Summer 7-2020

Three-Dimensional, Vision-Based Proportional Navigation for UAV Collision Avoidance

Guanhong Gao
Embry-Riddle Aeronautical University

THREE-DIMENSIONAL, VISION-BASED PROPORTIONAL NAVIGATION FOR UAV COLLISION AVOIDANCE

By

Guanhong Gao

A Thesis Submitted to the Faculty of Embry-Riddle Aeronautical University

In Partial Fulfillment of the Requirements for the Degree of

Master of Science in Aerospace Engineering

July 2020

Embry-Riddle Aeronautical University

Daytona Beach, Florida

THREE-DIMENSIONAL, VISION-BASED PROPORTIONAL NAVIGATION FOR UAV COLLISION AVOIDANCE

By

Guanhong Gao

This Thesis was prepared under the direction of the candidate's Thesis Committee Chair, Dr. Richard Prazenica, Department of Aerospace Engineering, and has been approved by the members of the Thesis Committee. It was submitted to the Office of the Senior Vice President for Academic Affairs and Provost, and was accepted in the partial fulfillment of the requirements for the Degree of Master of Science in Aerospace Engineering.

THESIS COMMITTEE

Chairman, Dr. Richard Prazenica	Member, Dr. Hever Moncayo
Member, Dr. Troy Henderson	
Graduate Program Coordinator, Dr. Magdy Attia	Date
Dean of the College of Engineering, Dr. Maj Mirmirani	Date
Associate Provost of Academic Support, Dr. Christopher Grant	Date

ACKNOWLEDGEMENTS

My deepest gratitude goes first and foremost to Dr. Richard Prazenica, for his constant encouragement and guidance. He has walked me through all the stages of the writing of this thesis. Without his consistent and illuminating instruction, this thesis could not have reached its present form.

Second, I would like to express my heartfelt gratitude to my committee member, Dr. Troy Henderson and Dr. Hever Moncayo. They gave me good advice to help me improve my thesis. This thesis could not have a clear development direction without their advising.

Last my thanks would go to my beloved family for their loving considerations and great confidence in me all through these years. I also owe my sincere gratitude to my friends and my fellow classmates who gave me their help and time in listening to me and helping me work out my problems during the difficult course of the thesis.

ABSTRACT

As the number of potential applications for Unmanned Aerial Vehicles (UAVs) keeps rising steadily, the chances that these devices will operate in close proximity to static or dynamic obstacles also increases. Therefore, collision avoidance is an important challenge to overcome for Unmanned Aerial Vehicle operations. Electro-optical devices have several advantages such as light weight, low cost, low algorithm requirements with respect to computational power and possibly night vision capabilities. Therefore, vision-based Unmanned Aerial Vehicle collision avoidance has received considerable attention. Although much progress has been made in collision avoidance systems (CAS), most approaches are focused on two-dimensional environments. In order to operate in complex three-dimensional urban environments, three-dimensional collision avoidance systems are required. This thesis develops a three-dimensional vision-based collision avoidance system to providesense and avoid capabilities for unmanned aerial vehicles (UAVs) operating in complex urban environments with multiple static and dynamic collision threats. This collision avoidance system is based on the principle of proportional navigation (Pro-Nav), which states that a collision will occur when the line-of-sight (LOS) angles to another object remain constant. According to this guidance law, monocular electro-optical devices can be implemented on Unmanned Aerial Vehicles, which can provide measurements of the line-of-sight angles, indicating potential collision threats. In this thesis, the guidance laws were applied to a nonlinear, six degree-of-freedom Unmanned Aerial Vehicles model in different two-dimensional or three-dimensional simulation environments with a varying number of static and dynamic obstacles.

TABLE OF CONTENTS

LIST OF FIGURES

Figure Page

Figure Page

LIST OF TABLES

Table Page

NOMENCLATURE

ACAS	Airborne Collision Avoidance System
ADS-B	Automatic Dependent Surveillance-Broadcast
CAS	Collision Avoidance System
FOV	Field of View
LIDAR	Light Detection and Ranging
LPF	Low Pass Filter
LOS	Line of Sight
RADAR	Radio Detection and Ranging
SAA	Sense and Avoid
SONAR	Sound Navigation and Ranging
TCAS	Traffic Collision Avoidance System
UAS	Unmanned Aircraft System
UAV	Unmanned Aerial Vehicle

1. Introduction

An Unmanned Aerial Vehicle (UAV), also known as a drone, has no on-board pilot or crew. Unmanned aerial vehicles are a part of an unmanned aircraft system (UAS), which also comprises a communication system between the UAV and the ground controller. The operator's remote control or on-board computer can give the UAV varying degrees of autonomy.

UAVs were initially used in military applications, but the potential of UAVs to be widely used in civil applications is also considerable, and their use has rapidly expanded to business. UAVs have many advantages, such as low cost, safety benefits, and mobility. Due to those benefits, UAVs can potentially replace crewed aerial vehicles in many tasks that are too dangerous or dirty for the human to complete. On the other hand, some features of UAV operations can cause problems and challenges because they are not directly controlled by a human during flight. One of the biggest challenges is collision avoidance. In order to complete a successful flight mission, collision avoidance ability is necessary for both static and dynamic obstacles. Fundamentally, UAV avoidance and other air traffic, ground traffic, or robots have some similarities but also retain their characteristics, like flight dynamics or collision detection, which makes them a unique and interesting basis for research (Pham, Smolka, Stoller, Phan, & Yang, 2015).

1.1. Sense-and-Avoid functionality

After removing the pilot from the traditional aircraft control system, to overcome the most prominent challenge of collision avoidance, UAVs need to have Sense-and-Avoid (SAA) functionality or a SAA system.

An effective SAA system should meet the following requirements：

1. Be able to detect obstacles and potential collisions.
2. Division of levels of risk to determine imminent collisions.
3. Determine the right time to react to the potential collisions and determine the particular maneuver and flight trajectory based on geometric or optical measurements.
4. Generate the command to the controller and complete the maneuver.

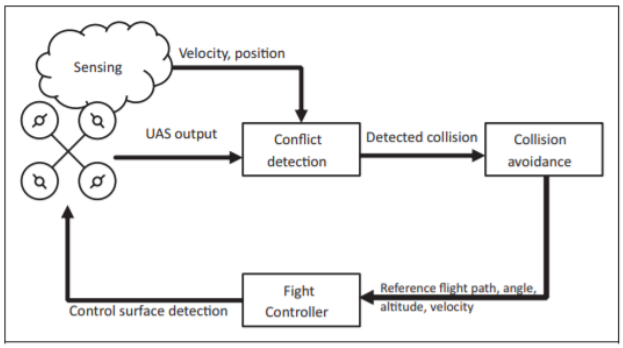

Figure 1.1 SAA Components (Skowron, Chmielowiec, lowacka , Krupa, & Srebro, 2019).

A typical SAA system has four components, including sensing, conflict detection, collision avoidance, and the flight controller, as shown in Figure 1 (Skowron, Chmielowiec, lowacka , Krupa, & Srebro, 2019). The last part is common for both unmanned vehicles and manned vehicles; thus, in this thesis, we limit the discussion to the first three components, and they will be explained in detail later.

1.2. Sensing and Detection

Sensing and detection are the first step of a SAA or Collision Avoidance System (CAS); the sensing ability is how the UAVs collect useful information from the

surrounding environment. In contrast to trajectory planning, collision avoidance systems focus more on perceived risks and reacting to the dangers as soon as possible. Thus, the sensor does not have to obtain all the information of the obstacles; only the information required to indicate a collision is needed. In general, the SAA sensing can be divided into two different types: Cooperative Sensing (e.g., ADS-B, TCAS, ACAS) and Noncooperative Sensing (e.g., vision sensors, RADAR and FOV).

1.2.1. Cooperative Sensing

As discussed by Skowron et al. (2019), cooperative sensing is based on several aircraft, which are equipped with similar devices, cooperating with each other, with the information transmitted between the aircraft. The most common cooperative sensors are ADS-B, TCAS, ACAS. In the beginning, those sensing solutions were designed for large scale manned aircraft to avoid midair collisions, but with further development, their size has been modified to fit mid or small-scale vehicles like UAVs.

Airservices Australia introduces the working principle of the Automatic Dependent Surveillance-Broadcast (ADS-B) system in detail in “How ADS-B works”.
ADS-B periodically broadcasts the precise position of aircraft, which is obtained by satellite navigation (2012). The position information is transmitted between aircraft, which can use the data to find out the position of the aircraft, such that RADAR is no longer required. The system does not need any pilot input and can continuously provide all the necessary flight data like position and velocity to the other aircraft. ADS-B can effectively reduce collisions and ensure safety and is regarded as essential in the future Unmanned Aircraft Systems (UAS) and Sense-and-Avoid (SAA) systems.

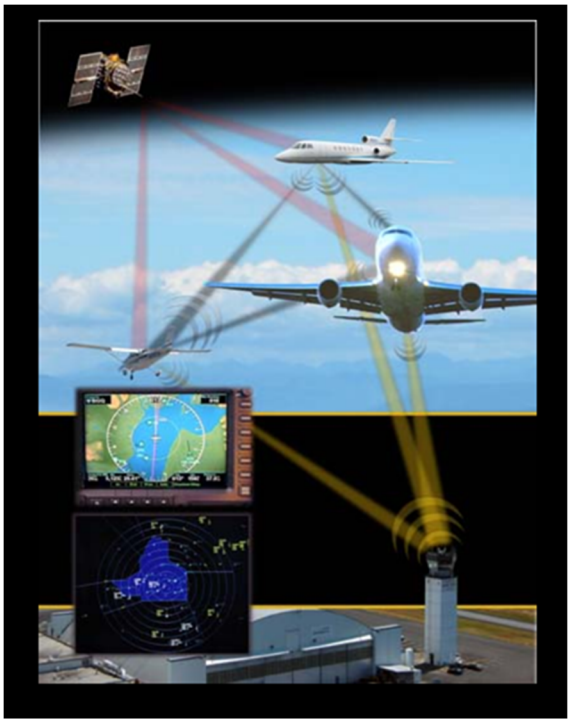

Figure 1.2 ADS-B schematic diagram ("Automatic dependent surveillance – broadcast", n.d.).

Collision avoidance systems have been under development for many years. As early as the late 20th century, the Traffic Collision Avoidance System (TCAS) or Traffic Alert and Collision Avoidance System was originated, which tries to prevent mid-air collisions between two aircraft. TCAS can monitor the airspace including all aircraft that are equipped with a corresponding active transponder. The monitoring system will independently warn pilots of the existence of those transponder-equipped aircraft and the potential of mid-air collision (MAC).

TACS can effectively prevent collisions, but its deterministic logic also results in several limitations. For example, TCAS does not account for uncertainty caused by the pilot. Different from TCAS, the Airborne Collision Avoidance System (ACAS, usually pronounced as ay-kas) can warn pilots of uncertainties and risk of collision threats by

applying probabilistic models to avoid false positives. When the system detects high risk, the system will commence a maneuver to avoid any imminent collision (Essen & Giannakopoulou, 2014).

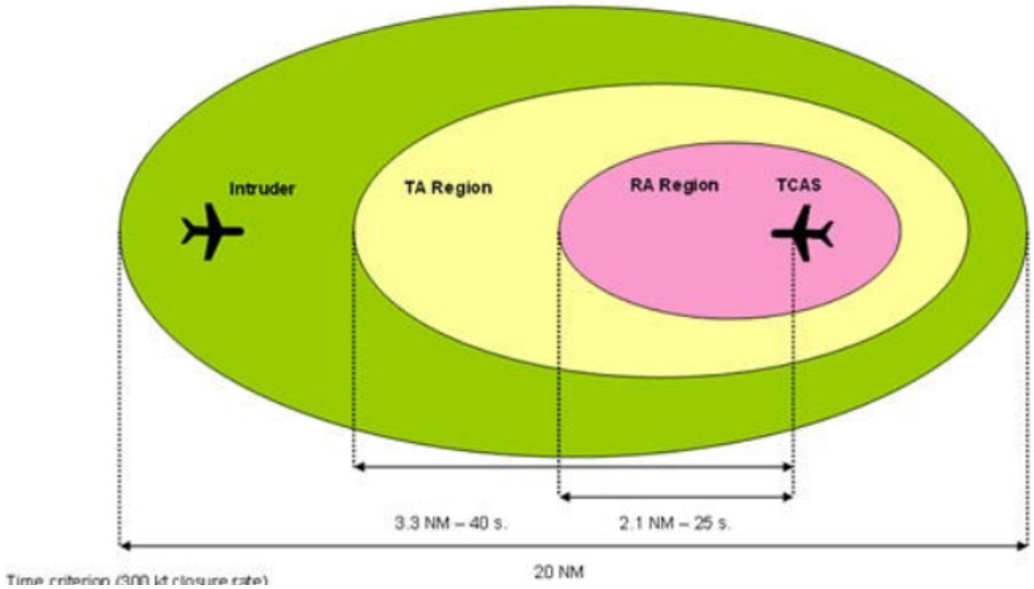

Figure 1.3 TCAS schematic diagram (Essen & Giannakopoulou, 2014).

1.2.2. Noncooperative Sensing

Compared with cooperative sensing, noncooperative sensing does not need other aircraft to participate in cooperation or share flight data with others. Aircraft equipped with this sensing solution deal with collision risk independently by analyzing signals from on-board sensors. Due to its independence, noncooperative sensing is suitable for small scale UAV collision avoidance at low altitude, and the technology development over the past several years has made this approach possible for UAV applications. Some common non-cooperative sensors include electro-optical sensor and RADAR.

Vision sensors, which can provide images containing the coordinates of obstacles are commonly used sensors for collision avoidance. The SAA system can get useful

information directly from the image plane. For example, Saha, Natra and Waharte (2014) developed an SAA system by using a monocular camera to measure the real distance between the obstacles and the UAV. Vision sensors have the following advantages (Pham et al., 2015):

1. They are small and lightweight, making them suitable for integration on small UAVs to complete avoidance maneuvers.
2. They are easy to operate, which is beneficial to civil and business applications.
3. Vision sensors can directly measure LOS angles for specific avoidance algorithms.
4. This type of sensor resembles our natural vision, which makes it promising and gives researchers significant intuition.

Radio Detection and Ranging (RADAR) is another representative sensing system which can provide useful information based on electromagnetic waves in the radio or microwave spectrum. According to Viquerat (2007), not only the obstacle velocity and range, but also the angular location with respect to the observer can be returned by the continuous wave microwave doppler RADAR under certain circumstances (pp. 245-254). Compared to cameras, a RADAR system has a larger detection range.

Similar to RADAR, Light Detection and Ranging (LIDAR) and Sound Navigation and Ranging (SONAR) are also UAV-friendly active methods. LIDAR sensors illuminate laser light impulses and measure the reflected light to calculate the distance and create a 3D representation of a digital image. SONAR uses acoustic waves reflection to accomplish a measurement. It is important to note that light impulses can help the sensing system to perform high-precision calculations, but they are also more susceptible to

atmospheric conditions. In contrast, SONAR performance also depends on the weather, but it can maintain high accuracy while keeping a lower cost. Therefore, SONAR is more popular in UAV applications (Skowron et al., 2019).

1.3. Detection and Maneuver Approach

After sensing obstacle information, the next important step is determining if a collision will occur; then the collision avoidance system can make a decision to generate an avoidance command. Whether the impact can be accurately detected is a critical quality of the system, which depends on the sensing information and the algorithms used to process the sensor data. At present, the most commonly used methods are trajectory calculation and distance estimation (Pham et al., 2015).

Trajectory calculation and distance estimation methods have the same basic idea, which is checking the distance between the UAV and the obstacle when the distance is less than a setup threshold value. The detection system will regard the obstacle as an imminent collision. These methods are widely used in CAS; both of them will calculate the shortest distance between the objects. Therefore, it requires the sensors to have the ability to provide an accurate position or distance to obstacles. Based on this theory, a monocular detection method was proposed, which could merge the approximate outline of obstacles from multi-scale-oriented patches (MOPS) and the spatial coordinates of feature points calculated by the scale-invariant feature transform (SIFT) algorithm (Lee, Lee, Park, Im, & Park, 2011). A quadcopter frontal obstacle detection and avoidance in a GPS denied environment was achieved by Saha et al. (2014).

A bearing angle approach method was proposed by Saunders and Beard (2008). In this paper, the field of view (FOV) was considered as a “no obstacle” range. Thus, when

the obstacle can be detected within the camera FOV, the UAV will generate a maneuver by changing the bearing angle until the targets disappeared from the FOV.

In this thesis, inspired by recent research from Clark (2017), a proportional navigation collision detection algorithm is developed, which is based on the vision information from a monocular camera. In this case, the guidance system can determine avoidance maneuvers by using standard camera hardware to measure line-of-sight angles.

1.4. Proportional Navigation

In this thesis, based on camera sensing, proportional navigation is implemented to achieve detection and avoidance. Proportional navigation, also known as Pro-Nav, is a guidance law that was proposed initially for missile interception. The research on proportional navigation can be traced back to World War II. In 1950, the first missile to use proportional navigation was tested successfully (Zarchan, 1994). Until now, the studies on proportional navigation have made great progress. There are six proportional navigation laws in common use including true proportional navigation (TPN), realistic version of true proportional navigation (RTPN), generalized proportional navigation (GTPN), ideal proportional navigation (IPN), pure proportional navigation (PPN) and optimal proportional navigation (OPN) (Li et al., 2013).

Since the 1940s, 2D proportional navigation has been studied in a lot of literature and that research also achieved great success. The fundamental law was posted by Northwestern Polytechnical University.

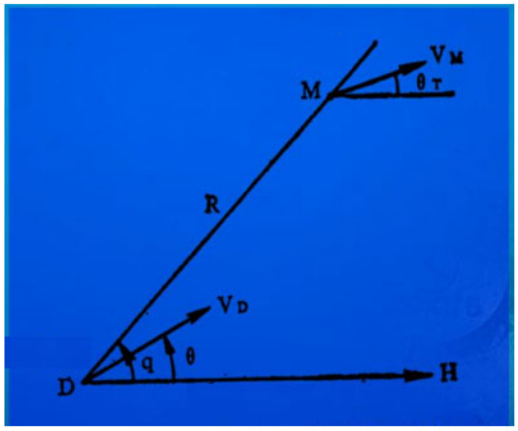

Figure 1.4 2-D proportional navigation (Northwestern Polytechnical Univ., 2012).

As shown in Figure 1.4, D is the missile with a constant velocity V_D and M is the target with a constant velocity V_M. The vector pointing from missile to target $\vec{R}$ is the line-of-sight vector; by defining a reference direction vector $\overrightarrow{DH}$, we can obtain the LOS angle $\angle q$. The proportional navigation law states that a collision will occur when the relative velocity VR is aligned with the LOS; thus, q will remain constant throughout the end game (Murtaugh & Criel, 1966).

