



저작자표시 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.
- 이차적 저작물을 작성할 수 있습니다.
- 이 저작물을 영리 목적으로 이용할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#) 

공학박사학위논문

**Reactive Collision Avoidance of Unmanned
Aerial Vehicles Using Vision Sensors**

영상센서를 이용한 무인항공기 반응적 충돌회피 기법 연구

2014년 8월

서울대학교 대학원

기계항공공학부

최 현 진

Reactive Collision Avoidance of Unmanned Aerial Vehicles Using Vision Sensors

영상센서를 이용한 무인항공기 반응적 충돌회피 기법 연구

지도교수 김 유 단

이 논문을 공학박사 학위논문으로 제출함

2014년 7월

서울대학교 대학원

기계항공공학부

최 현 진

최현진의 공학박사 학위논문을 인준함

2014년 7월

위 원 장 : _____ (인)

부위원장 : _____ (인)

위 원 : _____ (인)

위 원 : _____ (인)

위 원 : _____ (인)

Reactive Collision Avoidance of Unmanned Aerial Vehicles Using Vision Sensors

Supervisor : Professor YOUDAN KIM

A Dissertation by

HYUNJIN CHOI

July 2014

Department of Mechanical and Aerospace Engineering
Seoul National University

Submitted to Seoul National University in partial fulfillment of
the requirements for the degree of DOCTOR OF PHILOSOPHY

Approved as to style and content by

Professor Changdon Kee, Chairman

Professor Youdan Kim, Vice-Chairman

Professor H. Jin Kim, Member

Professor Changook Park, Member

Professor Hyochoong Bang, Member

Abstract

Reactive Collision Avoidance of Unmanned Aerial Vehicles Using Vision Sensors

Hyunjin Choi

Department of Mechanical and Aerospace Engineering

The Graduate School

Seoul National University

Vision sensors have the potential to deal with collision avoidance of unmanned aerial vehicles (UAVs) due to their passive and noncooperative characteristics, and therefore they are suited to the “Sense-and-Avoid” of the UAV for avoiding intruders. However, the vision sensors cannot directly provide range information of the obstacles, and therefore it is difficult to obtain the states of the obstacles. To deal with the “Sense-and-Avoid” of the vision-based UAV, a vision-based reactive collision avoidance algorithm is proposed in this dissertation. The collision avoidance algorithm adopts a measurement model of a single vision sensor and the concept of the closest-point-of-approach to obtain useful characteristics for collision avoidance. By utilizing the characteristics and assuming the size of the intruder, a robust and safe collision avoidance direction is provided. A time condition for collision avoidance is also proposed, which prevents unnecessary trajectory extension caused by the size assumption.

On the other hand, the UAV guidance law should be compatible with the collision avoidance algorithm. In this dissertation, several UAV guidance problems are considered; waypoint tracking, waypoint tracking with line tracing, and general trajectory tracking problems. Speed, heading angle, and flight path angle commands, which guarantee convergence with a desired trajectory, are generated for the guidance problems. The guidance commands are combined with the collision avoidance commands by adopting a command switching scheme which uses the command filter. The effectiveness of the proposed method is demonstrated through numerical simulations.

Keywords: Unmanned aerial vehicle, Collision avoidance, Single vision sensor, Moving obstacle, Closest-point-of-approach, Trajectory tracking

Student Number: 2010-30200

Contents

Chapter 1 Introduction	1
1.1 Background and Objectives	1
1.2 Literature Survey.....	6
1.3 Research Contributions	12
1.4 Dissertation Organization	14
Chapter 2 Guidance Law of Unmanned Aerial Vehicles	15
2.1 Model of Unmanned Aerial Vehicles	15
2.2 Guidance Problems of Unmanned Aerial Vehicles	19
2.2.1 Waypoint Tracking Problem	19
2.2.2 Waypoint Tracking with Line Tracing Problem.....	20
2.2.3 Trajectory Tracking Problem	23
2.3 Command Tracking Guidance Law	26
2.3.1 Guidance Command.....	26
2.3.2 Stability Analysis	31
2.4 Simulations on Guidance	33
Chapter 3 Reactive Collision Avoidance.....	41
3.1 Collision Detection	41
3.2 Solutions for Collision Avoidance	45
3.3 Command Switching Scheme using Command Filter	53
Chapter 4 Vision-Based Reactive Collision Avoidance	56

4.1	Single Vision Sensor and Its Limitations	57
4.2	Closest-Point-of-Approach (CPA)	61
4.3	Collision Avoidance Algorithm	66
4.3.1	Collision Avoidance Direction	67
4.3.2	Collision Avoidance Time	77
Chapter 5 Simulations on Collision Avoidance		82
5.1	Simulations on Reactive Collision Avoidance	82
5.2	Simulations on Vision-Based Reactive Collision Avoidance	92
Chapter 6 Conclusions		109
6.1	Concluding Remarks	109
6.2	Future Works	111
Bibliography		113
Appendix A Lyapunov Stability Theorem		122
Appendix B Perspective Projection of Vision Sensor		124
초 록		126

List of Tables

Table 2.1	Parameters used in the UAV guidance.....	33
Table 4.1	UAV wingspan data with respect to the airspace classes of the National Airspace System [55].....	65
Table 5.1	Parameters used in the RCA algorithm.....	83
Table 5.2	Miss distance results.....	89
Table 5.3	Parameters used in the VRCA algorithm.....	92

List of Figures

Figure 2.1	Configuration of position and velocity vectors.....	16
Figure 2.2	Configuration of waypoint tracking.....	21
Figure 2.3	Configuration of waypoint tracking with line tracing	21
Figure 2.4	Configuration of trajectory tracking	23
Figure 2.5	Configuration of the states of the UAV and the reference states of the UAV	26
Figure 2.6	Trajectory of the waypoint tracking (Command tracking guidance law).....	36
Figure 2.7	Trajectory of the waypoint tracking with line tracing (Command tracking guidance law).....	36
Figure 2.8	Position error between the UAV and the reference on the line (Command tracking guidance law).....	37
Figure 2.9	Time histories of reference-command-state (Command tracking guidance law).....	37
Figure 2.10	Trajectory of a circular trajectory tracking (Command tracking guidance law).....	38
Figure 2.11	Position error between the UAV and the reference on the circular trajectory (Command tracking guidance law)	38
Figure 2.12	Trajectory of the waypoint tracking with line tracing (Follow-the-carrot)	39
Figure 2.13	Position error between the UAV and the reference on the line (Follow-the-carrot).....	39
Figure 2.14	Trajectory of a circular trajectory tracking (Follow- the-carrot)	40
Figure 2.15	Position error between the UAV and the reference on the circular trajectory (Follow-the-carrot)	40
Figure 3.1	Relative geometry of the UAV and intruder	42

Figure 3.2	Safety boundary projection.....	46
Figure 3.3	Viewpoint of the UAV towards the position of the intruder	51
Figure 3.4	Block diagram of the UAV system	54
Figure 3.5	Command filter.....	54
Figure 3.6	Data flow of Guidance/RCA system.....	55
Figure 4.1	Measurement model of a vision sensor	58
Figure 4.2	Relative recognition according to the intruder's size assumption.....	63
Figure 4.3	Geometry of CPA.....	67
Figure 4.4	Path frame and geometry of the CPA for collision avoidance direction.....	69
Figure 4.5	Avoidance directions according to approaching configurations	76
Figure 4.6	Geometry of the single point avoidance	78
Figure 4.7	Data flow of Guidance/VRCA system.....	81
Figure 5.1	Trajectory of the GO approach (Intruder's heading: E).....	85
Figure 5.2	Trajectory of the GO approach (Intruder's heading: SE).....	85
Figure 5.3	Trajectory of the GO approach (Intruder's heading: S).....	86
Figure 5.4	Trajectory of the GO approach (Intruder's heading: SW)	86
Figure 5.5	Trajectory of the 3-D RCA (Intruder's heading: E).....	87
Figure 5.6	Trajectory of the 3-D RCA (Intruder's heading: SE).....	87
Figure 5.7	Trajectory of the 3-D RCA (Intruder's heading: S).....	88
Figure 5.8	Trajectory of the 3-D RCA (Intruder's heading: SW)	88
Figure 5.9	Trajectory of the UAV guidance with collision avoidance	90
Figure 5.10	Position error between the UAV and its path.....	90
Figure 5.11	Time histories of the heading angle command and filtered heading angle command.....	91
Figure 5.12	Trajectory of the GO approach (Noninteractive intruder,	

	$R_0 = 0.5R_r$, intruder's heading: E, failure case).....	95
Figure 5.13	Trajectory of the GO approach (Noninteractive intruder, $R_0 = 0.5R_r$, intruder's heading: SE).....	95
Figure 5.14	Trajectory of the GO approach (Noninteractive intruder, $R_0 = 0.5R_r$, intruder's heading: S).....	96
Figure 5.15	Trajectory of the GO approach (Noninteractive intruder, $R_0 = 0.5R_r$, intruder's heading: SW).....	96
Figure 5.16	Trajectory of the VRCA (Noninteractive intruder, $R_0 = 0.5R_r$, intruder's heading: E).....	97
Figure 5.17	Trajectory of the VRCA (Noninteractive intruder, $R_0 = 0.5R_r$, intruder's heading: SE).....	97
Figure 5.18	Trajectory of the VRCA (Noninteractive intruder, $R_0 = 0.5R_r$, intruder's heading: S).....	98
Figure 5.19	Trajectory of the VRCA (Noninteractive intruder, $R_0 = 0.5R_r$, intruder's heading: SW).....	98
Figure 5.20	Miss distance and maximum deviation results (Noninteractive intruder, $R_0 = 0.5R_r$).....	99
Figure 5.21	Trajectory of the GO approach (Interactive intruder, $R_0 = 0.5R_r$, intruder's heading: E, failure case).....	101
Figure 5.22	Trajectory of the GO approach (Interactive intruder, $R_0 = 0.5R_r$, intruder's heading: SE).....	101
Figure 5.23	Trajectory of the GO approach (Interactive intruder, $R_0 = 0.5R_r$, intruder's heading: S).....	102
Figure 5.24	Trajectory of the GO approach (Interactive intruder, $R_0 = 0.5R_r$, intruder's heading: SW).....	102
Figure 5.25	Trajectory of the VRCA (Interactive intruder, $R_0 = 0.5R_r$, intruder's heading: E).....	103

Figure 5.26	Trajectory of the VRCA (Interactive intruder, $R_0 = 0.5R_r$, intruder's heading: SE).....	103
Figure 5.27	Trajectory of the VRCA (Interactive intruder, $R_0 = 0.5R_r$, intruder's heading: S)	104
Figure 5.28	Trajectory of the VRCA (Interactive intruder, $R_0 = 0.5R_r$, intruder's heading: SW).....	104
Figure 5.29	Miss distance and maximum deviation results (Interactive intruder, $R_0 = 0.5R_r$)	105
Figure 5.30	Trajectory of the VRCA with time condition (Noninteractive intruder, $R_0 = 0.5R_r$, intruder's heading: SE).....	107
Figure 5.31	Trajectory of the VRCA with time condition (Noninteractive intruder, $R_0 = 0.5R_r$, intruder's heading: SW)	107
Figure 5.32	Miss distance results for noninteractive intruder	108
Figure 5.33	Maximum deviation results for noninteractive intruder	108
Figure B.1	Pinhole perspective projection model.....	124

Chapter 1

Introduction

1.1 Background and Objectives

Over the last few decades, plenty of studies on unmanned aerial vehicle (UAV) have been performed due to its utility and functionality. UAV has been also regarded as an alternative means of manned aircraft because it reduces risks to human life. Especially, UAV has various applications in the area of military service including reconnaissance, surveillance, attack, etc. To perform these missions, an autonomous UAV system is required, and therefore much research on the guidance and control system for UAV have been done [1-3]. Also, the function of collision avoidance is required for the safe operation of UAV.

Collision avoidance problems have been widely studied in the field of robotics, and various methods have been developed for the collision avoidance of industrial robots or mobile robots [4, 5]. In the field of robotics, most of the collision avoidance algorithms are

focused on the path planning which aims to find a collision-free path in known environments. Thus, these algorithms usually consider stationary obstacles such as geographical features.

However, the path planning algorithms are not suitable for the collision avoidance of UAV when obstacles are moving. Because most of the path planning algorithms provide a pre-designed path, the path might be useless in dynamically varying environments. Note that other aircraft or intruders can be regarded as moving obstacles in the field of aircraft. Therefore, the collision avoidance problem of UAV should consider the cases of dynamic obstacles, which are related to the situations of midair collision (MAC).

MAC is a case of important problems for the safe operation of air transportation system because the situation directly threatens human life, and thus there have been a large number of efforts to resolve the MAC situation. As a result, several safety rules of flight [6, 7] have been established and instruments for collision avoidance have been developed. In general, manned aircraft use specific equipments such as the Traffic Collision Avoidance System (TCAS) or Automatic Dependent Surveillance-Broadcast (ADS-B) [8-10] which can communicate and negotiate with other aircraft. Thus, they are operated

with adequate regulations and communication. Unlike manned aircraft, however, UAV is not equipped with these kinds of communication capabilities, yet. For these reasons, the “Sense-and-Avoid” concept has been proposed for UAV [11-14]. The “Sense-and-Avoid” concept can improve the safety of UAV by detecting and avoiding possible collision threats through a sensor system and corresponding collision avoidance algorithm.

For the “Sense-and-Avoid” of UAV, various sensors which can measure the states of external objects could be implemented. Laser sensors, radar sensors, vision sensors, and acoustic sensors can be considered for the UAV. These sensors are divided into noncooperative sensors and cooperative sensors in accordance with the communication capability. Noncooperative sensors use their own measurements and do not communicate or negotiate with other sensors, whereas cooperative sensors do. According to the effect on the environments, they are also divided into two types; passive and active. Passive sensors do not emit any energy to measure external objects, whereas active sensors do [11, 12].

