 3-D space. This is the simplest problem to apply the suggested OSA suboptimal guidance law and an improvement in the tracking performance can be clearly seen in the results. Simulation results with a simulated vision-based measurement are shown to illustrate the guidance performance improvement.

(a) Problem Formulation

A mission of the rendezvous with a stationary target at a given terminal time t_f can be formulated as a quadratic minimization problem given in (4.2) with the following parameters

$$A = O, \quad B = I, \quad S_f = \begin{bmatrix} s_x I & O \\ O & s_v I \end{bmatrix}, \quad \boldsymbol{x}_f = \mathbf{0} \tag{4.36}$$

where $s_x > 0$ and $s_v \geq 0$ are constant weights. The vision-based measurement is a target's center position in each image frame, which is formulated in (2.6) in Chapter 2. Because of a fixed image resolution, a noise on the vision-based measurement should depend on the range to the target from the camera. Hence, the measurement error covariance matrix R_k is modeled as follows in the EKF design.

$$R_k = \frac{\sigma^2}{\hat{X}_{c_k}^2} I \tag{4.37}$$

Since it is known that the target is stationary in this application, the EKF is formulated to estimate the relative position of the target with respect to the vehicle. Therefore, the estimation error covariance matrix P_k and the measurement matrix H_{k+1} in the EKF are 3 × 3 and 2 × 3 matrices respectively, and we need to replace the P_{k+1}^- and H_{k+1} matrices in (4.35) by

$$P_{k+1}^- = \begin{bmatrix} P_{k+1}^- & O \\ O & O \end{bmatrix}, \quad H_{k+1} = \begin{bmatrix} H_{k+1} & O \end{bmatrix}$$

(b) Guidance Design

The estimated optimal guidance law at t_k with the parameters given in (4.36) results in the following linear feedback controller.

$$\hat{\boldsymbol{a}}_{v_k}^*(t_k) = K_X \hat{\boldsymbol{X}}_k - K_V \boldsymbol{V}_v(t_k) \tag{4.38}$$

where

$$K_X = \frac{s_x(t_f - t_k) + \frac{s_x s_v}{2}(t_f - t_k)}{\Lambda_k}, \quad K_V = \frac{s_x(t_f - t_k)^2 + \frac{s_x s_v}{3}(t_f - t_k)^3 + s_v}{\Lambda_k}$$

$$\Lambda_k = 1 + \frac{s_x}{3}(t_f - t_k)^3 + \frac{s_x s_v}{12}(t_f - t_k)^4 + s_v(t_f - t_k)$$

In the OSA suboptimal guidance design, an additional guidance input $\Delta \boldsymbol{a}_k^1$ is added to $\hat{\boldsymbol{a}}_{v_k}^*(t_k)$ in order to create an extra motion which improves the estimation performance. For the measurement of the target position in an image, the measurement matrix in the EKF is given by

$$H_{k+1} = \frac{1}{\hat{X}_{c_{k+1}}^-} \begin{bmatrix} -\frac{\hat{Y}_{c_{k+1}}^-}{\hat{X}_{c_{k+1}}^-} & 1 & 0 \\ -\frac{\hat{Z}_{c_{k+1}}^-}{\hat{X}_{c_{k+1}}^-} & 0 & 1 \end{bmatrix} \tag{4.39}$$

where

$$\begin{aligned} \hat{\boldsymbol{X}}_{c_{k+1}}^- &= \begin{bmatrix} \hat{X}_{c_{k+1}}^- & \hat{Y}_{c_{k+1}}^- & \hat{Z}_{c_{k+1}}^- \end{bmatrix}^T = L_c(t_{k+1})\hat{\boldsymbol{X}}_{k+1}^- \\ &= L_c(t_{k+1})\left(\hat{\boldsymbol{X}}_k + \boldsymbol{V}_v(t_k)(t_{k+1} - t_k) - \frac{1}{2}\left(\hat{\boldsymbol{a}}^*(t_k) + \Delta \boldsymbol{a}_k^1\right)(t_{k+1} - t_k)^2\right) \end{aligned}$$

Therefore, the measurement matrix H_{k+1} is a function of the additional input $\Delta \boldsymbol{a}_k^1$ and the derivative (4.35) can be obtained. The resulting $\Delta \boldsymbol{a}_k^1$ is a function of the current estimate $\hat{\boldsymbol{X}}_k$, its error covariance P_k, the known vehicle velocity $\boldsymbol{V}_v(t_k)$ and camera attitude $L_c(t_{k+1})$.

(c) Simulation Results

Simulation results for the vision-based rendezvous with a stationary target are compared between two guidance policies: the OSA suboptimal guidance $\boldsymbol{a}_v^1$ and the estimated optimal guidance $\hat{\boldsymbol{a}}_v^*$. The vehicle is located at the origin with its velocity as $\boldsymbol{V}_v(0) = \begin{bmatrix} 10 & 2 & 0 \end{bmatrix}^T$ (ft/sec) at the initial time, and the target is fixed at $\boldsymbol{X}_t = \begin{bmatrix} 100 & 20 & 20 \end{bmatrix}^T$ (ft). The vehicle's mission is to make a rendezvous with the target at time $t_f = 20$ (sec). $s_x = 100$ and $s_v = 10$ are given. The initial estimation error of the target position is -20 (ft) in each axis, and the initial error covariance is $P_0 = 20^2 I$ (ft^2). For the measurement noise covariance matrix, $\sigma = 1$ is used. Figure 38 shows the vehicle trajectory, and Figure 39 is the vehicle velocity and acceleration which are generated by those two guidance laws. The results of the OSA suboptimal

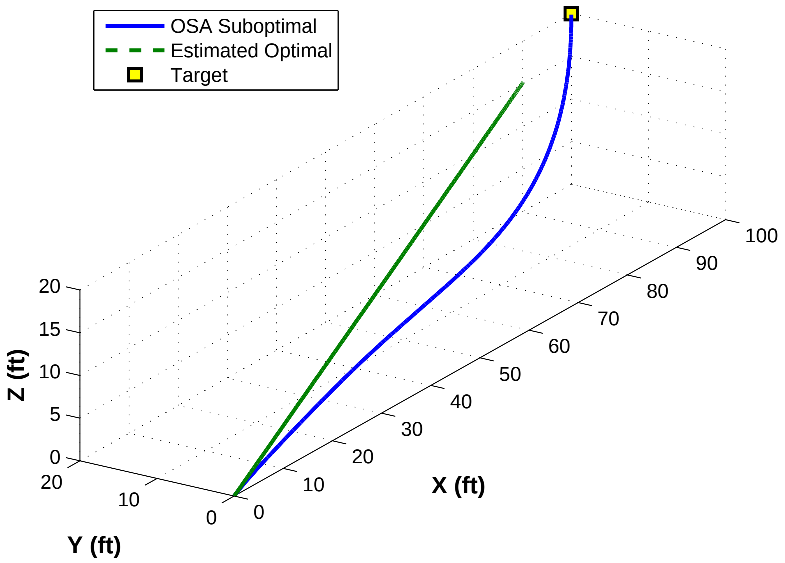


Figure 38: Vehicle Trajectory and Target Location (OSA Suboptimal vs. Estimated Optimal)

guidance law are shown in blue solid lines and those of the estimated optimal guidance law are shown in green dashed lines. Figure 40 summarizes the total cost which includes the terminal tracking error and the control cost for the entire mission. Figure 41 presents a convergence of the estimation error and its standard deviation. When using the estimated optimal guidance policy, the vehicle approaches almost straight to the target. Range observability is lost in such a case, and a large bias in the position estimation error remains. Due to the large bias, the vehicle fails to rendezvous with the target and its final miss distance is 14.8 (ft). On the other hand, the OSA suboptimal guidance law creates lateral motions to keep the range observability and hence improve the vision-based estimation performance, and it enables the vehicle to achieve the rendezvous mission with high accuracy (within 0.03 (ft)). Even though the control cost increases due to the additional input $\Delta \boldsymbol{a}_k^1$, the terminal tracking error and the total cost are significantly reduced by using the OSA suboptimal guidance policy as shown in Figure 40.

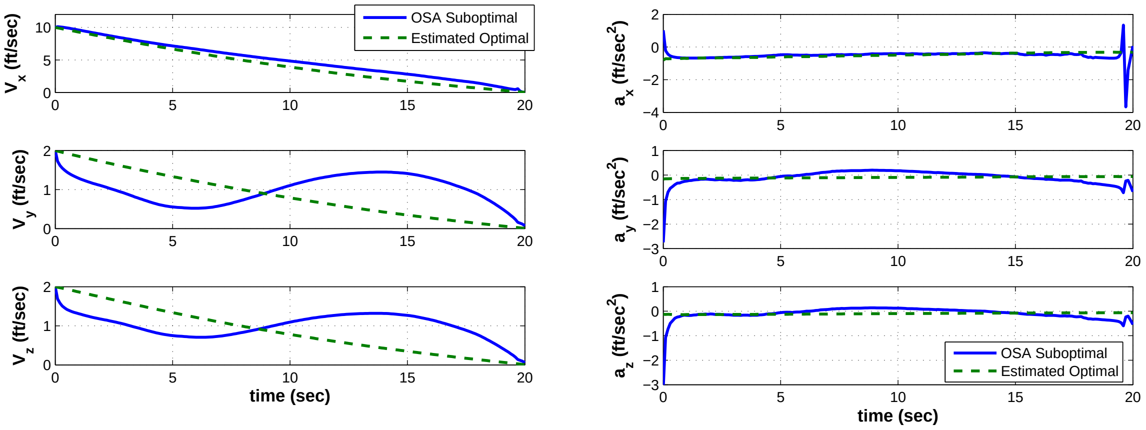


Figure 39: (Left) Vehicle Velocity and (Rihgt) Acceleration (OSA Suboptimal vs. Estimated Optimal)

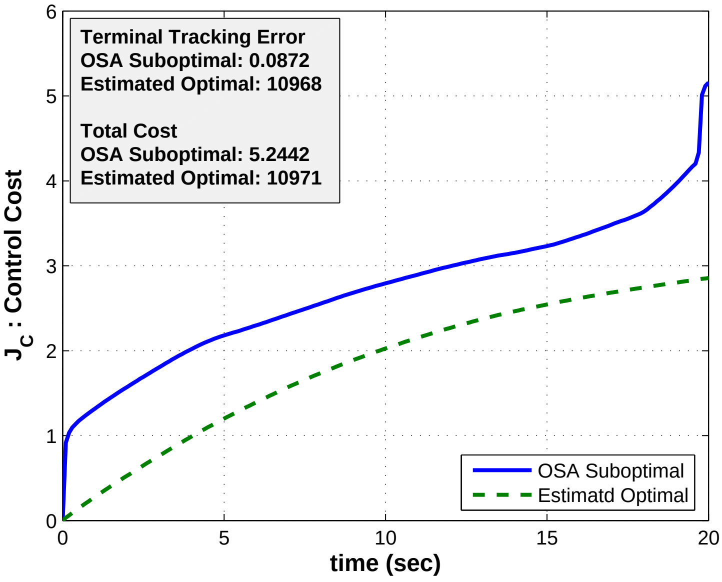


Figure 40: Control Cost, Terminal Tracking Error and Total Cost (OSA Suboptimal vs. Estimated Optimal)

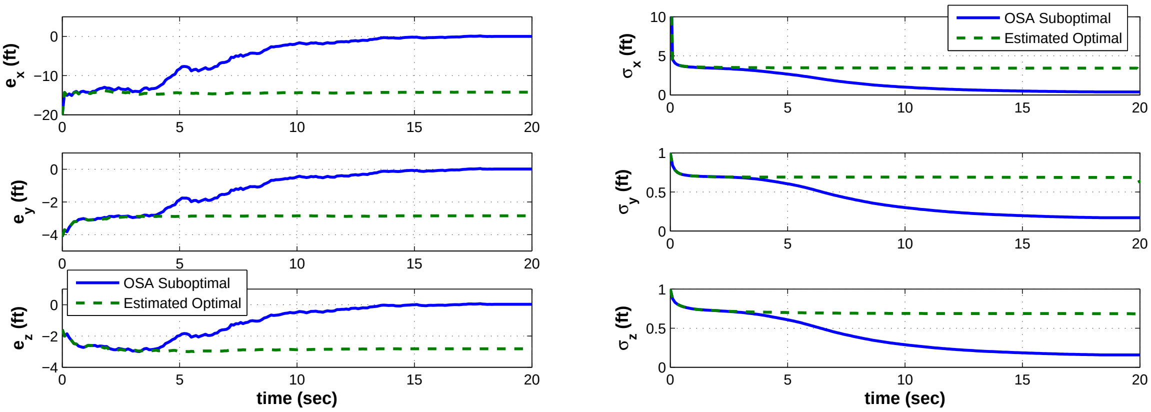


Figure 41: (Left) Position Estimation Error, (Right) Standard Deviation of the Estimation Error (OSA Suboptimal vs. Estimated Optimal)

4.2.4 Application 2 : Vision-Based Obstacle Avoidance

The second application is a combined mission of waypoint tracking and vision-based obstacle avoidance. A vehicle is required to visit a given waypoint while avoiding unforeseen obstacles on its way by using 2-D vision information. This problem has been already treated in Chapter 3. By applying the collision-cone approach developed in Section 3.3, a guidance for obstacle avoidance is reduced to a target tracking problem and the same guidance law that was used in the first application is applied.

(a) Problem Formulation

In this example, a constant known speed U in the X- direction is assumed. The waypoint tracking problem is given by (4.2) with

$$A = O, \quad B = \begin{bmatrix} \infty & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad S_f = \begin{bmatrix} s_x I & O \\ O & s_v I \end{bmatrix}, \quad \boldsymbol{x}_f = \begin{bmatrix} \mathbf{0} \\ -U \\ 0 \\ 0 \end{bmatrix}$$

where $s_x > 0$ and $s_v \geq 0$ are constant weights. Since the relative state between the vehicle and the waypoint is assumed to be completely known, optimal guidance can be realized if there is no obstacle. However, if there is an obstacle which is critical to the vehicle, the vehicle needs to take some avoiding maneuver. Same as in the first application, a vision-based measurement is an obstacle center position in each image frame. Stationary point obstacles are assumed in this problem, and the vehicle always needs to maintain a certain minimum separation distance d from every obstacles to avoid a collision with them.

(b) Guidance Design

Suppose there is no critical obstacle. Then the optimal guidance input for waypoint tracking is given by

$$\boldsymbol{a}^*_{wp}(t_k) = K_X \left(\boldsymbol{X}_{wp} - \boldsymbol{X}_v(t_k)\right) - K_V \boldsymbol{V}_v(t_k) \tag{4.40}$$

where K_X and K_V are the same gain as derived in the previous section. The terminal time is calculated by using a known speed U and the remaining X distance to the waypoint.

$$t_f = t_k + \frac{X_{wp} - X_v(t_k)}{U} \tag{4.41}$$

For obstacle avoidance, the collision-cone criteria developed in Section 3.3 is used to determine the most critical obstacle. Let $\boldsymbol{X}_{obs}$ denote an obstacle position. $\boldsymbol{X}_v$ and $\boldsymbol{V}_v$ are vehicle position and velocity. If the collision cone criteria is satisfied, an aiming point $\boldsymbol{X}_{ap}$ is specified at a tangential point of the collision cone and the obstacle's safety boundary (See Figure 21). Now the obstacle avoidance mission coincides with the aiming point tracking problem. Therefore, a similar guidance law from the waypoint tracking problem can be applied.

$$\hat{\boldsymbol{a}}^*_{oa}(t_k) = K_X \left(\hat{\boldsymbol{X}}_{ap} - \boldsymbol{X}_v(t_k)\right) - K_V \boldsymbol{V}_v(t_k) \tag{4.42}$$

$\hat{\boldsymbol{X}}_{ap}$ is used instead of $\boldsymbol{X}_{ap}$ since the aiming point is determined based on the estimated obstacle position $\hat{\boldsymbol{X}}_{obs}$ instead of the true obstacle's position $\boldsymbol{X}_{oba}$. The gains K_X and K_V are still the same but are calculated by using the terminal time

$$t_f = t_k + \frac{\hat{X}_{ap} - X_{vk}}{U} \tag{4.43}$$

In the OSA suboptimal guidance policy, we add the additional input $\Delta\boldsymbol{a}_k^1$ to the estimated optimal guidance (4.42). $\Delta\boldsymbol{a}_k^1$ is computed by (4.35) in a similar manner as done in the previous example. The overall guidance policy is to apply $\boldsymbol{a}_{wp}^*(t_k)$ when there is no critical obstacle, and to apply $\hat{\boldsymbol{a}}_{oa}^*(t_k) + \Delta\boldsymbol{a}_k^1$ when there is a critical obstacle.

(c) Simulation Results

Simulation results are compared between the OSA suboptimal guidance policy and the estimated optimal guidance policy. The vehicle is initially at the origin with its velocity $\boldsymbol{V}_v(0) = \begin{bmatrix} 10 & 2 & 2 \end{bmatrix}^T$ (ft/sec), and a constant speed in the X-direction is $U = 10$ (ft/sec). A waypoint is given at $\boldsymbol{X}_{wp} = \begin{bmatrix} 100 & 20 & 20 \end{bmatrix}^T$ (ft). There are two unforeseen obstacles on the way to the waypoint: Obstacle 1 at $\boldsymbol{X}_{obs_1} = \begin{bmatrix} 30 & 6 & 6 \end{bmatrix}^T$ (ft) and Obstacle 2 at $\boldsymbol{X}_{obs_2} = \begin{bmatrix} 60 & 20 & 15 \end{bmatrix}^T$ (ft). The minimum separation distance from obstacles is $d = 10$ (ft). The initial estimation error is +50% of the original relative position for each obstacle, and its initial error covariance is $P_0 = 20^2 I$ (ft^2). $s_x = 100$ and $s_v = 10$ are used. Figure 42 shows the vehicle trajectory, obstacle locations and their safety boundaries, and the waypoint location. Figure 43 is the vehicle velocities and accelerations. Figure 44 presents a time profile of distances from each obstacle and Figure 45 shows the control cost. The blue lines are the results using the suggested OSA suboptimal guidance, and the green dashed lines are the results using the estimated optimal guidance. We can see that there is a violation of Obstacle 1's safety boundary when the estimated optimal guidance law is used, and that the violation is removed when using the OSA suboptimal guidance. Therefore the guidance performance of obstacle avoidance is significantly improved by introducing the additional input $\Delta\boldsymbol{a}_k^1$. However, it is obvious that additional input increases a control cost and the total control cost is almost doubled when using the OSA suboptimal guidance policy. Figure 46 shows the estimation error of each obstacle

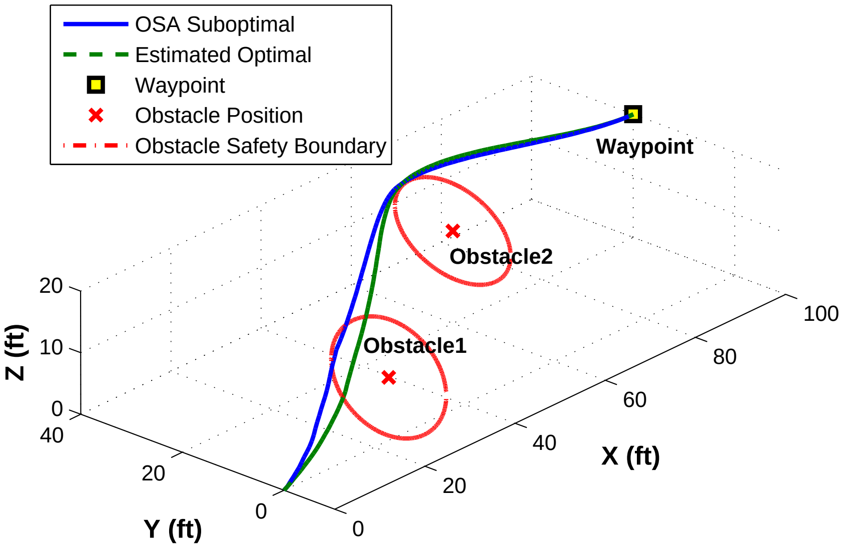


Figure 42: Vehicle Trajectory and Obstacles (OSA Suboptimal vs. Estimated Optimal)

position and its standard deviation. It is clear from the result that convergence of the estimation error is significantly improved by the OSA suboptimal guidance. That is why the guidance performance was improved.