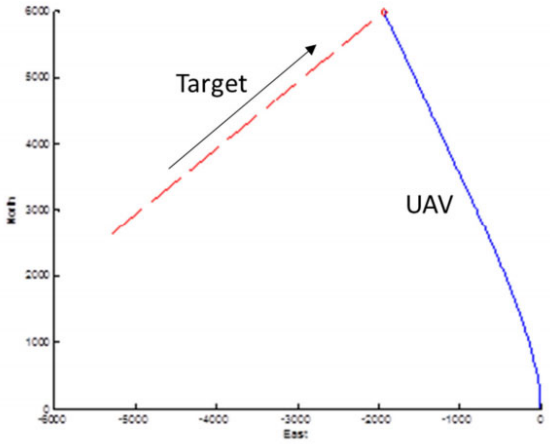

Figure 1.5 2-D frame proportional navigation intercept scenario-1 (Clark, 2017).

Simulation results using proportional navigation interception laws can be seen in Figures 1.5 and 1.6 (Clark, 2017). Figure 1.5 displays a typical interception scenario, in which the UAV represents an interceptor with an initial north heading. At the same time, the target starts at a point one half-mile north and one mile west of the UAV with a 26.6° heading and the same velocity as the UAV, which can be detected as a positive LOS rate. The controller reacts almost immediately to drive the UAV to intercept by changing the heading westward to the target.

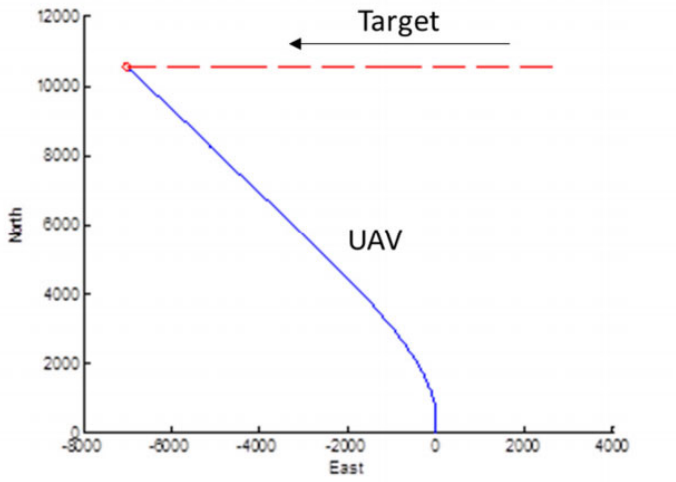

Figure 1.6 2-D frame proportional navigation intercept scenario-2 (Clark, 2017).

Another example is shown in Figure 1.6, in which the UAV is flying with the same initial condition; the target starts a mile and a half east of the UAV start point with the same constant velocity but a different -90 heading. In this case, the target is detected in a different direction, which provides a negative initial LOS rate. However, the UAV will still change heading by a negative angle to intercept with a constant LOS angle. In conclusion, the Pro-Nav guidance result is not affected by the bearing of the target but only by the sign and value of the LOS rate.

In the early literature, proportional navigation was divided into two basic types – pure proportional navigation (PPN) and true proportional navigation (TPN) (Guelman, 1976). The difference between the two guidance laws is that the command acceleration is applied along the normal vector of missile velocity and the target line-of-sight (LOS) in PPN and TPN, respectively. In 1976, the closed solution of the equations of motion according to the TPN was provided by Guelman, and it started research on the performance analysis of the TPN in a nonlinear framework. After this, the closed solution was also obtained from nonlinear equation of motion and trajectory analysis by several such researches by Yang et al. (1980) and Becker (1990).

However, all of the above literature made numerous assumptions when analyzing the equations of motion because of the highly nonlinear nature of those equations. In order to make the analysis results closer to reality, Dhar and Ghose investigated the realistic version of true proportional navigation (RTPN) by reducing the simplifying assumption on the closing velocity between the missile and the interception target.

The generalized proportional navigation (GTPN) was defined by Yang et al. (1987) and developed the closed form solution of equations of motion under this guidance law. The GTPN is a similar law as PN, but the commanded missile acceleration does not have to be the normal vector of LOS. In some circumstances, GTPN has a larger capture area and a shorter intercept time compared to TPN; therefore, GTPN can be considered as a better guidance law in these scenarios.

The traditional PN laws have been widely studied and developed, but the LOS referenced PN law was found to be hard to achieve practically (Sakula & Mahapatra, 1990). Therefore, a new guidance law called ideal proportional navigational (IPN) was

developed. In IPN, the commanded acceleration is along the normal vector of the relative velocity rather than the LOS vector and it is proportional to the product of the LOS rate and the relative velocity (Yuan & Chern, 1992). According to the closed form solution, IPN could be considered as the most effective guidance law out of all PN laws.

Since the real intercept happens in a three-dimensional environment, 3D Pro-Nav has also been studied extensively. The 3D generalized proportional navigation was introduced by Yang and Yang (1996); in their work, the analytical solution was also derived, which solved the difficulties existing in coupled nonlinear equations. In 1995, Guelman et al. provided a 3D minimum energy guidance law based on the pure proportional navigation (PPN). In contract to missile intercept guidance, proportional navigation has also been studied for 3D collision avoidance. A vision aided Pro-Nav law was developed for micro air vehicles by Beard et al. (2007). In their paper, the guidance laws were derived for both bank-to-turn and skid-to-turn equations of motion.

1.5. Thesis Objectives and Summary

This thesis extends the previous literature and applies the Pro-Nav to automatic collision avoidance based on the interception guidance law. This thesis focuses on the three-dimensional algorithm and designs a 3D collision avoidance system based on the pro-Nav law. In order to complete a 3D maneuver, a flight path selection switch was designed to make the UAV select vertical or lateral maneuvers to accomplish the avoidance. In addition, this thesis investigates the UAV performance with camera noise and develops a method to improve the performance. The rest of the thesis is organized as follows: Chapter 2 elaborates on the sensing and detection method, which includes the camera, UAV modeling and 3D line of sight definition. Chapter 3 describes the 3D

proportional navigation law and discusses the flight path selection law. This chapter also provides weighting function methods to generate guidance commands. By implementing this method, the flight path is optimized by minimizing cost functions. Chapter 4 provides simulation results in several scenarios by analyzing the resulting trajectories and some specific metrics. Chapter 5 and 6 discusses the influence of different factors on UAV performance, including noise in the image measurements, and then verifies the feasibility and robustness of the system.

2. Vision System and UAV Modeling

The collision avoidance system is shown in Figure 2.1 and has four parts: aircraft, sensor, guidance command generator, and controllers. In this thesis, a middle-scale fixed-wing UAV model is selected, and the 12 flight dynamics equations of motion are used to model non-linear, six-degree-of-freedom (DOF) UAV dynamics to calculate UAV states in a simulation environment.

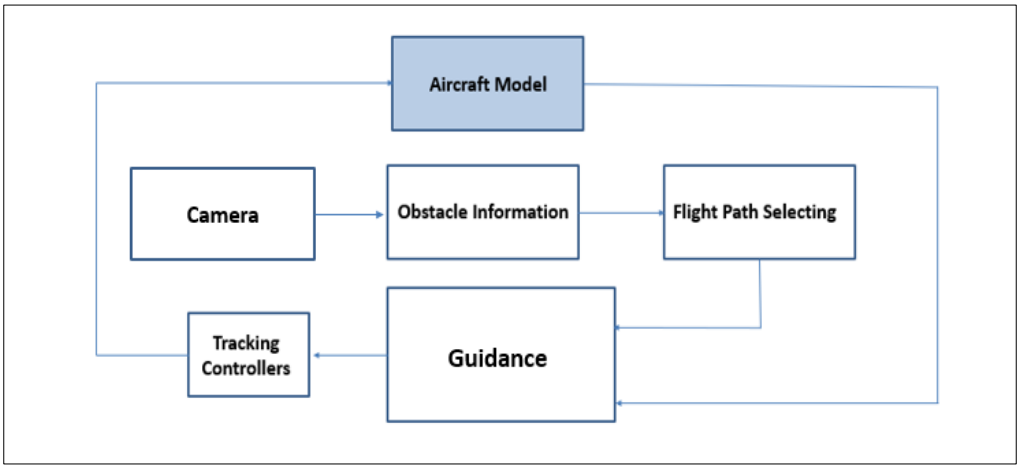

Figure 2.1 Diagram of collision avoidance system.

2.1. UAV Model

In this thesis, a Navion aircraft is selected as a representative medium-scale, fixed-wing UAV model. The wingspan is 33.4 ft, the weight is 2756 lbf and the wing surface is 184.1 ft^2. According to the size of the UAV, the collision occurs when the obstacle miss distance is within the half wingspan. Aerodynamic data for this aircraft at sea level are available in (Nelson, 1998). The aircraft mass and geometric parameters are provided in Table 2.1.

Table 2.1

UAV Model Parameters

Parameter	**Value**	**Unit**
Wingspan (b_w)	33.4	ft
Wing aspect ratio (A_w)	6.06	
Chord length (c_w)	5.7	ft
Wing surface area (S_w)	184.1	ft^2
Weight (W)	2756	lbf

The UAV maneuvers are controlled by several PID controllers, which include rolling, pitching, heading and velocity controllers. The UAV velocity and rudder deflection are maintained at a trim condition and zero, respectively, during maneuvering. Rolling and pitching angles are controlled by rolling and pitching PID controllers to accomplish the guidance commands generated by the Pro-Nav guidance law to complete avoidance maneuvers.

The guidance and control system is designed to generate two uncoupled avoidance maneuvers: vertical maneuvers and lateral maneuvers. The vertical avoidance is accomplished by pitching the UAV to change the vertical flight path angle, and lateral avoidance is performed using a bank-to-turn maneuver generated by a heading change. The bank-to-turn maneuver, also known as a banking turn, is a fundamental motion for fixed-wing aircraft. As shown in Figure 2.2, turning is started by inclining the wings to let the aircraft roll. Therefore, the unopposed component of the lift will generate a side force perpendicular to the flight path, which will make the aircraft change its heading. In UAV simulation model, the UAV rudder is fixed to ensure the heading change is only generated by the bank-to-turn maneuver.

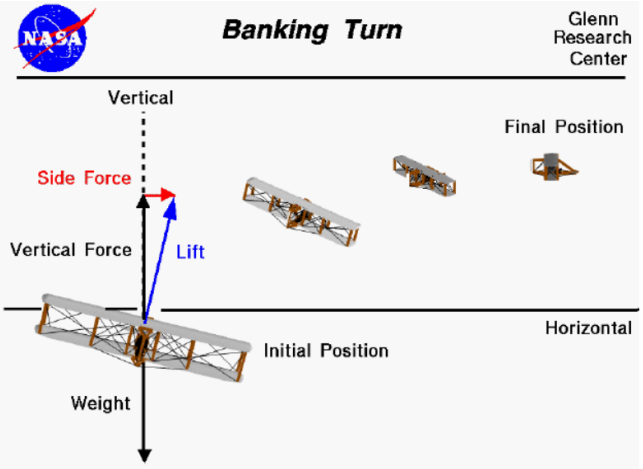

Figure 2.2 Aircraft bank-to-turn maneuver (NASA Glenn Research Center, 2015).

The commanded roll change $\Delta\phi_{com}$ is proportional to the commanded heading change $\Delta\psi_{com}$, and $\Delta\phi_{com}$ can be expressed in Equation (1):

$$\Delta\phi_{com} = K_{\phi_{com}} \Delta\psi_{com} \tag{1}$$

$K_{\phi_{com}}$ is a proportional gain, which can convert the commanded heading change $\Delta\psi_{com}$ to a commanded roll angle change $\Delta\phi_{com}$. Then, the roll PID controller output will command the aileron actuator to generate a deflection, which is shown in Figure 2.3.

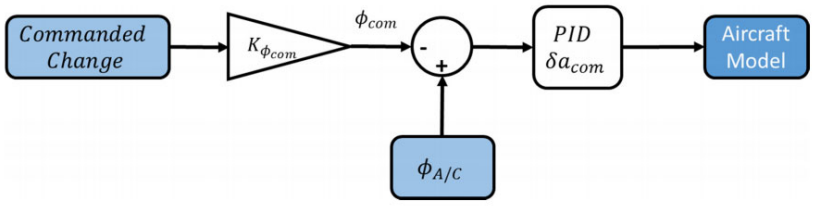

Figure 2.3 Diagram of Bank-to-turn control method for lateral UAV maneuvers.

The pitching maneuver is accomplished by changing the vertical flight path angle Γ, which can be expressed as:

$$\Gamma = \theta_{A/C} - \alpha \tag{2}$$

α is the UAV angle of attack (AOA); thus, the vertical flight path angle can be represented by the difference between the UAV pitch angle and AOA.

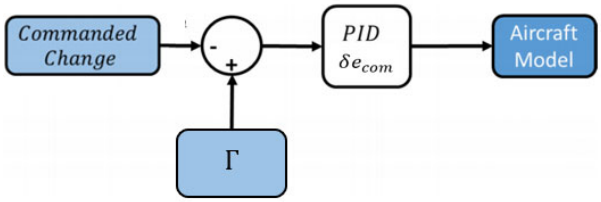

Figure 2.4 Diagram of vertical maneuver control method for UAV simulations.

Then, the pitch PID controller output will command the elevator actuator to generate a deflection, which is shown in Figure 2.4.

2.2. Camera Model

In this thesis, a monocular camera with [-30°, 30°] field of view and 1280 x 1280 pixel resolution is selected and modeled. Compared to other vision devices like stereo cameras, the monocular camera has a simpler structure, which means it can provide the necessary information while saving cost. Besides, the measurement range of stereo is limited by the distance between two cameras; this shortcoming does not exist in monocular applications. Those advantages make monocular vision a suitable choice for small to middle scale UAV applications. However, monocular vision also has a big limitation in that it cannot resolve range information. A monocular camera lacks the ability to locate the exact position and determine the real size of targets. Therefore, a proportional navigation algorithm (Pro-Nav) was developed that only uses the line of sight (LOS) angles, which can be measured using monocular vision and line of sight rates, rather than the 3D target location, to accomplish the avoidance.

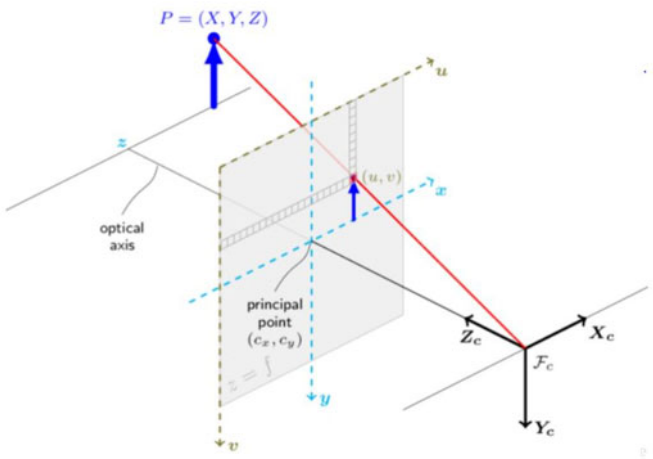

Figure 2.5 Camera mapping model (CSDN, n.d.).

2.3. Coordinate Transformation

The LOS vector is defined as the vector pointing from the UAV center of mass to the target in the inertial (NED) frame, which can be expressed as:

$$LOS = r_P^E - r_C^E = \begin{bmatrix} \hat{X} \\ \hat{Y} \\ \hat{Z} \end{bmatrix} \tag{3}$$

in which r_P^E and r_C^E are the target and UAV positions in the NED frame and, $\hat{X}$, $\hat{Y}$ and $\hat{Z}$ are the projections of the LOS vector on the north, east, and Z axes. It is imperative to convert all the calculations to the inertial frame, because all measurements need to be in a common reference frame in order to compute LOS rates.

According to the relationship between the coordinates mentioned above, the back-projecting method can be utilized to map the targets from the pixel frame to the NED frame. The mapping starts from the pixel frame, where the obstacle information is captured.

As shown in Figure 2.6, (u, v) are the pixel frame coordinates, whose origin $O_{uv,}$ is located at the left top corner, and (x, y) are the image frame coordinates, whose origin O is located at the center of the image plane.

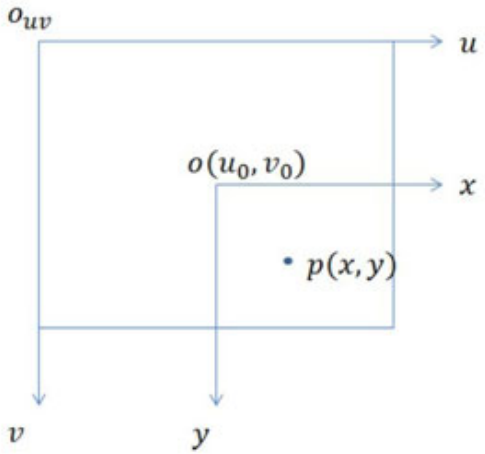

Figure 2.6 Target location in the pixel and image frames (CSDN, n.d.).

The transformation from pixel to image coordinates can be expressed as:

$$\begin{cases} u = \dfrac{x}{dx} + u_0; \\ v = \dfrac{y}{dy} + v_0; \end{cases} \tag{4}$$

where 1 $pixel = dx\ mm$. This result in the following transformation:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \mathrm{K} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \dfrac{1}{dx} & 0 & u_0 \\ 0 & \dfrac{1}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{5}$$

In equation (5), K is defined as the intrinsic calibration matrix to transform the coordinates from the pixel frame into the image frame.

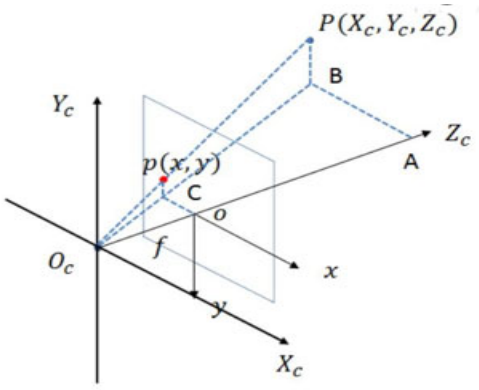

Figure 2.7 Target location in the image and camera frames (CSDN, n.d.).