The noncooperative and passive characteristics contribute to the improvement of UAV’s autonomy and functionality by enabling

stealth and self-defense capabilities. Since collision avoidance algorithms for the “Sense-and-Avoid” are required to resolve noncooperative collision situations, these noncooperative and passive sensors are appropriate to the UAV. In particular, noncooperative characteristics enable the UAV to avoid unspecified intruders, thereby they have the potential to deal with various collision situations.

The vision sensor is one of the noncooperative and passive sensors, and is widely applied to the UAV system due to the light weight and cheap price. Among the vision sensors, a single vision sensor has a loss-of-depth problem because it projects 3-dimensional objects onto a 2-dimensional image plane. Thus, it is difficult to obtain the position and velocity information of the objects by the vision sensor [15]. The loss-of-depth problem of the vision sensor can be dealt with by using additional sensors which can measure the range. In this case, the UAV cannot fully take advantage of the noncooperative and passive characteristics of the sensor. On the other hand, the stereovision technique using two vision sensors, which is similar to human eyes, can solve the loss-of-depth problem. However, the stereovision technique is computationally expensive, which makes it difficult to use for real-time applications [15, 16]. In the case of a single vision

sensor, the range estimation scheme also deals with the loss-of-depth problem [17]. However, this method has a convergence problem for moving obstacles because of bad observability conditions, and it requires unnatural oscillatory maneuvers of the UAV to satisfy the observability conditions.

In summary, the collision avoidance problem using a single vision sensor still has many difficulties for moving obstacles, but collision avoidance algorithms for the problem will contribute to the “Sense-and-Avoid” of UAV in efficiency and cost aspects. For those reasons, collision avoidance of the UAV using a single vision sensor is considered in this study, which is aiming at avoiding a moving intruder.

1.2 Literature Survey

In the field of robotics, collision avoidance has been studied to solve global path planning problems or local collision avoidance problems. The global path planning problems focus on relatively large-scale obstacles such as geographical features, and therefore the path planning problems consider a collision-free path in a known and stationary environment. Various path planning methods were introduced in Ref. [4]. Especially, potential function and roadmap methods are representative path planning methods. The potential function method makes a collision-free path by applying artificial repulsive forces for the obstacles and an attractive force for the goal. The roadmap method provides an organized map such as visibility graph or Voronoi graph to find a collision-free path. For optimal path planning, there exists A* algorithm which finds the shortest path of a graph structure. Lavelle proposes a sampling-based path planning algorithm called rapidly-exploring random tree (RRT) [5], which considers nonholonomic constraints of mobile vehicles.

However, the global path planning is not suitable for moving obstacles, and more direct collision avoidance methods such as

reactive collision avoidance algorithms are required. The concept of the reactive collision avoidance is that the vehicle decides and performs its own collision avoidance maneuver in real-time whenever the collision situation is updated. For stationary obstacles, vector field histogram [18] and nearness diagram navigation [19] methods were proposed for mobile robots. For moving obstacles, Chakravarty and Ghose proposed an obstacle avoidance algorithm which is called as the collision cone approach [20], where collision detection and avoidance scheme was proposed using the collision cone geometry. Similarly, Fiorini and Shiller proposed the velocity obstacle method for moving obstacle avoidance, which also adopted the collision cone geometry [21]. In the velocity obstacle method, the vehicle decides its own maneuver using the updated information of the obstacle motion.

On the other hand, in the field of aircraft, moving obstacle avoidance is more important than stationary obstacle avoidance because it is directly related to the MAC. Kuchar and Yang performed comparative analysis of the various collision avoidance algorithms for aircraft [22]. In Ref. [22], the collision avoidance solutions were classified into prescribed, optimized, and force field solutions. The force field approach [23] is similar to the potential function approach.

The collision cone [20] and velocity obstacle [21] approaches are kinds of optimized approaches because an optimized geometry of the collision cone is considered. Bilimoria considered a similar geometry and proposed a geometric optimization approach for aircraft [24]. Hwang et al. proposed a protocol-based conflict resolution algorithm for air traffic control using the result of the geometric optimization approach [25]. Using the collision cone geometry, decentralized reactive collision avoidance [26], missile guidance based UAV guidance [27], and 3-dimensional geometric collision avoidance [28] methods were also proposed. Those methods are computationally efficient.

The collision cone geometry is related to the closest-point-of-approach (CPA) of two moving aircraft. Mujumdar and Padhi proposed a nonlinear geometric guidance law and a differential geometric guidance law using the geometry of the CPA [29]. Gates proposed real-time guidance laws based on the geometric properties of the CPA [30] and a stereo projection concept for the cockpit display of aircraft [31]. Choi et al. proposed a 3-dimensional reactive collision avoidance algorithm which can avoid multiple obstacles by using a projection concept and the CPA geometry [32].

On the other hand, model predictive control (MPC) can be also implemented to avoid obstacles in dynamic environments [33]. MPC adopts the receding horizon concept, and solves a trajectory optimization problem. However, this method requires a high-performance computer because of the computational load.

As mentioned in the previous section, the collision avoidance algorithm of UAV should be compatible with sensors for the “Sense-and-Avoid”. For the sensors applicable to the UAV, vision sensors are the most suitable. In general, vision sensors have plenty of information, which can be extracted through image processing [15]. For image processing, Ha et al. proposed the active contour algorithm which provides information of the object’s size [34]. However, a single vision sensor has the loss-of-depth problem, and therefore it is hard to obtain range information of objects. To overcome the loss-of-depth problem, several range estimation methods based on Kalman filter [35, 36], neural network [37], and adaptive observer [38, 39] have been proposed. These methods utilize the subtended angle, which can be obtained from the size of the intruder, to estimate the unknown range.

There have been several efforts to resolve the collision situation

using the range estimation. Watanabe et al. proposed an EKF-based adaptive estimation method [40] and Dippold et al. proposed a nonlinear estimator [41] for stationary obstacles. Shakernia et al. used EKF to estimate the states of moving obstacles [42]. In the cases of stationary obstacles, collision avoidance maneuvers of UAV satisfy observability conditions of the estimation problem. In the cases of moving obstacles, however, the UAV should generate oscillatory maneuvers to satisfy the conditions. Since the considered problem inherently has bad observability conditions because of the relatively long distance between the UAV and obstacle, the range estimation approach is faced with the convergence problem in the cases of moving obstacles. Meanwhile, optical flow can be also considered for the collision avoidance [43]. However, this approach is valid only for stationary obstacles because of the loss-of-depth problem of the vision sensor. Therefore, collision avoidance using a single vision sensor for avoiding moving obstacles has not yet been sufficiently studied and is still a challenging problem [44].

On the other hand, UAV has its own dynamics and includes various autopilots designed by the proportional-integral-derivative (PID) controller [45], linear controller [46], and nonlinear controller

[47-49]. According to the model of a fixed-wing UAV, it has kinematic constraints which are kinds of nonholonomic constraints [50], and these constraints influence on the design of guidance law. In the cases of trajectory or path tracking problems, conventional approaches using a look-ahead point such as the follow-the-carrot [51, 52] have lacks of convergence analysis. Therefore, a guidance law which guarantees convergence with a desired trajectory or path is required. The guidance problems including the collision avoidance will be also considered in this study.

1.3 Research Contributions

The contributions of this dissertation are summarized as follows.

First, a guidance law for a nonholonomic UAV system which guarantees convergence with a desired trajectory is proposed. In general, a fixed-wing UAV has kinematic constraints on speed and turning rate because of aircraft dynamics. These constraints restrict the motion of the UAV and yield specific trajectories. The proposed guidance law generates a bounded command set which consists of speed, heading angle, and flight path angle commands in waypoint tracking and trajectory tracking problems. According to the guidance law, the position error between the UAV and the reference trajectory converges to zero when the states of the UAV are converged to the guidance commands.

Second, a reactive collision avoidance algorithm and a command switching scheme of the UAV are proposed. The reactive collision avoidance algorithm utilizes the CPA-based collision detection and avoidance scheme. The collision avoidance solution provides speed, heading angle, and flight path angle commands whenever the collision situation is updated. The commands for guidance and collision

avoidance can be chosen in accordance with the collision situation. In addition, the command filter is adopted to mitigate sharp command switching which might result in abrupt maneuvers.

Finally, a vision-based reactive collision avoidance algorithm of the UAV is proposed. In relation to the loss-of-depth problem of a single vision sensor, the collision avoidance algorithm is proposed using the CPA geometry. The detailed features of the proposed collision avoidance algorithm are as follows:

- The characteristics of the CPA with respect to the intruder's size are analyzed using a measurement model of a single vision sensor.
- A direction condition that is safe and robust with respect to the motion of the intruder is proposed.
- A time condition to regulate the side effect of the loss-of-depth problem is developed.

The proposed algorithm utilizes the characteristics of the CPA to account for the limitation of the single vision sensor, and thus, the algorithm makes it possible to avoid an intruder without estimating the unknown size or range of the intruder.

1.4 Dissertation Organization

This dissertation is organized as follows:

Chapter 1 describes background and objectives of collision avoidance of UAV. Also, related research and contributions of this dissertation are summarized. Chapter 2 constructs guidance problems of the UAV including waypoint tracking and trajectory tracking problems. Then, a guidance law for a nonholonomic UAV system is proposed. Chapter 3 introduces a reactive collision avoidance algorithm for 3-dimensional collision situations, and a command switching scheme is proposed for guidance and collision avoidance of the UAV. Chapter 4 deals with a reactive collision avoidance algorithm of the UAV using a single vision sensor. The proposed collision avoidance algorithm is based on the characteristics of the vision sensor and the CPA geometry. Chapter 5 provides results of numerical simulations, and concluding remarks and future works are addressed in Chapter 6.

Chapter 2

Guidance Law of Unmanned Aerial Vehicles

In general, a fixed-wing aircraft is a nonholonomic system because the aircraft system cannot move in arbitrary directions by kinematic constraints. To make a UAV perform given missions, a proper guidance law should be designed. Thus, the guidance law of the UAV is considered in this chapter.

2.1 Model of Unmanned Aerial Vehicles

Motions of aircraft are directly related to the changes of aircraft's speed, heading angle, and flight path angle, which yield accelerating, decelerating, turning, climbing, and descending maneuvers. Therefore, desired values of the speed, heading angle, and flight path angle can be considered as guidance commands of the UAV. The UAV guidance can be achieved by autopilots which make the states of the UAV follow the guidance commands [45].

Let us consider the north-east-down (NED) frame to represent the position and velocity vectors as shown in Fig. 2.1. The velocity of the UAV, \mathbf{V} , can be described as follows.

$$\mathbf{V} = [V \cos \gamma \cos \chi \quad V \cos \gamma \sin \chi \quad -V \sin \gamma]^T \quad (2.1)$$

where V is a speed, χ is a heading angle, and γ is a flight path angle of the UAV, respectively

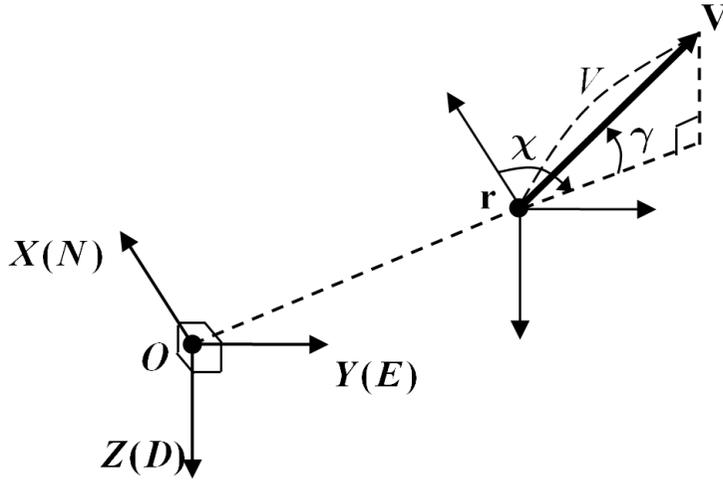


Figure 2.1 Configuration of position and velocity vectors

In this study, the following 3-dimensional point-mass model is considered as a model of the UAV.

$$\dot{V} = g(u_1 - \sin \gamma) \quad (2.2)$$

$$\dot{\chi} = \frac{g}{V \cos \gamma} u_2 \quad (2.3)$$

$$\dot{\gamma} = \frac{g}{V}(u_3 - \cos \gamma) \quad (2.4)$$

where u_1 , u_2 , and u_3 are non-dimensional force terms with respect to the axis of the UAV's body frame, respectively [47].

Let us define a command set $(V_{cmd}, \chi_{cmd}, \gamma_{cmd})$. The objective of control is to make the UAV track the corresponding command value. For this, the following control inputs can be proposed.

$$u_1 = \sin \gamma + \frac{\dot{V}_{cmd} - k_V(V - V_{cmd})}{g} \quad (2.5)$$

$$u_2 = \frac{\dot{\chi}_{cmd} - k_\chi(\chi - \chi_{cmd})}{g} V \cos \gamma \quad (2.6)$$

$$u_3 = \cos \gamma + \frac{\dot{\gamma}_{cmd} - k_\gamma(\gamma - \gamma_{cmd})}{g} V \quad (2.7)$$

where k_V , k_χ , and k_γ are positive constant gains. Stability of the closed-loop system can be easily proved by using the Lyapunov stability theorem [53, 54].