4.3 Extended Suboptimal Optimization Strategies

In this section, two different suboptimal optimization techniques are studied as extended approaches of the OSA suboptimal guidance law to solve the stochastic optimization problem established in Subsection 4.1.3. The previous section suggests the OSA suboptimal guidance and the simulation results showed that it significantly improves the guidance performance compared to the conventional guidance. Also, this guidance strategy does not require the iterative computation and is real-time applicable. However, the OSA assumption used in the design is far from reality and an optimality of the solution has not been investigated. Therefore, this section aims to develop suboptimal guidance strategies that can well-approximate the optimal

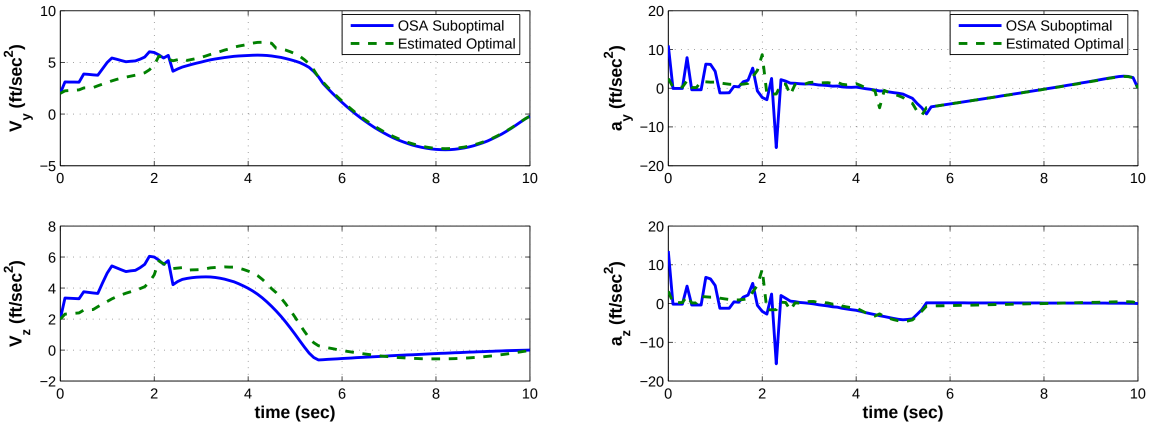


Figure 43: (Left) Vehicle Velocity and (Right) Acceleration (OSA Suboptimal vs. Estimated Optimal)

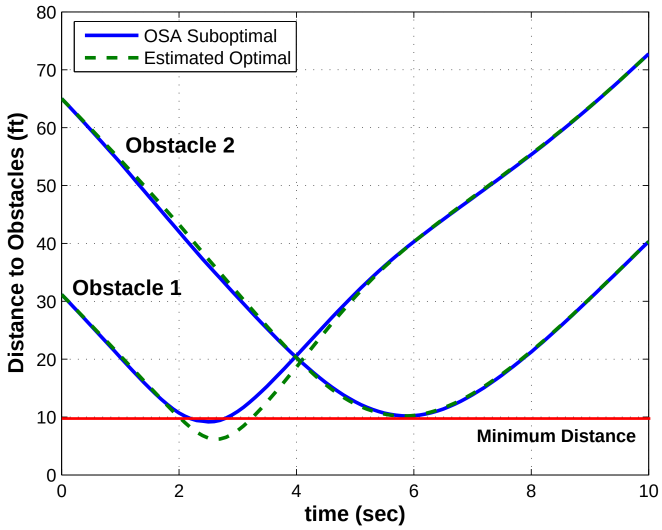


Figure 44: Distance from Obstacles (OSA Suboptimal vs. Estimated Optimal)

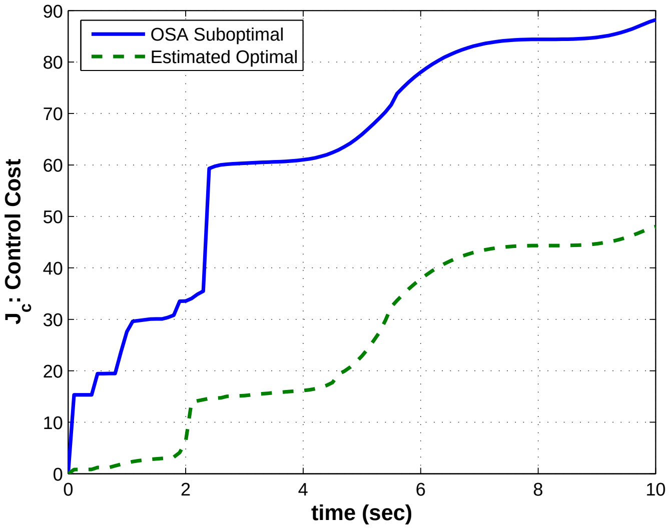


Figure 45: Control Cost (OSA Suboptimal vs. Estimated Optimal)

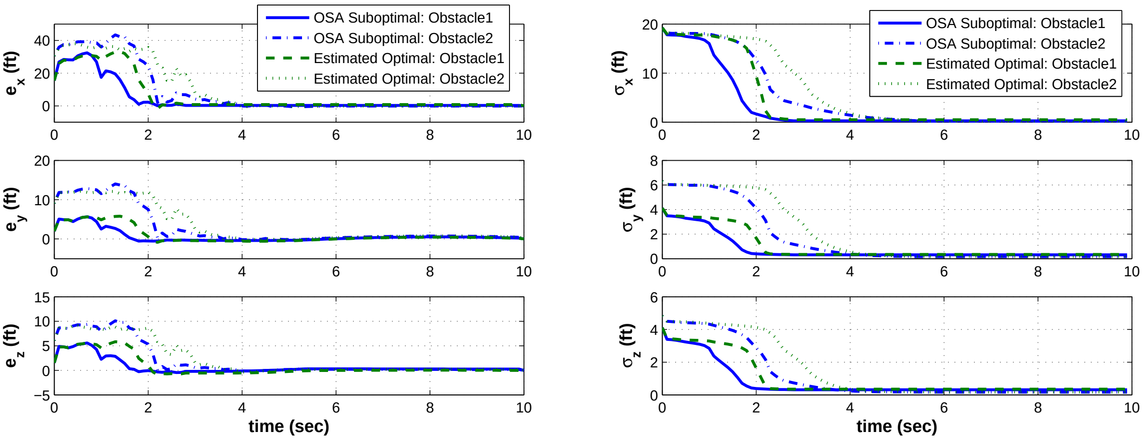


Figure 46: (Left) Position Estimation Error, (Right) Standard Deviation of the Estimation Error (OSA Suboptimal vs. Estimated Optimal)

solution. First, the OSA optimization is generalized as the n-step-ahead (nSA) optimization for an arbitrary number n. Then, it is extended to the p %-ahead suboptimal guidance by changing a value for n at each time step. The nSA and the p %-ahead suboptimal guidance designs are studied with an application to a very simple problem of 2-D vision-based target tracking. Simulation results of the suggested two suboptimal guidance designs are compared with results of the conventional guidance given in Subsection 4.1.2, and also with the optimal solution which is numerically obtained.

4.3.1 n-Step-Ahead Optimization

The first suboptimal guidance strategy proposed in this thesis is called n-step-ahead (nSA) optimization. This approach is a generalization of the OSA optimization approach discussed in the previous section with an arbitrary number of n instead of $n = 1$. In the nSA optimization, the optimization is performed under the assumption that the observer anticipates n more final measurements for n time steps ahead from the current time. The optimal solution of (4.14) can be obtained by setting a sufficiently large number for n. The solutions for different values of n are derived numerically, and their optimality and computational cost will be investigated through the simulation results in Subsection 4.4.

(a) Approach

As done in the previous section, consider the problem of determining a vehicle's acceleration input at the k-th time step t_k given the estimated relative state $\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}(t_k)$ and its error covariance matrix $P_k = P(t_k)$. In the nSA optimization approach, the optimization is performed under the assumption of having measurements for only the first n steps ahead from t_k. Let $\boldsymbol{a}_v^n(t)$ denote the nSA-based suboptimal guidance input computed at t_k. Let

$$\boldsymbol{a}_v^n(t) = \hat{\boldsymbol{a}}^*_{v_{k+i}}(t) + \Delta\boldsymbol{a}^n_{k+i}, \quad t_{k+i} \leq t \leq t_{k+i+1} \tag{4.44}$$

be the vehicle acceleration input for $0 \leq i \leq n-1$. The input $\hat{\boldsymbol{a}}^*_{v_{k+i}}(t)$ denotes the estimated optimal guidance law which is recalculated at t_{k+i} by using the last updated estimate $\hat{\boldsymbol{x}}_{k+i}$ and its error covariance matrix P_{k+i}. $\Delta\boldsymbol{a}^n_{k+i}$ is an additional input which is for estimation improvement. Note that the estimated optimal guidance law at t_k, $\hat{\boldsymbol{a}}^*_{v_k}(t)$, coincides with the zero-step-ahead ($n = 0$) suboptimal guidance $\boldsymbol{a}^0_v(t)$ given in (4.11). Under the nSA optimization assumption, the estimation accuracy will not be improved after the time step t_{k+n} because of the lack of measurements beyond this time. Therefore, the estimated optimal guidance law which was recalculated at t_{k+n} will be applied after t_{k+n}.

$$\boldsymbol{a}^n_v(t) = \hat{\boldsymbol{a}}^*_v(t), \quad t_{k+n} \leq t \leq t_f \tag{4.45}$$

Then, the expected cost $\boldsymbol{E}\left[J(t_k)\right]$ can be derived by assuming $\boldsymbol{E}\left[\tilde{\boldsymbol{x}}(t_{k+i})\right] = \boldsymbol{0}$, $\boldsymbol{E}\left[\tilde{\boldsymbol{x}}(t_{k+i})\tilde{\boldsymbol{x}}^T(t_{k+i})\right] = P_{k+i}$ and $Q_{k+i} = O$. The expected cost becomes a function of the current estimate $\hat{\boldsymbol{x}}_k$ and P_k and the additional acceleration input $\Delta\boldsymbol{a}^n = \begin{bmatrix}\Delta\boldsymbol{a}^{n^T}_k & \Delta\boldsymbol{a}^{n^T}_{k+1} & \cdots & \Delta\boldsymbol{a}^{n^T}_{k+n-1}\end{bmatrix}^T$. The minimization problem (4.14) now becomes a minimization over a single vector $\Delta\boldsymbol{a}^n$, and the optimal $\Delta\boldsymbol{a}$ which minimizes $\boldsymbol{E}\left[J(t_k)\right]$ can be derived by solving the following algebraic equation.

$$\frac{\partial\boldsymbol{E}\left[J(t_k)\right]}{\partial\Delta\boldsymbol{a}^n} = \boldsymbol{0} \tag{4.46}$$

The resulting $\Delta\boldsymbol{a}^n$ can be expressed in terms of the given estimate $\hat{\boldsymbol{x}}_k$ and its error covariance matrix P_k.

(b) Special Case of Terminal Tracking

The special case of terminal tracking (i.e., $A = O$) is considered. As derived in Equation (4.12), the estimated optimal guidance or the zero-step-ahead suboptimal guidance calculated at t_k by using the estimate $\hat{\boldsymbol{x}}(t_k) = \hat{\boldsymbol{x}}_k$ is given as follows.

$$\hat{\boldsymbol{a}}^*_{v_k}(t) = \boldsymbol{a}^0_v(t) = -B^{-1}G^T e^{F^T(t_f-t)}S_f\left(I + \mathcal{G}_k S_f\right)^{-1}\left(e^{F(t_f-t_k)}\hat{\boldsymbol{x}}_k - \boldsymbol{x}_f\right) \tag{4.47}$$

When applying $\boldsymbol{a}_v^0(t)$ for $t_k \le t \le t_f$, the relative state at t_{k+i} is given by

$$\begin{aligned}\boldsymbol{x}^0(t_{k+i}) &= e^{F(t_{k+i}-t_k)}\boldsymbol{x}(t_k) + \int_{t_k}^{t_{k+i}} e^{F(t_{k+i}-s)} G\hat{\boldsymbol{a}}_v^0(s)ds \\ &= e^{F(t_{k+i}-t_k)}\left(\hat{\boldsymbol{x}}_k + \tilde{\boldsymbol{x}}_k\right) \\ &\quad -e^{-F(t_f-t_{k+i})}\left(\mathcal{G}_k - \mathcal{G}_{k+i}\right) S_f \left(I + \mathcal{G}_k S_f\right)^{-1}\left(e^{F(t_f-t_k)}\hat{\boldsymbol{x}}_k - \boldsymbol{x}_f\right) \end{aligned} \quad (4.48)$$

In the nSA suboptimal optimization approach, the guidance input for $t_k \le t \le t_{k+1}$ is a sum of the estimated optimal guidance $\hat{\boldsymbol{a}}_{v_k}^*(t)$ and an additional input $\Delta\boldsymbol{a}_k^n$.

$$\boldsymbol{a}_v^n(t) = \hat{\boldsymbol{a}}_{v_k}^*(t) + \Delta\boldsymbol{a}_k^n = \boldsymbol{a}_v^0(t) + \Delta\boldsymbol{a}_k^n, \quad t_k \le t \le t_{k+1} \quad (4.49)$$

Then, the true relative state at the next time step t_{k+1} becomes

$$\boldsymbol{x}^n(t_{k+1}) = \boldsymbol{x}^0(t_{k+1}) + \Gamma_{k+1}\Delta\boldsymbol{a}_k^n = \boldsymbol{x}^0(t_{k+1}) + \Delta\boldsymbol{x}_{k+1}^n \quad (4.50)$$

where Γ_{k+1} is defined in (4.17) and $\Delta\boldsymbol{x}_{k+1}^n = \Gamma_{k+1}\Delta\boldsymbol{a}_k^n$. Similar as (4.27), the estimated optimal guidance is recalculate at t_{k+1} as follows.

$$\hat{\boldsymbol{a}}_{v_{k+1}}^*(t) = \boldsymbol{a}_v^0(t) - B^{-1}G^T e^{F^T(t_f-t)} S_f \left(I + \mathcal{G}_{k+1}S_f\right)^{-1} e^{F(t_f-t_{k+1})}\left(\Delta\boldsymbol{x}_{k+1}^n + \Delta\tilde{\boldsymbol{x}}_{k+1}\right) \quad (4.51)$$

where $\Delta\tilde{\boldsymbol{x}}_{k+1} = \tilde{\boldsymbol{x}}_{k+1}^- - \tilde{\boldsymbol{x}}_{k+1} = \Phi_{k+1}\tilde{\boldsymbol{x}}_k - \tilde{\boldsymbol{x}}_{k+1}$ denotes an improvement in estimation by the measurement obtained at t_{k+1}. Now, the nSA suboptimal guidance input for $t_{k+1} \le t \le t_{k+2}$ is determined by the sum of $\hat{\boldsymbol{a}}_{v_{k+1}}^*(t)$ and the additional input $\Delta\boldsymbol{a}_{k+1}^n$.

$$\boldsymbol{a}_v^n(t) = \hat{\boldsymbol{a}}_{v_{k+1}}^*(t) + \Delta\boldsymbol{a}_{k+1}^n, \quad t_{k+1} \le t \le t_{k+2} \quad (4.52)$$

By iterating the same procedure for n steps until t_{k+n}, the nSA suboptimal guidance $\boldsymbol{a}_v^n(t)$ for $t_{k+i} \le t \le t_{k+i+1}$ $(0 \le i \le n-1)$ can be derived as

$$\begin{aligned}\boldsymbol{a}_v^n(t) &= \hat{\boldsymbol{a}}_{v_{k+i}}^*(t) + \Delta\boldsymbol{a}_{k+i}^n \\ &= \boldsymbol{a}_v^0(t) + \Delta\boldsymbol{a}_{k+i}^n - B^{-1}G^T e^{F^T(t_f-t)} S_f \sum_{j=1}^{i}\left(I + \mathcal{G}_{k+j}S_f\right)^{-1} e^{F(t_f-t_{k+j})}\Delta\boldsymbol{x}_{k+j}^n \\ &\quad -B^{-1}G^T e^{F^T(t_f-t)} S_f \sum_{j=1}^{i}\left(I + \mathcal{G}_{k+j}S_f\right)^{-1} e^{F(t_f-t_{k+j})}\Delta\tilde{\boldsymbol{x}}_{k+j} \end{aligned} \quad (4.53)$$

where

$$\begin{aligned} \Delta \boldsymbol{x}_{k+j}^{n} &= \Gamma_{k+j} \Delta \boldsymbol{a}_{k+j}^{n} && (4.54) \\ \Delta \tilde{\boldsymbol{x}}_{k+j} &= \tilde{\boldsymbol{x}}_{k+j}^{-} - \tilde{\boldsymbol{x}}_{k+j} = \Phi_{k+j} \tilde{\boldsymbol{x}}_{k+j-1} - \tilde{\boldsymbol{x}}_{k+j}, \quad 1 \leq j \leq n && (4.55) \end{aligned}$$

Then, the true relative state at t_{k+n} is given by

$$\begin{aligned} \boldsymbol{x}^{n}(t_{k+n}) &= \boldsymbol{x}^{0}(t_{k+n}) \\ &+ e^{-F(t_f - t_{k+n})} \left(I + \mathcal{G}_{k+n} S_f \right) \sum_{j=1}^{n} \left(I + \mathcal{G}_{k+j} S_f \right)^{-1} e^{F(t_f - t_{k+j})} \Delta \boldsymbol{x}_{k+j}^{n} \\ &- e^{-F(t_f - t_{k+n})} \sum_{j=1}^{n-1} \left(\mathcal{G}_{k+j} - \mathcal{G}_{k+n} \right) S_f \left(I + \mathcal{G}_{k+j} S_f \right)^{-1} e^{F(t_f - t_{k+j})} \Delta \tilde{\boldsymbol{x}}_{k+j} \end{aligned} \tag{4.56}$$

After the n-time-step ahead from the current time, it is assumed that the measurements are not available, and hence there is no need to add extra maneuvers for estimation performance improvement. Therefore, the estimated optimal guidance law calculated at t_{k+n} is applied for the rest of the time interval $t_{k+n} \leq t \leq t_f$.

$$\begin{aligned} \boldsymbol{a}_v^{n}(t) &= \hat{\boldsymbol{a}}_{v_{k+n}}^{*}(t), \quad t_{k+n} \leq t \leq t_f \\ &= \boldsymbol{a}_v^{0}(t) - B^{-1} G^T e^{F^T(t_f - t)} S_f \sum_{j=1}^{n} \left(I + \mathcal{G}_{k+j} S_f \right)^{-1} e^{F(t_f - t_{k+j})} \Delta \boldsymbol{x}_{k+j}^{n} \\ &\quad - B^{-1} G^T e^{F^T(t_f - t)} S_f \sum_{j=1}^{n} \left(I + \mathcal{G}_{k+j} S_f \right)^{-1} e^{F(t_f - t_{k+j})} \Delta \tilde{\boldsymbol{x}}_{k+j} \end{aligned} \tag{4.57}$$

When applying the nSA suboptimal guidance (4.53) and (4.57), the terminal tracking error will be

$$\begin{aligned} \boldsymbol{x}^{n}(t_f) - \boldsymbol{x}_f &= e^{F(t_f - t_{k+n})} \boldsymbol{x}^{n}(t_{k+n}) + \int_{t_{k+n}}^{t_f} e^{F(t_f - s)} G \hat{\boldsymbol{a}}_{v_{k+n}}^{*}(s) ds - \boldsymbol{x}_f \\ &= \left(\boldsymbol{x}^{0}(t_f) - \boldsymbol{x}_f \right) + \sum_{j=1}^{n} \left(I + \mathcal{G}_{k+j} S_f \right)^{-1} e^{F(t_f - t_{k+j})} \Delta \boldsymbol{x}_{k+j}^{n} \\ &\quad - \sum_{j=1}^{n} \mathcal{G}_{k+j} S_f \left(I + \mathcal{G}_{k+j} S_f \right)^{-1} e^{F(t_f - t_{k+j})} \Delta \tilde{\boldsymbol{x}}_{k+j} \end{aligned} \tag{4.58}$$