Figure 2.7 shows the calibration from the image frame into the camera frame. (X_c, Y_c, Z_c) represents the target with respect to the camera frame, the origin of which is the optical center. The triangles constituted by the target points, image plane and optical center comply with the similar triangle theorem, which results in:

$$\frac{AB}{oC} = \frac{AO_c}{oO_c} = \frac{PB}{pC} = \frac{X_c}{x} = \frac{Z_c}{f} = \frac{Y_c}{y} \tag{6}$$

$$x = f\frac{X_c}{Z_c}, y = f\frac{Y_c}{Z_c} \tag{7}$$

where f is the focal distance. Assuming the unknown value Z_c as 1, the following physical relationship is obtained:

$$R_I^C \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{8}$$

Finally, the LOS vector can be mapped into the inertial frame as:

$$\begin{bmatrix} \hat{X} \\ \hat{Y} \\ \hat{Z} \end{bmatrix} = [DCM]^{-1} R_C^B R_I^C K \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

$$= [DCM]^{-1} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} \frac{1}{dx} & 0 & u_0 \\ 0 & \frac{1}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad (9)$$

R_C^B is the transformation matrix from camera to body-fixed axes. In this thesis, the camera is aligned with the body-fixed axes, so the transformation simply reorders the coordinate axes. $[DCM]$ is the commonly known Direction Cosine Matrix that defines the transformation from the inertial to the body-fixed frame:

$$[DCM] = \begin{bmatrix} cos\theta cos\psi & cos\theta sin\psi & -sin\theta \\ -cos\phi sin\psi + sin\phi sin\theta cos\psi & cos\phi cos\psi + sin\phi sin\theta sin\psi & sin\phi cos\theta \\ sin\phi sin\psi + cos\phi sin\theta cos\psi & -sin\phi cos\psi + cos\phi sin\theta sin\psi & cos\phi cos\theta \end{bmatrix}$$

where ϕ, ψ, θ represent the UAV roll, yaw and pitch angles. Without loss of generality, $\hat{X}$ is assumed to be 1 to rescale the LOS vector; then we can obtain the LOS direction vector:

$$los = \begin{bmatrix} 1 \\ \frac{\hat{Y}}{\hat{X}} \\ \frac{\hat{Z}}{\hat{X}} \end{bmatrix} = \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{bmatrix} \quad (10)$$

2.4. Line of Sight Vector

For implementing the Pro-Nav law in a 3D environment, the LOS angles must be computed in the same reference frame. Therefore, a frame called the vehicle frame is defined to achieve that. As shown in Figure 2.8, the vehicle frame, whose axes are denoted by (Xv, Yv, Zv), is oriented identically to the inertial frame, but its origin is at the vehicle center of mass (Beard et al., 2007).

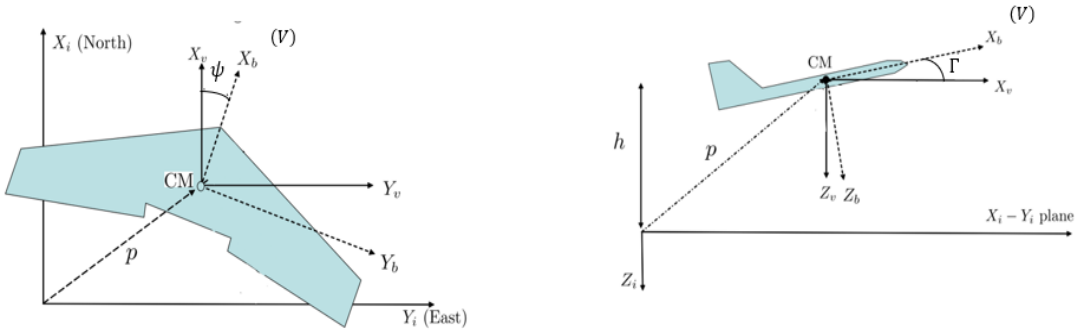

Figure 2.8 Vehicle frame and body frame (Beard et al., 2007).

Then, the 3D LOS angle can also be decomposed into the two projection planes to yield a horizontal LOS angle χ and a vertical LOS angle γ, which can be derived as follows:

$$\chi = \tan^{-1}\left(\frac{\hat{y}}{\hat{x}}\right) \tag{11}$$

$$\gamma = \tan^{-1}\left(\frac{\hat{z}}{\hat{x}}\right) \tag{12}$$

Assuming the sideslip is approximately equal to 0, the projection of the angle between X_v and X_b on (X_v, Y_v) coordinates can represent the UAV heading angle ψ, and the projection of the angle between X_v and X_b on (X_v, Z_v) can represent the UAV vertical flight path angle Γ.

The lateral and longitudinal LOS rates can be estimated as follows:

$$\dot{\chi}(t_k) = \frac{\chi(t_k) - \chi(t_{k-1})}{\Delta t} \tag{13}$$

$$\dot{\gamma}(t_k) = \frac{\gamma(t_k) - \gamma(t_{k-1})}{\Delta t} \tag{14}$$

where Δt is the image sampling step.

Equations (13) and (14) provide the basis for an ideal vision-based pro-Nav intercept law, in which an interceptor will try to collide with a target with a constant velocity and heading angle.

3. Proportional Navigation Law – Weighting Function Approach

2D Pro-Nav guidance laws are be reversely applied to avoid collisions by detecting the impact threat and deriving commands away from the ideal interception command. To achieve autonomous collision avoidance, the system needs to satisfy the following properties:

1. Being able to detect the collisions.
2. The guidance should consider every target that is a potential collision threat.
3. Have the ability to determine the maneuver type: lateral or longitudinal.

3.1. Flight Path Selection

To complete a collision avoidance maneuver, a very important step is flight path selection. In other words, the CAS has to determine whether a lateral or vertical maneuver should be performed. Based on the two kinds of LOS projected angles, we can also decouple a 3D maneuver into two 2D maneuvers with respect to the ψ and Γ angles, which are rolling maneuvers and pitching maneuvers.

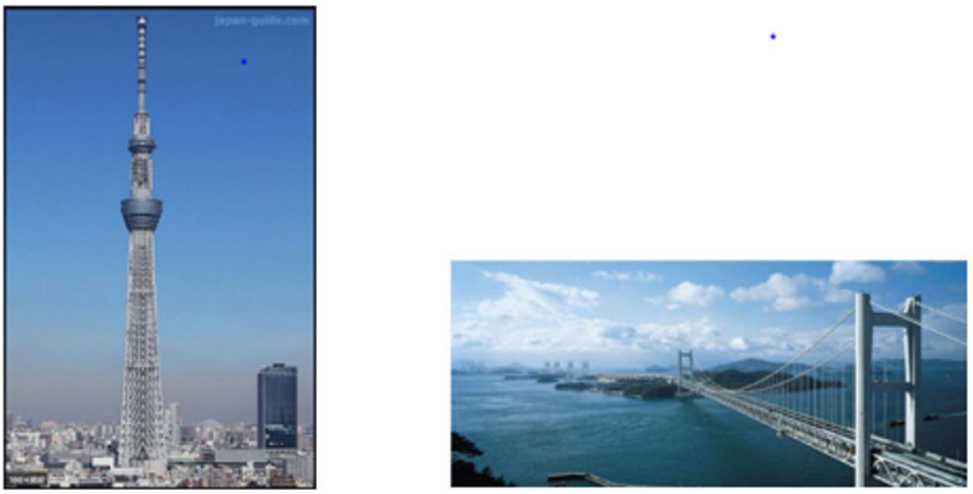

Figure 3.1 Typical building obstacles in urban environments (‘Tower’, n.d.).

Figure 3.1 shows two typical scenarios in urban environments respectively. The left side of Figure 3.1 represents a "tower" and the right side represents a "bridge". In both cases, the object is represented in terms of a number of discrete feature points as might be identified in the image plane. It can be noticed that the sum of horizontal LOS angles of the feature points is less than the sum of vertical LOS angles ($\sum_i^N \boldsymbol{\chi_i} < \sum_i^N \boldsymbol{\gamma_i}$) for the tower; in other words, it is "tall" which suggests that a lateral avoidance maneuver would be optimal. For the bridge, the sum of horizontal LOS angles is greater than the sum of vertical LOS angles ($\sum_i^N \boldsymbol{\chi_i} > \sum_i^N \boldsymbol{\gamma_i}$) suggesting that a vertical maneuver would be a better option. In a real 3D urban environment, UAVs should try to avoid obstacles with a safety margin; we do not want to be too close for an optimal trajectory and minimum energy cost. Therefore, Figure 3.2 provides the ideal avoidance maneuvers for these scenarios.

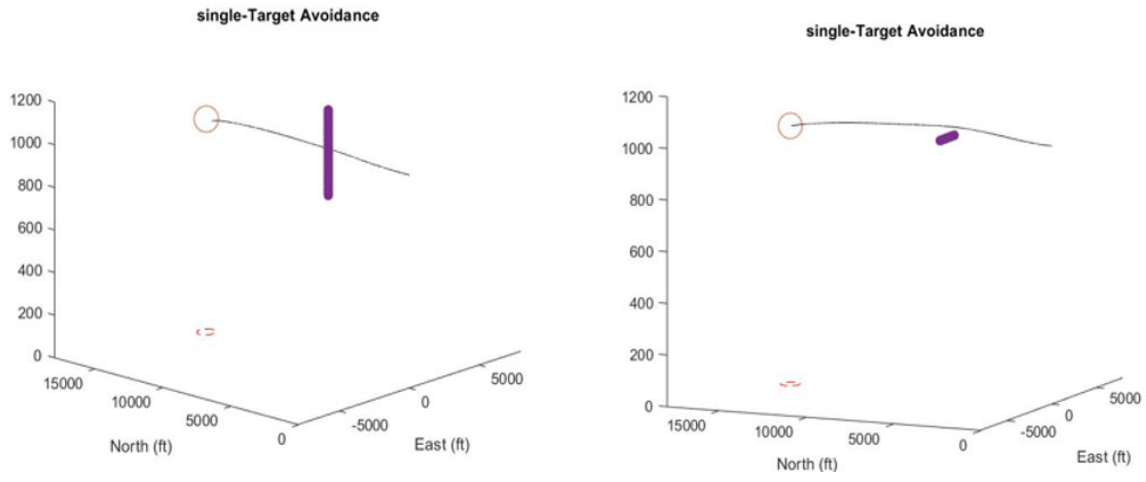

Figure 3.2 Ideal optimal Pro-Nav avoidance trajectories.

Figure 3.2 provides the optimal trajectories, in which the UAV bypasses the tower from one side and crosses the bridge from above. Thus, those two scenarios provide a

logic that, if $\sum_i^N \boldsymbol{\chi_i} < \sum_i^N \boldsymbol{\gamma_i}$ the UAV will perform a lateral maneuver to bypass obstacles; on the other hand, if $\sum_i^N \boldsymbol{\chi_i} > \sum_i^N \boldsymbol{\gamma_i}$ the UAV will generate a vertical maneuver to cross the obstacles. The basic concept of this last result is decomposing a complex 3D maneuver into two 2D maneuvers, which also conforms to the 3D Pro-Nav logic to decompose the 3D scenario into two 2D scenarios.

3.2. 3D Pro-Nav Law

To apply the 2D equation to a 3D environment, a 3D maneuver has to be regarded as a composition of Pro-Nav guidance laws in maneuvers on different planes. In other words, a 3D command angle also needs to be decomposed to multiple projection angles on the planes similar to the LOS angles. Thus, in this thesis, a 3D maneuver command is orthogonally projected onto the (X_v, Y_v) and (X_v, Z_v) planes to become ψ_{com} and Γ_{com}. Then expanding the Pro-Nav law to the 3D environment to generate commanded heading and vertical flight path angles, we obtain:

$$\dot{\psi}_{\mathrm{com}} = \mathrm{N_H}\dot{\chi} \tag{15}$$

$$\dot{\Gamma}_{\mathrm{com}} = \mathrm{N_V}\dot{\gamma} \tag{16}$$

where ψ_{com} and Γ_{com} represent the ideal interception commanded heading angle and flight path angle. The proportional gains $\mathrm{N_H}$ and $\mathrm{N_V}$ are constants that can be tuned to achieve an optimal intercept trajectory.

$$\Delta\psi_{com} = N_H\dot{\chi}\Delta t \tag{17}$$

$$\Delta\Gamma_{com} = N_V\dot{\gamma}\Delta t \tag{18}$$

According to Equations (15) and (16), the commanded angle change within a time step is provided in Equations (17) and (18). The time step Δt represents the sample step, or frame rate, of the camera sensor.

Recall the baseline Pro-Nav guidance law defined previously. Substituting Equation (1) into Equation (17) simplifies the control law further, resulting in:

$$\Delta\phi_{com} = K_{\phi_{com}} N_H \dot{\chi} \Delta t \tag{19}$$

$$\Delta\Gamma_{com} = N_V \dot{\gamma} \Delta t \tag{20}$$

Equations (19) and (20) show that the Pro-Nav intercept commands for a fixed-wing UAV can be executed by rolling and pitching maneuvers; the command angles are proportional to the lateral and vertical LOS angles respectively. Rolling and pitching maneuvers are independent with each other at least to first order, such that when the target and UAV are in the same plane, only one maneuver is needed, which can greatly reduce calculation and energy cost.

3.3. Collision Avoidance: Weighting Function Method

The last section provides the baseline guidance law for the Pro-Nav interception. While forcing a UAV to intercept a target is not the goal of this work, knowledge of the optimal intercept angles provides information that can be used for collision avoidance. For example, when the UAV is on a collision course, the LOS rates to the target should be zero; meanwhile the ideal intercept command angle should be also zero. Therefore, computing the intercept angles provides knowledge of maneuvers that will result in a collision. The entire UAV trajectory includes two main components – avoiding obstacles and approaching a desired waypoint. Thus, the overall guidance law needs to balance the maneuver between avoiding multiple collision threats while still achieving the mission goal.

To consider all the potential collision threats and the desired waypoint to obtain an optimal trajectory, a cost function approach is implemented that consists of every target

(waypoint and obstacles). These targets are given different weights based on the Pro-Nav factors, LOS rates and Pro-Nav constants.

Before deriving the cost function, the collision requirements are modified to account for uncertainty in the LOS rates. Define a threshold $\epsilon > 0$, with ϵ chosen to be a very small value, and $\dot{\chi}$ and $\dot{\gamma}$ are the obstacles' LOS rates with respect to the UAV. According to the Pro-Nav law, a collision is likely to occur when both $|\dot{\chi}|$ and $|\dot{\gamma}| < \epsilon$.

To derive the UAV guidance commands to avoid each collision threat while approaching the desired waypoint along an optimal trajectory, the baseline guidance law is applied in an optimization based approach as follows:

$$\Delta\phi_{com} = argmin(J_1(\Delta\phi_{com}, \dot{\chi_{wp}}, \dot{\chi_{obi}})) \tag{21}$$

$$\Delta\Gamma_{com} = argmin(J_2(\Delta\Gamma_{com}, \dot{\gamma_{wp}}, \dot{\gamma_{obi}})) \tag{22}$$

where $\dot{\chi_{wp}}$ and $\dot{\gamma_{wp}}$ are the LOS rates to the goal waypoint. $\{\dot{\chi_{obi}}\}_{i=1}^{N}$ and $\{\dot{\gamma_{obi}}\}_{i=1}^{N}$ are the LOS rates to each collision threat.

Then the lateral and vertical maneuver cost functions are defined as:

$$J_1 = W_{wpH}\left(\Delta\phi_{com} - N_H\dot{\chi_{wp}}\Delta t\right)^2 + W_{obH}\sum_{i}^{N} f\left(\Delta\phi_{com},\ \dot{\chi_{wpi}}, \Delta t\right)^2 \tag{23}$$

$$J_2 = W_{wpV}\left(\Delta\Gamma_{com} - N_H\dot{\gamma_{wp}}\Delta t\right)^2 + W_{obV}\sum_{i}^{N} f\left(\Delta\Gamma_{com},\ \dot{\gamma_{wpi}}, \Delta t\right)^2 \tag{24}$$

In these cost functions,

$$W_{wpH}, W_{wpV} > 0 \tag{25}$$

$$\begin{cases} W_{obH} > 0,\ \dot{\chi} \le \epsilon \cup \dot{\gamma} \le \epsilon \cup \sum_{i}^{N}\chi_i < \sum_{i}^{N}\gamma_i \\ W_{obH} = 0,\ \dot{\chi} > \epsilon \cap \dot{\gamma} > \epsilon \cap \sum_{i}^{N}\chi_i < \sum_{i}^{N}\gamma_i \end{cases} \tag{26}$$

$$\begin{cases} W_{obV} > 0,\ \dot{\chi} \leq \epsilon \cup \dot{\gamma} \leq \epsilon \cup \sum_{i}^{N} \chi_i \geq \sum_{i}^{N} \gamma_i \\ W_{obV} = 0,\ \ \dot{\chi} > \epsilon \cap \dot{\gamma} > \epsilon \cap \sum_{i}^{N} \chi_i \geq \sum_{i}^{N} \gamma_i \end{cases} \tag{27}$$

Equation (23) and (24) are lateral and vertical weighting functions, respectively, and are minimized independently. The goal is to compute $\Delta\phi_{com}$ and $\Delta\Gamma_{com}$ in a specific range by minimizing J_1 and J_2, thus providing an optimal trajectory. In this thesis, J_1 and J_2 are minimized in Matlab by using the function 'fminbnd'. This function can find the minimum of a single-variable function on a fixed interval. For example, x = fminbnd(fun,x1,x2), where x is the command angle, 'fun' is the weighting function and x1, x2 are the design range of the command angle. The two parts of the cost function represent approaching the waypoint and avoiding obstacles by approaching zero respectively to result a minimum value of J. Therefore, setting and tuning different weighting values like W_{wp} and W_{ob} can balance the maneuver between collision avoidance and waypoint navigation.

In this thesis, the waypoint navigation has a fixed format. For example, when $\Delta\phi_{com}$ is closer to the ideal waypoint interception command $N_H\dot{\chi_{wp}}\Delta t$, the first part of the weighting function approaches zero, which means the UAV is on the proper heading to the waypoint, and similarly for the vertical case. On the other hand, several different forms for the avoidance term were considered in this work.

Both $\Delta\phi_{com}$ and $\Delta\Gamma_{com}$ are computed at each time step, but obstacle information is only included for one or the other based on the maneuver type. Since the UAV should approach the waypoint throughout the trajectory, the weights W_{wpH} and W_{wpV} should

always be greater than zero. Then, recalling the flight path selection logic mentioned in the previous chapter, if a pitching maneuver is expected, the weight of the vertical avoidance components should be greater than zero and the lateral avoidance component should be equal to zero. On the contrary, the opposite series occurs when a lateral maneuver is expected. When $\sum_i^N \chi_i = \sum_i^N \gamma_i$, the equations are determined by comparing the vertical maneuver trajectory and the lateral maneuver trajectory in a single point scenario, which will be discussed in the next chapter.