Substituting Eqs. (2.5)-(2.7) into Eqs. (2.2)-(2.4) yields

$$\dot{V} = \dot{V}_{cmd} - k_V(V - V_{cmd}) \quad (2.8)$$

$$\dot{\chi} = \dot{\chi}_{cmd} - k_\chi(\chi - \chi_{cmd}) \quad (2.9)$$

$$\dot{\gamma} = \dot{\gamma}_{cmd} - k_\gamma(\gamma - \gamma_{cmd}) \quad (2.10)$$

Equations (2.8)-(2.10) can be regarded as a simplified autopilot

dynamics of the UAV. Note that a more detailed UAV model is the 6-DOF (degree of freedom) nonlinear model with aerodynamics coefficients [45, 46]. Usually, conventional PID controllers can be adopted to design the autopilot of the UAV. Otherwise, a high performance controller for the 6-DOF nonlinear aircraft model should be designed by using nonlinear control methods such as sliding mode control, backstepping control, etc.

In this study, guidance and collision avoidance of the UAV are mainly focused, and therefore the simplified autopilot model is considered. In addition to the autopilot dynamics, the following constraints on state variables are considered.

$$0 < V_{\min} \leq V \leq V_{\max} \quad (2.11)$$

$$-\gamma_{\max} \leq \gamma \leq \gamma_{\max} \quad (2.12)$$

$$-\dot{V}_{\max} \leq \dot{V} \leq \dot{V}_{\max} \quad (2.13)$$

$$-\dot{\chi}_{\max} \leq \dot{\chi} \leq \dot{\chi}_{\max} \quad (2.14)$$

$$-\dot{\gamma}_{\max} \leq \dot{\gamma} \leq \dot{\gamma}_{\max} \quad (2.15)$$

Equations (2.11) and (2.12) are from the kinematic characteristics of the fixed-wing UAV, which are related to the stall of the aircraft. Equations (2.13)-(2.15) are from the dynamic constraints of the UAV.

2.2 Guidance Problems of Unmanned Aerial Vehicles

In this section, three guidance problems are considered to make a UAV achieve the desired tracking performance.

2.2.1 Waypoint Tracking Problem

To design a UAV guidance, a waypoint tracking problem can be considered, which is a conventional guidance problem of UAV. In the waypoint tracking problem, a set of waypoints is defined, and guidance commands are generated to make UAV track each waypoint one by one.

Let us define the position vector of the UAV, \mathbf{r} , and the position vector of the waypoint, \mathbf{r}_w , as follows.

$$\mathbf{r} \triangleq [x \ y \ z]^T \quad (2.16)$$

$$\mathbf{r}_w \triangleq [x_w \ y_w \ z_w]^T \quad (2.17)$$

Figure 2.2 shows a simple configuration of the waypoint tracking problem. As shown in this configuration, the UAV only has to head its

direction to the direction of the waypoint. After reaching the waypoint, the waypoint is switched to the next waypoint. The command set for the waypoint tracking can be designed as follows.

$$\chi_{cmd} = \tan^{-1} \left(\frac{y_w - y}{x_w - x} \right) \quad (2.18)$$

$$\gamma_{cmd} = \sin^{-1} \left(\frac{-(z_w - z)}{\|\mathbf{r}_w - \mathbf{r}\|} \right) \quad (2.19)$$

In this case, the speed command, V_{cmd} , can be taken as an arbitrary constant satisfying Eq. (2.11). Usually, it is set as the cruise speed of the UAV.

2.2.2 Waypoint Tracking with Line Tracing Problem

Now, let us consider another waypoint tracking problem. Figure 2.3 shows a configuration of the waypoint tracking with line tracing problem. This problem is more difficult than the simple waypoint tracking problem because the UAV should be aligned with the virtual line between the previous waypoint and current waypoint while heading the waypoint.

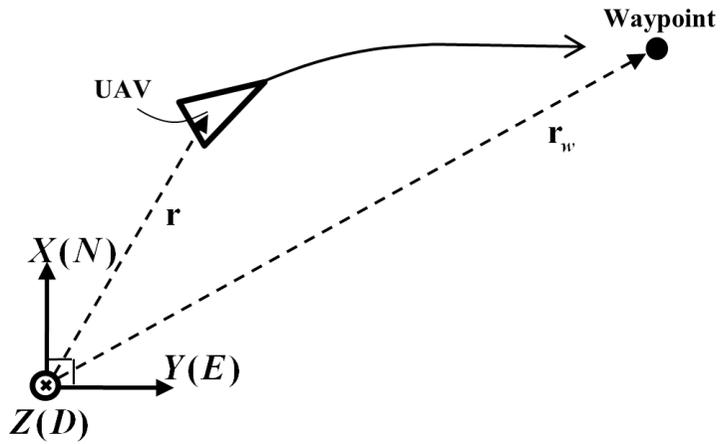


Figure 2.2 Configuration of waypoint tracking

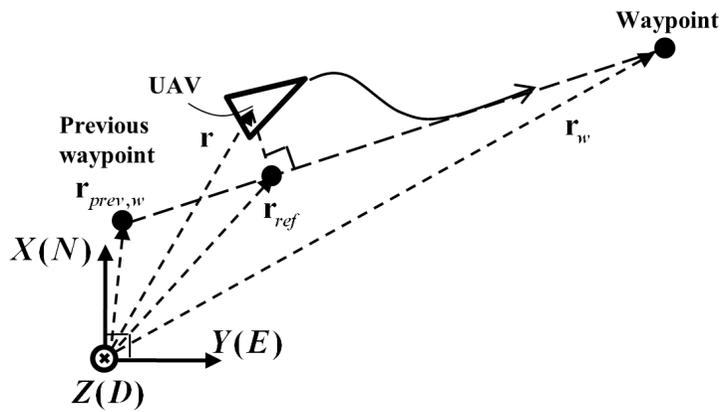


Figure 2.3 Configuration of waypoint tracking with line tracing

In Fig. 2.3, $\mathbf{r}_{prev,w}$ is the position of the previous waypoint, which is defined as follows.

$$\mathbf{r}_{prev,w} \triangleq [x_{prev,w} \ y_{prev,w} \ z_{prev,w}]^T \quad (2.20)$$

Let us consider a reference position of the UAV, which is on the line between the previous waypoint and the current waypoint as

$$\mathbf{r}_{ref} \triangleq [x_{ref} \ y_{ref} \ z_{ref}]^T \quad (2.21)$$

If the vector $\mathbf{r} - \mathbf{r}_{prev,w}$ is projected on the line, the reference position, which is the projected position of the UAV, can be obtained as follows.

$$\mathbf{r}_{ref} = \mathbf{r}_{prev,w} + \frac{(\mathbf{r} - \mathbf{r}_{prev,w})^T (\mathbf{r}_w - \mathbf{r}_{prev,w})}{\|\mathbf{r}_w - \mathbf{r}_{prev,w}\|^2} (\mathbf{r}_w - \mathbf{r}_{prev,w}) \quad (2.22)$$

Using a reference command set $(V_{ref}, \chi_{ref}, \gamma_{ref})$ and Eq. (2.1), the reference velocity of the UAV can be represented as follows.

$$\mathbf{V}_{ref} \triangleq \begin{bmatrix} V_{ref} \cos \gamma_{ref} \cos \chi_{ref} \\ V_{ref} \cos \gamma_{ref} \sin \chi_{ref} \\ -V_{ref} \sin \gamma_{ref} \end{bmatrix} \quad (2.23)$$

Since the reference velocity should be headed to the current waypoint,

χ_{ref} and γ_{ref} can be set as

$$\chi_{ref} = \tan^{-1} \left(\frac{y_w - y_{prev,w}}{x_w - x_{prev,w}} \right) \quad (2.24)$$

$$\gamma_{ref} = \sin^{-1} \left(\frac{-(z_w - z_{prev,w})}{\|\mathbf{r}_w - \mathbf{r}_{prev,w}\|} \right) \quad (2.25)$$

Again, V_{ref} can be set as a cruise speed of the UAV. If the position and velocity of the UAV correspond with the reference position and velocity in Eqs. (2.22) and (2.23), the tracking satisfies the alignment constraint and finally the UAV will head to the waypoint.

2.2.3 Trajectory Tracking Problem

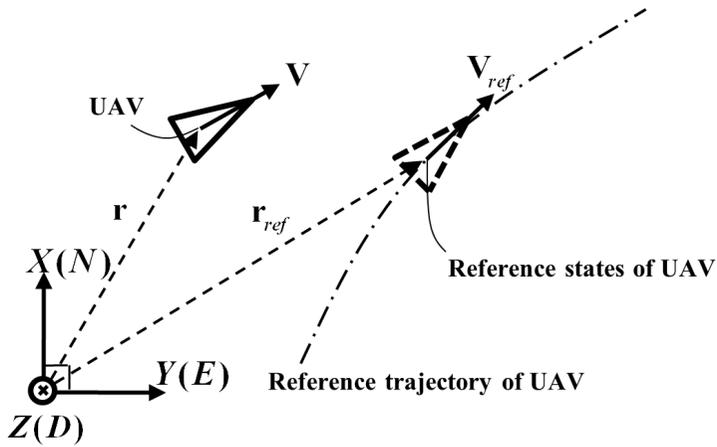


Figure 2.4 Configuration of trajectory tracking

Similar to the waypoint tracking with line tracing problem, a trajectory tracking problem can be considered. This problem is a generalized version of the waypoint tracking problem. Figure 2.4 shows the configuration of the trajectory tracking problem.

As shown in Fig. 2.4, the trajectory tracking is achieved when the position and velocity of the UAV are identical to the reference position and velocity on the trajectory. The reference trajectory consists of position, direction, and curvature information. At the reference position, \mathbf{r}_{ref} , on the trajectory, the reference heading angle, χ_{ref} , and flight path angle, γ_{ref} , provide the direction information. If the reference speed, V_{ref} , is set, then the reference velocity vector, \mathbf{V}_{ref} , can be obtained.

Now, let us consider a horizontal curvature radius, ρ_H , and a vertical curvature radius, ρ_V , respectively. The curvature radii have the following relations.

$$\rho_H = \frac{V_{ref} \cos \gamma_{ref}}{\dot{\chi}_{ref}} \quad (2.26)$$

$$\rho_V = \frac{V_{ref}}{\dot{\gamma}_{ref}} \quad (2.27)$$

Using Eqs. (2.26) and (2.27), $\dot{\chi}_{ref}$ and $\dot{\gamma}_{ref}$ can be obtained for a specified reference trajectory $(V_{ref}, \rho_H, \rho_V)$. Note that the reference position, velocity, and acceleration vectors on the trajectory can be obtained from the information of the trajectory including the position, direction, and curvature, which will be used to generate a guidance law of the UAV.

2.3 Command Tracking Guidance Law

2.3.1 Guidance Command

The UAV guidance is focused on command generation related to the desired speed, heading angle, and flight path angle $(V_{cmd}, \chi_{cmd}, \gamma_{cmd})$, which makes the states of the UAV converge to the reference states on the trajectory. Note that speed, heading angle, and flight path angle are components of the velocity vector.

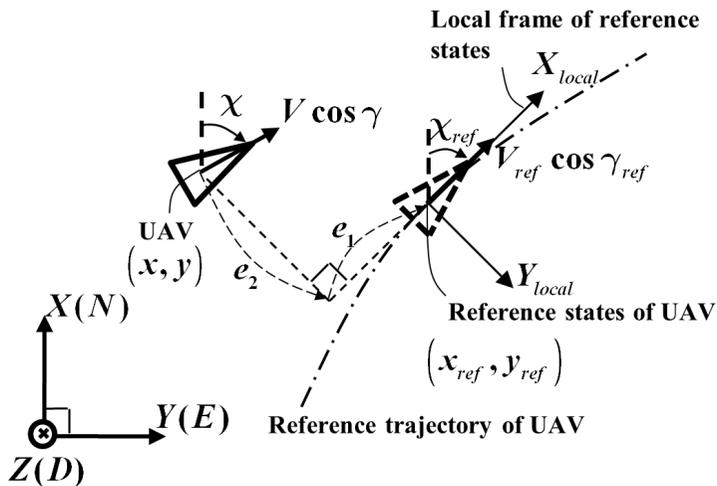


Figure 2.5 Configuration of the states of the UAV and the reference states of the UAV

Figure 2.5 shows the configuration of the states of the UAV and the reference states. Let us define the error vector as follows.

$$\mathbf{e} = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} \triangleq \begin{bmatrix} \cos \chi_{ref} & \sin \chi_{ref} & 0 \\ -\sin \chi_{ref} & \cos \chi_{ref} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{ref} - x \\ y_{ref} - y \\ z_{ref} - z \end{bmatrix} \quad (2.28)$$

Differentiating Eq. (2.28) with respect to time, and using Eqs. (2.1) and (2.23) in the resulting equation yield the following equation.

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} 0 & \dot{\chi}_{ref} & 0 \\ -\dot{\chi}_{ref} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} + \begin{bmatrix} V_{ref} \cos \gamma_{ref} \\ 0 \\ -V_{ref} \sin \gamma_{ref} \end{bmatrix} - \begin{bmatrix} V \cos \gamma \cos(\chi_{ref} - \chi) \\ -V \cos \gamma \sin(\chi_{ref} - \chi) \\ -V \sin \gamma \end{bmatrix} \quad (2.29)$$

Consider the following desired error dynamics.

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} -k_1 & \dot{\chi}_{ref} & 0 \\ -\dot{\chi}_{ref} & -k_2 & 0 \\ 0 & 0 & -k_3 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} \quad (2.30)$$

where k_1 , k_2 , and k_3 are positive constants. If the error dynamics satisfies Eq. (2.30), then the errors converge to zeros because the matrix in Eq. (2.30) has negative eigenvalues. Let us consider a command set $(V_{cmd}, \chi_{cmd}, \gamma_{cmd})$ satisfying the desired error dynamics.

Substituting Eq. (2.30) into Eq. (2.29) yields the following relation.

$$\begin{bmatrix} V_{ref} \cos \gamma_{ref} \\ 0 \\ -V_{ref} \sin \gamma_{ref} \end{bmatrix} - \begin{bmatrix} V \cos \gamma \cos(\chi_{ref} - \chi) \\ -V \cos \gamma \sin(\chi_{ref} - \chi) \\ -V \sin \gamma \end{bmatrix} = - \begin{bmatrix} k_1 e_1 \\ k_2 e_2 \\ k_3 e_3 \end{bmatrix} \quad (2.31)$$

If (V, χ, γ) satisfies Eq. (2.31), then the error dynamics converge to zero. Therefore, the guidance command can be generated to satisfy Eq. (2.31). By making the states of the UAV follow the guidance command, the UAV will move to the reference trajectory.