Now the expected value of the total cost J is calculated by assuming $\boldsymbol{E}\left[\Delta\tilde{\boldsymbol{x}}_{k+j}\right] = \mathbf{0}$ and approximating its covariance matrix by

$$\boldsymbol{E}\left[\Delta\tilde{\boldsymbol{x}}_{k+j}\Delta\tilde{\boldsymbol{x}}_{k+i}^T\right] = \begin{cases} P_{k+j}^- - P_{k+j} = K_{k+j}H_{k+j}P_{k+j}^-, & i = j \\ O, & i \neq j \end{cases}$$

where P_{k+j}^- is the predicted estimation error covariance matrix calculated in the EKF prediction procedure (4.16). From the nSA guidance input given in (4.53) and (4.57) and the terminal tracking error in (4.58), the expected cost $\boldsymbol{E}\left[J(t_k)\right]$ can derived as

$$J^n(t_k) = \boldsymbol{E}\left[J(t_k)\right] = J^0(t_k) - \frac{1}{2}\sum_{j=1}^{n} trK_{k+j}H_{k+j}P_{k+j}^-\mathcal{S}_{k+j} + \frac{1}{2}\sum_{j=1}^{n}\Delta\boldsymbol{a}_{k+j-1}^{n^T}\mathcal{B}_{k+j}\Delta\boldsymbol{a}_{k+j-1}^{n} \tag{4.59}$$

where

$$\begin{aligned} \mathcal{S}_{k+j} &= e^{F^T(t_f - t_{k+j})}S_f\mathcal{G}_{k+j}S_f\left(I + \mathcal{G}_{k+j}S_f\right)^{-1}e^{F(t_f - t_{k+j})} \\ \mathcal{B}_{k+j} &= \Gamma_{k+j}^T e^{F^T(t_f - t_{k+j})}S_f\left(I + \mathcal{G}_{k+j}S_f\right)^{-1}e^{F(t_f - t_{k+j})}\Gamma_{k+j} + B(t_{k+j} - t_{k+j-1}) \end{aligned}$$

The first term in (4.59), $J^0(t_k) = \hat{J}^*(t_k)$ is the expected cost when using the estimated optimal guidance law given in (4.13). The second term represents an increase in control cost due to the additional input $\Delta\boldsymbol{a}_{k+j}^n$, and the third term represents a decrease in terminal tracking error due to the estimation improvement by the measurement updates at t_{k+j} $(j = 1, \cdots, n)$. The main concept of the optimal guidance design in this thesis is to decrease the total expected cost (4.59) by improving the overall guidance accuracy by making some favorable maneuvers to improve the estimation accuracy. In this example, it means that we would like to make the decrease in cost (the second term in (4.59)) larger than the increase in cost (the third term). Hence, the nSA suboptimal guidance policy is to find a vector $\Delta\boldsymbol{a}^n = \begin{bmatrix}\Delta\boldsymbol{a}_k^{n^T} & \Delta\boldsymbol{a}_{k+1}^{n^T} & \cdots & \Delta\boldsymbol{a}_{k+n-1}^{n^T}\end{bmatrix}^T$ which maximizes the decrease in cost.

$$\max_{\Delta\boldsymbol{a}^n}\Delta J^n(t_k) = \left(J^0(t_k) - J^n(t_k)\right) \tag{4.60}$$

Since this is a maximization problem over a single vector $\Delta\boldsymbol{a}^n$, the optimal $\Delta\boldsymbol{a}^n$ can be obtained by solving

$$\frac{\partial \Delta J^n(t_k)}{\partial \Delta\boldsymbol{a}^n} = \frac{\partial}{\partial \Delta\boldsymbol{a}^n}\frac{1}{2}\sum_{j=1}^{n}\Big(trK_{k+j}H_{k+j}P^-_{k+j}\mathcal{S}_{k+j} - \Delta\boldsymbol{a}^{n^T}_{k+j-1}\mathcal{B}_{k+j}\Delta\boldsymbol{a}^n_{k+j-1}\Big) = \mathbf{0}$$

$$\therefore \quad \Delta\boldsymbol{a}^n_{k+i-1} = \frac{1}{2}\mathcal{B}^{-T}_{k+i}\left[\frac{\partial}{\partial \Delta\boldsymbol{a}^n_{k+i-1}}\sum_{j=1}^{n}\big(trK_{k+j}H_{k+j}P^-_{k+j}\mathcal{S}_{k+j}\big)\right]^T, \quad i = 1, 2, \cdots, n \tag{4.61}$$

In particular, the additional input at the current time step t_k can be obtained by solving (4.61) when $i = 1$. By substituting the Kalman gain K_{k+j} with the form given in (4.20), $\Delta\boldsymbol{a}^n_k$ can be rewritten by

$$\Delta\boldsymbol{a}^n_k = \frac{1}{2}\mathcal{B}^{-T}_{k+1}\left[\frac{\partial}{\partial \Delta\boldsymbol{a}^n_k}\sum_{j=1}^{n}\Big(trP^-_{k+j}\mathcal{S}_{k+j}P^-_{k+j}H^T_{k+j}\left(H_{k+j}P^-_{k+j}H^T_{k+j} + R_{k+j}\right)^{-1}H_{k+j}\Big)\right]^T \tag{4.62}$$

By applying the EKF update and prediction laws, it can be proved that the matrix $P^-_{k+j}\mathcal{S}_{k+j}P^-_{k+j}$ includes a constant matrix

$$\bar{\mathcal{S}}_{k+j} = P_k e^{F^T(t_{k+j}-t_k)}\mathcal{S}_{k+j}e^{F(t_{k+j}-t_k)}P_k \tag{4.63}$$

As explained in Subsection 4.2.2, this $\bar{\mathcal{S}}_{k+j}$ matrix plays the role of weighing the additional input $\Delta\boldsymbol{a}^n_k$ and having the small estimation error covariance matrix results in the small additional input.

The matrix H_{k+j} in (4.62) is the measurement matrix that is defined by the Jacobian of the measurement model $\boldsymbol{h}(\boldsymbol{x})$ evaluated at a predicted estimate $\boldsymbol{x} = \hat{\boldsymbol{x}}^-_{k+j}$ when applying the nSA guidance $\boldsymbol{a}^n(t)$. From the EKF prediction (4.15), it is obvious that the predicted state $\hat{\boldsymbol{x}}^-_{k+j}$ is a function of the last updated estimate $\hat{\boldsymbol{x}}_k$ and the additional inputs $\Delta\boldsymbol{a}^n_{k+i}$ for $i = 0, 1, \cdots, j-1$. Hence, the function $trK_{k+j}H_{k+j}P^-_{k+j}\mathcal{S}_{k+j}$ is highly nonlinear with respect to $\Delta\boldsymbol{a}^n_{k+i-1}$ and the derivative on the right hand side of (4.62) is difficult to solve analytically. This thesis suggests applying a numerical derivation method.

(c) Numerical Derivation

The derivative in (4.61) is numerically solved by using the Secant approximation[41]. The Secant approximation is simply an one-sided first order approximation of the derivative. For a given i, consider the derivative

$$\frac{\partial}{\partial \Delta \boldsymbol{a}^n_{k+i-1}} \sum_{j=1}^{n} \left(trK_{k+j} H_{k+j} P^-_{k+j} \mathcal{S}_{k+j} \right) \tag{4.64}$$

evaluated at $\boldsymbol{a}_v(t)$ with known $\hat{\boldsymbol{x}}_k$ and P_k. First, the traces $trK_{k+j}H_{k+j}P^-_{k+j}\mathcal{S}_{k+j}$ when applying $\boldsymbol{a}_v(t)$, denoted by $trK_{k+j}H_{k+j}P^-_{k+j}\mathcal{S}_{k+j}|\boldsymbol{a}_v$, is computed for each j by iterating the EKF prediction and update process until the time step t_{k+n}. This computation is performed assuming the residual $\left(\boldsymbol{z}_{k+j} - \boldsymbol{h}(\hat{\boldsymbol{x}}^-_{k+j})\right)$ is zero. Let

$$\delta \boldsymbol{a}_{\{1,2,3\}}(t) = \begin{cases} \begin{bmatrix} \delta a \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ \delta a \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ \delta a \end{bmatrix}, & t_{k+i-1} \le t \le t_{k+i} \\ \mathbf{0}, & otherwise \end{cases}$$

be three different deviations for small δa. In the same way, the traces when applying $\boldsymbol{a}_{\{1,2,3\}}(t) = \boldsymbol{a}_v(t) + \delta \boldsymbol{a}_{\{1,2,3\}}(t)$ are computed. Then, the derivative (4.64) is approximated by

$$\frac{1}{\delta a} \sum_{j=i}^{n} \begin{bmatrix} trK_{k+j}H_{k+j}P^-_{k+j}\mathcal{S}_{k+j}\Big|_{\boldsymbol{a}_1(t)} - trK_{k+j}H_{k+j}P^-_{k+j}\mathcal{S}_{k+j}\Big|_{\boldsymbol{a}_v(t)} \\ trK_{k+j}H_{k+j}P^-_{k+j}\mathcal{S}_{k+j}\Big|_{\boldsymbol{a}_2(t)} - trK_{k+j}H_{k+j}P^-_{k+j}\mathcal{S}_{k+j}\Big|_{\boldsymbol{a}_v(t)} \\ trK_{k+j}H_{k+j}P^-_{k+j}\mathcal{S}_{k+j}\Big|_{\boldsymbol{a}_3(t)} - trK_{k+j}H_{k+j}P^-_{k+j}\mathcal{S}_{k+j}\Big|_{\boldsymbol{a}_v(t)} \end{bmatrix} \tag{4.65}$$

Since $\Delta \boldsymbol{a}^n_{k+i-1}$ only affects the terms $K_{k+j}H_{k+j}P^-_{k+j}$ for $j \ge i$, the summation in (4.64) can be replaced by a summation taken only from $j = i$ to n. By using the Secant approximation to derive the derivatives, the equation (4.61) is solved for $\Delta \boldsymbol{a}^n$ by the following iterative procedure:

1. First iteration :

   (i) Determine $\Delta\boldsymbol{a}_k^n$ from (4.61) by evaluating the derivative at

   $$\boldsymbol{a}_v(t) = \hat{\boldsymbol{a}}_{v_k}^*(t) = \boldsymbol{a}_v^0(t), \quad t_k \leq t \leq t_f$$

   $\boldsymbol{a}_v^0(t)$ is the estimated optimal guidance given in (4.47).

   (ii) Iteratively, determine $\Delta\boldsymbol{a}_{k+j}^n$ for $j = 1, 2, \cdots, n-1$ from (4.61) by evaluating the derivative at

   $$\boldsymbol{a}_v(t) = \begin{cases} \hat{\boldsymbol{a}}_{v_{k+i-1}}^*(t) + \Delta\boldsymbol{a}_{k+i-1}^n = \boldsymbol{a}_v^n(t), & t_{k+i-1} \leq t \leq t_{k+i} \quad (i = 1, 2, \cdots, j) \\ \hat{\boldsymbol{a}}_{v_{k+j}}^*(t), & t_{k+j} \leq t \leq t_f \end{cases}$$

   where $\boldsymbol{a}_v^n(t)$ is given by (4.53) with $\Delta\boldsymbol{x}_{k+i}^n = \Gamma_{k+i}\Delta\boldsymbol{a}_{k+i-1}^n$ and $\Delta\tilde{\boldsymbol{x}}_{k+i} = \mathbf{0}$. $\Delta\boldsymbol{a}_{k+i-1}^n$ is the result from the previous j steps. $\hat{\boldsymbol{a}}_{v_{k+j}}^*(t)$ is the estimated optimal guidance calculated at t_{k+j} assuming $\Delta\tilde{\boldsymbol{x}}_{k+i} = \mathbf{0}$ for all $i = 1, 2, \cdots j$.

2. Second and further iterations :

   (i) Set $\Delta\boldsymbol{a}^{n^0} = \Delta\boldsymbol{a}^n$, which was calculated in the previous iteration.

   (ii) Iteratively, determine $\Delta\boldsymbol{a}_{k+j}^n$ for $j = 0, 1, 2, \cdots, n-1$ from (4.61) by evaluating the derivative at

   $$\boldsymbol{a}_v(t) = \begin{cases} \hat{\boldsymbol{a}}_{v_{k+i-1}}^*(t) + \Delta\boldsymbol{a}_{k+i-1}^n = \boldsymbol{a}_v^n(t), & t_{k+i-1} \leq t \leq t_{k+i} \quad (i = 1, 2, \cdots, n) \\ \hat{\boldsymbol{a}}_{v_{k+n}}^*(t), & t_{k+n} \leq t \leq t_f \end{cases}$$

   where $\boldsymbol{a}_v^n(t)$ is given by (4.53) with $\Delta\boldsymbol{x}_{k+i}^n = \Gamma_{k+i}\Delta\boldsymbol{a}_{k+i-1}^n$ and $\Delta\tilde{\boldsymbol{x}}_{k+i} = \mathbf{0}$. $\Delta\boldsymbol{a}_{k+i-1}^n$ for $i = 1, 2, \cdots, j-1$ is the result from the previous j steps and that for $i = j, j+1, \cdots, n$ is the result from the previous iteration. $\hat{\boldsymbol{a}}_{v_{k+n}}^*(t)$ is the estimated optimal guidance calculated at t_{k+j} assuming $\Delta\tilde{\boldsymbol{x}}_{k+i} = \mathbf{0}$ for all $i = 1, 2, \cdots n$.

(iii) If $\|\Delta \boldsymbol{a}^n - \Delta \boldsymbol{a}^{n^0}\| \leq \epsilon$, stop the iteration. Otherwise, start the next iteration. If the number of iterations is larger than N_{max}, stop the iteration.

$\epsilon > 0$ is a given tolerance. N_{max} is a maximum limit on the number of iterations. Hence, the EKF prediction and update process will be repeated at most nN_{max} times.

4.3.2 p %-Ahead Optimization

In this subsection, the nSA suboptimal guidance law discussed in Subsection 4.3.1 is modified by changing the value of n at each time step. This thesis suggests p %-ahead optimization as an alternative method to the nSA optimization. In this approach, the optimization is performed under the assumption that the measurements are available only for the first p % of the rest of the vehicle path. At the current time step t_k, the value of n should be determined as follows.

$$n = round\left(\frac{p}{100} \cdot \frac{t_f - t_k}{t_k - t_{k-1}}\right) \tag{4.66}$$

where $round()$ is a rounding function which returns the nearest integer. Therefore, the number n decreases as time goes to the terminal time t_f. Once n is specified, the guidance input is calculated numerically in the same way as shown in the previous subsection.

4.4 Simulation Results

This section studies the nSA and the p %-ahead suboptimal optimization strategies developed in Section 4.3 through simulations of a 2-D vision-based target tracking problem. The simulation results using these suboptimal guidance strategies are compared with the numerically obtained optimal solution in order to see how well the suboptimal strategies approximate it. The optimal solution can be derived by setting a sufficiently large value for n or p in the nSA or the p %-ahead suboptimal guidance laws. Furthermore, the processing time required to compute the suboptimal guidance solution is compared for different values for n and p.

4.4.1 Simulation Settings

(a) Problem Formulation

In this subsection, 2-D vision-based target tracking problem is considered. In this problem, a vehicle is required to reach a target at a given terminal time t_f. A stationary target is assumed and its location is $\boldsymbol{X}_t = \begin{bmatrix} 100 & 0 \end{bmatrix}^T$ (ft). The vehicle is initially located at the origin $\boldsymbol{X}_v(0) = \begin{bmatrix} 0 & 0 \end{bmatrix}^T$ (ft) with the initial velocity $\boldsymbol{V}_v(0) = \begin{bmatrix} 10 & 0 \end{bmatrix}^T$ (ft/sec). Let $\boldsymbol{X} = \boldsymbol{X}_t - \boldsymbol{X}_v$ be the relative position. The target tracking problem is formulated as a minimization problem with $S_f = s_x I$, $A = O$ and $B = I$.

$$\min_{\boldsymbol{a}_v} J = \frac{s_x}{2} \boldsymbol{X}^T(t)\boldsymbol{X}(t) + \frac{1}{2}\int_0^{t_f} \boldsymbol{a}_v^T(t)\boldsymbol{a}_v(t)dt \tag{4.67}$$

s_x is a constant weight on the terminal tracking error. $s_x = 1$ and $t_f = 10$ (sec) are used in the simulations. If the target's location $\boldsymbol{X}_t$ is known to the vehicle, the optimal guidance input is $\boldsymbol{a}_v^*(t) = \boldsymbol{0}$ (ft/sec^2) for $0 \leq t \leq t_f$ and the optimal cost is $J^*(0) = 0$. However, in this problem, the target location is unknown and is estimated using 2-D vision information. In simulation, the fake image processor outputs are created by adding a range-related noise to the true image coordinate of the target. Since a target's size appears smaller when the range from the camera to the target is larger, it is assumed that the measurement noise is inversely related to the range. The camera's attitude is controlled so that its optical axis is always aligned with the inertial X-axis in this problem. The image processor output at time t_k is given by

$$z_k = \frac{Y(t_k)}{X(t_k)} + \frac{\sigma_x}{X(t_k)}\nu_k \tag{4.68}$$

where ν_k is a standard white Gaussian noise and $\sigma_x = 1$ (ft) is used. The sampling time used in the simulation is $\Delta t = 0.1$ (sec).

(b) EKF Settings

An EKF is applied to estimate the relative position $\boldsymbol{X}$ from the vision-based measurement z_k given in (4.68). The EKF is initialized by using the first measurement

z_0 and a 20 (ft) under-estimated range $r_0 = 80$ (ft). The initial estimated state and its error covariance matrix are set as follows.

$$\hat{\boldsymbol{X}}_0 = r_0 \begin{bmatrix} 1 \\ z_0 \end{bmatrix}, \quad P_0 = 20^2 \begin{bmatrix} 1 & 0 \\ 0 & z_0^2 \end{bmatrix} \tag{4.69}$$

The measurement and process noise covariance matrices used in the EKF are

$$R_k = \left(\frac{\sigma_z}{\hat{X}_k} \right)^2, \quad Q_k = \sigma^2 I \cdot \Delta t \tag{4.70}$$

where $\sigma_z = 1$ (ft) and $\sigma = 0.1$ (ft/sec).

4.4.2 Results 1 : n-Step-Ahead Suboptimal Guidance

First, the simulations are performed with the nSA suboptimal guidance law for different values of n's from $n = 0$ to $n = 100$. The case of $n = 0$ is the estimated optimal guidance (4.47). Since the sampling time is $\Delta t = 0.1$ (sec) and the simulation time is 10 (sec) in this simulation, the case of $n = 100$ corresponds to the optimal solution which was solved without any assumptions. The case of $n = 1$ is the OSA suboptimal guidance law established as a real-time applicable guidance strategy in Section 4.2. In the simulations, a series of random numbers was used for the measurement noise ν_k in (4.68). Hence, the average costs and processing times are taken over 100 simulation runs for each value of n. A left plot in Figure 47 compares examples of vehicle trajectories for cases of the different n's while simulating with the same series of random numbers for ν_k. A right figure of Figure 47 shows the average control effort J_c (in green), the average terminal tracking error J_f (in red), and the average total cost $J = J_c + J_f$ (in blue) versus the number of steps n. Figure 48 presents examples of the target position estimation error and its standard deviation from the same simulation in which the results of the vehicle trajectory in Figure 47 were obtained.