In order to study the application and performance of the guidance law for collision avoidance, two different types of weighting functions are developed and implemented. The specific formulas are developed in the next section.

3.4. Inversion Weighting Function

The inversion weighting cost functions for lateral and vertical guidance are defined as:

$$J_1(\Delta\phi_{com}) = W_{wpH}\left(\Delta\phi_{com} - N_{wpH}\dot{\chi_{wp}}\Delta t\right)^2 + W_{obH}\sum_i^N \frac{1}{\left(\Delta\phi_{com} - N_{obH}\dot{\chi_{obi}}\Delta t\right)^2}$$
$$J_2(\Delta\Gamma_{com}) = W_{wpV}\left(\Delta\Gamma_{com} - N_{wpV}\dot{\gamma_{wp}}\Delta t\right)^2 + W_{obV}\sum_i^N \frac{1}{\left(\Delta\Gamma_{com} - N_{obV}\dot{\gamma_{obi}}\Delta t\right)^2} \quad (28)$$

In Equation (28), the first term in the cost functions encourages guidance angles $\Delta\phi_{com}$ and $\Delta\Gamma_{com}$ to approach a goal waypoint, $\left(\Delta\phi_{com} - N_H\dot{\chi_{obi}}\Delta t\right)^2$ and $\left(\Delta\Gamma_{com} - N_H\dot{\gamma_{obi}}\Delta t\right)^2$ represent approaching obstacles in two directions, where $N_{obH}\dot{\chi_{ob}}\Delta t$ and $N_{obV}\dot{\gamma_{ob}}\Delta t$ are the ideal command angles to intercept the obstacles. In this case, those two terms are implemented in the denominator, which means the second term of the weighting function is repulsive to angles that would result in collisions. In this manner,

the weighting function achieves a balance between waypoint following and collision avoidance maneuvers to obtain an optimal trajectory.

3.5. Virtual Point Weighting Function

An alternative virtual point approach was developed:

$$J_1(\Delta\phi_{com}) = W_{wpH}\left(\Delta\phi_{com} - N_{wpH}\dot{\chi_{wp}}\Delta t\right)^2 + W_{obH}\sum_i^N\left(\Delta\phi_{com} - N_H\dot{\chi_{vi}}\Delta t\right)^2$$
$$J_2(\Delta\Gamma_{com}) = W_{wpV}\left(\Delta\Gamma_{com} - N_H\dot{\gamma_{wp}}\Delta t\right)^2 + W_{obV}\sum_i^N\left(\Delta\Gamma_{com} - N_H\dot{\gamma_{vi}}\Delta t\right)^2 \quad (29)$$

In contrast to the inversion method, virtual point uses a direct method to balance the waypoint following and obstacle avoidance maneuvers. Recalling the definition of the direction LOS vector, a virtual direction LOS vector is defined with a miss distance on the $\hat{y}$ and $\hat{z}$ components, which is expressed in equation (30).

$$\text{los}_V = \begin{bmatrix} \hat{x} \\ \hat{y} + M_y \\ \hat{z} + M_Z \end{bmatrix} = \begin{bmatrix} \hat{x}_V \\ \hat{y}_V \\ \hat{y}_V \end{bmatrix} \quad (30)$$

Then the virtual LOS vector can provide virtual LOS rates $\dot{\chi_v}$ and $\dot{\gamma_v}$, similar to $\Delta\phi_{com}$ and $\Delta\Gamma_{com}$; both the virtual LOS rates, $\dot{\chi_v}$ and $\dot{\gamma_v}$, are computed at each time step. The terms $\left(\Delta\phi_{com} - N_H\dot{\chi_v}\Delta t\right)^2$ and $\left(\Delta\Gamma_{com} - N_H\dot{\gamma_v}\Delta t\right)^2$ represent approaching a virtual waypoint, which is equivalent to an obstacle avoidance maneuver. Therefore, this kind of cost function achieves a balance between the two intercept maneuvers for waypoint and virtual obstacles.

4. SIMULATION RESULTS

The basic logic of avoidance and the weighting function approach have already been derived in the previous chapters. Several different scenarios are implemented in simulation in this chapter to investigate the feasibility and robustness of this guidance and control law.

MATLAB/Simulink is the elemental simulation environment in which the research is implemented in this thesis. The entire UAV model and guidance avoidance system are mostly modeled by Simulink blocks, cooperated with MATLAB coding functions, and called in Simulink.

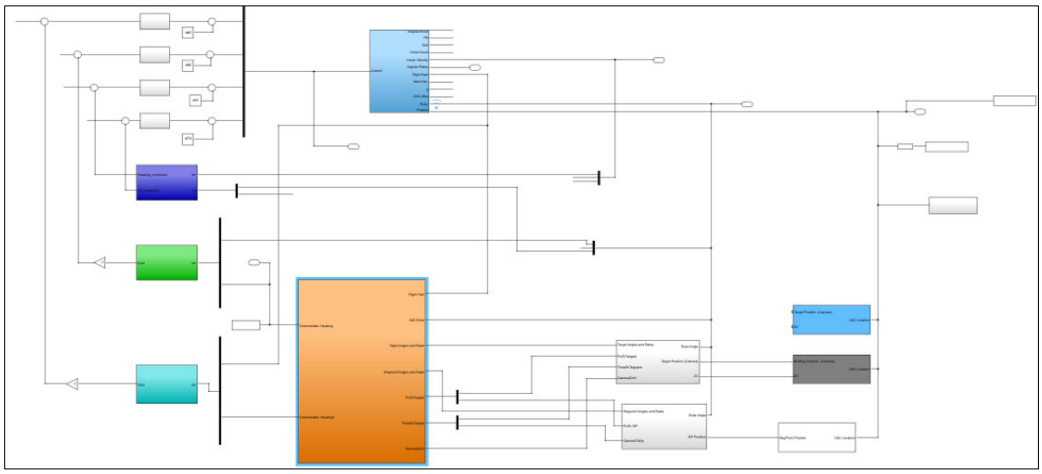

Figure 4.1 Diagram of overall Simulink block configuration.

As shown in Figure 4.1, the UAV is simulated as a 6-dof rigid body and modeled in the UAV dynamics block. The rest of the system is modeled in other main blocks, such as the guidance block, PID controller blocks, and target generating blocks.

The UAV dynamics are modeled using a component build-up method. The UAV dynamics simulation block is shown in Figure 4.2, which includes all coefficients to calculate aerodynamics and 12 aircraft rigid body equations of motion to calculate

translational velocity, angular velocity, attitude (roll, pitch and yaw angles), and inertial position. Although the true UAV dynamics are much more complex, with nonlinear aerodynamic coefficients, the equations of motion employed here are commonly used to model flight dynamics.

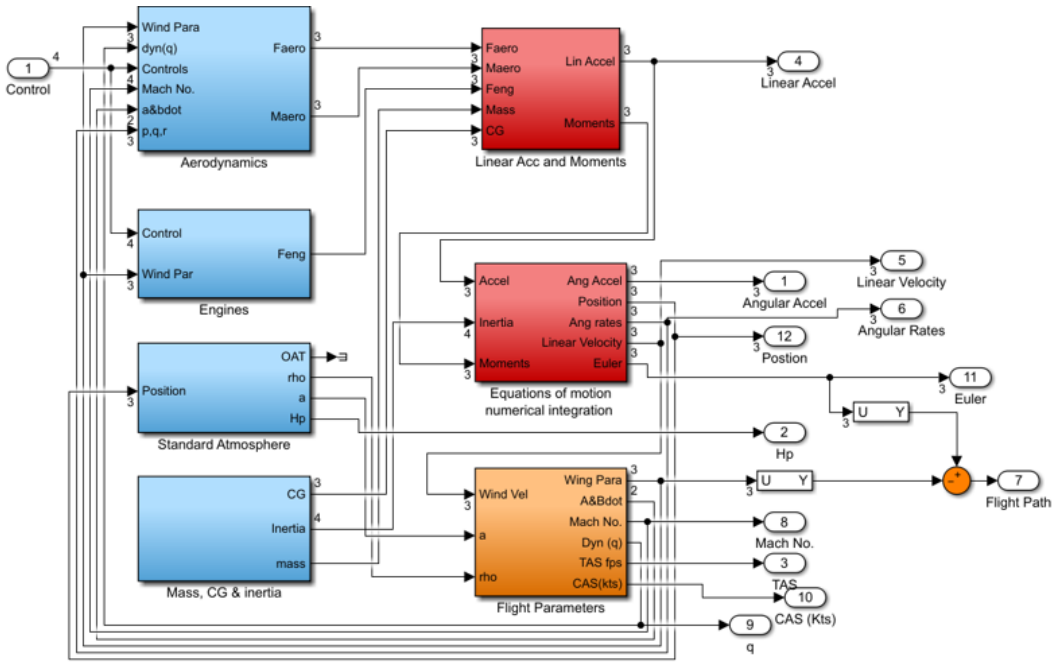

Figure 4.2 Diagram of UAV dynamics block.

4.1. Inversion Weighting Function Simulation Results

In order to have intuitive comparisons of control effort and verify the robustness of the guidance law with the inversion weighting functions, the same UAV initial trim condition, which starts from (0,0,1000) with 0 heading and constant velocity V_0=176ft/s is implemented in all scenarios. The variables in the weighting functions are provided in Table 4.2.

Table 4.1

Coefficients in weighting function

Variables	**Value**
$\boldsymbol{W_{wpH}}$	1
$\boldsymbol{W_{obH}}$	0.4
$\boldsymbol{W_{wpV}}$	1
$\boldsymbol{W_{obV}}$	0.3
$\boldsymbol{N_{wpH}}$	250
$\boldsymbol{N_{obH}}$	9000000
$\boldsymbol{N_{wpV}}$	180
$\boldsymbol{N_{obV}}$	90000000

The UAV performance is evaluated by three metrics: minimum miss distance, time difference and control energy. The minimum miss distance represents the minimum distance between the UAV and obstacle and the time difference represents the difference between the time required to fly to the desired waypoint with and without obstacles. According to the UAV size, the minimum miss distance should be at least greater than the half wingspan 16.7 ft to ensure a safe flight trajectory without impact. The control energy is a nondimensional value, which can be expressed in Equation (31):

$$E = \sum \delta_e^2 + \sum \delta_a^2 + \sum \delta_r^2 + \sum \delta_T^2 \tag{31}$$

where δ_e, δ_a, δ_r are elevator, aileron, and rudder deflection in every time step, δ_T is the thrust change. Therefore, the three metrics of the UAV performance can be evaluated by analyzing the trajectory and energy cost.

4.1.1. Scenario 1: Stationary Obstacle

The static obstacle is set as a point which is located at (5280,0,1000). The trajectory for vertical and lateral maneuvers are provided in Figure 4.3.

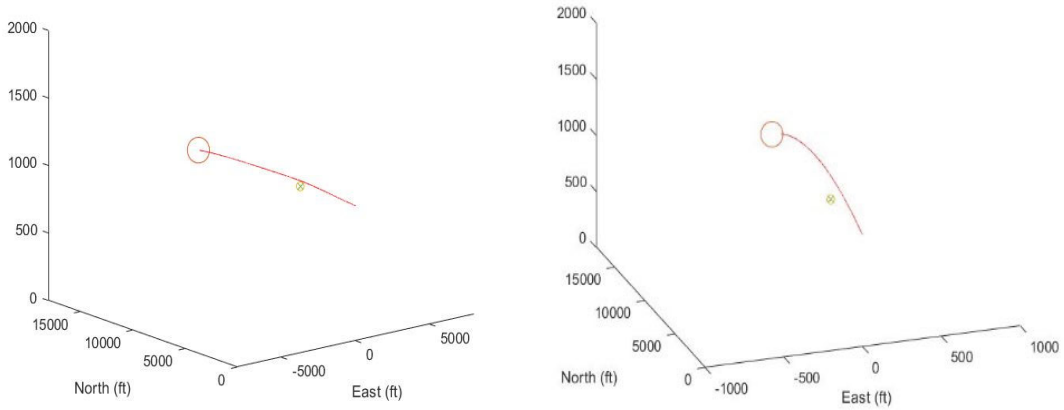

Figure 4.3 Stationary obstacle simulation: avoidance trajectories.

The metrics of the trajectories when the UAV is performing vertical and lateral maneuvers are shown in Table 4.2. The table shows that the vertical maneuver results in a trajectory with a lower minimum miss distance and less control energy cost. Therefore, the weighting function will generate a vertical maneuver command when $\sum_i^N \chi_i = \sum_i^N \gamma_i$.

Table 4.2

Metrics for the single point obstacle scenario trajectories

Metrics	**Vertical Maneuver**	**Lateral Maneuver**	**Unit**
Min Miss Distance	40.08	82.6	ft
Control Energy	4.4944	7.0595	
Time difference	-0.43	0.11	sec

4.1.2. Scenario 2: Moving Obstacles

In this scenario, the obstacle flies starting from a position of (5280, 2640, 1000) with -π heading and constant velocity $0.5V_0$. Therefore, if no avoidance maneuver is performed, a collision will occur at the position (0, 2640, 1000) in the NED frame .

The vertical LOS rates and command angle $\Delta\Gamma_{com}$ are shown in Figure 4.5. According to Figure 4.5, with the LOS rate lower than the threshold value, the command angle is generated and has a rapid change at the beginning to guide the UAV to avoid the obstacle; then the command plot will tend to be flat to guide the UAV to approach the waypoint.

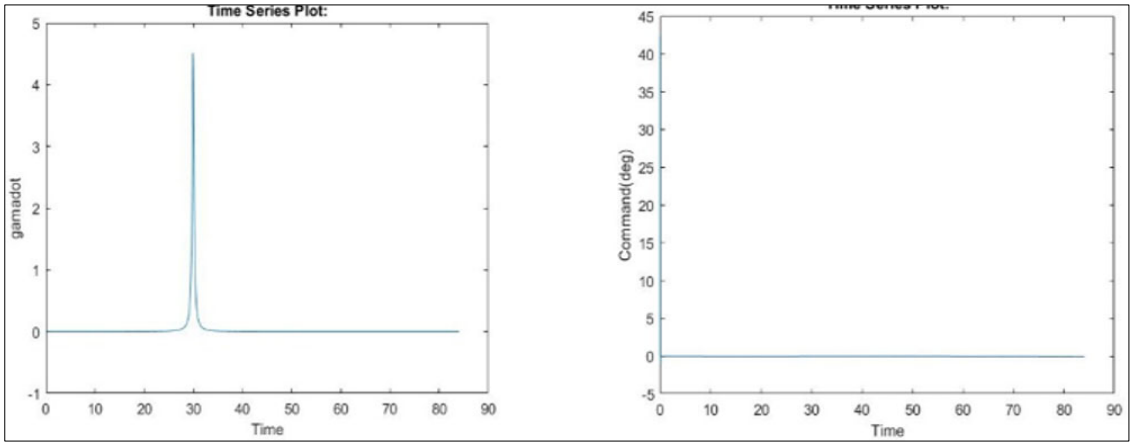

Figure 4.4 LOS rate and command angle in single moving obstacle scenario.

The UAV avoidance maneuver is shown in Figure 4.5. This figure, which represents a 3D view, top view, and side view of the trajectory, verifies the 3D flight path selection logic. After the obstacle is detected, a pitch-up command is generated, followed by a pitch-down command after the obstacle is out of the camera field of view.

Table 4.3 provides several metrics, including the minimum miss distance of 39.9 ft. The miss distance, which is greater than 20 ft, can be considered as a safe range depending on the size of the UAV. It is also worth mentioning that all the metrics depend on the coefficients in the weighting functions; these effects will be discussed in the next chapter.

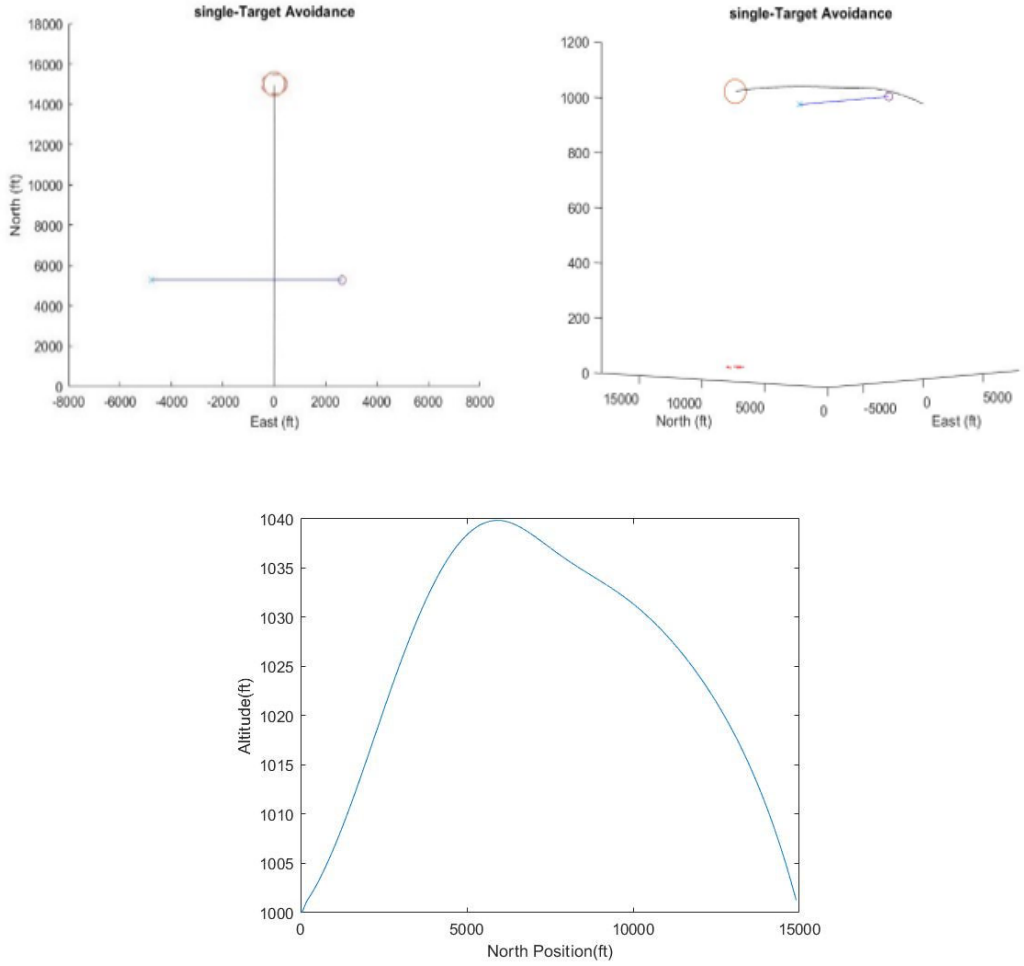

Figure 4.5 Single moving obstacle simulation: avoidance trajectory.