From Eq. (2.31), each component of the command set can be calculated as follows.

$$V_{cmd} = \sqrt{(V_{ref} \cos \gamma_{ref} + k_1 e_1)^2 + (k_2 e_2)^2 + (-V_{ref} \sin \gamma_{ref} + k_3 e_3)^2} \quad (2.32)$$

$$\chi_{cmd} = \chi_{ref} + \tan^{-1} \left(\frac{k_2 e_2}{V_{ref} \cos \gamma_{ref} + k_1 e_1} \right) \quad (2.33)$$

$$\gamma_{cmd} = \sin^{-1} \left(\frac{V_{ref} \sin \gamma_{ref} - k_3 e_3}{V_{cmd}} \right) \quad (2.34)$$

Note that V_{cmd} is the speed command, χ_{cmd} is the heading angle command, and γ_{cmd} is the flight path angle command.

Using the speed, heading angle, and flight path angle commands, the UAV can easily perform acceleration or steering under nonholonomic constraints such as speed and turn rate limits.

For a kinematically constrained UAV, k_1 , k_2 , and k_3 can be set as follows.

$$k_1 = \frac{a_1}{\sqrt{b_1^2 + e_1^2}} \quad (2.35)$$

$$k_2 = \frac{a_2}{\sqrt{b_2^2 + e_2^2}} \quad (2.36)$$

$$k_3 = \frac{a_3}{\sqrt{b_3^2 + e_3^2}} \quad (2.37)$$

where a_i and b_i are positive constants. Note that $k_1 e_1$, $k_2 e_2$, and $k_3 e_3$ are sigmoid functions with respect to e_1 , e_2 , and e_3 , respectively. By defining $K_1 \triangleq k_1 e_1$, $K_2 \triangleq k_2 e_2$, and $K_3 \triangleq k_3 e_3$, the command set $(V_{cmd}, \chi_{cmd}, \gamma_{cmd})$ can be rewritten as

$$V_{cmd} = \sqrt{(V_{ref} \cos \gamma_{ref} + K_1)^2 + K_2^2 + (-V_{ref} \sin \gamma_{ref} + K_3)^2} \quad (2.38)$$

$$\chi_{cmd} = \chi_{ref} + \tan^{-1} \left(\frac{K_2}{V_{ref} \cos \gamma_{ref} + K_1} \right) \quad (2.39)$$

$$\gamma_{cmd} = \sin^{-1} \left(\frac{V_{ref} \sin \gamma_{ref} - K_3}{V_{cmd}} \right) \quad (2.40)$$

Note that K_i is a bounded value in the range of $[-a_i, a_i]$, and therefore the components in Eqs. (2.38)-(2.40) have the following

bounds.

$$V_{ref} \cos \gamma_{ref} - a_1 \leq V_{ref} \cos \gamma_{ref} + K_1 \leq V_{ref} \cos \gamma_{ref} + a_1 \quad (2.41)$$

$$0 \leq K_2^2 \leq a_2^2 \quad (2.42)$$

$$\begin{aligned} V_{ref} \sin \gamma_{ref} - a_3 \operatorname{sgn}(\gamma_{ref}) &\leq V_{ref} \sin \gamma_{ref} - K_3 \\ &\leq V_{ref} \sin \gamma_{ref} + a_3 \operatorname{sgn}(\gamma_{ref}) \end{aligned} \quad (2.43)$$

Then, the commands have the following bounds for given reference states.

$$V_{cmd} \geq \sqrt{(V_{ref} \cos \gamma_{ref} - a_1)^2 + (V_{ref} \sin \gamma_{ref} - a_3 \operatorname{sgn}(\gamma_{ref}))^2} \quad (2.44)$$

$$\begin{aligned} V_{cmd} &\leq \sqrt{(V_{ref} \cos \gamma_{ref} + a_1)^2 + a_2^2 + (V_{ref} \sin \gamma_{ref} + a_3 \operatorname{sgn}(\gamma_{ref}))^2} \\ &\quad (2.45) \end{aligned}$$

$$\chi_{cmd} \geq \chi_{ref} - \tan^{-1} \left(\frac{a_2}{V_{ref} \cos \gamma_{ref} - a_1} \right) \quad (2.46)$$

$$\chi_{cmd} \leq \chi_{ref} + \tan^{-1} \left(\frac{a_2}{V_{ref} \cos \gamma_{ref} - a_1} \right) \quad (2.47)$$

$$\begin{aligned} \gamma_{cmd} &\geq -\sin^{-1} \left(\frac{V_{ref} \sin \gamma_{ref} + a_3 \operatorname{sgn}(\gamma_{ref})}{\sqrt{(V_{ref} \cos \gamma_{ref} - a_1)^2 + (V_{ref} \sin \gamma_{ref} + a_3 \operatorname{sgn}(\gamma_{ref}))^2}} \right) \\ &\quad (2.48) \end{aligned}$$

$$\gamma_{cmd} \leq \sin^{-1} \left(\frac{V_{ref} \sin \gamma_{ref} + a_3 \operatorname{sgn}(\gamma_{ref})}{\sqrt{(V_{ref} \cos \gamma_{ref} - a_1)^2 + (V_{ref} \sin \gamma_{ref} + a_3 \operatorname{sgn}(\gamma_{ref}))^2}} \right) \quad (2.49)$$

Equations (2.46)-(2.49) can be obtained from trigonometric configurations of the components in Eqs. (2.41)-(2.43). If the reference trajectory is properly designed and bounded, then the commands are also bounded according to Eqs. (2.44)-(2.49). Finally, the UAV can be controlled in accordance with the bounded command set.

2.3.2 Stability Analysis

Using the autopilot of the UAV, Eqs. (2.8)-(2.10), the states of the UAV (V, χ, γ) converge to the commands $(V_{cmd}, \chi_{cmd}, \gamma_{cmd})$. Then, the position errors (e_1, e_2, e_3) converge to zeros due to the error dynamics in Eq. (2.30), which can be also proved by the Lyapunov stability theorem [53, 54]. Let us consider the following Lyapunov candidate function.

$$L = \frac{1}{2}(e_1^2 + e_2^2 + e_3^2) \quad (2.50)$$

The derivative of the Lyapunov candidate function can be obtained as

$$\dot{L} = e_1\dot{e}_1 + e_2\dot{e}_2 + e_3\dot{e}_3 = -k_1e_1^2 - k_2e_2^2 - k_3e_3^2 \quad (2.51)$$

Except for the equilibrium point, $\mathbf{e} = \mathbf{0}$, $L > 0$ and $\dot{L} < 0$. At the equilibrium point, $L = 0$ and $\dot{L} = 0$. Therefore, the equilibrium point is asymptotically stable according to the Lyapunov stability theorem.

Note that the commands $(V_{cmd}, \chi_{cmd}, \gamma_{cmd})$ converge to the original reference states $(V_{ref}, \chi_{ref}, \gamma_{ref})$ when (e_1, e_2, e_3) converge to zeros.

2.4 Simulations on Guidance

To demonstrate the effectiveness of the guidance law, numerical simulations are performed. The speed of the UAV is set as 30m/s. The state constraints and other parameters are set as shown in Table 2.1. Constants k_V , k_χ , and k_γ in Eqs. (2.8)-(2.10) are set considering the point-mass model of the UAV. Note that the time constants of the UAV are inversely proportional to k_V , k_χ , and k_γ , respectively. According to the time constants, each error decreases about 95% after 3 seconds.

Table 2.1 Parameters used in the UAV guidance

UAV model and guidance	
Speed of UAV	$V = 30 \text{ m/s}$
Model constants	$k_V = k_\chi = k_\gamma = 1$ $V_{\max} = 40 \text{ m/s}, V_{\min} = 20 \text{ m/s}$
State constraints	$\dot{V}_{\max} = 2 \text{ m/s}^2, \dot{\chi}_{\max} = 30 \text{ deg/s}$ $\gamma_{\max} = 30 \text{ deg}, \dot{\gamma}_{\max} = 30 \text{ deg/s}$
Guidance command	$a_1 = a_2 = a_3 = 8$ $b_1 = b_2 = b_3 = 8$

In the first simulation, waypoint tracking is considered. Waypoint 1 is the previous waypoint, and Waypoint 2 is the current waypoint of the UAV. That is, the UAV is headed to Waypoint 2. In this scenario, the guidance commands are determined using Eqs. (2.16)-(2.19). Numerical result is shown in Fig. 2.6 In Fig. 2.6, it can be seen that the trajectory of the UAV is deviated from the line between Waypoint 1 and Waypoint 2 during the turn maneuver.

In the second simulation, waypoint tracking with line tracing is considered. In this problem, the UAV should be aligned to the line between the previous waypoint and the current waypoint. The reference speed V_{ref} is set as the cruise speed of the UAV, and other reference states are obtained by using Eqs. (2.21)-(2.25). Numerical results are shown in Figs. 2.7-2.9. Figure 2.7 shows the trajectory of the waypoint tracking with line tracing. The position error between the UAV and the reference is shown in Fig. 2.8. It can be seen that the UAV effectively passes the waypoint while tracking the virtual line. As shown in Fig. 2.9, the states of the UAV and commands converge to the reference states of the UAV.

In the third simulation, trajectory tracking is considered. Figures 2.10 and 2.11 show numerical results of a circular trajectory tracking.

Radius of the circular trajectory is set as 150m, and V_{ref} is set as the cruise speed of the UAV. Then, reference states on the circular trajectory can be defined using tangential directions and curvature radii of the trajectory. As shown in Fig. 2.11, the position error between the UAV and the reference converge to zero. According to the proposed command tracking guidance law, it can be stated that the UAV can track the trajectory without divergence by defining the reference position, velocity, and acceleration vectors on the trajectory.

Performance of the proposed guidance law can be compared with a conventional path tracking method called as the follow-the-carrot method [51, 52]. The follow-the-carrot method considers a look-ahead point, and UAV is guided to head the point. Figures 2.12-2.15 show numerical results of the line tracing problem and the trajectory tracking problem. The look-ahead distance, which denotes distance between the look-ahead point and the position of the UAV, is set as 30m. In the simulations, it can be seen that the follow-the-carrot method does not guarantee convergence with the path when the path has curvature, which is different to the proposed guidance law. Therefore, the proposed guidance law has merits in terms of convergence with a desired trajectory or path.

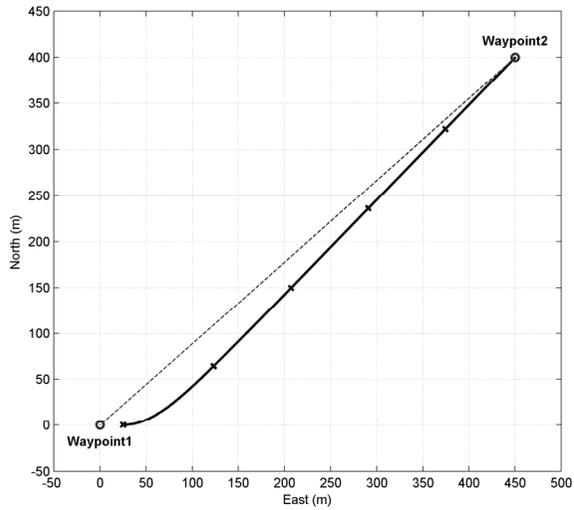


Figure 2.6 Trajectory of the waypoint tracking (Command tracking guidance law)

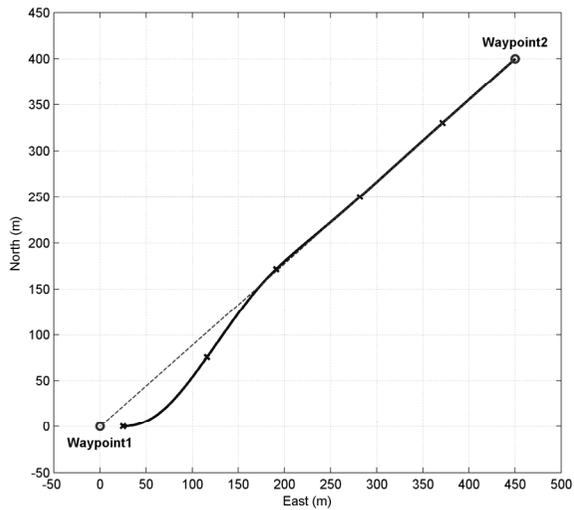


Figure 2.7 Trajectory of the waypoint tracking with line tracing (Command tracking guidance law)

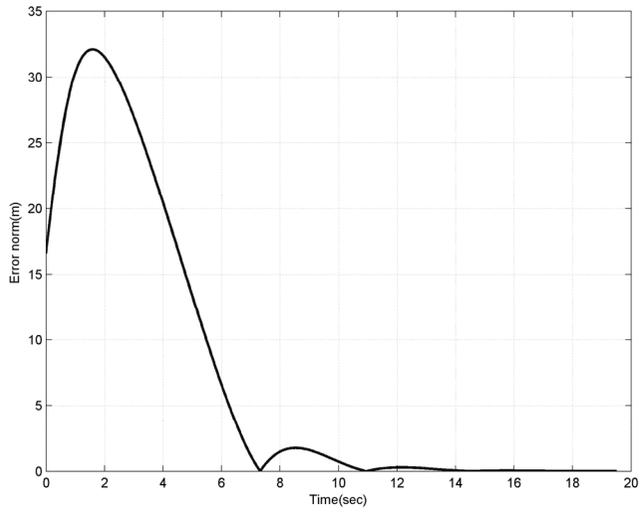


Figure 2.8 Position error between the UAV and the reference on the line (Command tracking guidance law)