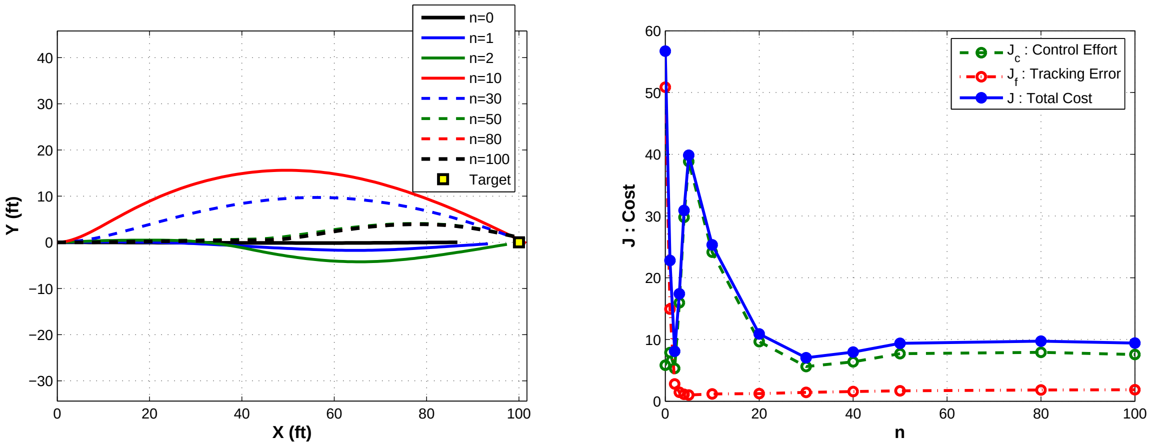


Figure 47: (Left) Example of Vehicle Trajectory, (Right) Control Effort, Terminal Tracking Error, and Total Cost (nSA)

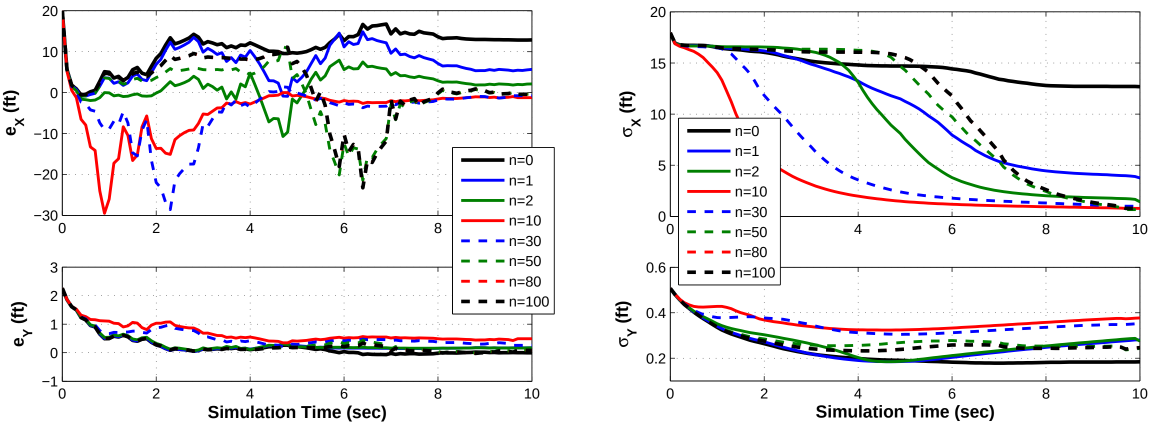


Figure 48: Example of (Left) Position Estimation Error and (Right) Error Standard Deviation (nSA)

Table 2: Average Costs (nSA : $n = 0,\ 1,\ 50,\ 100$)

n	Tracking Error : J_f	Control Effort : J_c	Total Cost : J
0	50.865	5.8310	56.696
1	14.917	7.873	22.791
50	1.678	7.705	9.383
100	1.867	7.5585	9.424

In Figures 47 and 48, the results of the estimated guidance ($n = 0$) and the optimal guidance ($n = 100$) are shown in a black solid line and in a black dashed line, respectively. The results of the OSA suboptimal guidance design is shown in a blue solid line. When $n = 0$, the vehicle moves straight towards the target and results in a large tracking error. By applying the idea of stochastic optimal guidance, a lateral maneuver which improves the range observability is created for the vehicle motion. From Figure 48, it is clear that the estimation performance in range X is improved by the suggested nSA suboptimal guidance design. Since this problem has a symmetric configuration, there exist two symmetric trajectory candidates for the optimal solution. However, one of them is specified to be optimal since the measurement noise ν_k breaks the symmetry. In the example shown in Figure 47, the initially estimated lateral position has a negative error, and hence the optimal guidance includes a lateral motion in the opposite (positive) side. As shown Figure 47, for cases of $n \leq 5$, the guidance law tries to reduce the tracking error J_f by increasing the control effort J_c. When $n = 5$, the guidance law achieves a very accurate tracking performance, but with a large control effort which is far from optimal. After $n = 5$, the resulting lateral motion in the vehicle trajectory becomes smaller to reduce the control effort as n increases. At the same time, the motion is created later in the trajectory, or at a closer distance to the target. This is because the variation in the vision-based measurements is still large with the smaller motion. The compensation between the control cost and the tracking error settles down after $n = 50$, and the optimal solution is well-approximated by the nSA suboptimal guidance for cases of $n \geq 50$. Table 2 compares the averaged costs between the estimated optimal ($n = 0$), the OSA suboptimal ($n = 1$), the 50-steps-ahead suboptimal ($n = 50$) and the optimal ($n = 100$) guidance laws. By compared to the optimal solution, the OSA suboptimal guidance law has a large tracking error. However, the total cost J is reduced by almost 60 % by using the OSA instead of the estimated guidance law.

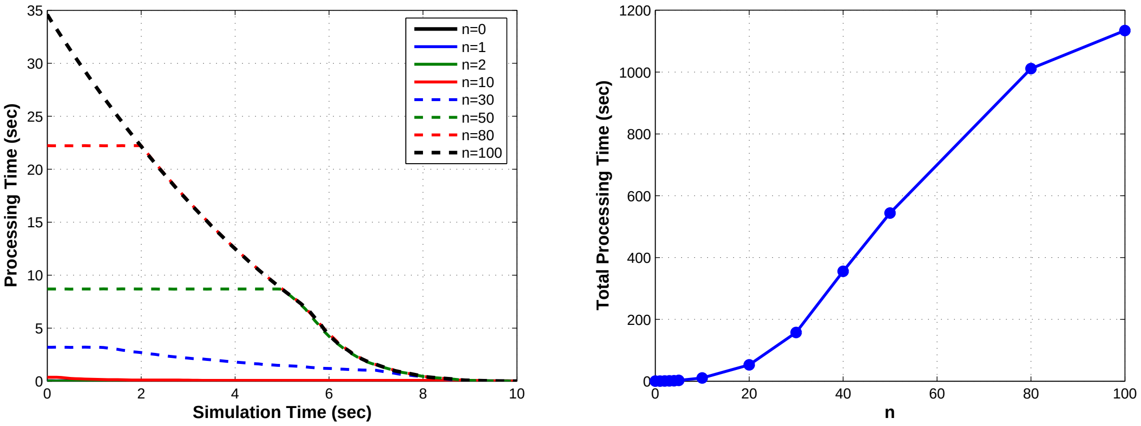


Figure 49: (Left) Processing Time at Each Time Step and (Right) Total Processing Time for 100 Time Steps (nSA)

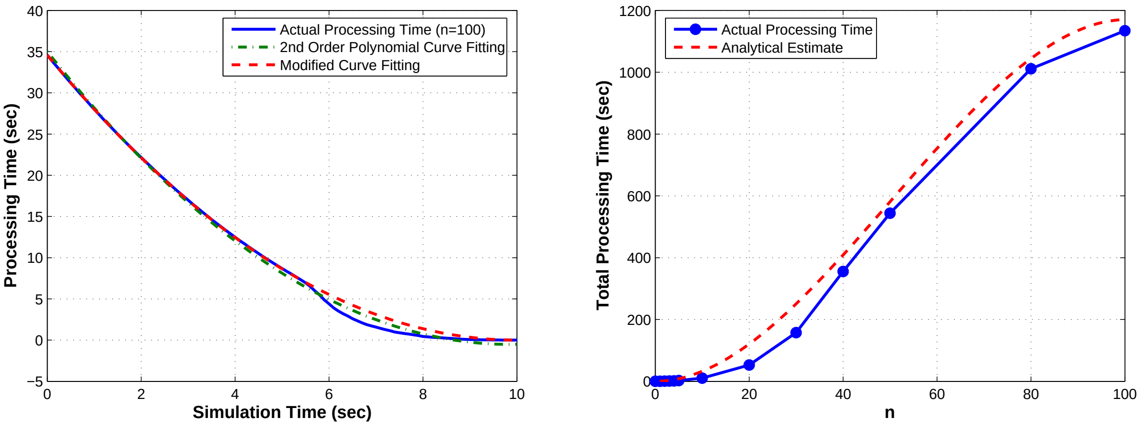


Figure 50: (Left) Curve Fitting for Processing Time at Each Time Step ($n = 100$), (Right) Estimated Total Processing Time

Table 3: Average Processing Time (nSA : $n = 0,\ 1,\ 50,\ 100$)

n	At Each Time Step (sec)	Total Time (sec)
0	2.132×10^{-4}	0.0215
1	0.0016	0.1585
50	8.6948	543.840
100	34.5938	1.134×10^{3}

Figure 49 presents the average processing time at each time step $t_k = 0.1k$ (sec) and the average total processing time for all 100 time steps. Simulations are performed by using MATLAB with a processor of Intel Core2Duo 1.6 MHz. The processing time at each time step is almost constant at the beginning, and it decreases after certain time since there are less than n steps left to reach the terminal time t_f beyond that time. The curve fitting is performed on the resulting processing time in the first plot of Figure 49 for the case of $n = 100$. When $n = 100$, a number of steps-ahead to calculate at the time step $t_k = 0.1k$ coincides with a number of steps left to the terminal time, i.e. $(100 - k)$. Therefore, theoretically, the processing time should decrease as a function of $(100 - k)$. By performing the 2nd order polynomial curve fitting on the result, the following parabolic curve is obtained.

$$\begin{aligned}\{\text{processing time at } t_k\} &= 0.3648t_k^2 - 7.2015t_k + 35.0171 \\ &= 0.3648\left(9.87 - t_k\right)^2 - 0.5222 \\ &= 0.003648\left(98.7 - k\right)^2 - 0.5222 \qquad (4.71)\end{aligned}$$

Hence, the processing time is almost linear to the square of the number of steps-ahead to calculate. A left plot of Figure 50 shows the actual average processing time and the fitted curve given by (4.71). However, the parabola (4.71) gives a negative value at and around $k = 99$ which is not appropriate. Therefore, the curve was modified by using the average processing time at t_0 as follows.

$$\{\text{processing time at } t_k\} = 0.3459\left(10 - t_k\right)^2 = 0.003459\left(100 - k\right)^2 \qquad (4.72)$$

The modified fitted curve (4.72) was also shown in Figure 50. There is a large mismatch when the number of steps left is small (after $t = 6$). This is due to the iteration stopping algorithm (iii) on the top of Page 96. Now, suppose that the processing time to calculate the $(100 - k)$-step-ahead optimization is given by (4.72). Consider the result of the total processing time for 100 time steps shown in a right plot of Figure

49. When using the nSA suboptimal guidance, a number of steps-ahead to calculate at the time step t_k is given by

$$\{\text{a number of steps-ahead to calculate at } t_k\} = \begin{cases} n, & k < 100 - n \\ 100 - k, & k \geq 100 - n \end{cases} \tag{4.73}$$

Then, for the nSA suboptimal guidance, the processing time at each time step is

$$\{\text{processing time at } t_k\} = \begin{cases} 0.003459n^2, & k < 100 - n \\ 0.003459\left(100 - k\right)^2, & k \geq 100 - n \end{cases} \tag{4.74}$$

By taking a summation of this, the total processing time can be estimated by

$$\begin{aligned} \{\text{total processing time}\} &= \sum_{k=0}^{100} \{\text{processing time at } t_k\} \\ &= 0.003459\left(\sum_{k=0}^{100-n-1} n^2 + \sum_{k=100-n}^{100} (100 - k)^2\right) \\ &= 0.003459\left((100 - n)n^2 + \frac{1}{6}n(n+1)(2n+1)\right) \end{aligned} \tag{4.75}$$

A right figure in Figure 50 compares the actual total processing time with its analytical estimate obtained by (4.75). Again, the difference between the results comes from the stopping algorithm.

Table 3 compares these processing times for the cases of $n = 0,\ 1,\ 50,\ 100$. In these simulation results, only the cases with $n \leq 20$ have a processing time at each time step less than the actual sampling time 0.1 (sec) and the total time less than the actual simulation time 10 (sec). From the results shown in Table 2 and Table 3, the OSA suboptimal guidance law can still be considered very efficient because it achieves the significant improvement in the guidance performance (which corresponds to the decrease in the total cost) with a very small increase in the processing time.

4.4.3 Results 2 : p %-Ahead Suboptimal Guidance

The p %-ahead suboptimal guidance proposed in Subsection 4.3.2 is applied to the same simulation. Similar as done for the nSA suboptimal guidance, simulation results are compared for different values of p from 0 to 100 (%). Again, the case of $p = 0$ (%) corresponds to the estimated optimal guidance law and the case of $p = 100$ (%) corresponds to the optimal guidance law. A left plot of Figure 51 shows examples of the vehicle trajectories for the cases of several different values of p while simulating with the same series of random numbers as used in simulations of Figure 47. A right plot of Figure 51 summarizes the control cost J_c, the tracking error J_f and the total cost J that are averaged over 100 simulations. Figure 52 present examples of the estimation performance. From these results, it can be said that the p %-ahead suboptimal solution well-approximates the optimal solution when $p \geq 60$ % in this problem. Table 4 compares the average costs for the cases of $p = 0,\ 10,\ 60,\ 100$ %.

We have seen that the simulation results of the p %-ahead suboptimal guidance law were similar as those of the nSA suboptimal guidance law. However, a main difference between the two appears in their computational loads. Figure 53 shows the average processing time at each time step $t_k = 0.1k$ (sec) and the average total processing time versus p. Since the number of steps n is determined by (4.66) in the p %-ahead suboptimal guidance law, the n linearly decreases from p to 0 as k increases from 0 to 100. Thus, as shown in Figure 53, the processing time at t_k decreases almost linearly with regard to t_k. The second plot in Figure 53 shows that the average total processing time of the p %-ahead optimization is always smaller than that of the nSA optimization for $n = p$. Table 5 shows the processing time at the initial time t_0, which is the maximum processing time at one time step, and the total processing time for the cases of $p = 0,\ 10,\ 60,\ 100$ (%). Comparing the total processing time for the case of $p = 60\%$ in this table and that for the case of $n = 50$ in Table 3, the well-approximated solution is obtained with about 25 % less computational load by using the p %-ahead suboptimal guidance law.

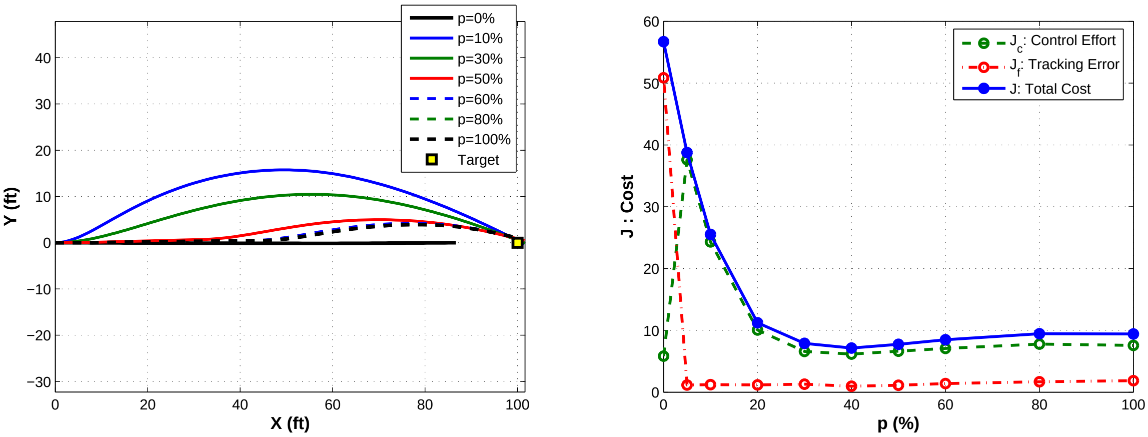


Figure 51: (Left) Example of Vehicle Trajectory, (Right) Control Effort, Terminal Tracking Error, and Total Cost (p %-ahead)

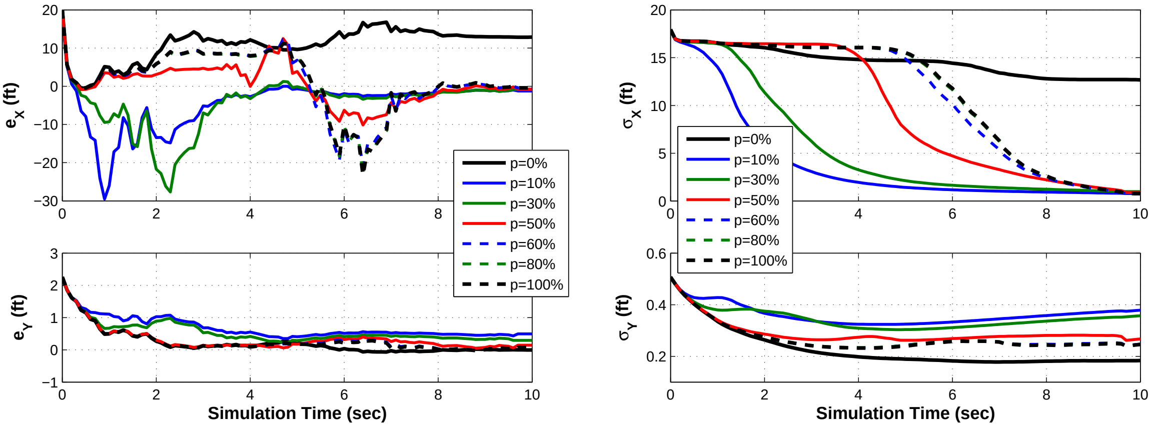


Figure 52: Example of (Left) Position Estimation Error and (Right) Error Standard Deviation (p %-ahead)

Table 4: Average Costs (p %-ahead : $p = 0,\ 10,\ 60,\ 100$ %)

p (%)	Tracking Error : J_f	Control Effort : J_c	Total Cost : J
0	50.865	5.8310	56.696
10	1.221	24.307	25.529
60	1.402	7.082	8.485
100	1.867	7.558	9.424

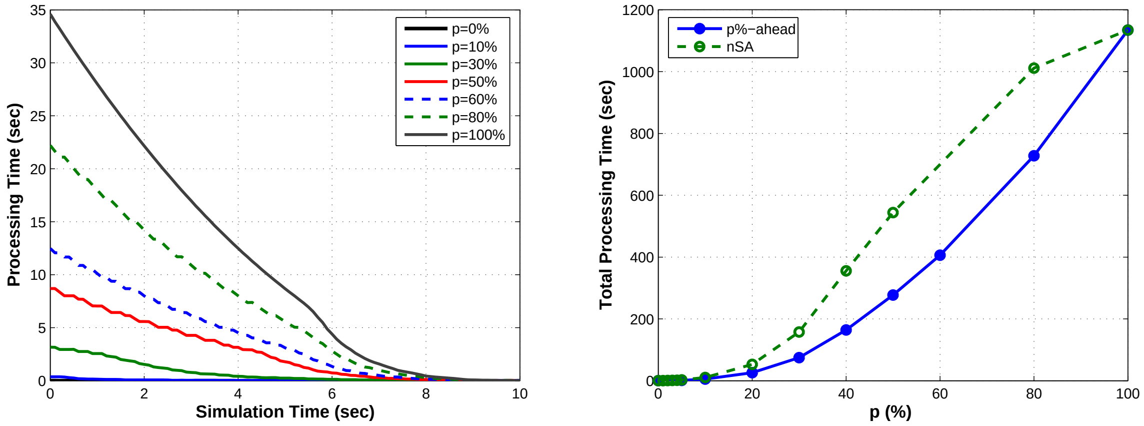


Figure 53: (Left) Processing Time at Each Time Step, (Right) Total Processing Time for 100 Time Steps (p %-ahead)

Table 5: Average Processing Time (p %-ahead : $p = 0,\ 10,\ 60,\ 100$ %)

p (%)	At Initial Time Step (sec)	Total Time (sec)
0	9.360×10^{-4}	0.0215
10	0.370	5.213
60	12.483	405.960
100	34.594	1.134×10^3

4.5 Conclusion

This chapter developed the stochastic optimal guidance design which improves the relative guidance performance under the condition of using the EKF to estimated the relative states from the 2-D vision-based measurements. The one-step-ahead (OSA) optimization approach was used to solve the stochastic optimization problem. The OSA suboptimal guidance law is applied to two different problems; the vision-based rendezvous with a stationary target and the vision-based obstacle avoidance. The significant improvement in the overall guidance performance was observed in the simulation results of both two applications. Furthermore, the OSA suboptimal guidance was generalized as the n-step-ahead (nSA) suboptimal guidance. The p %-ahead suboptimal guidance was also suggested as an extension of the nSA suboptimal guidance. Simulations of the 2-D vision-based target tracking problem were performed by using these guidance laws, and the results were compared for different values of n and p to examine the optimality and the computational cost. It has seen that the p %-ahead suboptimal guidance law can achieve the well-approximate optimal solution with less processing time compared with the nSA suboptimal guidance. From the simulation results of the nSA suboptimal guidance law, it can be concluded that the OSA suboptimal guidance law is sill effective in improving the guidance performance in real-time implementations relative to conventional approaches, but is not necessarily close to the optimal solution.