Table 4.3

Metrics for the single moving obstacle scenario

Metrics	**Value**	**Unit**
Min Miss Distance	39.9	ft
Control Energy	4.48	
Time Difference	-0.43	sec

The next section discusses the feasibility of the weighting function approach when dealing with multiple obstacles. In Figure 4.6, the first obstacle flies starting from a position of (5280, 2640, 1000) with -180° heading and constant velocity $0.5V_0$; the

second obstacle flies starting from (5280, 10560, 1000) with -180° heading and constant velocity $0.5V_0$. Thus, if no avoidance maneuver is performed, a collision will occur at (0, 2640, 1000).

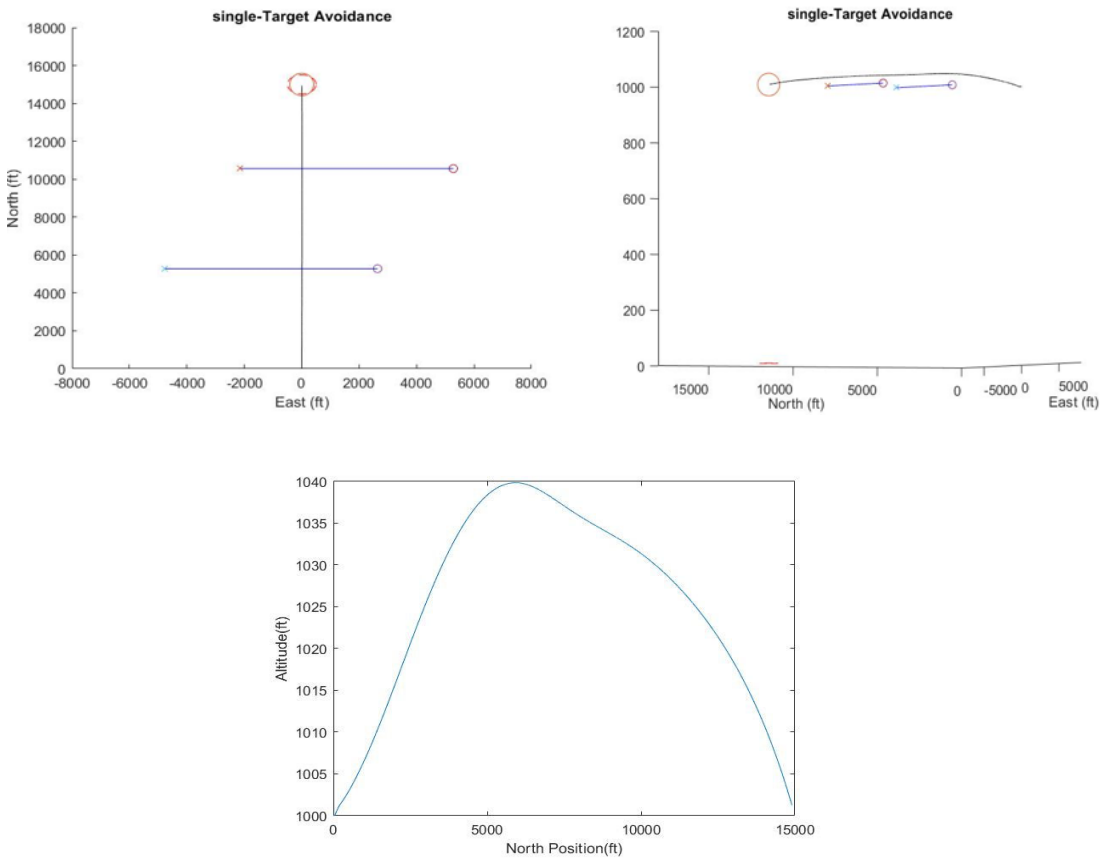

Figure 4.6 Multiple obstacle simulation trajectory

Figure 4.6 shows that the guidance and control law is capable of avoiding multiple obstacles and following the flight path selection logic. By analyzing the metrics in Table 4.4, the control law generates a similar trajectory with the same miss distance and control energy compared to the single obstacle scenario. When the two targets are in the same horizontal plane, the UAV can bypass both with the same flight path and energy cost because, when the first target has been avoided, the second obstacle is no longer a

collision threat. At this point, there is no need to perform an avoidance maneuver for the second obstacle, which shows that the weighting function method can optimize control energy cost while considering all targets.

Table 4.4

Metrics for the multiple moving obstacle scenario

Metrics	**Value**	**Unit**
Min Miss Distance	39.9	ft
Control Energy	4.48	
Time Difference	-0.43	sec

4.1.3. Static Obstacles

The initial implementation of the inversion weighting functions verifies the avoidance capability for single and multiple moving obstacles. This chapter will further investigate the effort in more complex static obstacle scenarios. In contrast to the high velocity, smaller size dynamic obstacles, static obstacles have zero speed and can be quite large, which means they might be detected by a vision system as a large quantity of feature points rather than a single point. Static obstacles can represent typical landmarks in real urban environments. Thus, these cases may provide more challenges to the collision avoidance system but also represent realistic scenarios where the CAS is implemented in real urban environments.

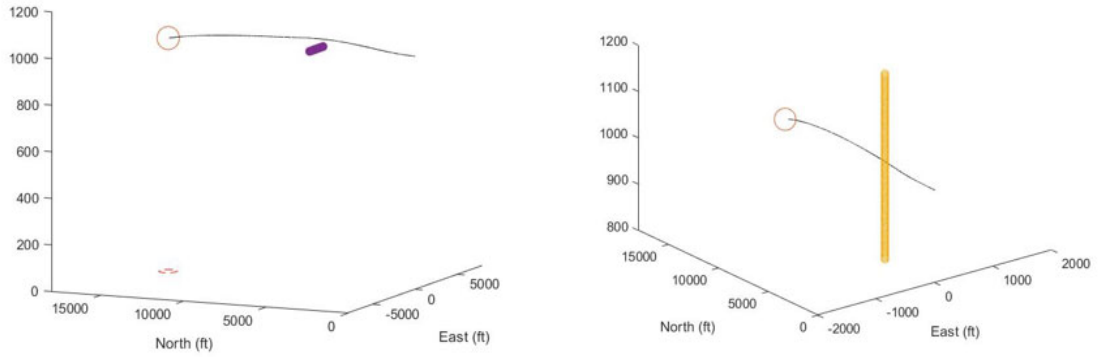

Figure 4.7 Static urban obstacle simulation trajectories.

Figure 4.7 models the static "bridge" and "tower" scenarios respectively. The "bridge" is composed of 128 feature points from -800ft to 800ft in the east direction at 1000ft altitude, and the "tower" is composed of 128 feature points from 800ft to 1200ft altitude at 6000ft north location. Figure 4.7 shows that the UAV accomplishes obstacle avoidance maneuvers (vertical and lateral maneuvers) respectively with different obstacle shapes and sizes while demonstrating automatic flight path selection. Table 4.5 shows the metrics for those two scenarios; both two maneuvers obtained a safe minimum miss distance within a short total time of around 30 sec.

Table 4.5

Metrics for the static urban obstacle scenarios

Metrics	**Bridge**	**Tower**	**Unit**
Min Miss Distance	44.7	197.6	ft
Control Energy	5.4671	21.4505	
Time Difference	-0.48	-0.32	Sec

4.1.4. Complex Scenarios and Combination Obstacle

To further verify obstacle avoidance capabilities, more complex scenarios are implemented to represent more realistic urban obstacles. The first scenario, which

represents a combination of the tower and bridge obstacles, confirms the reliability of the flight path selection logic. The simulation trajectory is shown in Figure 4.8, which illustrates a successful avoidance trajectory with a pitch maneuver followed by a roll maneuver. The control inputs and Euler angles are shown in Figures 4.9 and 4.10.

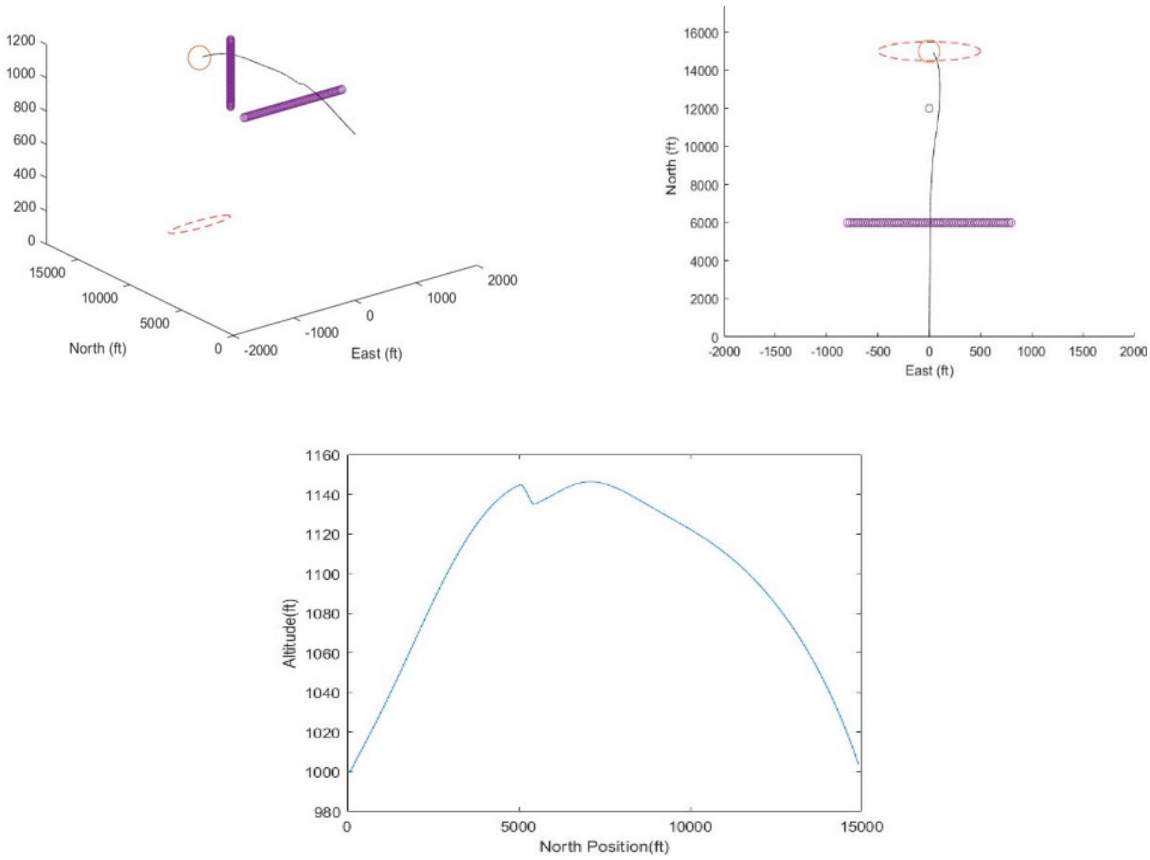

Figure 4.8 Complex obstacle simulation trajectory.

The metrics are shown in Table 4.6; the results show that the flight has a safe minimum miss distance greater than the half wingspan and a flight time shorter than the no obstacle case. The control energy has a significant increase compared to the simple static obstacle scenario, which might be caused by the altitude increase in the maneuver.

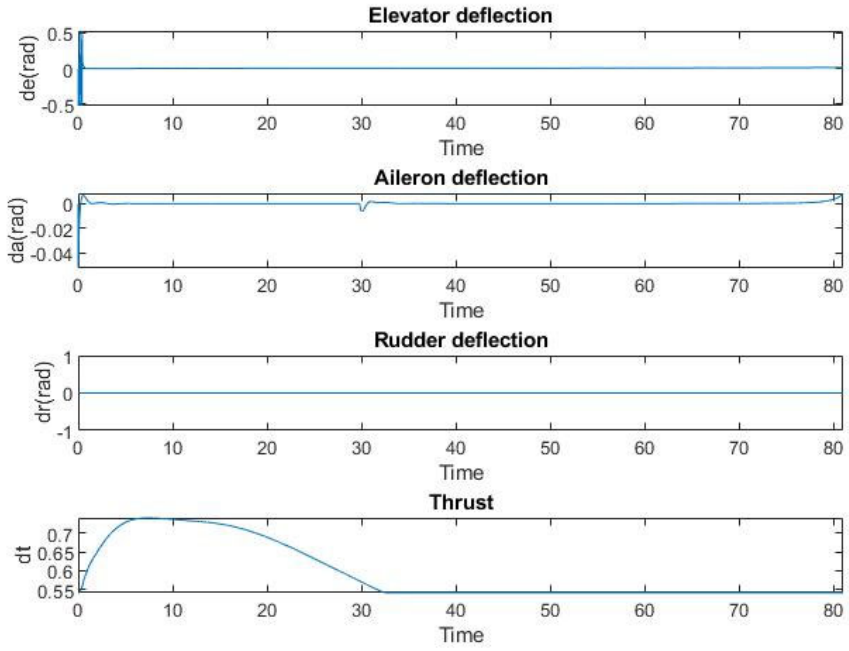

Figure 4.9 UAV control inputs for the complex scenario.

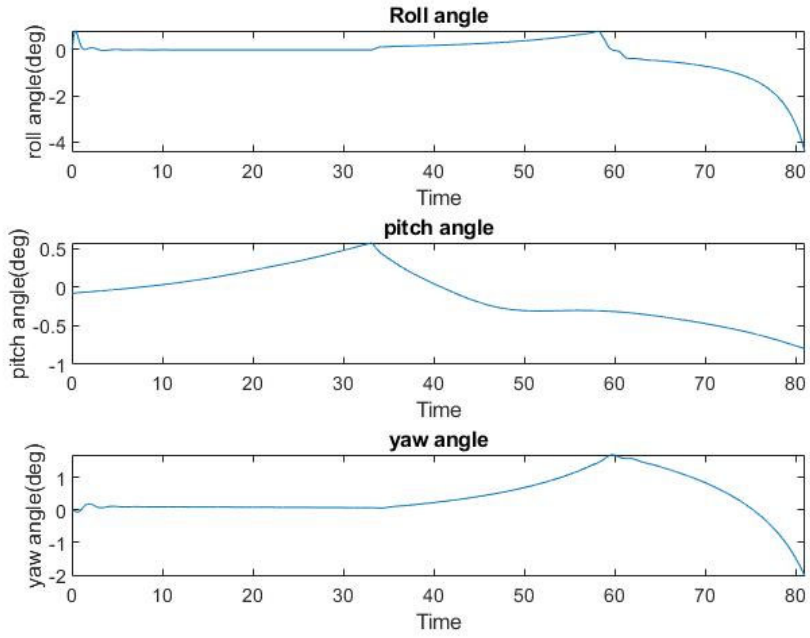

Figure 4.10 UAV Euler angles for the complex scenario.

Table 4.6

Metrics for the combined obstacle scenario

Metrics	**Value**	**Unit**
Min Miss Distance	35.9	ft
Control Energy	75.081	
Time Difference	-1.78	sec

It is worth mentioning that, according to the top and side views in Figure 4.8 b and c, the two different maneuvers are completed separately and independently. The overall trajectory can be regarded as a 2D maneuver in each view plane; thus, when a target is detected, the avoidance maneuver type is chosen based on the shape of the obstacles, which further confirms the feasibility and reliability of the flight path selection logic.

The second scenario further confirms the robustness of the weighting function method. This case includes multiple “towers” to simulate a curved street between two walls of an urban canyon.

In this scenario, the UAV is flying towards a different waypoint, which is located at (9000, -1500, 1000) in the NED frame. Therefore, to approach the waypoint, a left turn, or negative roll is required. The UAV must also avoid the simulated buildings in the urban canyon, representing a challenging scenario. In this case, the UAV selects a trajectory to pass through the curved urban canyon by performing lateral rolling maneuvers, as shown in Figure 4.11.

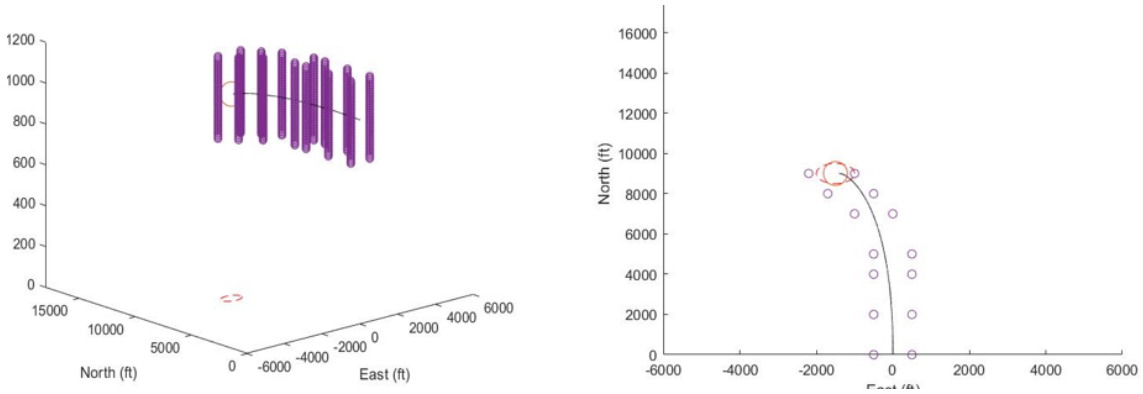

Figure 4.11 Urban canyon obstacle simulation trajectory.

Table 4.7 presents a sufficient miss distance to obstacles to ensure a flight trajectory in the middle of the curved urban canyon. However, due to the more aggressive maneuver and larger quantity of feature points, the control energy and total flight time have substantially increased compared to simpler scenarios.

Table 4.7

Metrics for the urban canyon obstacle scenario

Metrics	**Value**	**Unit**
Min Miss Distance	157.6589	ft
Control Energy	134.8507	
Time Difference	-0.08	sec

In conclusion, the system using the inversion weighting function can select an appropriate flight path according to different types of obstacles. This CAS system has feasibility and robustness when the UAV is facing complex obstacle environments.

4.2. Virtual Point Weighting Function Simulation Results

Since the inversion weighting functions' feasibility and robustness have been verified, this section will discuss the same properties by simulating the same scenarios for

the virtual point approach. The coefficients selected for the virtual point method are shown in Table 4.8.