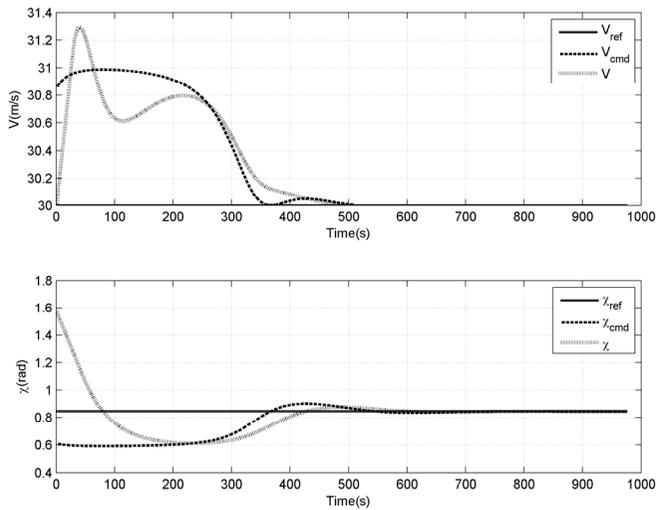


Figure 2.9 Time histories of reference-command-state (Command tracking guidance law)

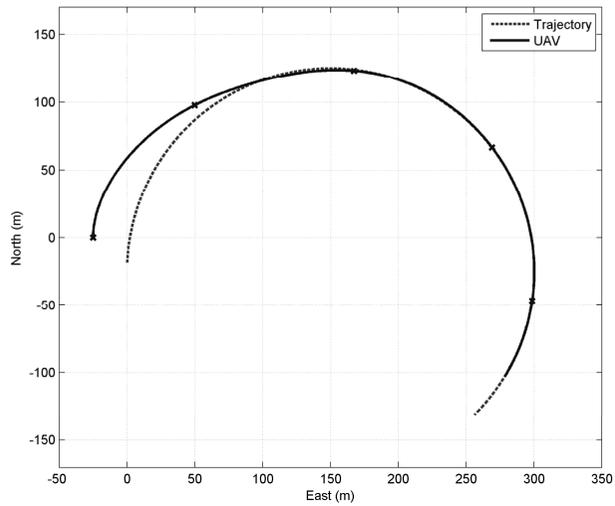


Figure 2.10 Trajectory of a circular trajectory tracking (Command tracking guidance law)

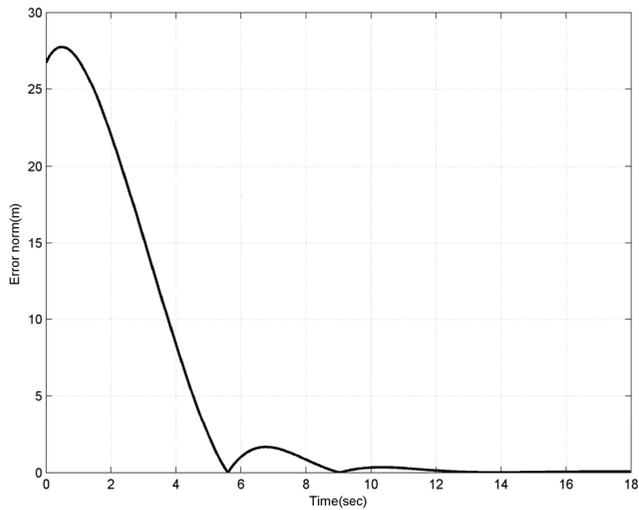


Figure 2.11 Position error between the UAV and the reference on the circular trajectory (Command tracking guidance law)

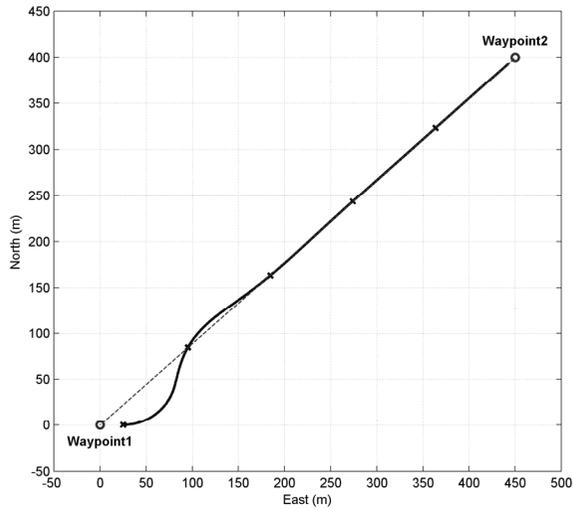


Figure 2.12 Trajectory of the waypoint tracking with line tracing (Follow-the-carrot)

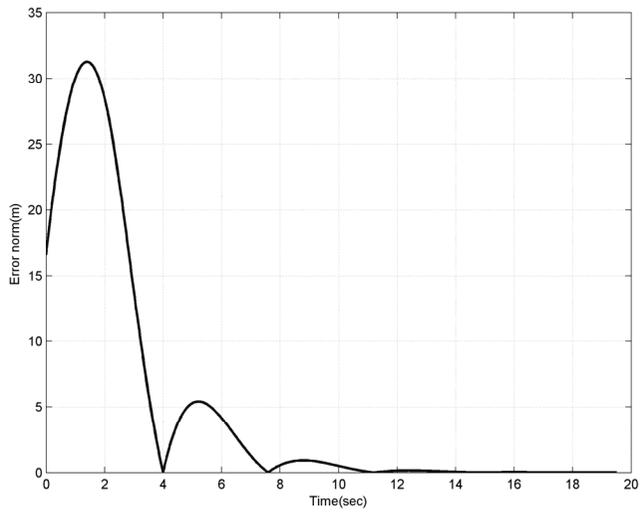


Figure 2.13 Position error between the UAV and the reference on the line (Follow-the-carrot)

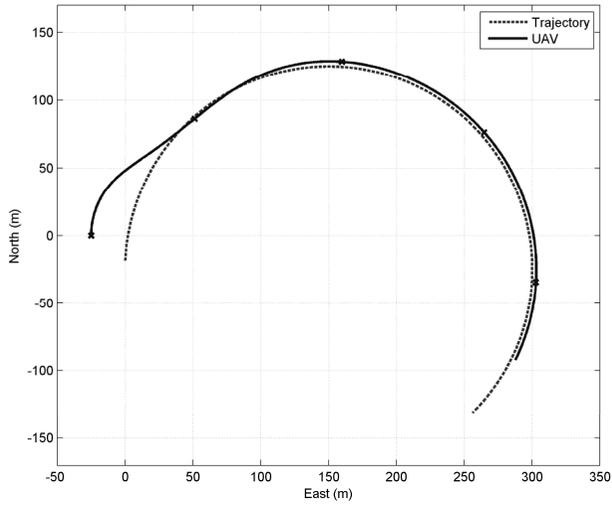


Figure 2.14 Trajectory of a circular trajectory tracking (Follow-the-carrot)

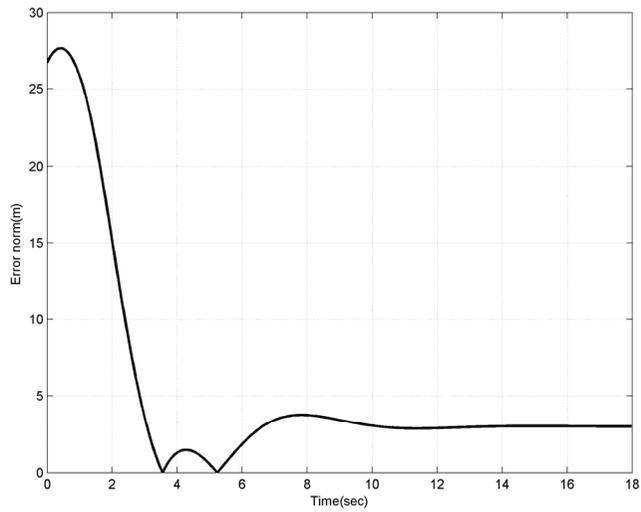


Figure 2.15 Position error between the UAV and the reference on the circular trajectory (Follow-the-carrot)

Chapter 3

Reactive Collision Avoidance

3.1 Collision Detection

During the last few decades, lots of studies on geometric collision avoidance approaches have been performed to avoid dynamic obstacles. Collision cone [20], velocity obstacle [21], and geometric optimization [24] methods are representative geometric collision avoidance approaches. These methods adopt the following assumption for collision detection because aircraft cruise at a constant speed for most of the flying time.

Assumption 3.1. The velocities of vehicles and obstacles are assumed to be constant.

Under Assumption 3.1, the collision detection schemes based on the collision cone have been developed, which can be interpreted in the concept of CPA (closest-point-of-approach). Using the current

position and velocity of the obstacle, one's own vehicle can predict the future states of the two aircraft, namely itself and the intruder, by dynamic propagation. The closest point can be calculated using the predicted states, which will be explained as follows

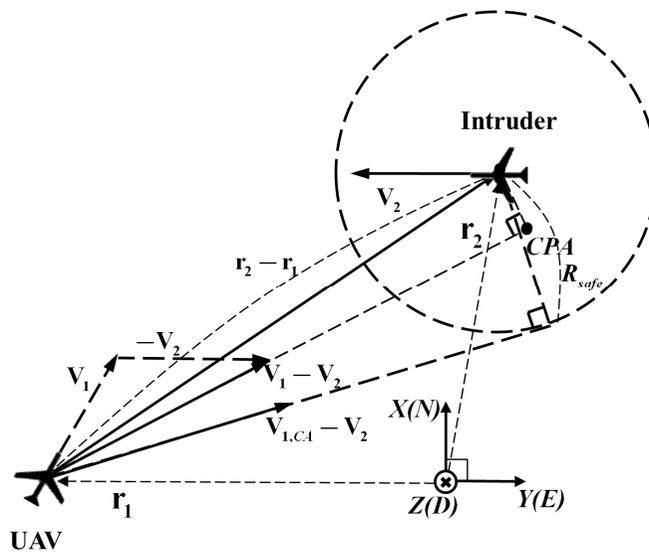


Figure 3.1 Relative geometry of the UAV and intruder

Let us consider a collision situation that consists of a UAV and an intruder as shown in Fig. 3.1. Collision detection is performed by considering the collision cone geometry. In Fig. 3.1, \mathbf{r}_1 and \mathbf{V}_1 are defined as the position and velocity vectors of the UAV, and \mathbf{r}_2 and \mathbf{V}_2 are defined as the corresponding vectors of the intruder. Let us

define the current relative position and velocity of the intruder with respect to the UAV as

$$\tilde{\mathbf{r}} \triangleq \mathbf{r}_2 - \mathbf{r}_1 \quad (3.1)$$

$$\tilde{\mathbf{V}} \triangleq \mathbf{V}_2 - \mathbf{V}_1 \quad (3.2)$$

The CPA positions of the UAV and the intruder are defined as $\mathbf{r}_{1,CPA}$ and $\mathbf{r}_{2,CPA}$, respectively, which are obtained by extrapolating the trajectories of the two aircraft from the current states. The time of the CPA, t_{CPA} , can be obtained by calculating the time at which the relative distance between the UAV and the intruder has a minimum value. Therefore, the closest point between the two aircraft satisfies the following condition.

$$\frac{d}{dt} \left(\|\tilde{\mathbf{r}}(t)\|^2 \right)_{t=t_{CPA}} = 0 \quad (3.3)$$

By setting current time to be zero, the above condition can be rewritten as follows under Assumption 3.1.

$$\tilde{\mathbf{r}}^T(t_{CPA}) \dot{\tilde{\mathbf{r}}}(t_{CPA}) = \left(\tilde{\mathbf{r}} + \tilde{\mathbf{V}} t_{CPA} \right)^T \tilde{\mathbf{V}} = 0 \quad (3.4)$$

Then, t_{CPA} can be obtained as

$$t_{CPA} = -\frac{\tilde{\mathbf{r}}^T \tilde{\mathbf{V}}}{\tilde{\mathbf{V}}^T \tilde{\mathbf{V}}} \quad (3.5)$$

Now, the CPA positions of the two aircraft can be obtained as follows.

$$\mathbf{r}_{1,CPA} = \mathbf{r}_1 + \mathbf{V}_1 t_{CPA} \quad (3.6)$$

$$\mathbf{r}_{2,CPA} = \mathbf{r}_2 + \mathbf{V}_2 t_{CPA} \quad (3.7)$$

Assume that the current distance between the two aircraft is $d = \|\tilde{\mathbf{r}}\|$.

Then, the norm of the relative distance between the two aircraft at time t_{CPA} can be obtained as follows.

$$\begin{aligned} \|\mathbf{r}_{2,CPA} - \mathbf{r}_{1,CPA}\|^2 &= \|\tilde{\mathbf{r}} + \tilde{\mathbf{V}} t_{CPA}\|^2 = \tilde{\mathbf{r}}^T \tilde{\mathbf{r}} - \tilde{\mathbf{r}}^T \tilde{\mathbf{V}} \frac{\tilde{\mathbf{r}}^T \tilde{\mathbf{V}}}{\tilde{\mathbf{V}}^T \tilde{\mathbf{V}}} \\ &= d^2 - \|\tilde{\mathbf{V}} t_{CPA}\|^2 \end{aligned} \quad (3.8)$$

Note that Eq. (3.8) provides the square of the closest distance.

To define the situation of the collision, a collision criterion related to a safety radius, R_{safe} , should be considered. If the closest distance is less than the safety radius for a non-negative t_{CPA} , it is declared that each aircraft is in the situation of a collision. This criterion can be formulated as the following inequality equation for $t_{CPA} \geq 0$.

$$\|\mathbf{r}_{2,CPA} - \mathbf{r}_{1,CPA}\|^2 = d^2 - \|\tilde{\mathbf{V}} t_{CPA}\|^2 < R_{safe}^2 \quad (3.9)$$

Note from Fig. 3.1 that collision detection based on the CPA is equivalent to the conventional geometric approaches.