CHAPTER V

THESIS CONTRIBUTIONS, CONCLUDING REMARKS AND RECOMMENDED FUTURE RESEARCH

This chapter summarizes the contributions and conclusions of this thesis, and also recommends future research. This thesis addressed the monocular vision-based relative navigation and guidance design for unmanned aerial vehicles (UAVs). Unlike a stereo-vision system, the major challenge in the monocular vision-based estimation problem is that the range (or depth) information is not necessarily observable. One way to ensure the observability is to extract information about the target other than its image coordinates, such as the target's size. The relative navigation design for the vision-based formation flight discussed in Section 2.3 utilized the image size of the leader airplane's wingspan as an additional measurement to guarantee range observability. The other way is to create camera motion that is perpendicular to the line-of-sight to the target. This approach is known as *Motion Stereo*. Chapter 4 of this thesis formulated the stochastic optimal guidance problem which maximizes the accuracy of the target tracking performance under the condition of using the EKF-based navigation system. In this work, the vehicle trajectory was optimized to minimize the weighted sum of the tracking error and the control effort. For real-time implementation of the optimal guidance law, the suboptimal optimization technique was suggested and its optimality and computational cost were examined.

Another challenge of vision-based control problems is the real-time implementation of all the algorithms. In this thesis, the relative navigation and guidance systems have been integrated with the real-time image processing algorithms, and the integrated system was applied in various vision-based control applications of UAVs including formation flight, obstacle modeling and avoidance, and target tracking. In particular, Chapter 3 focuses on the vision-based obstacle avoidance and waypoint tracking problem. The navigation filter design was based on the EKF, and the guidance law was derived based on the collision-cone approach and minimum-effort guidance. These were implemented and evaluated in image-in-the-loop simulations and flight tests.

5.1 Contributions and Conclusions

Three main contributions of this thesis are listed in this section.

5.1.1 Real-Time Implementation of the Monocular Vision-Based Relative Navigation Filter

Chapter 2 presented the vision-based relative navigation filter design using an extended Kalman filter (EKF). The EKF-based navigation design has been integrated with the real-time image processor, and applied to two different vision-based control and navigation applications; the formation flight between two UAVs and the 3-D obstacle modeling problem. The navigation performance of the suggested estimator designs are verified in simulations and flight tests.

In the first application of vision-based formation flight, the follower aircraft is required to maintain a position command relative to the leader aircraft by using vision-based information. The EKF is formulated to estimate the leader aircraft's relative states from the measurement of its image coordinates and size in each image frame. The leader's size on the image is used as a measurement to guarantee the range observability. The originality of the EKF design for this target tracking application is that the unit vector from the follower to the leader aircraft is chosen as the estimated state, instead of the bearing and elevation angles that are commonly used, to avoid a singularity. The estimation accuracy of the EKF is verified in the image-in-the-loop

6 DoF multi-aircraft flight simulation. The highlight of this work is that closed-loop vision-based formation flight has been achieved with the unmanned helicopter, the GTMax, as the follower and the unmanned model airplane, the GTEdge, as the leader. This may have been the first time automated formation flight based on vision only and without communication between two aircraft has ever been done. By using the same image processor and navigation system, the closed-loop vision-based formation between the two fixed-wing airplanes was also achieved. These successful flights proved that the EKF-based relative navigation filter is effective in vision-based control applications of UAVs.

Section 2.4 of this thesis proposed the line-based estimator design for the vision-based 3-D obstacle modeling problem. The suggested line-based obstacle modeling system recovers the 3-D obstacle edges from the measurements of their 2-D projections onto the image plane. Such measurements can be obtained by an image segmentation technique. The benefit of using the line-based estimator instead of using a feature point-based estimator, which is used in most related work, is that it can make use of the connectivity information between the feature points. In this application, the relative states of multiple objects need to be estimated. In order to solve the correspondence problem associating the measurements to the estimates, this thesis suggested applying the statistical z-test. After performing the measurement assignment using the z-test algorithm, an EKF is applied to estimate two endpoint positions of each obstacle edge from residuals between the measured line segment and the estimated edge that is projected onto an image plane. In addition, line addition and extension algorithms were developed in this thesis to enable the estimator to update the 3-D estimated obstacle edges that are partially visible in one image frame. Simulation results with the simulated vision-based measurements were presented to verify the capability of the suggested line-based estimator to recover 3-D objects. Furthermore, this estimator design was implemented with the real-time image segmentation algorithm in an image-in-the-loop simulation of vision-based 3-D grid terrain modeling, where it correctly estimated the 3-D terrain model.

5.1.2 3-D Collision Criteria and Minimum-Effort Guidance for Vision-Based Obstacle Avoidance

Chapter 3 considered the vision-based obstacle avoidance and waypoint tracking problem. In this problem, the vehicle is required to reach a given waypoint while avoiding unforeseen obstacles on its path to the waypoint by detecting the obstacles positions using an onboard camera. One of the contributions of this thesis is that it established the 3-D collision criteria. A large amount of research work has been done for autonomous obstacle avoidance guidance for UAVs. However, most of these restrict the algorithms to a case of 2-D planar flight at constant altitude, and do not take advantage of the vehicle's full maneuver capability. Unlike these approaches, the guidance strategy for obstacle avoidance designed in this thesis creates the 3-D avoidance maneuvers. In Section 3.3, the collision-cone approach was extended from 2-D to 3-D by considering the 2-D plane formed by the relative position and velocity vectors from the vehicle to the obstacle. Each obstacle is examined if it is critical to the vehicle by imposing the 3-D collision-cone and the time-to-go criteria. For the critical obstacle, the aiming point location is specified and the vehicle is guided to reach the waypoint via this aiming point.

Another contribution of this thesis is that it applied the minimum-effort guidance (MEG) for multiple target tracking to the guidance design and it showed that the control effort require to reach the waypoint can be reduced by using the suggested MEG-based guidance. The conventional approach was to apply proportional navigation (PN) for single target tracking sequentially, to achieve the aiming point tracking and then to achieve the waypoint tracking. While that sequential PN guidance law minimizes the control effort over a path from the vehicle's current position to the aiming point, and then over a path from the aiming point to the waypoint separately, the MEG-based guidance law minimizes it over the vehicle's entire flight path with the interior position constraint at the aiming point. Therefore, the vehicle can achieve the waypoint tracking and obstacle avoidance mission more efficiently with the suggested MEG approach than with the conventional PN guidance law. This was also verified by the simulation results.

The vision-based navigation and MEG-based guidance law for obstacle avoidance and waypoint tracking problem have been implemented in an image-in-the-loop 6 DoF UAV flight simulation, and their performances were evaluated in a realistic configuration using a realistic vehicle model. This simulation was in direct preparation for flight tests, and finally closed-loop vision-based obstacle avoidance was achieved using the GTmax and the balloon obstacle.

5.1.3 Stochastic Optimal Guidance Design

Chapter 4 proposed the stochastic optimal guidance design for the vision-based target tracking problem. Since the vision-based relative navigation performance highly depends on vehicle motion (or camera motion) relative to the target, the guidance law is designed to achieve the target tracking mission while creating the vehicle motion that improves the vision-based estimation performance. It is well-known that motion lateral or vertical to the line-of-sight to target improves observability and hence improves the estimation accuracy. However, the excessive maneuvers created result in large control effort. Therefore, this thesis suggested a stochastic optimal guidance which minimizes the expected cost defined by a weighted sum of the tracking error and control cost. The optimization was performed subject to the EKF process. In many studies on observer trajectory optimization, an estimation performance cost is chosen by design and is minimized or maximized subject to the vehicle's dynamics. The suggested stochastic optimal guidance design takes a different approach from these other studies in that it does not require selection of the performance cost in the design procedure.

Due to the nonlinear measurement model, this optimization problem is also nonlinear and can only be solved numerically. For the purpose of real-time implementation of the guidance law, this thesis suggested applying the suboptimal optimization technique based on the one-step-ahead (OSA) optimization in order to solve the stochastic optimization problem. In the OSA optimization approach, the stochastic optimization problem is solved under the assumption that only one more measurement is available at the next time step. By applying the OSA optimization, the suboptimal guidance law was obtained without any iterative computations. In this thesis, the OSA suboptimal guidance law was applied to two different vision-based target tracking applications; the vision-based rendezvous with a stationary target and the vision-based obstacle avoidance. The simulation results were compared with those due to the conventional guidance law which uses the optimal guidance law solved with the assumption of full-information. The results illustrated the significant improvement in the estimation and hence overall guidance performance by using the suggested guidance law in both the two applications.

Furthermore, to examine the optimality of the OSA suboptimal guidance law, the OSA optimization approach was generalized as the n-step-ahead (nSA) optimization with an arbitrary number for n. The nSA suboptimal guidance was derived by using a numerical derivation method. The suboptimal solution approaches the optimal solution as the number n increases, and the optimal solution can be numerically obtained when n is sufficiently large. However, the computational load becomes large when n is large. By performing a 2nd order polynomial curve fitting, it was shown that the processing time increases quadratically with n. The good match between the analytically estimated processing time and the actual processing time was also shown. In addition, the nSA suboptimal guidance is modified by changing the number

n at each time step, and the p %-ahead suboptimal guidance was developed. The nSA and the p %-ahead suboptimal guidance laws were studied in a simulation of vision-based target tracking in 2-D plane. Their optimality and computational costs were examined for different values of n and p. In particular, by comparing the OSA suboptimal guidance solution with the optimal solution, it was concluded that the OSA suboptimal guidance law is effective for real-time implementation even though it is not a very well-approximated optimal solution.

5.2 Recommended Future Research

Several recommended future works related to the research topics studied in this thesis are provided in this section.

5.2.1 Relative Navigation with Unknown Target Maneuvers

Throughout this thesis, the target motion is assumed to be known in the vision-based relative navigation design. In all the applications in this thesis except for the formation flight in Section 2.3, stationary targets were assumed. Even in the formation flight application, the leader airplane's maneuver is assumed to be known (either non-accelerating or circling). However, for many practical applications such as adversarial target tracking, the target motion is unknown to the own-ship vehicle and the relative motion dynamics include uncertainty associated with the target motion. There are two issues to be solved in the relative navigation design with unknown target maneuvers. One is how to model the target motion dynamics, and the other is how to compensate the uncertainty. There are several approaches that suggest augmenting the EKF-based estimator design with neural network-based adaptation to solve this problem[49][66]. It would be a challenging but important research topic to design the navigation filter in the presence of unknown target maneuvers and investigate its robustness against these maneuvers.

5.2.2 Image Processing with Feedback Loop from the Navigation Filter

As stated in the introduction part, developing real-time image processing algorithms is very difficult. This thesis made use of the algorithm that has been developed based on the active contour, and this image processor is designed independently from the rest of the system. During the real-time image processing, in order to reduce the processing time, a region of interest is chosen from the entire image plane and the image data is processed only in that region. Since the relative navigation filter estimates and predicts the target states relative to the vehicle, this information could be useful for the image processor in choosing the region of interest. One suggested approach is to feed back the predicted relative position of the target from the estimator to the image processor and to project it onto the current image plane so that the image processor can expect the target to appear around that location. However, this approach can cause a problem in finding the target in the image plane when estimation accuracy is poor. Therefore, the estimation error covariance can also be used to determine if the estimate is reliable. This would improve the selection of the region of interest in the image processing.

5.2.3 Guidance Design for Obstacle Avoidance

In this thesis, the guidance law for obstacle avoidance problem was developed based on the minimum-effort guidance with the assumption of a constant speed in the X-axis. This approach cannot deal with a case in which an obstacle is detected at a short distance. In such a case, the vehicle should slow down the approaching speed and take an avoiding maneuver. Therefore, more generally, the guidance problem for obstacle avoidance can be formulated as a minimum time problem with a control limit constraint. In this approach, the vehicle is able to use its full maneuver capability to avoid obstacles and to reach the waypoint as soon as possible. Therefore, it would be very interesting to derive the guidance law by formulating and solving the minimum time optimization problem for the obstacle avoidance problem.

5.2.4 Experimental Validation of Obstacle Modeling System

The line-based 3-D obstacle modeling system developed in Section 2.4 in this thesis has been integrated with the image segmentation algorithm, and its performance was evaluated in simulations. For the next step, the entire system should be tested with recorded onboard video images of an actual object such as a building. After validation with actual image data, the system should be implemented in the onboard computer and evaluated in flight test. It could also be integrated with the guidance system for obstacle avoidance.

5.2.5 Suboptimal Guidance Design for Real-Time Implementation

This thesis studied the stochastic optimal guidance design for vision-based control applications of UAVs. The OSA suboptimal guidance design has been proposed and its effectiveness was verified. However, it has been shown that the optimal solution is not very well-approximated by the suggested OSA optimization approach via the optimality analysis from the simulation results. Therefore, the further development of a suboptimal guidance design that can achieve the well-approximated optimal solution with small computational load will be needed.

In addition, the suboptimal guidance design proposed in this thesis was validated in simulation using simulated image processor outputs. Immediate future work would be to integrate the guidance design with the real image processor, and evaluate the guidance performance in more realistic simulations and in flight tests.

APPENDIX A

DERIVATION OF MINIMUM-EFFORT GUIDANCE FOR OBSTACLE AVOIDANCE

The minimum-effort guidance (MEG) law for obstacle avoidance is derived in this appendix. The MEG-based guidance law (3.33) is obtained by minimizing the control effort with an interior point position constraint at a given aiming point and with a terminal position constraint at a given waypoint. Since the vehicle is assumed to have a constant speed in the inertial X direction, only its lateral (Y) and vertical (Z) motion needs to be considered in this problem. Therefore, for convenience, we will redefine the position, velocity and acceleration vectors by truncating their fist component in this section. For example,

$$\boldsymbol{X}_v = \begin{bmatrix} Y_v \\ Z_v \end{bmatrix}, \quad \boldsymbol{V}_v = \begin{bmatrix} V_v \\ W_v \end{bmatrix}, \quad \boldsymbol{a}_v = \begin{bmatrix} a_y \\ a_z \end{bmatrix} \tag{A.1}$$

A.1 Optimization Problem

As discussed in Section 3.4, the MEG-based guidance can be obtained by solving the following minimization problem.

$$\min_{\boldsymbol{a}_v} J = \frac{1}{2}\int_{t_k}^{t_f} \boldsymbol{a}_v^T(t)\boldsymbol{a}_v(t)dt \tag{A.2}$$

subject to the linear vehicle dynamics

$$\dot{\boldsymbol{X}}_v = \boldsymbol{V}_v, \quad \dot{\boldsymbol{V}}_v = \boldsymbol{a}_v \tag{A.3}$$

with the following initial, interior and terminal constraints.

$$\begin{cases} \boldsymbol{X}_v(t_k) = \boldsymbol{X}_{v_k}, \quad \boldsymbol{V}_v(t_k) = \boldsymbol{V}_{v_k} \\ \boldsymbol{X}_v(\hat{t}_{go}) = \hat{\boldsymbol{X}}_{ap} \\ \boldsymbol{X}_v(t_f) = \boldsymbol{X}_{wp} \end{cases} \tag{A.4}$$

The terminal time t_f and the interior time $\hat{t}_{go}$ are both fixed.

A.2 Euler-Lagrange Equations

Euler-Lagrange equations for solving the optimization problem given in Section A.1 are formulated in this section. Define an adjoint minimization cost function by

$$\begin{aligned} \bar{J} &= \boldsymbol{\mu}_1^T \left(\boldsymbol{X}_v(\hat{t}_{go}) - \hat{\boldsymbol{X}}_{ap} \right) + \boldsymbol{\mu}_2^T \left(\boldsymbol{X}_v(t_f) - \boldsymbol{X}_{wp} \right) \\ &\quad + \int_{t_k}^{t_f} \left(H - \boldsymbol{\lambda}_1^T \dot{\boldsymbol{X}}_v - \boldsymbol{\lambda}_2^T \dot{\boldsymbol{V}}_v \right)(t)\ dt \end{aligned} \tag{A.5}$$

where $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ are some constants, $\boldsymbol{\lambda}_1$ and $\boldsymbol{\lambda}_2$ are constants called Lagrange multipliers, and $H(t)$ is a Hamiltonian defined by

$$H(t) = \frac{1}{2} \boldsymbol{a}_v^T(t) \boldsymbol{a}_v(t) + \boldsymbol{\lambda}_1^T(t) \boldsymbol{V}_v(t) + \boldsymbol{\lambda}_2^T(t) \boldsymbol{a}_v(t) \tag{A.6}$$

By splitting the integral over a time interval $[t_k,\ t_f]$ in (A.5) into two integrals over $[t_k,\ \hat{t}_{go}^-]$ and over $[\hat{t}_{go}^+,\ t_f]$, $\bar{J}$ can be rewritten as follows.

$$\begin{aligned} \bar{J} &= \boldsymbol{\mu}_1^T \left(\boldsymbol{X}_v(\hat{t}_{go}) - \hat{\boldsymbol{X}}_{ap} \right) + \int_{t_k}^{\hat{t}_{go}^-} \left(H - \boldsymbol{\lambda}_1^T \dot{\boldsymbol{X}}_v - \boldsymbol{\lambda}_2^T \dot{\boldsymbol{V}}_v \right)(t)\ dt \\ &\quad + \boldsymbol{\mu}_2^T \left(\boldsymbol{X}_v(t_f) - \boldsymbol{X}_{wp} \right) + \int_{\hat{t}_{go}^+}^{t_f} \left(H - \boldsymbol{\lambda}_1^T \dot{\boldsymbol{X}}_v - \boldsymbol{\lambda}_2^T \dot{\boldsymbol{V}}_v \right)(t)\ dt \end{aligned} \tag{A.7}$$

Furthermore, it can be expanded by applying an integration by parts.