Table 4.8

Coefficients for virtual point weighting function

Variables	Value
W_{wpH}	1
W_{obH}	0.4
W_{wpV}	1
W_{obV}	0.8
N_{wpH}	250
N_{obH}	70000
N_{wpV}	180
N_{obV}	8000

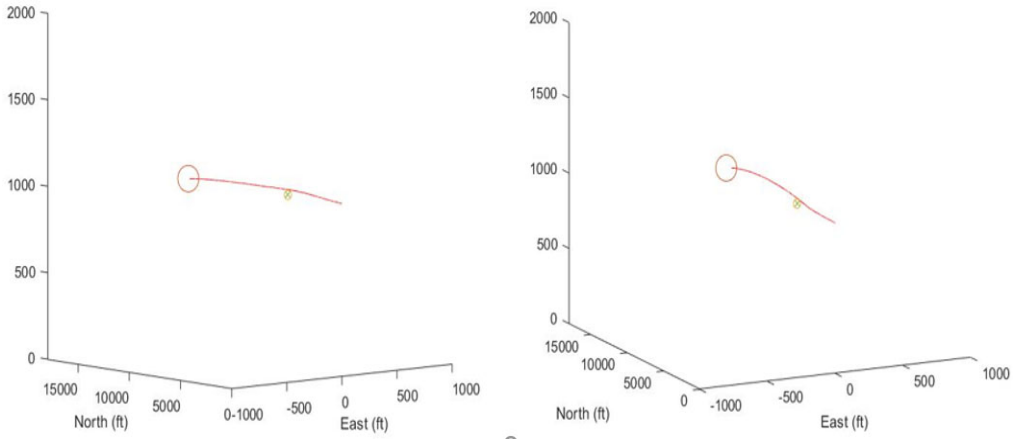

Figure 4.12 Stationary obstacle avoidance trajectory for virtual point method

A static obstacle is first set as a point which is located at (5280, 0, 1000). The trajectories for vertical and lateral maneuvers are provided in Figure 4.12. The metrics of the trajectories when the UAV is performing vertical and lateral maneuvers are shown in Table 4.9. The table shows that the vertical maneuver results in a trajectory with a shorter minimum miss distance, less control energy cost and shorter time to finish the flight.

Therefore, the weighting function will generate a vertical maneuver command when $\sum_i^N \chi_i = \sum_i^N \gamma_i$.

Table 4.9

Virtual point method: Metrics for the single point obstacle scenario trajectory

Metrics	**Vertical Maneuver**	**Lateral Maneuver**	**Unit**
Min Miss Distance	29.7	44.01	ft
Control Energy	2.4062	3.02	
Time difference	-0.33	0.05	sec

4.2.1. Moving Obstacles

As shown in Figures 4.13 and 4.14, the obstacles start from (5280, 2640, 1000) and (5280, 10560, 1000) with -180º heading and constant velocity $0.5V_0$. Similar to using inversion weighting functions, implementing virtual point weighting functions can also result in an ideal trend of avoidance, which is a similar pitch-up and pitch-down maneuver in the North-Down plane only. This trajectory is shown in Figure 4.13, which shows the system reaction of multiple obstacles. Therefore, the avoidance ability and flight path selection ability for the virtual point weighting function are verified.

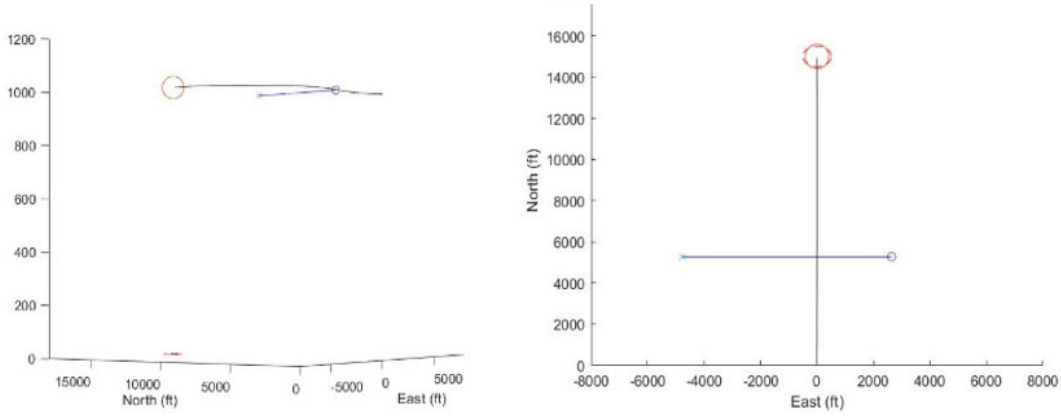

Figure 4.13 Single moving obstacle simulation trajectories for virtual point weighting function.

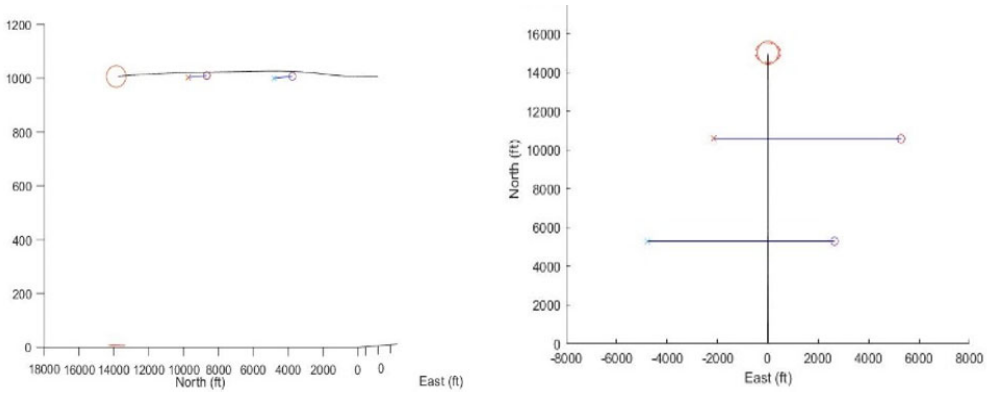

Figure 4.14 Multiple moving obstacle simulation trajectory for virtual point weighting function.

Table 4.10 provides several metrics for this scenario, including a 21.5 ft minimum miss distance. Compared with the same scenarios in the previous chapter, the UAV avoids the same obstacle with smaller miss distance and less energy cost but a longer maneuver time.

Table 4.10

Metrics for moving obstacle scenario, virtual point method

Metrics	**Single**	**Multiple**	**Unit**
Min Miss Distance	21.7	21.7	ft
Control Energy	1.3556	1.3556	
Time Difference	-0.2	-0.24	sec

The vertical LOS rates and command angle $\Delta\Gamma_{com}$ are shown in Figure 4.15. The command angle is also generated since the beginning and increases when the UAV gets closer to the obstacle, then decreases to drive the UAV to approach the waypoint. Compared to the inversion weighting function, the virtual point method reacts slower than the inversion method and results in a smoother command angle plot, which does not have a significant change.

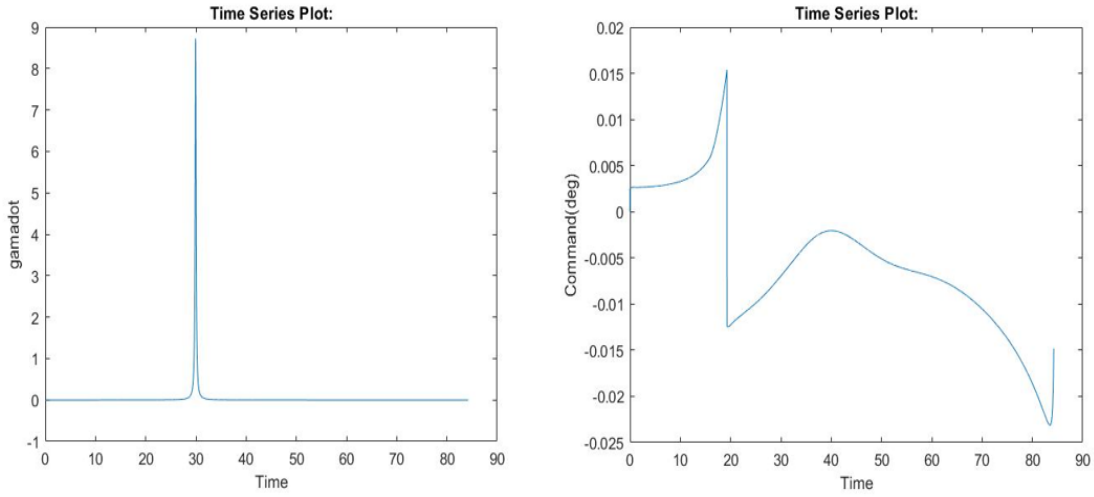

Figure 4.15 LOS rate and command angle in single moving obstacle scenario for virtual point method.

4.2.2. Static Obstacles

After verifying avoidance for small moving obstacles, the virtual point function method is applied to more complex static obstacles. Virtual point simulation trajectories for the bridge and tower obstacles are shown in Figure 4.16.

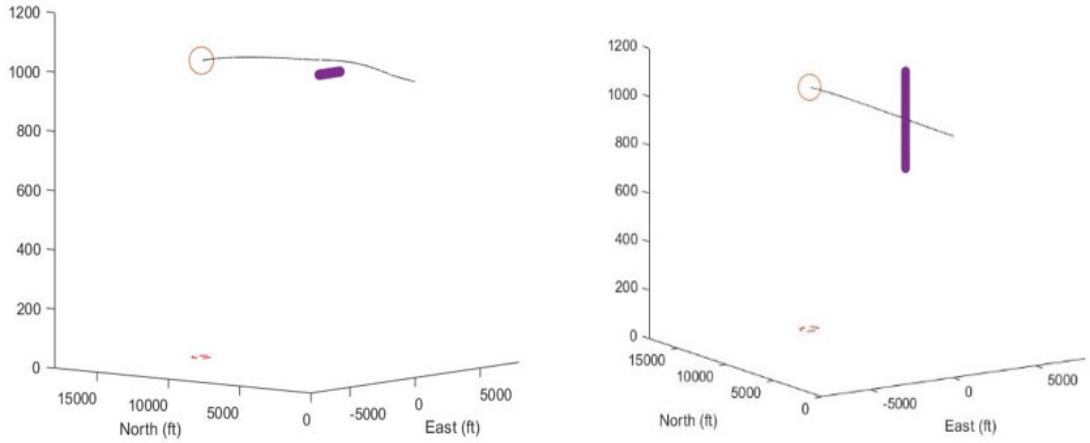

Figure 4.16 Static obstacle simulation trajectories for virtual point functions.

Similar to use the inversion functions, the UAV avoids the “bridge” with a pitching maneuver and avoids the “tower” by rolling. This means that the virtual point weighting functions completed with flight path selection have the ability to avoid obstacles with different shapes and sizes while selecting lateral or vertical flight maneuvers automatically.

Table 4.11

Metrics for the static obstacle scenarios using virtual point method

Metrics	**Bridge**	**Tower**	**Unit**
Min Miss Distance	43.5	91.5	ft
Control Energy	5.94	9.59	
Time Difference	-0.52	0.12	sec

As shown in Table 4.11, for each obstacle, the virtual point functions provide trajectories with a safe miss distance. Compared with the inversion function simulation results, the trajectory with the “bridge” obstacle is finished with a smaller miss distance and a shorter maneuver time but require a little more energy cost; the trajectory with the “tower” obstacle is finished with a much smaller miss distance and a shorter maneuver time and requires much less energy cost. These results imply that the virtual point functions provide more optimal trajectories than the inversion method.

4.2.3. Complex Scenarios and Combination Obstacles of Virtual Point Function

In order to further verify the feasibility and robustness of the virtual point method, the same complex scenarios are implemented to represent more realistic urban obstacles. In this case, the combined “tower” and “bridge” obstacle scenario is considered.

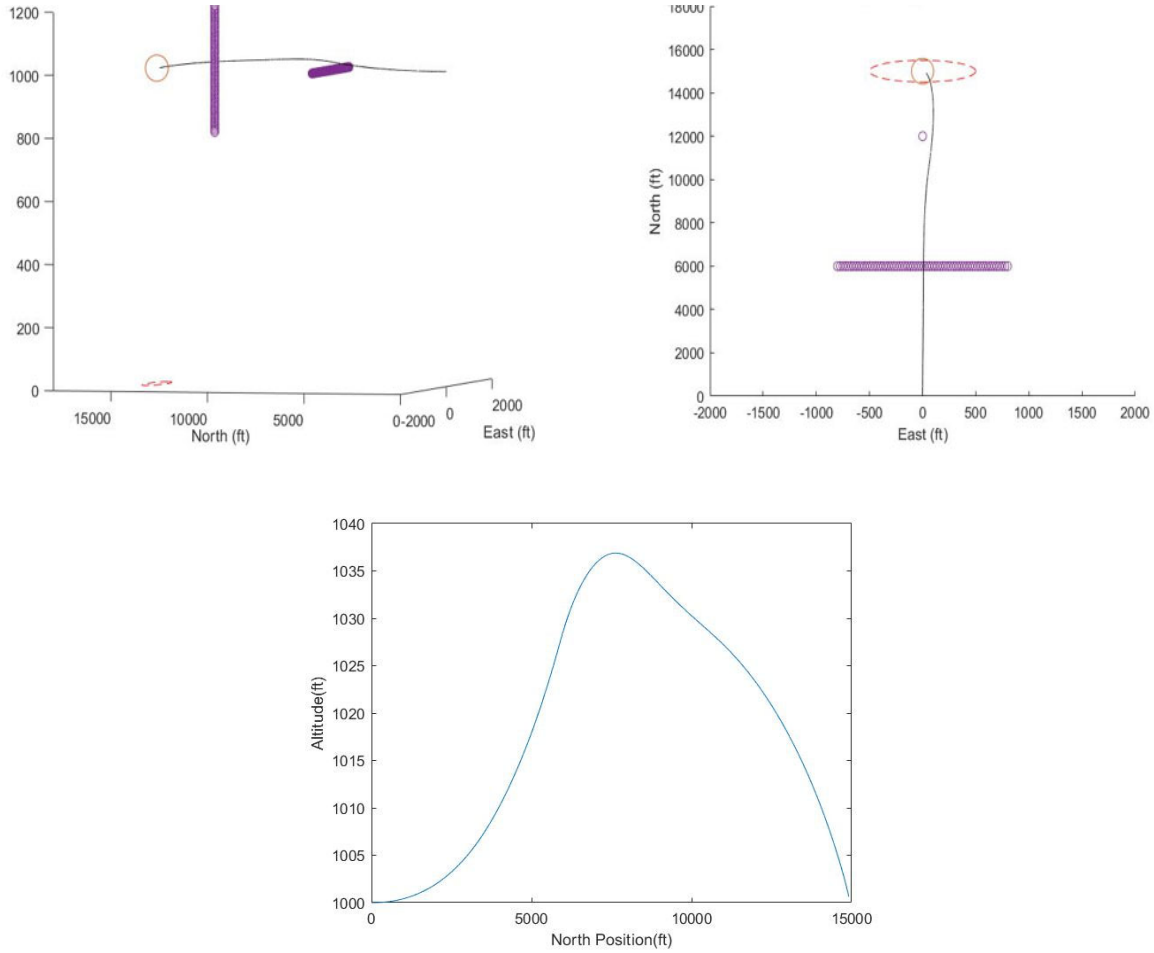

Figure 4.17 Complex obstacles simulation trajectory for virtual point functions.

The scenario with the combined tower and bridge confirms the reliability of the flight path selection ability of the virtual point functions. A successful simulation trajectory is provided in Figure 4.17. The UAV control inputs and Euler angles are shown in Figures 4.18 and 4.19, further confirming the flight trajectory. Compared to the inversion weighting function, the virtual point weighting function results in smoother plots for thrust and elevator deflection, which shows the reaction delay for the virtual point method.

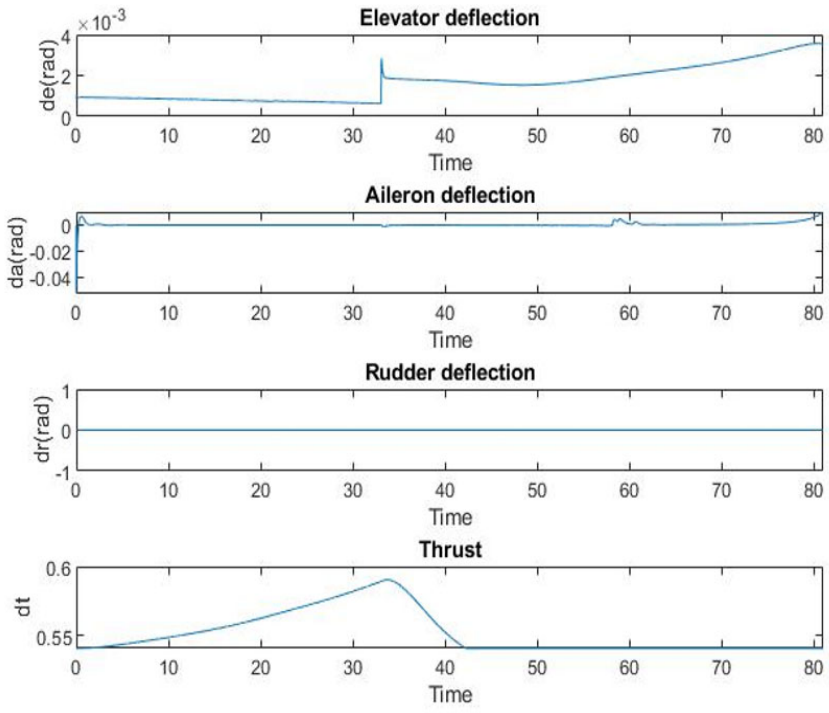

Figure 4.18 UAV control inputs in the complex scenario for virtual point method.

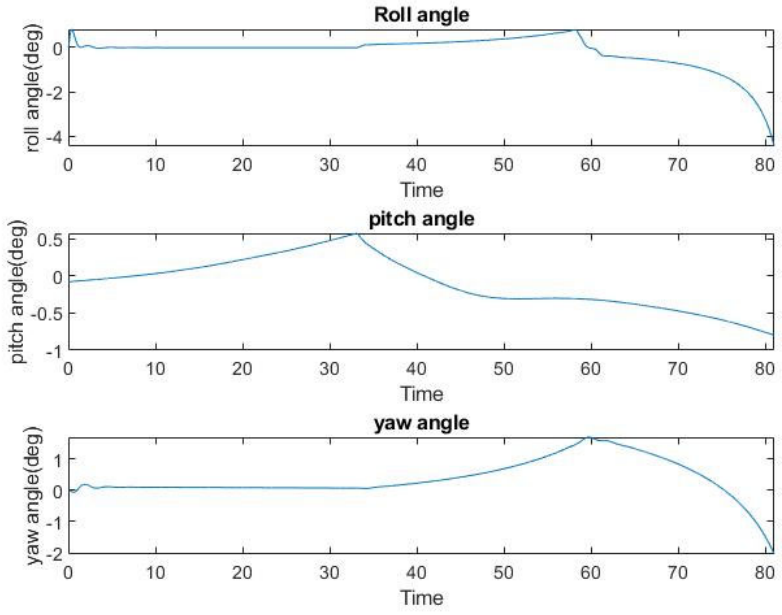

Figure 4.19 UAV Euler angles for the complex scenario for virtual point method.