3.2 Solutions for Collision Avoidance

When a collision is detected according to Eq. (3.9), the UAV should steer to other directions to avoid the collision. The reactive collision avoidance can be achieved by performing the collision detection and avoidance process in real-time. Thus, the collision avoidance can be described as a problem that finds $\mathbf{V}_{1,CA}$ which satisfies the following condition whenever the collision is detected:

$$d^2 - \|\tilde{\mathbf{V}}_{CA} t_{CPA,CA}\|^2 \geq R_{safe}^2 \quad (3.10)$$

where $\tilde{\mathbf{V}}_{CA} \triangleq \mathbf{V}_2 - \mathbf{V}_{1,CA}$ and $t_{CPA,CA} = -\frac{\tilde{\mathbf{r}}^T \tilde{\mathbf{V}}_{CA}}{\tilde{\mathbf{V}}_{CA}^T \tilde{\mathbf{V}}_{CA}}$.

To deal with this problem, the safety boundary projection concept as shown in Fig. 3.2 is adopted [32]. Let us project a spherical boundary of the intruder on the path of the UAV. Then, the UAV can easily find the direction for collision avoidance. To do this, an explicit formulation of $\mathbf{V}_{1,CA}$ is required.

Using Eq. (3.5), the following equation can be obtained.

$$\|\tilde{\mathbf{V}} t_{CPA}\|^2 = \frac{(\tilde{\mathbf{V}}^T \tilde{\mathbf{r}})(\tilde{\mathbf{r}}^T \tilde{\mathbf{V}})}{\tilde{\mathbf{V}}^T \tilde{\mathbf{V}}} \quad (3.11)$$

where $\tilde{\mathbf{r}}$ and $\tilde{\mathbf{V}}$ are the relative position and velocity vectors at the

current time. Substituting Eq. (3.11) into Eq. (3.9) gives the following inequality condition.

$$\tilde{\mathbf{V}}^T \mathbf{Q} \tilde{\mathbf{V}} > 0 \quad (3.12)$$

where

$$\mathbf{Q} = \tilde{\mathbf{r}}\tilde{\mathbf{r}}^T - (d^2 - R_{safe}^2)\mathbf{I}$$

$$= \begin{bmatrix} R_{safe}^2 - \tilde{y}^2 - \tilde{z}^2 & \tilde{x}\tilde{y} & \tilde{x}\tilde{z} \\ \tilde{y}\tilde{x} & R_{safe}^2 - \tilde{x}^2 - \tilde{z}^2 & \tilde{y}\tilde{z} \\ \tilde{z}\tilde{x} & \tilde{z}\tilde{y} & R_{safe}^2 - \tilde{x}^2 - \tilde{y}^2 \end{bmatrix} \quad (3.13)$$

with $\tilde{\mathbf{r}} = (\tilde{x}, \tilde{y}, \tilde{z})^T$, and $d = \sqrt{\tilde{x}^2 + \tilde{y}^2 + \tilde{z}^2}$. Substituting Eq. (3.2) into Eq. (3.12), we have

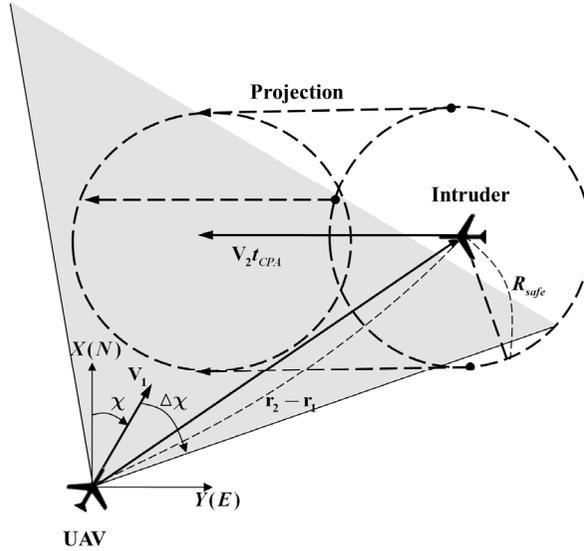


Figure 3.2 Safety boundary projection

$$\mathbf{V}_1^T \mathbf{Q} \mathbf{V}_1 - 2\mathbf{V}_2^T \mathbf{Q} \mathbf{V}_1 + \mathbf{V}_2^T \mathbf{Q} \mathbf{V}_2 > 0 \quad (3.14)$$

Therefore, the safety boundary satisfies the following condition.

$$\mathbf{V}_{1,CA}^T \mathbf{Q} \mathbf{V}_{1,CA} - 2\mathbf{V}_2^T \mathbf{Q} \mathbf{V}_{1,CA} + \mathbf{V}_2^T \mathbf{Q} \mathbf{V}_2 = 0 \quad (3.15)$$

Note from Eq. (3.15) that the safety boundary has a surface of quadratic form. Points on the safety boundary can be candidate solutions for the collision avoidance.

The speed of the UAV is V_1 , that is the velocity of the UAV should satisfy the following equation.

$$\mathbf{V}_{1,CA}^T \mathbf{V}_{1,CA} = V_1^2 \quad (3.16)$$

Therefore, the cross-sectional region between the quadratic surface, Eq. (3.15), and the sphere, Eq. (3.16), are candidates for the solution of the collision avoidance.

Considering the relative position $\tilde{\mathbf{r}}$, \mathbf{Q} in Eq. (3.13) could have various forms. As a result of this, it is not easy to obtain the safety boundary. To deal with this difficulty, the following Euler rotation matrices [46] are adopted.

$$\mathbf{C}(\gamma, 2) \triangleq \begin{bmatrix} \cos \gamma & 0 & -\sin \gamma \\ 0 & 1 & 0 \\ \sin \gamma & 0 & \cos \gamma \end{bmatrix} \quad (3.17)$$

$$\mathbf{C}(\chi, 3) \triangleq \begin{bmatrix} \cos \chi & \sin \chi & 0 \\ -\sin \chi & \cos \chi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.18)$$

Let us consider a coordinate transformation with respect to the line-of-sight (LOS) angles of the intruder. Through the coordinate transformation, the relative position and velocity vectors with respect to the LOS angles can be obtained as follows.

$$\tilde{\mathbf{r}}_{LOS} \triangleq \mathbf{C}(\gamma_{LOS}, 2)\mathbf{C}(\chi_{LOS}, 3)\tilde{\mathbf{r}} = [d \ 0 \ 0]^T \quad (3.19)$$

$$\mathbf{V}_{1,CA,LOS} \triangleq \mathbf{C}(\gamma_{LOS}, 2)\mathbf{C}(\chi_{LOS}, 3)\mathbf{V}_{1,CA} \triangleq [u_1 \ v_1 \ w_1]^T \quad (3.20)$$

$$\mathbf{V}_{2,LOS} \triangleq \mathbf{C}(\gamma_{LOS}, 2)\mathbf{C}(\chi_{LOS}, 3)\mathbf{V}_2 \triangleq [u_2 \ v_2 \ w_2]^T \quad (3.21)$$

Then, corresponding \mathbf{Q}_{LOS} becomes a diagonal matrix as follows.

$$\mathbf{Q}_{LOS} = \begin{bmatrix} R_{safe}^2 & 0 & 0 \\ 0 & R_{safe}^2 - d^2 & 0 \\ 0 & 0 & R_{safe}^2 - d^2 \end{bmatrix} \quad (3.22)$$

Substituting Eqs. (3.20)-(3.22) into Eq. (3.15), we have

$$R_{safe}^2 (u_1 - u_2)^2 = (d^2 - R_{safe}^2)(v_1 - v_2)^2 + (d^2 - R_{safe}^2)(w_1 - w_2)^2 \quad (3.23)$$

Let us define a constant variable β as

$$\beta \triangleq \sqrt{\frac{d^2 - R_{safe}^2}{R_{safe}^2}} \quad (3.24)$$

Then, Eq. (3.23) can be rewritten as

$$(u_1 - u_2)^2 = \beta^2(v_1 - v_2)^2 + \beta^2(w_1 - w_2)^2 \quad (3.25)$$

Equation (3.25) represents a conic surface, and β is a slope of the conic surface. The solutions of Eq. (3.25) satisfy the following relation.

$$\begin{bmatrix} u_1 - u_2 \\ v_1 - v_2 \\ w_1 - w_2 \end{bmatrix} = \begin{bmatrix} \beta h \\ h \cos \mu \\ h \sin \mu \end{bmatrix} \quad (3.26)$$

or

$$\mathbf{V}_{1,CA,LOS} = \mathbf{V}_{2,LOS} + \begin{bmatrix} \beta h \\ h \cos \mu \\ h \sin \mu \end{bmatrix} \quad (3.27)$$

Using Eq. (3.27) in the constraint Eq. (3.16), the following quadratic equation is obtained.

$$(1 + \beta^2)h^2 + 2(u_2\beta + v_2 \cos \mu + w_2 \sin \mu)h + V_2^2 - V_1^2 = 0 \quad (3.28)$$

Now, to have the real-value solution of h , the following condition should be satisfied.

$$(u_2\beta + v_2 \cos \mu + w_2 \sin \mu)^2 - (1 + \beta^2)(V_2^2 - V_1^2) \geq 0 \quad (3.29)$$

Then, two solutions of Eq. (3.28) can be obtained as follows.

$$h(\mu) = -\frac{(u_2\beta + v_2 \cos \mu + w_2 \sin \mu)}{1 + \beta^2} \pm \frac{\sqrt{(u_2\beta + v_2 \cos \mu + w_2 \sin \mu)^2 - (1 + \beta^2)(V_2^2 - V_1^2)}}{1 + \beta^2} \quad (3.30)$$

where $\mu \in [0, 2\pi)$.

By determining μ and substituting $h(\mu)$ into Eq. (3.27), the velocity vector $\mathbf{V}_{1,CA}(\mu)$ for the collision avoidance can be obtained as follows.

$$\begin{aligned} \mathbf{V}_{1,CA}(\mu) &\triangleq \mathbf{C}(-\chi_{LOS}, 3)\mathbf{C}(-\gamma_{LOS}, 2)\mathbf{V}_{1,CA,LOS}(\mu) \\ &\triangleq [V_{1,CA,x}(\mu) \ V_{1,CA,y}(\mu) \ V_{1,CA,z}(\mu)]^T \end{aligned} \quad (3.31)$$

Note that the solution of Eq. (3.31) is shown in Fig. 3.1. Then, the following commands for the collision avoidance can be obtained.

$$V_{cmd} = \|\mathbf{V}_{1,CA}(\mu)\| = V_1 \quad (3.32)$$

$$\chi_{cmd} = \tan^{-1} \left(\frac{V_{1,CA,y}(\mu)}{V_{1,CA,x}(\mu)} \right) \quad (3.33)$$

$$\gamma_{cmd} = \sin^{-1} \left(\frac{-V_{1,CA,z}(\mu)}{\|\mathbf{V}_{1,CA}(\mu)\|} \right) \quad (3.34)$$

The command set $(V_{cmd}, \chi_{cmd}, \gamma_{cmd})$ in Eqs. (3.32)-(3.33) can be delivered to the autopilot of the UAV in Chapter 2.1, which is a 3-dimensional solution of the collision avoidance problem.

Let us consider the direction of the current relative velocity $\mathbf{V}_1 - \mathbf{V}_2$ in Fig. 3.1. The current relative velocity can be expressed in the viewpoint of the UAV towards the position of the intruder as shown in Fig. 3.3. Using the coordinate transformation with respect to the LOS angles, the direction of the current relative velocity, μ_o , in Fig. 3.3 can be obtained as

$$\mu_o = \tan^{-1} \left(\frac{w_{1,current} - w_2}{v_{1,current} - v_2} \right) \quad (3.35)$$

where $[u_{1,current} \ v_{1,current} \ w_{1,current}]^T \triangleq \mathbf{C}(\gamma_{LOS}, 2)\mathbf{C}(\chi_{LOS}, 3)\mathbf{V}_1$.

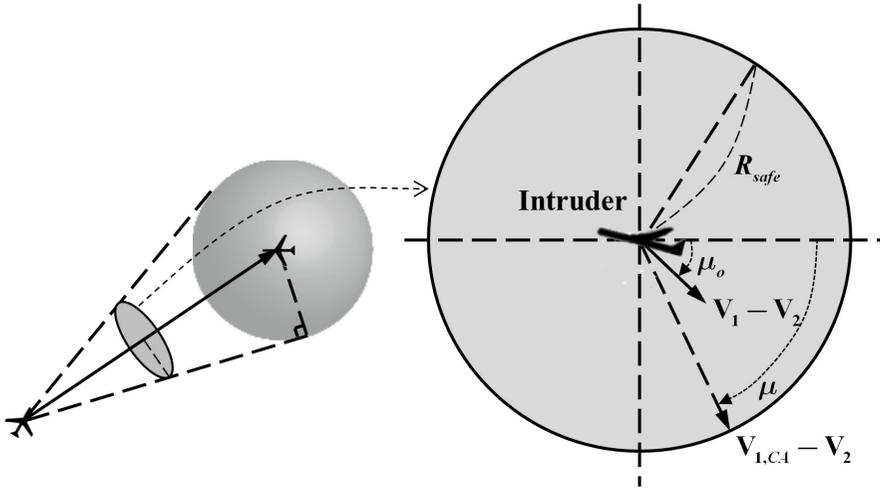


Figure 3.3 Viewpoint of the UAV towards the position of the intruder

In the case of $\mu = \mu_o$, deviation of the relative velocity vector is minimal. If the speed of the UAV is constant and the collision situation is set in a 2-dimensional plane, then this solution is identical to the solution of the conventional geometric collision avoidance approaches such as collision cone, velocity obstacle, or geometric optimization approach. Note that these geometric collision avoidance approaches are designed for 2-dimensional collision avoidance. In other words, the reactive collision avoidance is a 3-dimensionally expanded collision avoidance method, and the avoidance direction can be chosen by determining μ .