$$\begin{aligned} \bar{J} &= \boldsymbol{\mu}_1^T \left(\boldsymbol{X}_v(\hat{t}_{go}) - \hat{\boldsymbol{X}}_{ap} \right) + \boldsymbol{\mu}_2^T \left(\boldsymbol{X}_v(t_f) - \boldsymbol{X}_{wp} \right) \\ &\quad - \left[\boldsymbol{\lambda}_1^T \boldsymbol{X}_v + \boldsymbol{\lambda}_2^T \boldsymbol{V}_v \right]_{t_k}^{\hat{t}_{go}^-} + \int_{t_k}^{\hat{t}_{go}^-} \left(H + \dot{\boldsymbol{\lambda}}_1^T \boldsymbol{X}_v + \dot{\boldsymbol{\lambda}}_2^T \boldsymbol{V}_v \right)(t)\ dt \\ &\quad - \left[\boldsymbol{\lambda}_1^T \boldsymbol{X}_v + \boldsymbol{\lambda}_2^T \boldsymbol{V}_v \right]_{\hat{t}_{go}^+}^{t_f} + \int_{\hat{t}_{go}^+}^{t_f} \left(H + \dot{\boldsymbol{\lambda}}_1^T \boldsymbol{X}_v + \dot{\boldsymbol{\lambda}}_2^T \boldsymbol{V}_v \right)(t)\ dt \end{aligned} \tag{A.8}$$

Now consider the first variation of $\bar{J}$ due to variations in the acceleration input $\boldsymbol{a}_v(t)$. Because of the initial condition, the interior position constraint and the terminal position constraint given in (A.4), the corresponding variations $\delta\boldsymbol{X}_v(t_k)$, $\delta\boldsymbol{V}_v(t_k)$, $\delta\boldsymbol{X}_v(\hat{t}_{go})$ and $\delta\boldsymbol{X}_v(t_f)$ become zero. Therefore, the variation of $\bar{J}$ can be written as follows.

$$\begin{aligned}
\delta\bar{J} \;=\;& \left(\boldsymbol{\lambda}_2(\bar{t}_{go}^{+}) - \boldsymbol{\lambda}_2(\bar{t}_{go}^{-})\right)^T \delta\boldsymbol{V}_v(\hat{t}_{go}) + \boldsymbol{\lambda}_2^T(t_f)\delta\boldsymbol{V}_v(t_f) \\
&+ \int_{t_k}^{\hat{t}_{go}^{-}} \left(\frac{\partial H}{\partial \boldsymbol{a}_v}\delta\boldsymbol{a}_v + \left(\frac{\partial H}{\partial \boldsymbol{X}_v} + \dot{\boldsymbol{\lambda}}_1^T\right)\delta\boldsymbol{X}_v + \left(\frac{\partial H}{\partial \boldsymbol{V}_v} + \dot{\boldsymbol{\lambda}}_2^T\right)\delta\boldsymbol{V}_v\right)(t)\;dt \\
&+ \int_{\hat{t}_{go}^{+}}^{t_f} \left(\frac{\partial H}{\partial \boldsymbol{a}_v}\delta\boldsymbol{a}_v + \left(\frac{\partial H}{\partial \boldsymbol{X}_v} + \dot{\boldsymbol{\lambda}}_1^T\right)\delta\boldsymbol{X}_v + \left(\frac{\partial H}{\partial \boldsymbol{V}_v} + \dot{\boldsymbol{\lambda}}_2^T\right)\delta\boldsymbol{V}_v\right)(t)\;dt
\end{aligned}\tag{A.9}$$

As discussed in [13], the optimal solution of (A.2) can be found as $\boldsymbol{a}^*$ which attains $\delta\bar{J} = 0$. Since $\delta\boldsymbol{V}_v(\hat{t}_{go})$, $\delta\boldsymbol{V}_v(t_f)$, $\delta\boldsymbol{a}_v(t)$, $\delta\boldsymbol{X}_v(t)$ and $\delta\boldsymbol{V}_v(t)$ do not need to be zero, their coefficients should be all zero. Euler-Lagrange equations are formulated by doing so. For each of $t_k \le t \le \hat{t}_{go}^{-}$ and $\hat{t}_{go}^{+} \le t \le t_f$,

$$\frac{\partial H}{\partial \boldsymbol{a}_v}(t) = \boldsymbol{a}_v^T(t) + \boldsymbol{\lambda}_2^T(t) = \boldsymbol{0} \tag{A.10}$$

$$\dot{\boldsymbol{\lambda}}_1(t) = -\left(\frac{\partial H}{\partial \boldsymbol{X}_v}\right)^T(t) = \boldsymbol{0} \tag{A.11}$$

$$\dot{\boldsymbol{\lambda}}_2(t) = -\left(\frac{\partial H}{\partial \boldsymbol{V}_v}\right)^T(t) = -\boldsymbol{\lambda}_1(t) \tag{A.12}$$

with the following constraints.

$$\boldsymbol{\lambda}_2(\hat{t}_{go}^{-}) = \boldsymbol{\lambda}_2(\hat{t}_{go}^{+}), \quad \boldsymbol{\lambda}_2(t_f) = \boldsymbol{0} \tag{A.13}$$

In addition, since the vehicle's states should be continuous over time,

$$\boldsymbol{X}_v(\hat{t}_{go}^{-}) = \boldsymbol{X}_v(\hat{t}_{go}^{+}), \quad \boldsymbol{V}_v(\hat{t}_{go}^{-}) = \boldsymbol{V}_v(\hat{t}_{go}^{+}) \tag{A.14}$$

A.3 Optimal Solution

The optimal solution of the problem given in Section A.1 can be obtained by solving the differential equations (A.3) and (A.11,A.12) with the constraints (A.4), (A.13) and (A.14). First, consider the first time interval $t_k \le t \le \hat{t}_{go}^-$. From (A.11,A.12),

$$\dot{\boldsymbol{\lambda}}_1(t) = \mathbf{0} \;\;\Rightarrow\;\; \boldsymbol{\lambda}_1(t) = \boldsymbol{c}_1$$

$$\dot{\boldsymbol{\lambda}}_2(t) = -\boldsymbol{\lambda}_1(t) = -\boldsymbol{c}_1 \;\;\Rightarrow\;\; \boldsymbol{\lambda}_2(t) = -\boldsymbol{c}_1 t + \boldsymbol{d}_1$$

From (A.3),

$$\dot{\boldsymbol{V}}_v(t) = \boldsymbol{a}_v(t) = \boldsymbol{\lambda}_2(t) = -\boldsymbol{c}_1 t + \boldsymbol{d}_1 \;\;\Rightarrow\;\; \boldsymbol{V}_v(t) = -\frac{1}{2}\boldsymbol{c}_1 t^2 + \boldsymbol{d}_1 t + \boldsymbol{e}_1$$

$$\dot{\boldsymbol{X}}_v(t) = \boldsymbol{V}_v(t) = -\frac{1}{2}\boldsymbol{c}_1 t^2 + \boldsymbol{d}_1 t + \boldsymbol{e}_1 \;\;\Rightarrow\;\; \boldsymbol{X}_v(t) = -\frac{1}{6}\boldsymbol{c}_1 t^3 + \frac{1}{2}\boldsymbol{d}_1 t^2 + \boldsymbol{e}_1 t + \boldsymbol{f}_1$$

where $\boldsymbol{c}_1$, $\boldsymbol{d}_1$, $\boldsymbol{e}_1$ and $\boldsymbol{f}_1$ are constant vectors. Similarly, for the second time interval $\hat{t}_{go}^+ \le t \le t_f$, with constant vectors $\boldsymbol{c}_2$, $\boldsymbol{d}_2$, $\boldsymbol{e}_2$ and $\boldsymbol{f}_2$,

$$\boldsymbol{\lambda}_1(t) = \boldsymbol{c}_2 \tag{A.15}$$

$$\boldsymbol{\lambda}_2(t) = -\boldsymbol{c}_2 t + \boldsymbol{d}_2 \tag{A.16}$$

$$\boldsymbol{V}_v(t) = -\frac{1}{2}\boldsymbol{c}_2 t^2 + \boldsymbol{d}_2 t + \boldsymbol{e}_2 \tag{A.17}$$

$$\boldsymbol{X}_v(t) = -\frac{1}{6}\boldsymbol{c}_2 t^3 + \frac{1}{2}\boldsymbol{d}_2 t^2 + \boldsymbol{e}_2 t + \boldsymbol{f}_2 \tag{A.18}$$

There are sixteen unknown constants in the equations (A.15-A.18). They can be solved by applying the sixteen constraints (A.4,A.13,A.14). From the terminal conditions,

$$\begin{aligned}
\boldsymbol{\lambda}_2(t_f) &= -\boldsymbol{c}_2 t_f + \boldsymbol{d}_2 = \mathbf{0} \quad\Rightarrow\quad \boldsymbol{d}_2 = \boldsymbol{c}_2 t_f \\
\boldsymbol{X}_v(t_f) &= -\frac{1}{6}\boldsymbol{c}_2 t_f^3 + \frac{1}{2}\boldsymbol{c}_2 t_f^3 + \boldsymbol{e}_2 t_f + \boldsymbol{f}_2 \\
&= \frac{1}{3}\boldsymbol{c}_2 t_f^3 + \boldsymbol{e}_2 t_f + \boldsymbol{f}_2 = \boldsymbol{X}_{wp} \quad\Rightarrow\quad \boldsymbol{f}_2 = \boldsymbol{X}_{wp} - \frac{1}{3}\boldsymbol{c}_2 t_f^3 - \boldsymbol{e}_2 t_f
\end{aligned}$$

From the interior point constraint,

$$\begin{aligned}
\boldsymbol{X}_v(\hat{t}_{go}^+) &= -\frac{1}{6}\boldsymbol{c}_2 \hat{t}_{go}^3 + \frac{1}{2}\boldsymbol{c}_2 t_f \hat{t}_{go}^2 + \boldsymbol{e}_2 \hat{t}_{go} - \frac{1}{3}\boldsymbol{c}_2 t_f^3 - \boldsymbol{e}_2 t_f + \boldsymbol{X}_{wp} \\
&= -\frac{1}{6}\boldsymbol{c}_2\left(2t_f^3 - 3t_f\hat{t}_{go}^2 + \hat{t}_{go}^3\right) - \boldsymbol{e}_2\left(t_f - \hat{t}_{go}\right) + \boldsymbol{X}_{wp} = \hat{\boldsymbol{X}}_{ap} \\
\Rightarrow \qquad \boldsymbol{e}_2 &= \frac{\boldsymbol{X}_{wp} - \hat{\boldsymbol{X}}_{ap}}{t_f - \hat{t}_{go}} - \frac{1}{6}\boldsymbol{c}_2 \frac{2t_f^3 - 3t_f\hat{t}_{go}^2 + \hat{t}_{go}^3}{t_f - \hat{t}_{go}}
\end{aligned}$$

Then $\boldsymbol{\lambda}_2(\hat{t}_{go}^+)$ and $\boldsymbol{V}_v(\hat{t}_{go}^+)$ result in

$$\begin{aligned}
\boldsymbol{\lambda}_2(\hat{t}_{go}^+) &= -\boldsymbol{c}_2\hat{t}_{go} + \boldsymbol{c}_2 t_f = \boldsymbol{c}_2\left(t_f - \hat{t}_{go}\right) && \text{(A.19)}\\
\boldsymbol{V}_v(\hat{t}_{go}^+) &= -\frac{1}{2}\boldsymbol{c}_2\hat{t}_{go}^2 + \boldsymbol{c}_2 t_f\hat{t}_{go} - \frac{1}{6}\boldsymbol{c}_2\frac{2t_f^3 - 3t_f\hat{t}_{go}^2 + \hat{t}_{go}^3}{t_f - \hat{t}_{go}} + \frac{\boldsymbol{X}_{wp} - \hat{\boldsymbol{X}}_{ap}}{t_f - \hat{t}_{go}} &&\\
&= -\frac{1}{3}\boldsymbol{c}_2\left(t_f - \hat{t}_{go}\right)^2 + \frac{\boldsymbol{X}_{wp} - \hat{\boldsymbol{X}}_{ap}}{t_f - \hat{t}_{go}} && \text{(A.20)}
\end{aligned}$$

From the initial conditions,

$$\begin{aligned}
\boldsymbol{V}_v(t_k) &= -\frac{1}{2}\boldsymbol{c}_1 t_k^2 + \boldsymbol{d}_1 t_k + \boldsymbol{e}_1 = \boldsymbol{V}_{v_k} \quad \Rightarrow \quad \boldsymbol{e}_1 = \boldsymbol{V}_{v_k} + \frac{1}{2}\boldsymbol{c}_1 t_k^2 - \boldsymbol{d}_1 t_k\\
\boldsymbol{X}_v(t_k) &= -\frac{1}{6}\boldsymbol{c}_1 t_k^3 + \frac{1}{2}\boldsymbol{d}_1 t_k^2 + \left(\boldsymbol{V}_{v_k} + \frac{1}{2}\boldsymbol{c}_1 t_k^2 - \boldsymbol{d}_1 t_k\right) t_k + \boldsymbol{f}_1\\
&= \frac{1}{3}\boldsymbol{c}_2 t_k^3 - \frac{1}{2}\boldsymbol{d}_1 t_k^2 + \boldsymbol{f}_1 + \boldsymbol{V}_{v_k} t_k = \boldsymbol{X}_{v_k} \quad \Rightarrow \quad \boldsymbol{f}_1 = \boldsymbol{X}_{v_k} - \boldsymbol{V}_{v_k} t_k - \frac{1}{3}\boldsymbol{c}_1 t_k^3 + \frac{1}{2}\boldsymbol{d}_1 t_k^2
\end{aligned}$$

From the interior point constraint,

$$\begin{aligned}
\boldsymbol{X}_v(\hat{t}_{go}^-) &= -\frac{1}{6}\boldsymbol{c}_1\hat{t}_{go}^3 + \frac{1}{2}\boldsymbol{d}_1\hat{t}_{go}^2 + \left(\boldsymbol{V}_{v_k} + \frac{1}{2}\boldsymbol{c}_1 t_k^2 - \boldsymbol{d}_1 t_k\right)\hat{t}_{go} + \boldsymbol{X}_{v_k} - \boldsymbol{V}_{v_k} t_k - \frac{1}{3}\boldsymbol{c}_1 t_k^3 + \frac{1}{2}\boldsymbol{d}_1 t_k^2\\
&= -\frac{1}{6}\boldsymbol{c}_1\left(\hat{t}_{go}^3 - 3\hat{t}_{go}t_k^2 + 2t_k^3\right) + \frac{1}{2}\boldsymbol{d}_1\left(\hat{t}_{go} - t_k\right)^2 + \boldsymbol{V}_{v_k}\left(\hat{t}_{go} - t_k\right) + \boldsymbol{X}_{v_k} = \hat{\boldsymbol{X}}_{ap}\\
\Rightarrow \qquad & \boldsymbol{d}_1 = \frac{1}{3}\boldsymbol{c}_1\frac{\hat{t}_{go}^3 - 3\hat{t}_{go}t_k^2 + 2t_k^3}{\left(\hat{t}_{go} - t_k\right)^2} + 2\frac{\hat{\boldsymbol{X}}_{ap} - \boldsymbol{X}_{v_k}}{\left(\hat{t}_{go} - t_k\right)^2} - 2\frac{\boldsymbol{V}_{v_k}}{\hat{t}_{go} - t_k}
\end{aligned}$$

Then $\boldsymbol{\lambda}_2(\hat{t}_{go}^-)$ and $\boldsymbol{V}_v(\hat{t}_{go}^-)$ can be written as functions of $\boldsymbol{c}_1$ as follows.

$$\begin{aligned}
\boldsymbol{\lambda}_2(\hat{t}_{go}^-) &= -\boldsymbol{c}_1\hat{t}_{go} + \frac{1}{3}\boldsymbol{c}_1\frac{\hat{t}_{go}^3 - 3\hat{t}_{go}t_k^2 + 2t_k^3}{\left(\hat{t}_{go} - t_k\right)^2} + 2\frac{\hat{\boldsymbol{X}}_{ap} - \boldsymbol{X}_{v_k}}{\left(\hat{t}_{go} - t_k\right)^2} - 2\frac{\boldsymbol{V}_{v_k}}{\hat{t}_{go} - t_k} &&\\
&= -\frac{2}{3}\boldsymbol{c}_1\left(\hat{t}_{go} - t_k\right) + 2\frac{\hat{\boldsymbol{X}}_{ap} - \boldsymbol{X}_{v_k}}{\left(\hat{t}_{go} - t_k\right)^2} - 2\frac{\boldsymbol{V}_{v_k}}{\hat{t}_{go} - t_k} && \text{(A.21)}\\
\boldsymbol{V}_v(\hat{t}_{go}^-) &= -\frac{1}{2}\boldsymbol{c}_1\hat{t}_{go}^2 + \boldsymbol{d}_1\left(\hat{t}_{go} - t_k\right) + \boldsymbol{V}_{v_k} + \frac{1}{2}\boldsymbol{c}_1 t_k^2 &&\\
&= -\frac{1}{2}\boldsymbol{c}_1\left(\hat{t}_{go}^2 - t_k^2\right) + \frac{1}{3}\boldsymbol{c}_1\frac{\hat{t}_{go}^3 - 3\hat{t}_{go}t_k^2 + 2t_k^3}{\hat{t}_{go} - t_k} + 2\frac{\hat{\boldsymbol{X}}_{ap} - \boldsymbol{X}_{v_k}}{\hat{t}_{go} - t_k} - 2\boldsymbol{V}_{v_k} + \boldsymbol{V}_{v_k} &&\\
&= -\frac{1}{6}\boldsymbol{c}_1\left(\hat{t}_{go} - t_k\right)^2 + 2\frac{\hat{\boldsymbol{X}}_{ap} - \boldsymbol{X}_{v_k}}{\hat{t}_{go} - t_k} - \boldsymbol{V}_{v_k} && \text{(A.22)}
\end{aligned}$$

From the continuity between (A.19,A.20) and (A.21,A.22), the remaining constant vectors $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ can be derived. From $\boldsymbol{\lambda}_2(\hat{t}_{go}^+) = \boldsymbol{\lambda}_2(\hat{t}_{go}^-)$,

$$\boldsymbol{c}_2\left(t_f - \hat{t}_{go}\right) = -\frac{2}{3}\boldsymbol{c}_1\left(\hat{t}_{go} - t_k\right) + 2\frac{\hat{\boldsymbol{X}}_{ap} - \boldsymbol{X}_{v_k}}{\left(\hat{t}_{go} - t_k\right)^2} - 2\frac{\boldsymbol{V}_{v_k}}{\hat{t}_{go} - t_k} \tag{A.23}$$

By substituting (A.23) into (A.20),

$$\begin{aligned}\boldsymbol{V}_v(\hat{t}_{go}^+) &= \frac{2}{9}\left(\boldsymbol{c}_1\left(\hat{t}_{go}-t_k\right)-3\frac{\hat{\boldsymbol{X}}_{ap}-\boldsymbol{X}_{v_k}}{\left(\hat{t}_{go}-t_k\right)^2}+3\frac{\boldsymbol{V}_{v_k}}{\hat{t}_{go}-t_k}\right)\left(t_f-\hat{t}_{go}\right)+\frac{\boldsymbol{X}_{wp}-\hat{\boldsymbol{X}}_{ap}}{t_f-\hat{t}_{go}} \\ &= \frac{2}{9}\left(\boldsymbol{c}_1\left(\hat{t}_{go}-t_k\right)-\hat{\boldsymbol{a}}_{v_k}^{PN}\right)\left(t_f-\hat{t}_{go}\right)+\frac{\boldsymbol{X}_{wp}-\hat{\boldsymbol{X}}_{ap}}{t_f-\hat{t}_{go}} \qquad \text{(A.24)}\end{aligned}$$

where $\hat{\boldsymbol{a}}_{v_k}^{PN}$ is the proportional guidance input derived in (3.29). Since this should coincide with (A.22),

$$\begin{aligned}&\frac{1}{18}\boldsymbol{c}_1\left(\hat{t}_{go}-t_k\right)\left(4\left(t_f-\hat{t}_{go}\right)+3\left(\hat{t}_{go}-t_k\right)\right) \\ = \quad & 2\frac{\hat{\boldsymbol{X}}_{ap}-\boldsymbol{X}_{v_k}}{\hat{t}_{go}-t_k}-\boldsymbol{V}_{v_k}+\frac{2}{9}\hat{\boldsymbol{a}}_{v_k}^{PN}\left(t_f-\hat{t}_{go}\right)-\frac{\boldsymbol{X}_{wp}-\hat{\boldsymbol{X}}_{ap}}{t_f-\hat{t}_{go}}\end{aligned}$$

Finally, the MEG-based guidance input at the current time step t_k can be obtained as follows.

$$\begin{aligned}\hat{\boldsymbol{a}}_{v_k}^{MEG} &= \boldsymbol{\lambda}_2(t_k)=-\boldsymbol{c}_1 t_k+\frac{1}{3}\boldsymbol{c}_1\frac{\hat{t}_{go}^3-3\hat{t}_{go}t_k^2+2t_k^3}{\left(\hat{t}_{go}-t_k\right)^2}+2\frac{\hat{\boldsymbol{X}}_{ap}-\boldsymbol{X}_{v_k}}{\left(\hat{t}_{go}-t_k\right)^2}-2\frac{\boldsymbol{V}_{v_k}}{\hat{t}_{go}-t_k} \\ &= \frac{1}{3}\boldsymbol{c}_1\left(\hat{t}_{go}-t_k\right)+\frac{2}{3}\hat{\boldsymbol{a}}_{v_k}^{PN} \\ &= \frac{6}{4\left(t_f-\hat{t}_{go}\right)+3\left(\hat{t}_{go}-t_k\right)}\left(2\frac{\hat{\boldsymbol{X}}_{ap}-\boldsymbol{X}_{v_k}}{\hat{t}_{go}-t_k}-\boldsymbol{V}_{v_k}-\frac{\boldsymbol{X}_{wp}-\hat{\boldsymbol{X}}_{ap}}{t_f-\hat{t}_{go}}\right) \\ &\quad +\frac{2}{3}\hat{\boldsymbol{a}}_{v_k}^{PN}\left(\frac{2\left(t_f-\hat{t}_{go}\right)}{4\left(t_f-\hat{t}_{go}\right)+3\left(\hat{t}_{go}-t_k\right)}+1\right) \\ &= \hat{\boldsymbol{a}}_{v_k}^{PN}+\frac{3}{4\left(t_f-\hat{t}_{go}\right)+3\left(\hat{t}_{go}-t_k\right)}\left(3\frac{\hat{\boldsymbol{X}}_{ap}-\boldsymbol{X}_{v_k}}{\hat{t}_{go}-t_k}-2\frac{\boldsymbol{X}_{wp}-\hat{\boldsymbol{X}}_{ap}}{t_f-\hat{t}_{go}}-\boldsymbol{V}_{v_k}\right) \\ &= \hat{\boldsymbol{a}}_{v_k}^{PN}+\Delta\boldsymbol{a}_{v_k}^{MEG} \qquad \text{(A.25)}\end{aligned}$$

where $\Delta\boldsymbol{a}_{v_k}^{MEG}$ is the same as defined in (3.33).