Table 4.12

Metrics for the combined obstacles scenario

Metrics	**Bridge**	**Unit**
Min Miss Distance	28.93	Ft
Control Energy	6.26	
Time Difference	-0.34	Sec

As shown in Table 4.12, the minimum miss distance is 28.93 ft, which is greater than the 0.5 wingspan; thus, the avoidance is safe and successful. According to the top view and side view, the 3D trajectory is completely decoupled into a pitching maneuver and rolling maneuver by projecting onto North-Down and North-East planes, respectively. Although both types of maneuver can drive the UAV to avoid the combined obstacle, in contrast to the inversion function simulation trajectory, there is no spike in the virtual point function simulation trajectory, especially on the North-East projection. Using the inversion functions, the UAV has a tendency to maneuver back to the waypoint altitude after avoiding the first "building", which does not occur in the inversion method simulation result. Thus, the second type of weighting function is more conducive smoother trajectories.

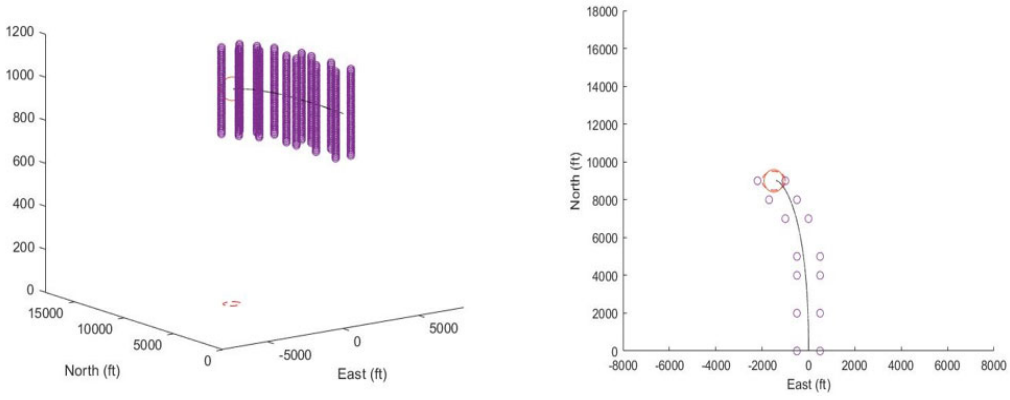

Figure 4.20 Curved urban canyon simulation trajectories for virtual point functions.

The last scenario implemented for the virtual point functions is the same curved urban canyon path. As shown in Figure 4.20, like the inversion function simulation result, the UAV opportunely avoids the obstacles on the sides of the path and approaches the waypoint.

The metrics for this scenario are shown in Table 4.13. Compared with the inversion function simulation results, virtual point functions generate a very similar trajectory. The miss distance and the total time in the two scenarios are basically the same, while the virtual point functions can reduce the control energy cost by approximately 12%.

Table 4.13

Metrics for the virtual point functions, urban canyon obstacle scenario

Metrics	**Value**	**Unit**
Min Miss Distance	156.85	ft
Control Energy	117.83	
Time Difference	-0.11	sec

4.3. Comparison Conclusion

After implementing six different scenarios using both the inversion and virtual point weighting functions, the commands of the two different methods in a single moving obstacle scenario are shown in Figure 4.21. The trajectory metrics of each case are collected in the Table 4.14.

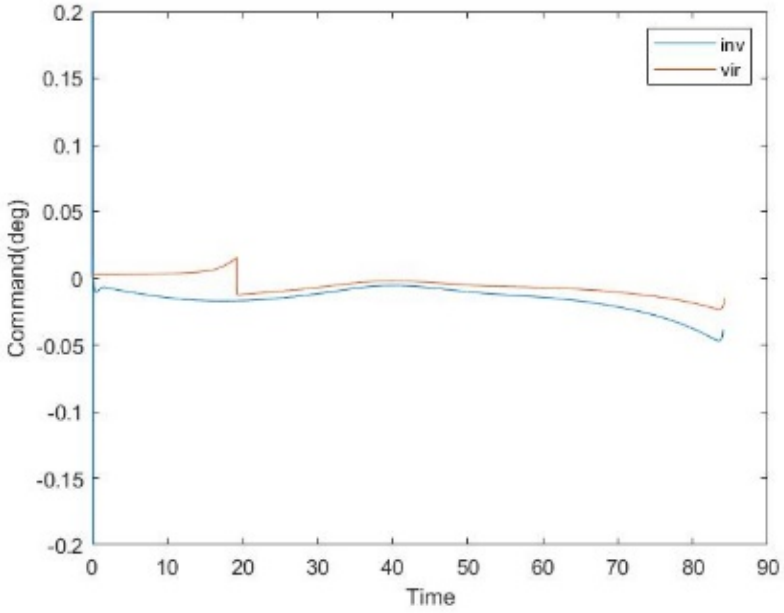

Figure 4.21 Command for inversion and virtual point functions in single moving obstacle scenario.

Table 4.14

Trajectory metrics for each obstacle scenario for inversion and virtual point method

	Min Miss Distance (ft)		**Control Energy (ft)**		**Time Difference (ft)**	
	Inv	**Vir**	**Inv**	**Vir**	**Inv**	**Vir**
Moving	39. 9	21.7	4.48	1 .35	-0.43	-0.24
Bridge	44.7	43.5	5.46	5.94	-0.48	-0.52
Tower	198	91.5	21.5	9.59	0.32	0.12
Complex Obstacle	35.9	28.93	75.08	6.26	-1.78	-0.38
Curved Urban Canyon	157.66	156.85	134.85	117.83	-0.08	-0.11

Then we can draw the following conclusions for the properties of the functions:

1. Both weighting functions are capable of making the UAV avoid both moving and static obstacles successfully.

2. Both weighting functions have similar control effort when the UAV is maneuvering in a narrow space.
3. Both weighting functions are verified to be robust and feasible to avoid obstacles of different sizes with autonomic flight path selection. For the complex scenario, both weighting functions result in a safe minimum miss distance; the inversion method has better performance in terms of flight time, but virtual point method results in much less control energy.
4. For vertical maneuvers, both weighting functions result in a similar performance in the bridge scenario. However, although the virtual point method results in a longer flight time, this method can reduce 46% of miss distance and 69% control energy in moving scenarios. For lateral maneuvers, the virtual point method can reduce 53% of miss distance, 56% control energy cost and shorter flight time in the tower obstacle.
5. In general, on the one hand, inversion weighting functions react faster and result in more aggressive maneuvers with larger miss distances and more control energy in most cases. On the other hand, virtual point functions react slower than the inversion method and result in smoother trajectories with smaller miss distances and less energy cost.
6. Although the virtual method results in several slower flights, the time differences between the flights are very small. Therefore, the virtual point method can generate more optimal trajectories.

5. PERFORMANCE ANALYSIS

The previous chapter has verified that the vision based Pro-Nav algorithm using both the inversion and virtual point weighting functions can guide the UAV to accomplish successful avoidances. However, in these scenarios, the coefficients in these functions were selected with minimal tuning. In real implementations, the weighting functions need to be tuned based on UAV size, velocity, and flight environment. In addition, to achieve the optimal maneuvers, the simulations were completed in an ideal environment without noise in the camera measurements. Thus, this chapter will discuss how the coefficients affect the performance of both types of weighting functions and investigate system performance with camera noise.

5.1. Weighting Coefficients Analysis

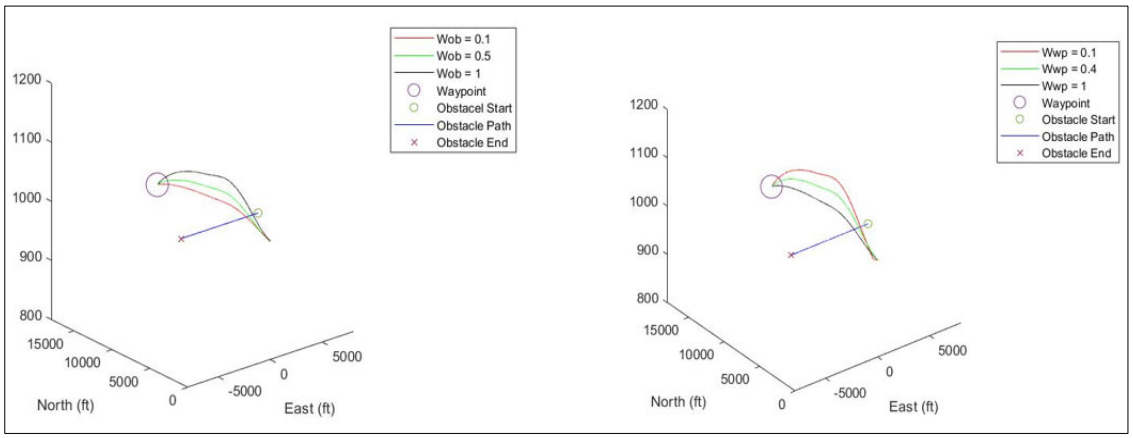

Figure 5.1 Trajectories for different weighting coefficients in the inversion functions.

To investigate the effects of varying the weighting coefficients in the inversion weighting functions, the single moving obstacle scenario is implemented. The effects of the weighting coefficients on the trajectories are shown in Figure 5.1. The left figure shows the effect of varying the avoidance weighting coefficients W_{obV}; the right figure

shows the effect of varying the waypoint guidance coefficient W_{wpV}. The metrics for the resulting trajectories are provided in Table 5.1.

Table 5.1

Metrics for the single moving obstacle scenario using the inversion weighting function

	W_{wpV}=1; N_{wpV}=180; N_{obV}=90000000			W_{obV}=1; N_{wpV}=180; N_{obV}=90000000			
Metrics	**W_{obV}=0.1**	**W_{obV}=0.5**	**W_{obV}=1**	**W_{wpV}=0.1**	**W_{wpV}=0.5**	**W_{wpV}=1**	**Unit**
Min Miss Distance	28.2	43.62	51.39	117.1	61.2056	51.39	ft
Control Energy	2.7047	5.3033	7.396	99.227	35.3456	7.396	
Total Time	-0.3	-0.48	-0.57	-1.93	-0.69	-0.57	sec

According to the results in Figure 5.1 and Table 5.1, when W_{wpV} is increased, the UAV will have a stronger tendency to approach the waypoint, which will result in a smaller miss distance, less control energy and more maneuver time; conversely, increased W_{obV} will result in more significant avoidance maneuvers, higher control energy and less maneuver time.

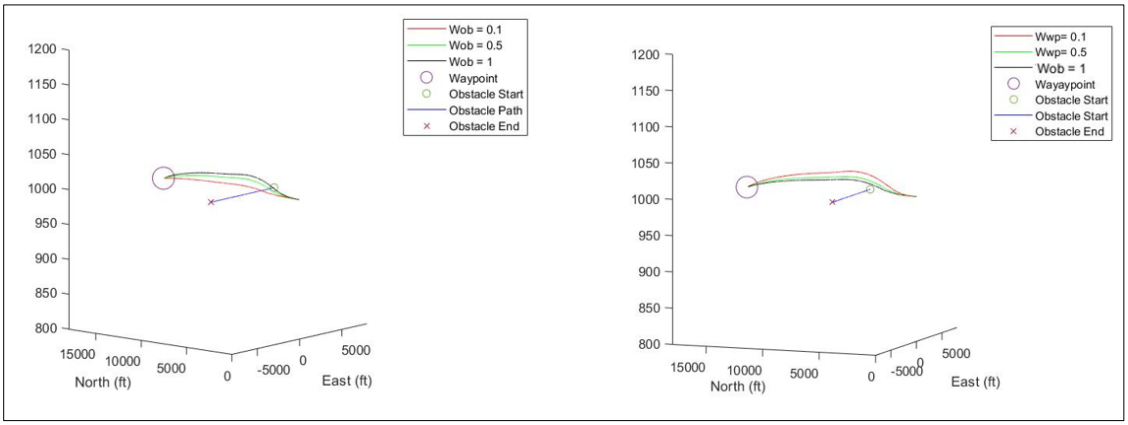

Figure 5.2 Trajectories for different weighting coefficients in the virtual functions.

The same law can be verified in the virtual point functions. As shown in Figure 5.2 and Table 5.2, greater W_{wpV} can increase the effect of approaching the waypoint and decrease the effect of avoiding obstacles; the effects of varying W_{obV} are just the opposite.

Table 5.2

Metrics for the single moving obstacle scenario using the virtual point function

	W_{wpV}=1; N_{wpV}=180; N_{obV}=8000			W_{obV}=0.5; N_{wpV}=180; N_{obV}=8000			
Metrics	**W_{obV}=0.1**	**W_{obV}=0.5**	**W_{obV}=1**	**W_{wpV}=0.1**	**W_{wpV}=0.5**	**W_{wpV}=0.9**	**Unit**
Min Miss Distance	9.78	18.31	24.21	30.59	27.07	21.85	ft
Control Energy	0.201	0.9394	1.6899	3.0917	2.1467	1.3582	
Total Time	-0.08	-0.2	-0.27	-0.36	-0.3	-0.24	sec

The metrics in Table 5.2 more clearly show the trends. Compared with the inversion functions, changing weighting coefficients has a similar influence on miss distance, energy cost and maneuvering time, but the overall effect is less for the virtual point approach.

5.2. Noise Analysis

Take the inversion weighting function as an example and the single moving obstacle scenario. In order to investigate the effect of noise on the system performance for vertical maneuvers, Gaussian noise with zero mean and 1 pixel standard deviation was applied on the pixel frame Y axis.

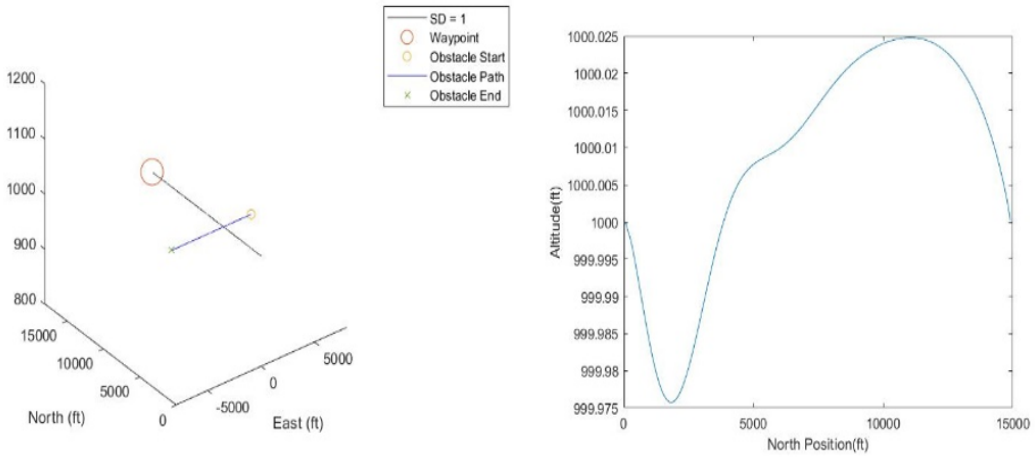

Figure 5.3 Simulated trajectory for inversion weighting function with 1 pixel SD noise.

As shown in Figure 5.3, the UAV does not maneuver to avoid the moving obstacle. The altitude change in Figure 5.3 shows that the UAV continues to fly at the trim altitude and does not maneuver to avoid the obstacle.

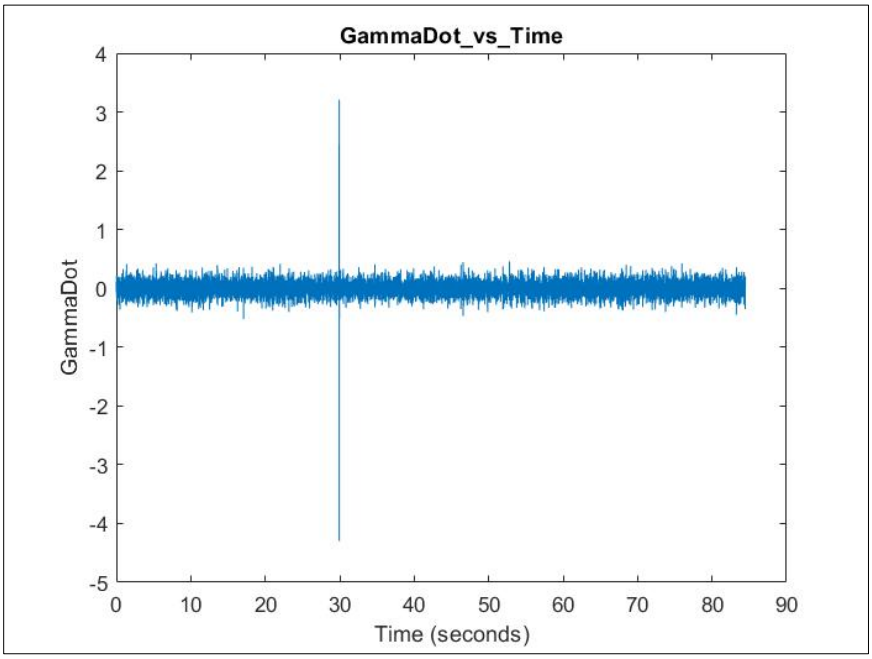

Figure 5.4 Vertical LOS rate with 1 pixel SD noise.

The vertical LOS rate ($\dot{\gamma_{ob}}$) is provided in Figure 5.4. It is obvious that the random noise in the pixel frame generates high frequency noise in the LOS rate, which results in the LOS rate being greater than the Pro-Nav threshold value at every time step; therefore, even though the obstacle poses a collision risk, the LOS rate cannot be calculated with sufficient accuracy to generate an avoidance command.