3.3 Command Switching Scheme using Command Filter

The tracking guidance law and the collision avoidance algorithm provide the command sets. Both command sets consist of V_{cmd} , χ_{cmd} , and γ_{cmd} , and the two command sets are selectively used according to the operation situations. If the collision threat is imminent, the command set for the collision avoidance is chosen. Otherwise, the UAV should fly towards the original waypoint or track the trajectory using the command set for the tracking guidance. To do this, the two command sets should be switched. In this study, a switching scheme is designed as shown in the block diagram of Fig. 3.4.

When the command set is switched from the tracking guidance law to the collision avoidance algorithm, or vice versa, then there may exist discontinuous command jumps which make the UAV perform abrupt maneuvers. To deal with this problem, the command filter [49] is used as shown in Fig. 3.4. The dynamics of the command filter is based on the second-order damping system with saturation functions as follows:

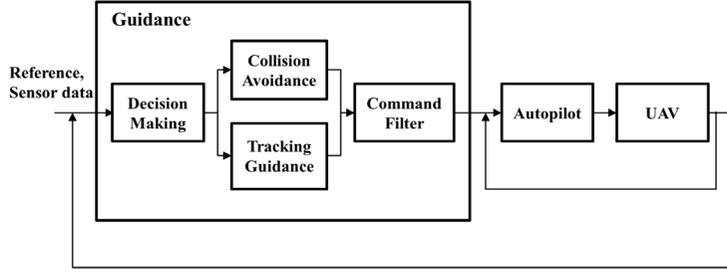


Figure 3.4 Block diagram of the UAV system

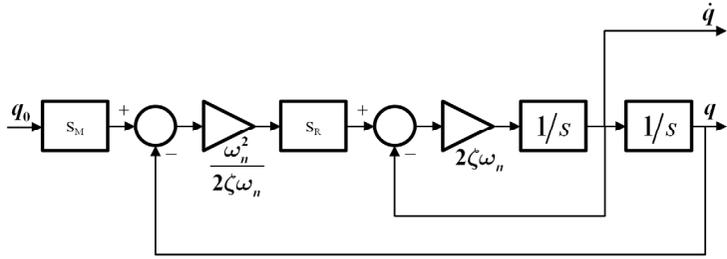


Figure 3.5 Command filter

$$\ddot{q} + 2\zeta\omega_n\dot{q} + 2\zeta\omega_n S_R \left(\frac{\omega_n^2}{2\zeta\omega_n} q - \frac{\omega_n^2}{2\zeta\omega_n} S_M(q_0) \right) = 0 \quad (3.36)$$

where q is a filtered command, q_0 is an original command, ζ is a damping ratio, ω_n is a natural frequency of the filter, and S_M and S_R are the saturation functions of the magnitude and rate as follows.

$$S_{(\cdot)}(x) = \begin{cases} x_{\max} & \text{if } x \geq x_{\max} \\ x & \text{if } x_{\min} < x < x_{\max} \\ x_{\min} & \text{otherwise} \end{cases} \quad (3.37)$$

The structure of the command filter is illustrated in Fig. 3.5.

For each command of the command set $(V_{cmd}, \chi_{cmd}, \gamma_{cmd})$, the corresponding command filter is used. The saturation functions confine each command and its derivatives to satisfy the state constraints in Eqs. (2.11)-(2.15). Note that the dynamics of the command filter is stable, and therefore the filtered commands can be substituted for $(V_{cmd}, \chi_{cmd}, \gamma_{cmd})$ and $(\dot{V}_{cmd}, \dot{\chi}_{cmd}, \dot{\gamma}_{cmd})$ without loss of generality.

In relation to the proposed command switching scheme, data flow of the proposed guidance and collision avoidance system can be illustrated as shown in Fig. 3.6.

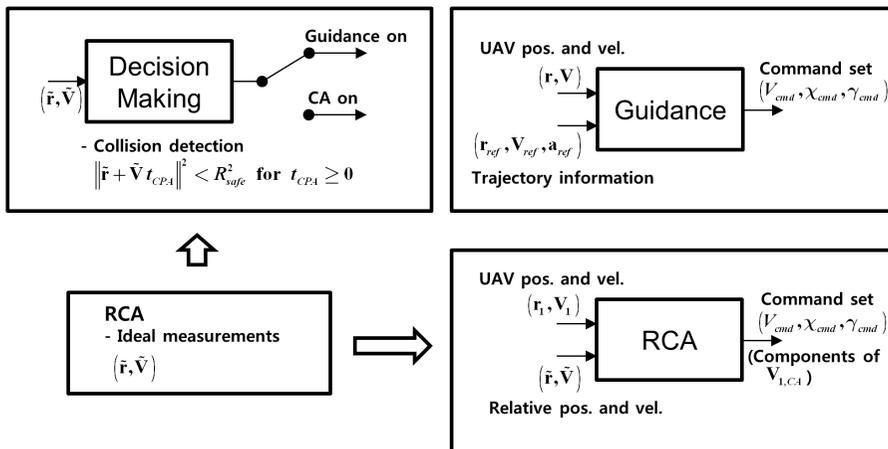


Figure 3.6 Data flow of Guidance/RCA system

Chapter 4

Vision-Based Reactive Collision

Avoidance

In this chapter, a collision avoidance algorithm using a single vision sensor is proposed. For the real-time operation, the algorithm adopts the geometry of the CPA [29-32]. First, the geometry of the CPA is incorporated with a measurement model of a single vision sensor, and several characteristics for the collision avoidance utilizing the CPA are presented. Then, a new reactive collision avoidance algorithm which can overcome the loss-of-depth problem of the vision sensor is proposed to avoid a moving intruder. The proposed algorithm is robust with respect to the motion of the intruder. The solution of the collision avoidance algorithm consists of the speed, heading angle, and flight path angle commands, $(V_{cmd}, \chi_{cmd}, \gamma_{cmd})$, which is reactively updated during the flight in accordance with the sensor measurements.

4.1 Single Vision Sensor and Its Limitations

To implement a single vision sensor for the collision avoidance algorithm, the following assumption is considered.

Assumption 4.1. The shape of the intruder is assumed as a circle in the image processing to obtain size information using the intruder's image.

Assumption 4.1 is the approximated assumption of the intruder's image considering small attitude changes of the intruder. Note that, the size of the intruder is related to the wingspan or the fuselage of UAV [40-44].

Figure 4.1 shows the measurement model of a single vision sensor in accordance with Assumption 4.1. In Fig. 4.1, l is the relative distance between the UAV and intruder and R_r is the actual size of the intruder, both of which are unknown because the sensor projects 3-dimensional objects onto a 2-dimensional image plane. Variable α is half the angle subtended at the sensor by the tangents to the opposite

ends of the intruder (represented as a circle) from the sensor [44]. χ_c and γ_c are relative azimuth and elevation angles, respectively. The variables α , χ_c , and γ_c are measurements of the vision sensor.

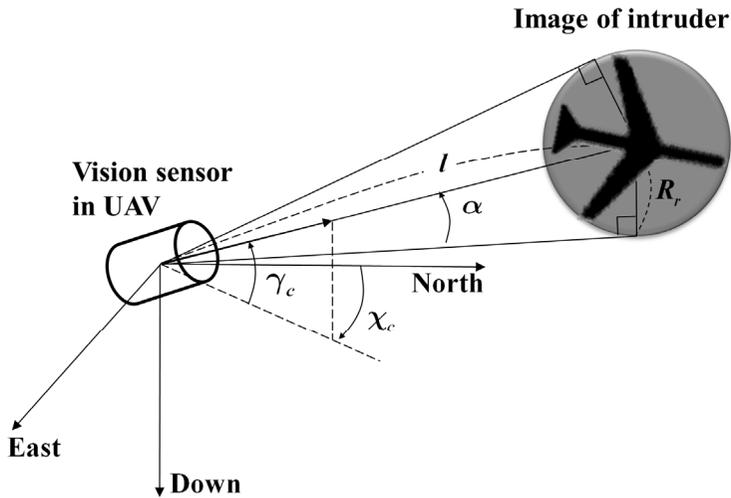


Figure 4.1 Measurement model of a vision sensor

According to the geometry and coordinates in Fig. 4.1, the actual relative position, $\tilde{\mathbf{r}}_r$, and actual relative velocity, $\tilde{\mathbf{V}}_r$, of the intruder with respect to the UAV are expressed as functions of (l, χ_c, γ_c) and their derivatives as:

$$\tilde{\mathbf{r}}_r = [l \cos \gamma_c \cos \chi_c \quad l \cos \gamma_c \sin \chi_c \quad -l \sin \gamma_c]^T \quad (4.1)$$

$$\tilde{\mathbf{V}}_r = \begin{bmatrix} \cos \gamma_c \cos \chi_c & -l \cos \gamma_c \sin \chi_c & -l \sin \gamma_c \cos \chi_c \\ \cos \gamma_c \sin \chi_c & l \cos \gamma_c \cos \chi_c & -l \sin \gamma_c \sin \chi_c \\ -\sin \gamma_c & 0 & -l \cos \gamma_c \end{bmatrix} \begin{bmatrix} \dot{l} \\ \dot{\chi}_c \\ \dot{\gamma}_c \end{bmatrix} \quad (4.2)$$

However, the UAV cannot obtain the states of the intruder using a single vision sensor only because l is unknown due to the loss-of-depth problem. Therefore, the conventional collision avoidance algorithms cannot be implemented directly. To deal with this problem, the angle α and an assumed size of the intruder, R_0 , are used in this study.

Using the assumed size of the intruder, l and \dot{l} can be assumed as

$$l = \frac{R_0}{\sin \alpha} \quad (4.3)$$

$$\dot{l} = -\frac{R_0}{\sin \alpha \tan \alpha} \dot{\alpha} \quad (4.4)$$

Substituting Eqs. (4.3) and (4.4) into Eqs. (4.1) and (4.2), the relative position and velocity can be assumed as

$$\tilde{\mathbf{r}} = \frac{R_0}{\sin \alpha} [\cos \gamma_c \cos \chi_c \quad \cos \gamma_c \sin \chi_c \quad -\sin \gamma_c]^T \quad (4.5)$$

$$\tilde{\mathbf{V}} = \frac{R_0}{\sin \alpha} \begin{bmatrix} \frac{\cos \gamma_c \cos \chi_c}{\tan \alpha} & -\cos \gamma_c \sin \chi_c & -\sin \gamma_c \cos \chi_c \\ \frac{\cos \gamma_c \sin \chi_c}{\tan \alpha} & \cos \gamma_c \cos \chi_c & -\sin \gamma_c \sin \chi_c \\ \frac{\sin \gamma_c}{\tan \alpha} & 0 & -\cos \gamma_c \end{bmatrix} \begin{bmatrix} \dot{\alpha} \\ \dot{\chi}_c \\ \dot{\gamma}_c \end{bmatrix} \quad (4.6)$$

Now, collision detection and avoidance can be performed using R_0 . Note that adopting R_0 does not guarantee the safety of the algorithm because R_0 is not the actual value, and the corresponding avoidance maneuver could be inadequate because of the loss-of-depth problem. In addition, vision sensors cannot estimate the intruder's intent because they are noncooperative sensors. For these reasons, the algorithm could be sensitive to the intruder's motion. In this study, let us define two types of intruders. One is an interactive intruder that has the same collision avoidance capability as the UAV, and the other is a noninteractive intruder that does not perform avoidance maneuvers. The collision avoidance algorithm using the vision sensor will be designed to avoid the interactive intruder as well as the noninteractive intruder with considerations for safety and robustness.

4.2 Closest-Point-of-Approach (CPA)

From Eqs. (4.5) and (4.6), several characteristics related to the CPA and R_0 can be obtained as follows.

Lemma 4.1. The time of the CPA between the UAV and the intruder, t_{CPA} , is independent of R_0 .

Proof. Consider the actual size of the intruder, R_r , and the corresponding relative position, $\tilde{\mathbf{r}}_r$, and velocity, $\tilde{\mathbf{V}}_r$. Because $\tilde{\mathbf{r}} = (R_0/R_r)\tilde{\mathbf{r}}_r$ and $\tilde{\mathbf{V}} = (R_0/R_r)\tilde{\mathbf{V}}_r$, the following relation exists.

$$t_{CPA} = -\frac{\tilde{\mathbf{r}}^T \tilde{\mathbf{V}}}{\tilde{\mathbf{V}}^T \tilde{\mathbf{V}}} = -\frac{\tilde{\mathbf{r}}_r^T \tilde{\mathbf{V}}_r}{\tilde{\mathbf{V}}_r^T \tilde{\mathbf{V}}_r} \quad (4.7)$$

Equation (4.7) indicates that t_{CPA} is independent of R_0 . (Q.E.D.)

Lemma 4.1 means t_{CPA} can be calculated from the measurements of the vision sensor and it does not require the range information. Then, the following three corollaries can be stated without proof.

Corollary 4.1. The UAV's CPA position, $\mathbf{r}_{1,CPA}$, can be determined

regardless of R_0 .

Corollary 4.2. The intruder's CPA position, $\mathbf{r}_{2,CPA}$, is located at $\tilde{\mathbf{r}} + \tilde{\mathbf{V}}t_{CPA}$ from $\mathbf{r}_{1,CPA}$ where $\|\tilde{\mathbf{r}} + \tilde{\mathbf{V}}t_{CPA}\|$ is dependent on R_0 but the direction of $\tilde{\mathbf{r}} + \tilde{\mathbf{V}}t_{CPA}$ is independent of R_0 .

Corollary 4.3. If the time of the CPA is positive, the trajectories of the UAV and the intruder based on the current states converge with a possibility of a collision.