APPENDIX B

DERIVATION OF LINEAR QUADRATIC OPTIMAL GUIDANCE

The optimal guidance law (4.4,4.5) which minimizes a quadratic cost is derived by using the Hamilton-Jacobi-Bellman equation in this appendix.

B.1 Optimization Problem

Recall the quadratic minimization problem given in Section 4.1 here. The optimal guidance law at a time step t_k is derived by solving

$$\begin{aligned}\min_{\boldsymbol{a}_v} J &= \frac{1}{2}\left(\boldsymbol{x}(t_f)-\boldsymbol{x}_f\right)^T S_f\left(\boldsymbol{x}(t_f)-\boldsymbol{x}_f\right) \\ &+\frac{1}{2}\int_{t_k}^{t_f}\Big(\left(\boldsymbol{x}-\boldsymbol{x}_c\right)^T A\left(\boldsymbol{x}-\boldsymbol{x}_c\right)+\boldsymbol{a}_v^T B\boldsymbol{a}_v\Big)(t)\;dt \qquad \text{(B.1)}\end{aligned}$$

subject to the linear relative dynamics $\dot{\boldsymbol{x}} = F\boldsymbol{x} + G\boldsymbol{a}_v$ with the initial condition $\boldsymbol{x}(t_k) = \boldsymbol{x}_k$.

B.2 Hamilton-Jacobi-Bellman Equation

Hamilton-Jacobi-Bellman equation is formulated based on dynamic programming theory or Bellman's principle of optimality. Suppose that a path *A*-*C* is an optimal path from Point *A* to Point *C* and Point *B* is an interior point on the path. Then the path *B*-*C* is the optimal path from Point *B* to Point *C*.

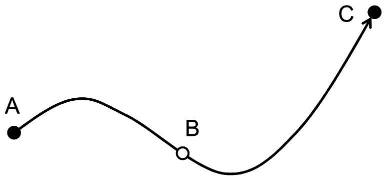


Figure 54: Bellman's Principle of Optimality

Define an optimal return function $J^0(\boldsymbol{x}, t)$ by the optimal cost of (B.1) as a function of the initial state $\boldsymbol{x}$ and the initial time t[13]. According to Bellman's principle of optimality, for small $\Delta t > 0$,

$$\begin{aligned} J^0(\boldsymbol{x}, t) &= J^0\left(\boldsymbol{x} + (F\boldsymbol{x} + G\boldsymbol{a}_v^*(t))\Delta t, t + \Delta t\right) \\ &\quad + \frac{1}{2}\left((\boldsymbol{x} - \boldsymbol{x}_c(t))^T A\,(\boldsymbol{x} - \boldsymbol{x}_c(t)) + \boldsymbol{a}_v^{*T}(t) B \boldsymbol{a}_v^*(t)\right)\Delta t \\ &= J^0(\boldsymbol{x}, t) + \frac{\partial J^0}{\partial \boldsymbol{x}}(F\boldsymbol{x} + G\boldsymbol{a}_v^*(t))\Delta t + \frac{\partial J^0}{\partial t}\Delta t \\ &\quad + \frac{1}{2}\left((\boldsymbol{x} - \boldsymbol{x}_c(t))^T A\,(\boldsymbol{x} - \boldsymbol{x}_c(t)) + \boldsymbol{a}_v^{*T}(t) B \boldsymbol{a}_v^*(t)\right)\Delta t \end{aligned} \tag{B.2}$$

where $\boldsymbol{a}_v^*$ is the optimal guidance input. It leads the following relationship.

$$\frac{\partial J^0}{\partial \boldsymbol{x}}(F\boldsymbol{x} + G\boldsymbol{a}_v^*(t)) + \frac{\partial J^0}{\partial t} + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}_c(t))^T A\,(\boldsymbol{x} - \boldsymbol{x}_c(t)) + \frac{1}{2}\boldsymbol{a}_v^{*T}(t) B \boldsymbol{a}_v^*(t) = 0 \tag{B.3}$$

When applying a non-optimal guidance $\boldsymbol{a}_v = \boldsymbol{a}_v^* + \Delta\boldsymbol{a}_v$ for $[t,\ t + \Delta t]$ and applying an optimal guidance for $[t + \Delta t,\ t_f]$, the resulting cost will become

$$\begin{aligned} J^1(\boldsymbol{x}, t) &= J^0\left(\boldsymbol{x} + (F\boldsymbol{x} + G\boldsymbol{a}_v(t))\Delta t, t + \Delta t\right) \\ &\quad + \frac{1}{2}\left((\boldsymbol{x} - \boldsymbol{x}_c(t))^T A\,(\boldsymbol{x} - \boldsymbol{x}_c(t)) + \boldsymbol{a}_v^T(t) B \boldsymbol{a}_v(t)\right)\Delta t \\ &= J^0(\boldsymbol{x}, t) + \frac{\partial J^0}{\partial \boldsymbol{x}}(F\boldsymbol{x} + G\left(\boldsymbol{a}_v^*(t) + \Delta\boldsymbol{a}_v\right))\Delta t + \frac{\partial J^0}{\partial t}\Delta t \\ &\quad + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}_c(t))^T A\,(\boldsymbol{x} - \boldsymbol{x}_c(t))\Delta t \\ &\quad + \frac{1}{2}\left(\boldsymbol{a}_v^*(t) + \Delta\boldsymbol{a}_v\right)^T B \left(\boldsymbol{a}_v^*(t) + \Delta\boldsymbol{a}_v\right)\Delta t \end{aligned} \tag{B.4}$$

By substituting (B.3) into (B.4),

$$J^1(\boldsymbol{x}, t) = J^0(\boldsymbol{x}, t) + \left(\frac{\partial J^0}{\partial \boldsymbol{x}}G + \boldsymbol{a}_v^{*T}(t) B\right)\Delta\boldsymbol{a}_v\Delta t + \frac{1}{2}\Delta\boldsymbol{a}_v^T B \Delta\boldsymbol{a}_v \Delta t \tag{B.5}$$

Since $J^0(\boldsymbol{x}, t)$ is the optimal cost, $J^1(\boldsymbol{x}, t) > J^0(\boldsymbol{x}, t)$ should be satisfied. For $\Delta t > 0$,

$$\left(\frac{\partial J^0}{\partial \boldsymbol{x}}G + \boldsymbol{a}_v^{*T}(t) B\right)\Delta\boldsymbol{a}_v + \frac{1}{2}\Delta\boldsymbol{a}_v^T B \Delta\boldsymbol{a}_v > 0 \tag{B.6}$$

The last term on the left hand side is always positive for non-zero $\Delta\boldsymbol{a}_v$ due to the positive definite weight matrix B. In order to ensure that the first two terms are non-negative for any $\Delta \boldsymbol{a}_v$, a sufficient condition is

$$\frac{\partial J^0}{\partial \boldsymbol{x}} G + \boldsymbol{a}_v^{*T}(t) B = \boldsymbol{0} \quad \Rightarrow \quad \boldsymbol{a}_v^*(t) = -B^{-1} G^T \left(\frac{\partial J^0}{\partial \boldsymbol{x}} \right)^T \tag{B.7}$$

Now this $\boldsymbol{a}_v^*(t)$ can be substituted back into (B.3).

$$\frac{\partial J^0}{\partial \boldsymbol{x}} F \boldsymbol{x} - \frac{1}{2} \left(\frac{\partial J^0}{\partial \boldsymbol{x}} \right) G B^{-1} G^T \left(\frac{\partial J^0}{\partial \boldsymbol{x}} \right)^T + \frac{\partial J^0}{\partial t} + \frac{1}{2} \left(\boldsymbol{x} - \boldsymbol{x}_c(t) \right)^T A \left(\boldsymbol{x} - \boldsymbol{x}_c(t) \right) = 0 \tag{B.8}$$

Assume the following quadratic form for the optimal return function $J^0(\boldsymbol{x}, t)$.

$$J^0(\boldsymbol{x}, t) = \frac{1}{2} \boldsymbol{x}^T S(t) \boldsymbol{x} + \boldsymbol{v}^T(t) \boldsymbol{x} + w(t) \tag{B.9}$$

where $S(t)$, $\boldsymbol{v}(t)$ and $w(t)$ are coefficient matrix, vector and scalar functions. Without loss of generality, $S(t)$ can be taken as a symmetric matrix. Then,

$$\frac{\partial J^0}{\partial \boldsymbol{x}} = \boldsymbol{x}^T S(t) + \boldsymbol{v}^T(t), \quad \frac{\partial J^0}{\partial t} = \frac{1}{2} \boldsymbol{x}^T \dot{S}(t) \boldsymbol{x} + \dot{\boldsymbol{v}}^T(t) \boldsymbol{x} + \dot{w}(t) \tag{B.10}$$

Substitute these into (B.8) and obtain

$$\begin{aligned} 0 \;=\; & \frac{1}{2} \boldsymbol{x}^T \left(\dot{S}(t) + S(t) F + F^T S(t) + A - S(t) G B^{-1} G^T S(t) \right) \boldsymbol{x} \\ & + \left(\dot{\boldsymbol{v}}(t) + \left(F^T - S(t) G B^{-1} G^T \right) \boldsymbol{v}(t) - A \boldsymbol{x}_c(t) \right)^T \boldsymbol{x} \\ & + \left(\dot{w}(t) - \frac{1}{2} \boldsymbol{v}^T(t) G B^{-1} G^T \boldsymbol{v}(t) + \frac{1}{2} \boldsymbol{x}_c^T(t) A \boldsymbol{x}_c(t) \right) \end{aligned} \tag{B.11}$$

Therefore, the coefficients $S(t)$, $\boldsymbol{v}(t)$ and $w(t)$ must satisfy the following first-order differential equations.

$$\dot{S}(t) + S(t) F + F^T S(t) + A - S(t) G B^{-1} G^T S(t) = O \tag{B.12}$$

$$\dot{\boldsymbol{v}}(t) + \left(F^T - S(t) G B^{-1} G^T \right) \boldsymbol{v}(t) - A \boldsymbol{x}_c(t) = \boldsymbol{0} \tag{B.13}$$

$$\dot{w}(t) - \frac{1}{2} \boldsymbol{v}^T(t) G B^{-1} G^T \boldsymbol{v}(t) + \frac{1}{2} \boldsymbol{x}_c^T(t) A \boldsymbol{x}_c(t) = 0 \tag{B.14}$$

Since a terminal condition of the optimal return function is given by

$$\begin{aligned} J_0(\boldsymbol{x}, t_f) &= \frac{1}{2} \boldsymbol{x}^T S(t_f) \boldsymbol{x} + \boldsymbol{v}^T(t_f) \boldsymbol{x} + w(t_f) \\ &= \frac{1}{2} \left(\boldsymbol{x} - \boldsymbol{x}_f \right)^T S_f \left(\boldsymbol{x} - \boldsymbol{x}_f \right) = \frac{1}{2} \boldsymbol{x}^T S_f \boldsymbol{x} - \boldsymbol{x}_f^T S_f \boldsymbol{x} + \frac{1}{2} \boldsymbol{x}_f^T S_f \boldsymbol{x}_f \end{aligned} \tag{B.15}$$

the terminal conditions on $S(t)$, $\boldsymbol{v}(t)$ and $w(t)$ are obtained as follows.

$$S(t_f) = S_f, \quad \boldsymbol{v}(t_f) = -S_f\boldsymbol{x}_f, \quad w(t_f) = \frac{1}{2}\boldsymbol{x}_f^T S_f \boldsymbol{x}_f \tag{B.16}$$

Then, the optimal guidance input at the current time t_k becomes

$$\boldsymbol{a}_v^*(t_k) = -B^{-1}G^T \left(\frac{\partial J^0}{\partial \boldsymbol{x}}\right)\Big|_{(\boldsymbol{x}(t_k),t_k)}^T = -B^{-1}G^T\left(S(t_k)\boldsymbol{x}(t_k) + \boldsymbol{v}(t_k)\right) \tag{B.17}$$

and the optimal cost is

$$J^*(t_k) = J_0(\boldsymbol{x}, t_k) = \frac{1}{2}\boldsymbol{x}^T(t_k)S(t_k)\boldsymbol{x}(t_k) + \boldsymbol{v}^T(t_k)\boldsymbol{x}(t_k) + w(t_k) \tag{B.18}$$

B.3 Optimal Solution for Terminal Tracking

Consider a special case of terminal tracking, which is the case of $A = O$. Define a vector function $\boldsymbol{u}(t) = S(t)\boldsymbol{x}(t) + \boldsymbol{v}(t)$. Then the optimal guidance law can be written by using $\boldsymbol{u}$ as follows.

$$\boldsymbol{a}_v^*(t) = -B^{-1}G^T\boldsymbol{u}(t) \tag{B.19}$$

A differential equation associated with $\boldsymbol{u}$ will be

$$\begin{aligned}
\dot{\boldsymbol{u}}(t) &= \dot{S}(t)\boldsymbol{x}(t) + S(t)\dot{\boldsymbol{x}}(t) + \dot{\boldsymbol{v}}(t) = \dot{S}(t)\boldsymbol{x}(t) + S(t)\left(F\boldsymbol{x}(t) + G\boldsymbol{a}_v^*(t)\right) + \dot{\boldsymbol{v}}(t) \\
&= \left(\dot{S}(t) + S(t)F\right)\boldsymbol{x}(t) - S(t)GB^{-1}G^T\boldsymbol{u}(t) + \dot{\boldsymbol{v}}(t) \\
&= -\left(F^T - S(t)GB^{-1}G^T\right)\left(S(t)\boldsymbol{x}(t) + \boldsymbol{v}(t)\right) - S(t)GB^{-1}G^T\boldsymbol{u}(t) \\
&= -F^T\boldsymbol{u}(t) \qquad \text{(B.20)}
\end{aligned}$$

with the terminal condition

$$\boldsymbol{u}(t_f) = S(t_f)\boldsymbol{x}(t_f) + \boldsymbol{v}(t_f) = S_f\boldsymbol{x}(t_f) - S_f\boldsymbol{x}_f = S_f\left(\boldsymbol{x}(t_f) - \boldsymbol{x}_f\right) \tag{B.21}$$

Therefore, the vector $\boldsymbol{u}(t)$ and the optimal solution can be solved as

$$\begin{aligned}
\boldsymbol{u}(t) &= e^{F^T(t_f-t)}\boldsymbol{u}(t_f) = e^{F^T(t_f-t)}S_f\left(\boldsymbol{x}(t_f) - \boldsymbol{x}_f\right) \qquad \text{(B.22)} \\
\therefore \quad \boldsymbol{a}_v^*(t) &= -B^{-1}G^Te^{F^T(t_f-t)}S_f\left(\boldsymbol{x}(t_f) - \boldsymbol{x}_f\right) \qquad \text{(B.23)}
\end{aligned}$$

From the linear relative dynamics,

$$\dot{\boldsymbol{x}}(t) = F\boldsymbol{x}(t) + G\boldsymbol{a}_v^*(t) = F\boldsymbol{x}(t) - GB^{-1}G^Te^{F^T(t_f-t)}S_f\left(\boldsymbol{x}(t_f) - \boldsymbol{x}_f\right) \tag{B.24}$$

This differential equation can be solved with a given initial state.

$$\boldsymbol{x}(t) = e^{F(t-t_k)}\boldsymbol{x}(t_k) - \int_{t_k}^{t} e^{F(t-s)}GB^{-1}G^Te^{F^T(t_f-s)}S_f\left(\boldsymbol{x}(t_f) - \boldsymbol{x}_f\right)ds \tag{B.25}$$

Then the term $(\boldsymbol{x}(t_f) - \boldsymbol{x}_f)$ is derived as

$$\begin{aligned}
\boldsymbol{x}(t_f) &= e^{F(t_f-t_k)}\boldsymbol{x}(t_k) - \int_{t_k}^{t_f} e^{F(t_f-s)}GB^{-1}G^Te^{F^T(t_f-s)}dsS_f\left(\boldsymbol{x}(t_f) - \boldsymbol{x}_f\right)\\
&= e^{F(t_f-t_k)}\boldsymbol{x}(t_k) - \mathcal{G}_kS_f\left(\boldsymbol{x}(t_f) - \boldsymbol{x}_f\right)\\
\Rightarrow &\quad \left(I + \mathcal{G}_kS_f\right)\left(\boldsymbol{x}(t_f) - \boldsymbol{x}_f\right) = e^{F(t_f-t_k)}\boldsymbol{x}(t_k) - \boldsymbol{x}_f\\
\Rightarrow &\quad \boldsymbol{x}(t_f) - \boldsymbol{x}_f = \left(I + \mathcal{G}_kS_f\right)^{-1}\left(e^{F(t_f-t_k)}\boldsymbol{x}(t_k) - \boldsymbol{x}_f\right)
\end{aligned} \tag{B.26}$$

where

$$\mathcal{G}_k = \int_{t_k}^{t_f} e^{F(t_f-s)}GB^{-1}G^Te^{F^T(t_f-s)}ds$$

By substituting this into (B.23), the optimal solution can be obtained as follows.

$$\boldsymbol{a}_v^*(t) = -B^{-1}G^Te^{F^T(t_f-t)}S_f\left(I + \mathcal{G}_kS_f\right)^{-1}\left(e^{F(t_f-t_k)}\boldsymbol{x}(t_k) - \boldsymbol{x}_f\right) \tag{B.27}$$

The optimal cost results in

$$\begin{aligned}
J^*(t_k) &= \frac{1}{2}\left(\boldsymbol{x}(t_f) - \boldsymbol{x}_f\right)^TS_f\left(\boldsymbol{x}(t_f) - \boldsymbol{x}_f\right) + \frac{1}{2}\int_{t_k}^{t_f}\boldsymbol{a}_v^{*T}(t)B\boldsymbol{a}_v^*(t)dt\\
&= \frac{1}{2}\left(e^{F(t_f-t_k)}\boldsymbol{x}(t_k) - \boldsymbol{x}_f\right)^T\left(I + \mathcal{G}_kS_f\right)^{-T}S_f\left(I + \mathcal{G}_kS_f\right)^{-1}\left(e^{F(t_f-t_k)}\boldsymbol{x}(t_k) - \boldsymbol{x}_f\right)\\
&\quad + \frac{1}{2}\left(e^{F(t_f-t_k)}\boldsymbol{x}(t_k) - \boldsymbol{x}_f\right)^T\left(I + \mathcal{G}_kS_f\right)^{-T}S_f\mathcal{G}_kS_f\left(I + \mathcal{G}_kS_f\right)^{-1}\left(e^{F(t_f-t_k)}\boldsymbol{x}(t_k) - \boldsymbol{x}_f\right)\\
&= \frac{1}{2}\left(e^{F(t_f-t_k)}\boldsymbol{x}(t_k) - \boldsymbol{x}_f\right)^T\left(I + \mathcal{G}_kS_f\right)^{-T}S_f\left(e^{F(t_f-t_k)}\boldsymbol{x}(t_k) - \boldsymbol{x}_f\right)
\end{aligned} \tag{B.28}$$

REFERENCES

[1] Aidala, V. and Hammel, S., "Utilization of Modified Polar Coordinates for Bearings-Only Tracking," *IEEE Transactions on Automatic Control*, vol. 28, no. 3, 1983.