To improve the UAV performance when the camera measurements have noise, a first order low pass filter is implemented to filter the high frequency noise in the measured LOS angle. The LPF is defined as in equation (32):

$$G(s) = \frac{0.2}{s + 0.2} \tag{32}$$

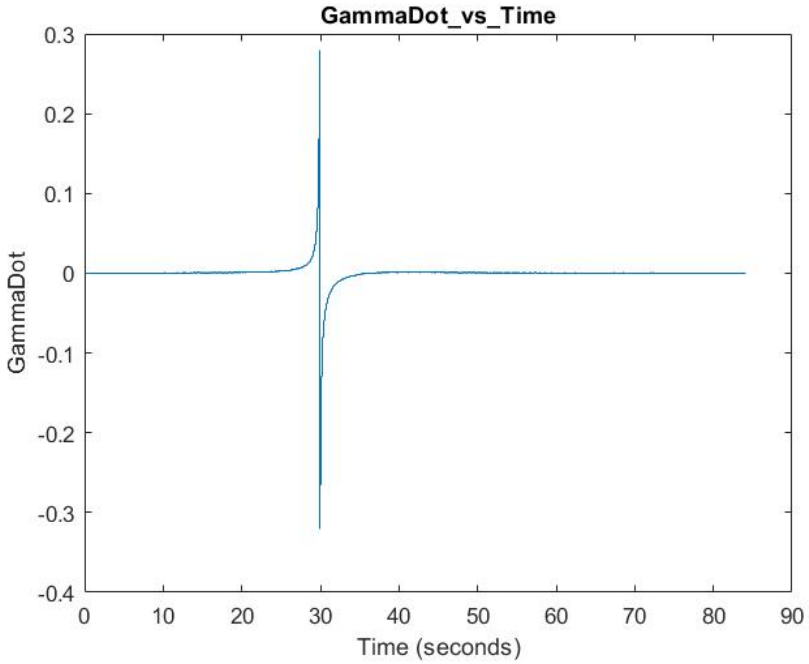

Figure 5.5 Filtered vertical LOS rate.

The filtered LOS rate is provided in Figure 5.5. According to Figure 5.5, the noise with frequency greater than the cut-off frequency of 50Hz is filtered from the LOS rate

signal. Based on this filtered LOS rate, the simulated UAV trajectory is generated as shown in Figure 5.6.

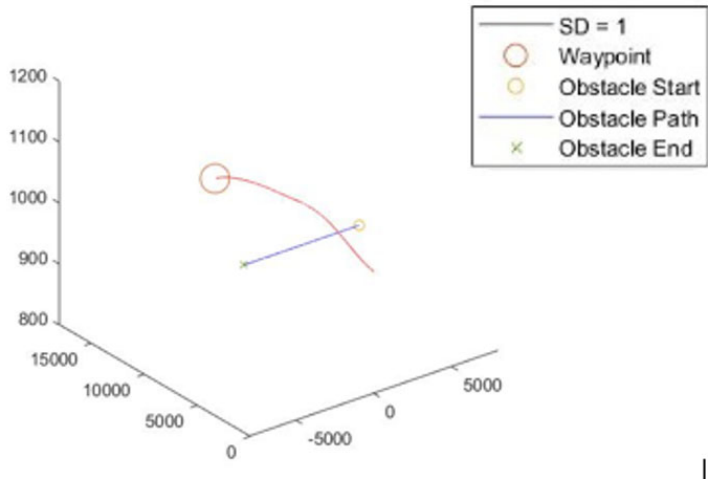

Figure 5.6 Simulated trajectory with filtered LOS rates.

Figure 5.6 provides the UAV trajectory using the first order filter. In the figure, the UAV accomplishes an avoidance maneuver by pitching to climb out of the collision path. According to Table 5.3, the UAV is able to successfully finish the avoidance maneuver. The LPF is therefore able to mitigate noise to ensure the normal operation of the collision avoidance system, but the minimum miss distance is less than the ideal noise-free case, and the control energy cost is also higher than the ideal case.

Table 5.3

Metrics for the single moving obstacle scenario with and without noise

Metrics	**With Noise**	**Without Noise**	**Unit**
Min Miss Distance	26.59	39.9	ft
Control Energy	2.018	4.48	
Total Time	-0.29	-0.43	sec

Since the LPF was shown to improve the system performance when the camera measurements have noise with a standard Gaussian distribution with 1 pixel standard deviation, the performance of the LPF is studied for varying noise levels.

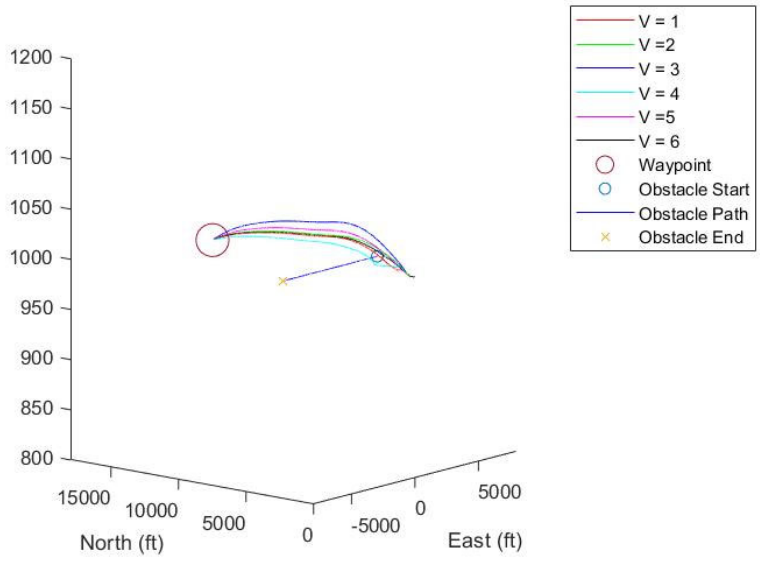

Figure 5.7 Multiple noise variance simulation trajectories with filtered LOS rate.

As shown in Figure 5.7, as the variance increases from 1 to 6 pixels, the UAV trajectories also change. In each case, the UAV can complete the avoidance maneuver by changing the pitch angle in a similar manner to the ideal scenario.

Table 5.4

Metrics for the single moving obstacle scenario with different noise levels

Metrics							**Unit**
Variance	1	2	3	4	5	6	ft
Min Miss Distance	26.59	28.63	41.96	18.98	33.0	27.65	
Control Energy	2.0079	3.46	6.5043	3.2872	3.7518	2.866	sec

The basic metrics for the simulation results are shown in Table 6.2. The minimum miss distances are controlled within the range of 18 ft to 33 ft, which can be considered as a safe range for the UAV in this scenario. Therefore, the LPF implemented in the single moving obstacle scenario was verified to be effective in improving the UAV performance when the camera has different levels of Gaussian noise.

To investigate the system performance on lateral maneuvers, the scenario "Tower" is implemented and Gaussian noise with zero mean and 1 pixel standard deviation was applied on the pixel frame X axis. To improve the UAV performance when the camera has noise, the first order LPF shown in equation (26) is implemented to filter the high frequency noise in the lateral LOS angle (λ).

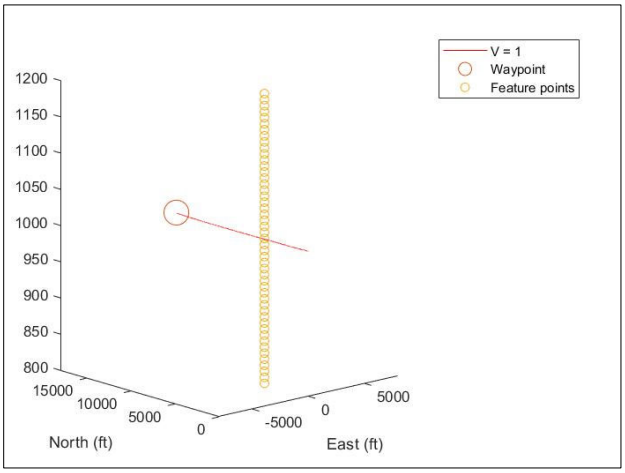

Figure 5.8 Tower obstacle simulation trajectory with 1 pixel SD noise.

Figure 5.8 provides the UAV trajectory using the first order filter. In the figure, the UAV accomplishes lateral avoidance maneuver by rolling. According to Table 5.5, the miss distance has a significant increase compared with the case where no filter is implemented. The UAV is able to finish a successful avoidance maneuver.

Table 5.5

Metrics for the tower obstacle scenario with 1 pixel noise

	Without filter	**With filter**	**Unit**
Variance	1	1	ft
Min Miss Distance	4.85	65.92	
Control Energy	0.0242	8.5599	Sec

Figure 5.9 provides the UAV trajectory using the first order filter. In the figure, the UAV accomplishes the avoidance maneuver by pitching and rolling. According to Table 5.6, the miss distance also has a significant increase compared with the case where no filter is implemented. The miss distance after using the filter is greater than half of the UAV wingspan, which means the UAV is able to finish a safe avoidance maneuver. This simulation results show that the LPF is able to ensure the normal operation of the collision avoidance system.

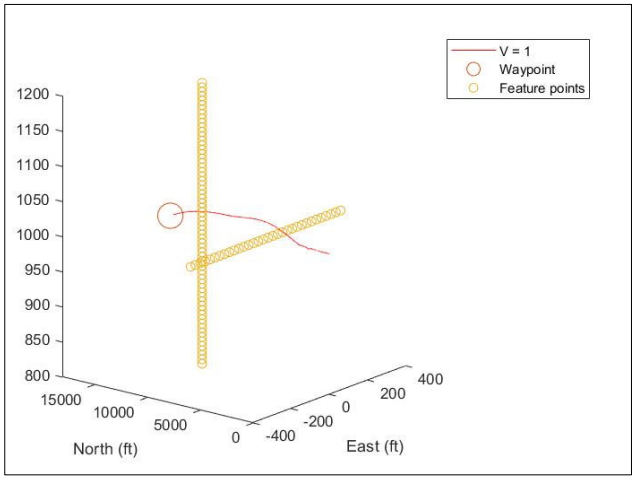

Figure 5.9 Complex obstacle simulation trajectory with 1 pixel SD noise.

Table 5.6

Metrics for the complex obstacle scenario with 1 pixel noise

	Without filter	**With filter**	**Unit**
Variance	1	1	ft
Min Miss Distance	6.7853	22.0806	
Control Energy	1.2558	3.951	Sec

According to the simulation results in this chapter, the guidance system is easily affected by high frequency noise, which can result in LOS rates that do not generate avoidance commands. For several specific scenarios, avoidance maneuvers were accomplished when the camera has noise with variance between 1 to 6 pixels after implementing a first order Low pass filter on the LOS angle measurements. This means the failures caused by vision noise within a specific range of variance can be reduced or eliminated by the LPF for several scenarios, but further simulation and tuning might be required.

6. CONCLUSION

This thesis has developed a 3D collision avoidance system based on the proportional navigation law. By implementing this avoidance law, the UAV is capable of avoiding single or multiple moving or static obstacles of different sizes automatically while approaching a designated waypoint.

The Pro-Nav law is first applied in collision detection by using a line of sight rate threshold value. According to the definition of the basic Pro-Nav guidance law, any obstacle that causes the absolute value of the LOS rate to be less than a defined threshold is considered as a potential collision threat. Meanwhile, a flight path selection logic is applied to decide the maneuver type. In general, if the obstacle LOS angles satisfy $\sum_i^N \boldsymbol{\chi_i} < \sum_i^N \boldsymbol{\gamma_i}$, the UAV will perform a lateral maneuver to bypass obstacles; on the other hand, if $\sum_i^N \boldsymbol{\chi_i} > \sum_i^N \boldsymbol{\gamma_i}$, the UAV will generate a vertical maneuver to cross the obstacles. Therefore, the collision avoidance system achieves lateral or vertical avoidance maneuvers for 3D obstacles.

The maneuver guidance command is generated by a weighting function method. This thesis has developed two types of functions – an inversion weighting function and a virtual point weighting function. Both weighting functions can balance the maneuver between approaching a desired waypoint and avoiding the obstacles.

Several scenarios are simulated to investigate the feasibility and performance of the weighting functions. According to the simulation results, both weighting functions are capable of making the UAV avoid both moving and static obstacles successfully. Both weighting functions have similar control effort when the UAV is maneuvering in a narrow space; both weighting functions are verified to be robust and feasible to avoid

obstacles of different sizes with automatic flight path selection. In general, the virtual point method was found to have significant advantages for lateral maneuvers and results in more optimal trajectories.

In addition, the UAV performance with camera pixel noise has been studied. According to the results, the high frequency noise will interfere with the calculation of the LOS rates, which can in turn lead to ignoring obstacle collision threats. A low pass filter (LPF) was applied to the measured LOS angles to address this issue, and it was verified that this filtering is able to reduce or eliminate failures caused by camera pixel noise within 6 pixel variance.

Future work for this collision avoidance system should be focused on flight test implementation and more complex 3D simulation scenarios. For simulation, virtual reality software, such as MetaVR Virtual Scene Generation, can be applied to provide more complex obstacle environments, which can also be coupled to image processing algorithms. In addition, the LPF can be further studied and the weighting functions can be further tuned to fit more complex camera noise scenarios or more complex filtering approaches such as the Kalman filter can be used instead of the LPF; these might improve the performance for higher noise levels. Different types of weighting functions can also potentially be developed for more optimal trajectories. Other performance metrics can be considered that include effects such as actuator saturation. Actuator saturation can be indirectly addressed in the guidance cost functions by enforcing upper and lower limits on the guidance angles based on actuator limitations; for example, aileron limits would lead to a maximum roll angle; elevator and thrust limits would lead to a maximum pitch angle.

The ultimate objective would be the implementation of this collision avoidance system, including a camera sensor and image processing algorithms, in UAV flight tests to prove the feasibility of the system in a real 3D environment. Therefore, real-time implementation would require alternative optimization tools that run faster than the MATLAB function.

REFERENCES

Airservices Australia. (2012). How ADS-B works. Retrieved from https://www.airservicesaustralia.com/projects/ads-b/how-ads-b-works/

Beard, R., Curtis, J., Eilders, M., Evers, J., & Cloutier, J. (2007). Vision aided proportional navigation for micro air vehicles. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, AIAA 2007-6609. doi: 10.2514/6.2007-6609

Clark, Matthew J. (2017). *Collision avoidance and navigation of UAS using vision-based proportional navigation.* (Unpublished master's thesis). Embry-Riddle Aeronautical University, Daytona Beach, FL.

CSDN, (n.d.). Computer vision: camera imaging principles: conversion between world coordinate system, camera coordinate system, image coordinate system, pixel coordinate system. Retrieved from https://blog.csdn.net/chentravelling/article/details/535580

Dhar, A., & Ghose, D. (1993). Capture region for a realistic TPN guidance law. *IEEE Transactions on Aerospace and Electronic Systems*, *29*(3), 995-1003. doi: 10.1109/7.220946

Automatic dependent surveillance – broadcast. (n.d.). In *Wikipedia*. Retrieved from https://en.wikipedia.org/wiki/Automatic_dependent_surveillance_-_broadcast

NASA Glenn Research Center (2015). Banking turn. Retrieved from https://www.grc.nasa.gov/www/k-12/airplane/turns.html

Tower (n.d.). In *Wikipedia*. Retrieved from https://en.wikipedia.org/wiki/Tower

Bridge (n.d.). In *Wikipedia*. Retrieved from https://en.wikipedia.org/wiki/Bridge

Guelman, M., Idan, M., & Golan, O. M. (1995). Three-dimensional minimum energy guidance. *IEEE Transactions on Aerospace and Electronic Systems*, *31*(2), 835-841. doi: 10.1109/7.381933

Guelman, M. (1976). The closed-form solution of true proportional navigation. *IEEE Transactions on Aerospace and Electronic Systems*, *12*(4), 472–82. doi: 10.1109/TAES.1976.308328.

Lee, J. O., Lee, K. H., Park, S. H., Im, S. G., & Park, J. (2011). Obstacle avoidance for small UAVs using monocular vision. *Aircraft Engineering and Aerospace Technology*, *83*(6), 397-406. doi: 10.1108/00022661111173270

Li, K., Zhang, T., & Chen, L. (2013). Ideal proportional navigation for exoatmospheric interception. *Chinese Journal of Aeronautics*, *26*(4), 976-985. doi: 10.1016/j.cja.2013.06.007

Murtaugh, S. A., & Criel, H. E. (1966). Fundamentals of proportional navigation. *IEEE Spectrum*, *3*(12), 75-85. doi: 10.1109/MSPEC.1966.5217080

Northwestern Polytechnical University. (2012). Modern control theory in missile guidance. Retrieved from http://www.doc88.com/p-890244997353.html

Pham, H., Smolka, S. A., Stoller, S. D., Phan, D., & Yang, J. (2015). A survey on unmanned aerial vehicle collision avoidance systems. arXiv preprint arXiv:1508.07723.

Saha, S., Natraj, A., & Waharte, S. (2014). A real-time monocular vision-based frontal obstacle detection and avoidance for low cost UAVs in GPS denied environment. In *2014 IEEE International Conference on Aerospace Electronics and Remote Sensing Technology*, 189-195. doi: 10.1109/ICARES.2014.7024382

Shukla, U. S., & Mahapatra, P. R. (1990). The proportional navigation dilemma-pure or true. *IEEE Transactions on Aerospace and Electronic Systems*, *26*(2), 382-392. doi: 10.1109/7.53445

Skowron, M., Chmielowiec, W., Glowacka, K., Krupa, M., & Srebro, A. (2019). Sense and avoid for small unmanned aircraft systems: Research on methods and best practices. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, *233*(16), 6044-6062. doi: 10.1177/0954410019867802

Viquerat, A., Blackhall, L., Reid, A., Sukkarieh, S., & Brooker, G. (2008). Reactive collision avoidance for unmanned aerial vehicles using doppler RADAR. In Laugier C., Siegwart R. (eds), *Field and Service Robotics. Springer Tracts in Advanced Robotics*, vol 42. Springer, Berlin, Heidelberg. doi: 10.1007/978-3-540-75404-6_23

von Essen, C., & Giannakopoulou, D. (2014). Analyzing the next generation airborne collision avoidance system. In Ábrahám E., Havelund K. (eds), *Tools and Algorithms for the Construction and Analysis of Systems. TACAS 2014. Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg. doi: 10.1007/978-3-642-54862-8_54

Yang, C. D., Hsiao, F. B., & Yeh, F. B. (1989). Generalized guidance law for homing missiles. *IEEE Transactions on Aerospace and Electronic Systems*, *25*(2), 197-211. doi: 10.1109/7.18681

Yang, C. D., & Yang, C. C. (1996). Analytical solution of generalized three-dimensional proportional navigation. *Journal of Guidance, Control, and Dynamics*, *19*(3), 721-724. doi: 10.2514/3.21685

Yang, C. D., Yeh, F. B., & Chen, J. H. (1987). The closed-form solution of generalized proportional navigation. *Journal of Guidance, Control, and Dynamics*, *10*(2), 216-218. doi: 10.2514/3.20205

Yuan, P. J., & Chern, J. S. (1992). Ideal proportional navigation. *Journal of Guidance, Control, and Dynamics*, *15*(5), 1161-1165. doi: 10.2514/3.20964

Zarchan, P. (1994). *Tactical and strategic missile guidance* (7th ed., vol. 1). Washington D.C., American Institute for Aeronautics and Astronautics.