[2] Belkhous, S., Azzouz, A., Nerquizian, C., Saad, M., and Nerquizian, V., "Trajectory Optimization in Both Static and Dynamic Environments," *IEEE International Conference on Industrial Technology*, 2004.

[3] Ben-Asher, J., "Minimum-Effort Interception of Multiple Targets," *AIAA Journal of Guidance, Control and Dynamics*, vol. 16, no. 3, 1993.

[4] Ben-Asher, J. and Yeash, I., *Advances in Missile Guidance Theory.* AIAA, 1998.

[5] Bernstein, D. and Haddad, W., *Control-System Synthesis: The Fixed-Structure Approach.* Georgia Institute of Technology, 1995.

[6] Betts, J., "Survey of Numerical Methods for Trajectory Optimization," *AIAA Journal of Guidance, Control and Dynamics*, vol. 21, no. 2, 1998.

[7] Blackmore, L., "A Probabilistic Particle Control Approach to Optimal, Robust Predictive Control," *AIAA Guidance, Navigation and Control Conference*, 2006.

[8] Blake, A. and Isard, M., *Active Contours.* Springer, 1998.

[9] Bone, E. and Bolkcom, C., "Unmanned Aerial Vehicles: Background and Issues for Congress," Tech. Rep. RL31872, Congressional Research Service, The Library of Congress, 2003.

[10] Brinker, J. and Wise, K., "Flight Testing of a Reconfigurable Flight Control Law on the X-36 Tailless Fighter Aircraft," *AIAA Guidance, Navigation and Control Conference*, 2000.

[11] B.Ristic, S.Arulampalam, and N.Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications.* Artech House Radar Library, 2004.

[12] Brown, R. and Hwang, P., *Introduction to Random Signals and Applied Kalman Filtering.* John Wiley & Sons, 1997.

[13] Bryson, E. and Ho, Y., *Applied Optimal Control.* Taylor & Francis, 1975.

[14] Calise, A., "Enforcing an Algebraic Constraint in Extended Kalman Filter," *AIAA Guidance, Navigation and Control Conference*, 2007.

[15] Chakravarthy, A. and Ghose, D., "Obstacle Avoidance in a Dynamic Environment: A Collision Cone Approach," *IEEE Transactions on Systems, Man and Cybernetics-Part A: Systems and Humans*, vol. 28, no. 5, 1998.

[16] Clough, B., "Unmanned Aerial Vehicles: Autonomous Control Challenges, a Researcher's Prospective," *AIAA Journal of Aerospace Computing, Information and Communication*, vol. 2, no. 8, 2005.

[17] Davidson, A., "Real-Time Simultaneous Localization and Mapping with a Single Vision Camera," *IEEE International Conference on Computer Vision*, 2003.

[18] de Wagter, C. and Mulder, J., "Towards Vision-Based UAV Situation Awareness," *AIAA Guidance, Navigation and Control Conference*, 2005.

[19] Fabiani, P., A.Piquereau, V.Fuertes, and Basset, P., "The ReSSAC Autonomous Rotorcraft: from autonomy demonstrations to out-of-sight flights," *The AHS Specialists' Meeting on Unmanned Rotorcraft*, 2005.

[20] Fabiani, P., V.Fuertes, A.Piquereau, Mampey, R., and Teichteil-Konigsbuch, F., "Autonomous Flight and Navigation of VTOL UAVs: from Autonomy Demonstrations to Out-of-Sight Flights," *Aerospace Science and Technology*, vol. 11, no. 2-3, 2007.

[21] Frew, E. and Sengupta, R., "Obstacle Avoidance with Sensor Uncertainty for Small Unmanned Aircraft," *IEEE Conference on Decision and Control*, 2004.

[22] Frew, E. and Rock, S., "Trajectory Generation for Constant Velocity Target Motion Estimating using Monocular Vision," *AIAA Guidance, Navigation and Control Conference*, 2003.

[23] Geyer, M. and Johnson, E., "An Integrated Top-Down Approach to 3D Obstacle Avoidance in Adversarial Environments for Unmanned Aerial Vehicles," *AIAA Guidance, Navigation and Control Conference*, 2007.

[24] Ha, J., Alvino, C., Prior, G., Niethammer, M., Johnson, E., and Tannenbaum, A., "Active Contours and Optical Flow for Automatic Tracking of Flying Vehicles," *American Control Conference*, 2004.

[25] Ha, J., Johnson, E., and Tannenbaum, A., "Real-time Visual Tracking Using Geometric Active Contours for the Navigation and Control of UAVs," *American Control Conference*, 2007.

[26] Hammel, S., Liu, P., Hilliard, E., and Gong, K., "Optimal Observer Motion for Localization with Bearing Measurements," *Computers and Mathematics with Applications*, vol. 18, no. 1-3, 1989.

[27] Han, S. and Bang, H., "Proportional Navigation-Based Optimal Collision Avoidance for UAVs," *2nd International Conference on Autonomous Robots and Agents*, 2004.

[28] Hassoun, G. and Lim, C., "Advanced Guidance Control System Design for Homing Missiles with Bearing-Only Measurements," *IEEE International Conference on Industrial Technology*, 1994.

[29] Hayter, A., *Probability and Statistics.* Duxbury, 2002.

[30] Hines, W., Montgomery, D., Goldman, D., and Borror, C., *Probability and Statistics in Engineering.* John Wiley & Sons, 2003.

[31] Hrabar, S., Sukhatme, G., Corke, P., Usher, K., and Roberts, J., "Combined Optic-Flow and Stereo-Based Navigation of Urban Canyons for a uav," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005.

[32] Hull, D., Speyer, J., and Burris, D., "Linear-Quadratic Guidance Law for Dual Control of Homing Missiles," *AIAA Journal of Guidance, Control and Dynamics*, vol. 13, no. 1, 1990.

[33] Ivey, G. and Johnson, E., "Investigation of Methods for Simultaneous Localization and Mapping Using Vision Sensors," *AIAA Guidance, Navigation and Control Conference*, 2006.

[34] Johnson, E., Calise, A., Sattigeri, R., and Watanabe, Y., "Approaches to Vision-Based Formation Control," *IEEE Conference on Decision and Control*, 2004.

[35] Johnson, E., Calise, A., Watanabe, Y., Ha, J., and Neidhoefer, J., "Real-Time Vision-Based Relative Aircraft Navigation," *AIAA Journal of Aerospace Computing Information, and Communication*, vol. 4, no. 4, 2007.

[36] Johnson, E. and Kannan, S., "Adaptive Flight Controller for an Autonomous Unnmanned Helicopter," *AIAA Guidance, Navigation and Control Conference*, 2002.

[37] Johnson, E. and Schrage, D., "The Georgia Tech Unmanned Aerial Research Vehicle: GTMax," *AIAA Guidance, Navigation and Control Conference*, 2003.

[38] Johnson, E., Watanabe, Y., Ha, J., Calise, A., and Tannenbaum, A., "Image Processor, Estimation, Guidance, and Flight Test of Vision-Based Formation Flight," *The 3rd International Symposium on Innovative Aerial/Space Flyer Systems*, 2006.

[39] Julier, S. and Uhlmann, J., "A New Extension of the Kalman Filter to Nonlinear Systems," *SPIE AeroSense Symposium*, 1997.

[40] Kalman, R., "A New Approach to Linear Filtering and Prediction Problems," *Transaction of the ASME - Journal of Basic Engineering*, vol. 82, 1960.

[41] Kelley, C., *Iterative Methods for Linear and Nonlinear Equations.* Society for Industrial and Applied Mathematics, 1995.

[42] Kim, J. and Rock, S., "Stochastic Feedback Controller Design Considering the Dual Effect," *AIAA Guidance, Navigation and Control Conference*, 2006.

[43] Koch, A., Wittich, H., and Thielecke, F., "A Vision-Based Navigation Algorithm for a VTOL-UAV," *AIAA Guidance, Navigation and Control Conference*, 2006.

[44] Kumar, B. and Ghose, D., "Radar-Assisted Collision Avoidance / Guidance Strategy for Planar Flight," *IEEE Transactions on Aerospace and Electronics Systems*, vol. 37, no. 1, 2001.

[45] Kwag, Y. and Kang, J., "Obstacle Awareness and Collision Avoidance Radar Sensor System for Low-Altitude Flying Smart UAV," *Digital Avionics Systems Conference*, 2004.

[46] Langelaan, J. and Rock, S., "Towards Autonomous UAV Flight in Forests," *AIAA Guidance, Navigation and Control Conference*, 2005.

[47] Logothetis, A., Isaksson, A., and Evans, R., "An Information Theoretic Approach to Observer Path Design," *IEEE Conference on Decision and Control*, 1997.

[48] Logothetis, A., Isaksson, A., and Evans, R., "Comparison of Suboptimal Strategies for Optimal Own-Ship Maneuvers in Bearings-Only Tracking," *American Control Conference*, 1998.

[49] Madyastha, V., *Adaptive Estimation for Control of Uncertain Nonlinear System with Applications to Target Tracking.* Ph.D. Thesis, Georgia Institute of Technology, 2005.

[50] Meingast, M., Gayer, C., and Sastry, S., "Vision Based Terrain Recovery for Landing Unmanned Aerial Vehicles," *IEEE Conference on Decision and Control*, 2004.

[51] Metthies, L. and Kanade, T., "Kalman Filter-based Algorithms for Estimating Depth from Image Sequences," *International Journal of Computer Vision*, 1989.

[52] Miller, J., "3-D Color Terrain Modeling System for Small Autonomous Helicopters," *Ph.D. Thesis, Carnegie Mellon University*, 2002.

[53] Miller, J. and Amidi, O., "3-D Site Mapping with the CMU Autonomous Helicopter," *The 5th International Conference on Inteligent Autonomous Systems*, 1998.

[54] Morita, T. and Kanade, T., "A Sequential Factorization Method for Recovering Shape and Motion from Images Streams," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 19, no. 8, 1997.

[55] Neumann, T. and Bulthoff, H., "Insect Inspired Visual Control of Translatory Flight," *European Conference on Artificial Life*, 2001.

[56] Nordberg, K., Doherty, P., Farneback, G., and Forssen, P., "Vision for a UAV Helicopter," *IEEE International Conference on Intelligent Robots and Systems*, 2002.

[57] Oh, S. and Johnson, E., "Development of UAV Navigation System Based on Unscented Kalman Filter," *AIAA Guidance, Navigation and Control Conference*, 2006.

[58] Oh, S. and Johnson, E., "Relative Motion Estimation for Vision-based Formation using Unscented Kalman Filter," *AIAA Guidance, Navigation and Control Conference*, 2007.

[59] Oshman, Y. and Davidson, P., "Optimal Observer Trajectories for Passive Traget Localization using Bearing-Only Measurements," *AIAA Guidance, Navigation and Control Conference*, 1996.

[60] Oshman, Y. and Davidson, P., "Optimization of Observer Trajectories for Bearings-Only Traget Localization," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 35, no. 3, 1999.

[61] Pollini, L., Cellini, M., Mati, R., and Innocenti, M., "Obstacle Avoidance for Unmanned Ground Vehicles in Unstructured Environments," *AIAA Guidance, Navigation and Control Conference*, 2007.

[62] Pollini, L., Grecco, F., Mati, R., and Innocenti, M., "Stereo Vision Obstacle Detection based on Scale Invariant Feature Transform Algorithm," *AIAA Guidance, Navigation and Control Conference*, 2007.

[63] Proctor, A. and Johnson, E., "Vision-Only Aircraft Flight Control Methods and Test Results," *AIAA Guidance, Navigation and Control Conference*, 2004.

[64] Rao, N., Phillips, N., Fu, S., and Conrardy, N., "Horizontal Plane Trajectory Optimization for Threat Avoidance and Waypoint Rendezvous," *IEEE Aerospace and Electoronics Conference*, 1990.

[65] Sato, A., "Research and Development and Civil Application of an Autonomous Unmanned Helicopter," *AHS International Forum 57*, 2001.

[66] Sattigeri, R., *Adaptive Estimation and Control with Application to Vision-Based Autonomous Formation Flight.* Ph.D. Thesis, Georgia Institute of Technology, 2007.

[67] Sattigeri, R. and Calise, A., "Neural Network Augmented Kalman Filtering in the Precense of Unknown System Inputs," *AIAA Guidance, Navigation and Control Conference*, 2006.

[68] Saunders, J., Beard, R., and McLain, T., "Obstacle Avoidance using Circular Paths," *AIAA Guidance, Navigation and Control Conference*, 2007.

[69] Schutte, K. and et al., "ARC: A Camcopter Based Mine Field Detection System," *The 5th International Airborne Remote Sensing Conference*, 2001.

[70] Sethian, J., "A Fast Marching Level Set Method for Monotonically Advancing Fronts," *The National Academy of Sciences of the United States of America*, 1996.

[71] Shim, D., Kim, H., and Sastry, S., "A Flight Control System for Aerial Robots: Algorithms and Experiments," *IFAC Control Engineering Practice*, 2003.

[72] Singer, R., "Estimating Optimal Tracking Filter Performance for Manned Maneuvering Targets," *IEEE Transactions on Aerospace amd Electronic Systems*, vol. 6, no. 4, 1970.

[73] Singh, S., Kantas, N., Vo, B., Doucet, A., and Evans, R., "Simulation-Based Optimal Sensor Scheduling with Application to Observer Trajectory Planning," *submitted to Automatica*, 2005.

[74] Sinopoli, B., Micheli, M., Donato, G., and Koo, T., "Vision-Based Navigation for an Unmanned Aerial Vehicle," *IEEE International Conference on Robotics and Automation*, 2001.

[75] Speyer, J., Hull, D., Tseng, C., and Larson, S., "Estimation Enhancement by Trajectory Modulation for Homing Missiles," *AIAA Journal of Guidance, Control and Dynamics*, vol. 7, no. 3, 1984.

[76] Srinivasan, M., Chahl, J., Weber, K., Venkatesh, S., Nagle, M., and Zhang, S., "Robot Navigation Inspired by Principles of Insect Vision," *Roboics and Autonomous Systems*, vol. 26, 1999.

[77] Srinivasan, M., S.Zhang, and Chahl, J., "Landing Strategies in Honeybees, and Possible Applications to Autonomous Airborne Vehicles," *Invertebrate Sensory Information Processing: Implications for Biologically Inspired Autonomous Systems*, 2001.

[78] Tannenbaum, A., "Three Snippets of Curve Evolution Theory in Computer Vision," *Mathmatical and Computer Modeling*, vol. 24, 1996.

[79] Thrun, S. and et al., "Stanley, the Robot That Won the DARPA Grand Challenge," *submitted to Journal of Field Robotics*, 2006.

[80] U.S. Air Force, "Predator RQ-1/MQ-1/MQ-9 Unmanned Aerial Vehicle (UAV), USA," tech. rep., http://www.airforce-technology.com/projects/predator/.

[81] Valasek, J., Kimmett, J., Hughes, D., Gunnam, K., and Junkins, J., "Vision Based Sensor and Navigation System for Autonomous Aerial Refueling," *AIAA the 1st UAV Conference*, 2002.

[82] Watanabe, Y., Calise, A., and Johnson, E., "Minimum -Effort Guidance for Vision-Based Collision Avoidance," *AIAA Atmospheric Flight Mechanics Conference*, 2006.

[83] Watanabe, Y., Calise, A., and Johnson, E., "Vision-Based Obstacle Avoidance for UAVs," *AIAA Guidance, Navigation and Control Conference*, 2007.

[84] Watanabe, Y., Johnson, E., and Calise, A., "Optimal 3-D Guidance from a 2-D Vision Sensor," *AIAA Guidance, Navigation and Control Conference*, 2004.

[85] Watanabe, Y., Johnson, E., and Calise, A., "Vision-Based Approach to Obstacle Avoidance," *AIAA Guidance, Navigation and Control Conference*, 2005.

[86] Watanabe, Y., Johnson, E., and Calise, A., "Stochastically Optimized Monocular Vision-Based Navigation and Guidance Design," *AIAA Guidance, Navigation and Control Conference*, 2007.

[87] Watanabe, Y., Madyastha, V., Johnson, E., and Calise, A., "Vision-Based Approaches to UAV Formation Flight and Obstacle Avoidance," *The 2nd International Symposium on Innovative Aerial/Space Flyer Systems*, 2005.

[88] Webb, T., Prazenica, R., Kurdila, A., and Lind, R., "Vision-Based State Estimation for Autonomous Micro-Air Vehicles," *AIAA Guidance, Navigation and Control Conference*, 2004.

[89] Welch, G. and Bishop, G., "An Introduction to the Kalman Filter," Tech. Rep. TR 95-041, University of North Carolina at Chapel Hill, 2004.

[90] Wu, A., Johnson, E., and Proctor, A., "Vision-Aided Inertial Navigation for Flight Control," *Journal of Aerospace Computing, Information, and Communication*, vol. 2, no. 9, 2005.

[91] Wu, A., Turbe, M., Kannan, S., Neidhoefer, J., and Johnson, E., "Flight Results of Autonomous Airplane Transitions to and from Vertical Hover," *AIAA Guidance, Navigation and Control Conference*, 2006.

[92] Zarchan, P., *Tactical and Strategic Missile Guidance.* AIAA, 1994.

[93] Zarchan, P. and Musoff, H., *Fundamentals of Kalman Filtering: A Practical Approach.* AIAA, 2004.

[94] Zhou, K., Doyle, J., and Glover, K., *Robust and Optimal Control*. Prentice Hall, 1996.

VITA

YOKO WATANABE

Yoko Watanabe was born on November 22, 1978 in Nara, Japan. She started her undergraduate study at Kyoto University, Kyoto, Japan in 1997 and received a Bachelor of Engineering in Engineering Science in March, 2001. The following April she entered Graduate School of Engineering at Kyoto University. She was awarded a Master of Engineering in Aeronautics and Astronautics from Kyoto University in March, 2003. She joined the graduate study program in School of Aerospace Engineering at Georgia Institute of Technology (Georgia Tech), Atlanta GA in the United States, in August, 2003. She received a Master of Science in Aerospace Engineering from Georgia Tech in May, 2005 and began a PhD program. She also accomplished a Master of Science in Mathematics in School of Mathematics at Georgia Tech in August, 2007. Her research interests include autonomous flight control, navigation and guidance system design for UAVs, optimal guidance and control, nonlinear estimation filter design, flight mechanics, and flight simulation and testing.