Lemma 4.2. If R_0 is smaller than R_r , the closest distance computed by the UAV is shorter than its actual value. Likewise, if R_0 is larger than R_r , the closest distance is longer than its actual value.

Proof. According to Eqs. (4.5) and (4.6), the UAV computes the relative position and velocity of the intruder using R_0 . Then, the closest distance has the following relation because $\tilde{\mathbf{r}} = (R_0/R_r)\tilde{\mathbf{r}}_r$ and $\tilde{\mathbf{V}} = (R_0/R_r)\tilde{\mathbf{V}}_r$.

$$\|\tilde{\mathbf{r}} + \tilde{\mathbf{V}}t_{CPA}\| = \frac{R_0}{R_r} \|\tilde{\mathbf{r}}_r + \tilde{\mathbf{V}}_r t_{CPA}\| \quad (4.8)$$

Therefore, the closest distance using R_0 is proportional to the ratio of R_0 and R_r as illustrated in Fig. 4.2. (Q.E.D.)

Corollary 4.1 means $\mathbf{r}_{1,CPA}$ is an exactly calculated point without the range or size information. According to Corollary 4.2, $\mathbf{r}_{2,CPA}$ can be assumed by using R_0 . Lemma 4.2 explains the concept of relative recognition of the UAV according to the assumption on R_0 . The following corollary can be stated in relation to the relative recognition concept and the expected collision avoidance result of the UAV.

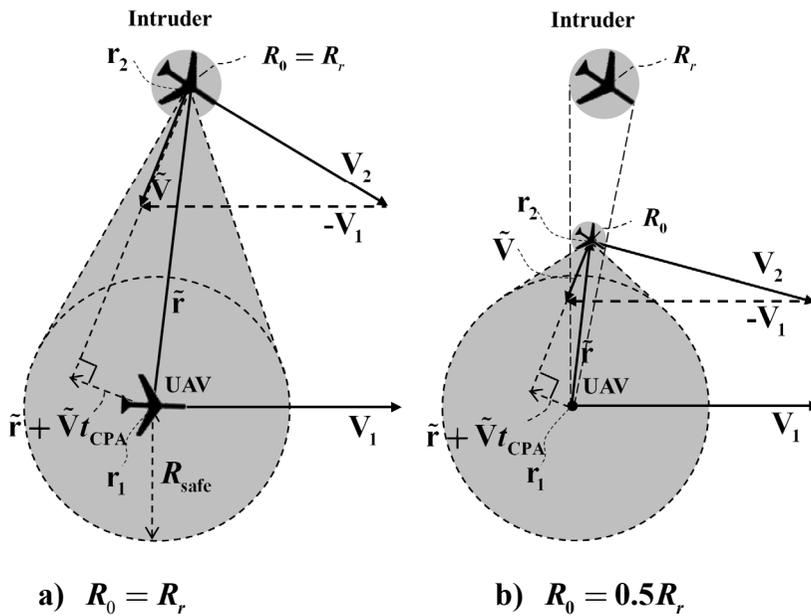


Figure 4.2 Relative recognition according to the intruder's size assumption

Corollary 4.4. If R_0 is assumed to be smaller than R_r , the expected result of the closest distance after collision avoidance is longer than R_{safe} .

The UAV is supposed to perform collision avoidance maneuvers to maintain R_{safe} according to the reactive collision avoidance algorithm. However, if the predicted closest distance is shorter than its actual value, the UAV recognizes that the situation is more dangerous. Then, the UAV will make more efforts to avoid the collision situation. Consequently, if the predicted closest distance, $\|\tilde{\mathbf{r}} + \tilde{\mathbf{V}}t_{CPA}\|$, approaches to R_{safe} , the actual distance, $\|\tilde{\mathbf{r}}_r + \tilde{\mathbf{V}}_r t_{CPA}\|$, will be much longer than R_{safe} . Therefore, the result of the closest distance will be much longer than the safety radius after the collision avoidance maneuver. In this case, the corresponding avoidance maneuver will be also safe, and its trajectory will be extended. This effect is conservative, and therefore it will be used in this study for the design of the reactive collision avoidance algorithm by assuming a small R_0 .

Remark 4.1. If R_0 is assumed to be too small, the UAV's maneuver could be drastic because the UAV overestimates the collision risk. To avoid drastic maneuvers, bounded size of the intruder is considered. The minimum and maximum values are defined as R_{\min} and R_{\max} , which are obtained from the information of the operating altitude of the aircraft. Table 4.1 summarizes the UAV wingspan data with respect to the different airspace classes of the United States National Airspace System [55], within which each UAV is designed to operate.

Table 4.1 UAV wingspan data with respect to the airspace classes of the National Airspace System [55]

Airspace Class (MSL, m)	UAV	Wingspan, m	Speed, m/sec
Class A (5,500~18,000)	Predator	14.8~20.1	36
	Fire Scout	8.4	0~64.3
Class E (365~5,500)	Shadow	3.9~4.3	30.9
	Hunter	8.9~10.5	45.8
Class G (~365)	Pointer	2.7	22.1
	Dragon Eye	1.2	18

(MSL: Mean Sea Level)

4.3 Collision Avoidance Algorithm

The proposed vision-based collision avoidance algorithm focuses on the characteristics discussed in the previous chapter. To deal with the loss-of-depth problem of the vision sensor, the following assumption related to Lemma 4.2, Corollary 4.4 and Remark 4.1 is adopted.

Assumption 4.2. R_0 is set as R_{\min} in a certain airspace class where the UAV is operated.

In this section, a collision avoidance algorithm is proposed by adopting Assumption 4.2. The proposed algorithm consists of a direction condition and a time condition. The direction condition provides a collision avoidance solution $(V_{cmd}, \chi_{cmd}, \gamma_{cmd})$ which is safe and robust to the motion of the intruder. The time condition considers the activation time of the algorithm to restrict the trajectory extension of the UAV. The detailed approaches are described as follows.

According to the geometry of the CPA, the collision region can be approximated as a spherical region at the CPA position. Then, avoidance direction can be determined considering the region. Let us define the path frame of the UAV (subscript P) which denotes the direction of the velocity vector of the UAV as shown in Fig. 4.4(a). Judging from the UAV's viewpoint, the direction can be chosen to avoid the shaded-sphere region in Fig. 4.4(a). Now, let us consider the intruder's relative velocity in the UAV's path frame as follows

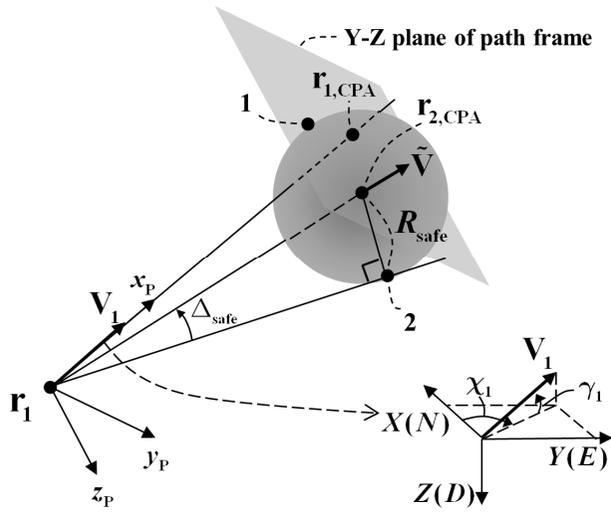
$$\tilde{\mathbf{V}}_P \triangleq \mathbf{C}(\gamma_1, 2)\mathbf{C}(\chi_1, 3)\tilde{\mathbf{V}} = [\tilde{u}_P \quad \tilde{v}_P \quad \tilde{w}_P]^T \quad (4.9)$$

where $\mathbf{C}(\gamma_1, 2)$ and $\mathbf{C}(\chi_1, 3)$ are Euler rotation matrices as defined in Eqs. (3.17) and (3.18). Then, \tilde{v}_P and \tilde{w}_P are independent of the speed of the UAV. From Fig. 4.4(b), a projected direction angle of the relative velocity, η_V , can be defined as follows:

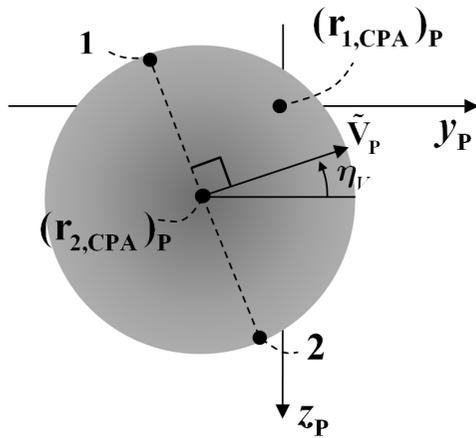
$$\eta_V \triangleq \tan^{-1} \left(\frac{-\tilde{w}_P}{\tilde{v}_P} \right) \quad (4.10)$$

where η_V indicates the intruder's path in the viewpoint of the UAV, which can be used to determine χ_{cmd} and γ_{cmd} .

Let us define an angle, Δ_{safe} , as shown in Fig. 4.4(a), which can be determined according to the safety radius R_{safe} as follows.



a) Path frame and geometry of CPA



b) Y-Z plane of the path frame

Figure 4.4 Path frame and geometry of the CPA for collision avoidance direction

$$\Delta_{safe} = \begin{cases} \frac{\pi}{2} & \text{if } \|\mathbf{r}_{2,CPA} - \mathbf{r}_1\| < R_{safe} \\ \sin^{-1}\left(\frac{R_{safe}}{\|\mathbf{r}_{2,CPA} - \mathbf{r}_1\|}\right) & \text{otherwise} \end{cases} \quad (4.11)$$

If the obstacle is not moving, any point on the circumference of a circle with radius R_{safe} can be chosen for the collision avoidance. Or, the avoidance point can be taken by considering the maneuver of the UAV which minimizes deviation from its nominal flight path.

On the other hand, collision avoidance of a moving intruder is more complex. Because the intruder is moving, and it could be interactive, most points on the circumference of a circle with radius R_{safe} are affected by the intruder's motion. For example, the directions related to η_V and $\eta_V + \pi$ in Fig. 4.4(b) do not guarantee safety avoidance because the UAV and the intruder cross each other's path. If the two aircraft have same collision avoidance algorithms, then those directions may induce other collision situations or may not resolve the situations effectively. Therefore, the following two directions that are less sensitive to the intruder's motion are chosen in this study.

$$\begin{bmatrix} \chi_{cmd1} \\ \gamma_{cmd1} \end{bmatrix} = \begin{bmatrix} \chi_{2,CPA} + \Delta_{safe} \cos(\eta_V + \frac{\pi}{2}) \\ \gamma_{2,CPA} + \Delta_{safe} \sin(\eta_V + \frac{\pi}{2}) \end{bmatrix} \quad (4.12)$$

$$\begin{bmatrix} \chi_{cmd2} \\ \gamma_{cmd2} \end{bmatrix} = \begin{bmatrix} \chi_{2,CPA} + \Delta_{safe} \cos(\eta_V - \frac{\pi}{2}) \\ \gamma_{2,CPA} + \Delta_{safe} \sin(\eta_V - \frac{\pi}{2}) \end{bmatrix} \quad (4.13)$$

where $\chi_{2,CPA} \triangleq \tan^{-1}\left(\frac{(y_{2,CPA} - y_1)}{(x_{2,CPA} - x_1)}\right)$, and $\gamma_{2,CPA} \triangleq \sin^{-1}\left(-\frac{(z_{2,CPA} - z_1)}{\|\mathbf{r}_{2,CPA} - \mathbf{r}_1\|}\right)$. x , y , and z denote the axis components of each aircraft's position vector, respectively.

These two directions are points 1 and 2 in Figs. 4.4(a) and 4.4(b). Because these directions avoid the intruder's path, they are robust to the intruder's motion. However, the direction that is furthest from $\mathbf{r}_{1,CPA}$ (point 2 in Fig. 4.4(b)) is not safe because of the uncertainty in $\mathbf{r}_{2,CPA}$ arising from the assumption on R_0 as discussed in Corollary 4.2. Therefore, the direction relatively close to $\mathbf{r}_{1,CPA}$ (point 1 in Fig. 4.4(b)) is chosen as

$$\begin{bmatrix} \chi_{cmd} \\ \gamma_{cmd} \end{bmatrix} = \begin{cases} \begin{bmatrix} \chi_{cmd1} \\ \gamma_{cmd1} \end{bmatrix} & \text{if } \left\| \begin{bmatrix} \chi_{cmd1} - \chi_1 \\ \gamma_{cmd1} - \gamma_1 \end{bmatrix} \right\| \leq \left\| \begin{bmatrix} \chi_{cmd2} - \chi_1 \\ \gamma_{cmd2} - \gamma_1 \end{bmatrix} \right\| \\ \begin{bmatrix} \chi_{cmd2} \\ \gamma_{cmd2} \end{bmatrix} & \text{otherwise} \end{cases} \quad (4.14)$$

Then, the solution is obtained from Eq. (4.14) and the constant speed V_{cmd} . Two possible singular cases exist in the solution by the proposed collision avoidance algorithm.

1) CPA singularity

The time of the CPA, t_{CPA} , cannot be computed when $\tilde{\mathbf{V}} = \mathbf{0}$ according to Eq. (4.7). Then, the UAV cannot detect the collision even though the collision threats exist. In this case, a predesignated collision avoidance maneuver, such as the coordinated turn, is required to avoid the singularity.

2) Directional singularity

The collision avoidance direction cannot be determined when $\tilde{v}_p = 0$ and $\tilde{w}_p = 0$, which may happen when the velocity vectors of the two aircraft are parallel to each other. This can be solved by using another criterion on the relative position at the CPA. A relative position vector at the time of the CPA can be considered as:

$$\tilde{\mathbf{r}}_p \triangleq \mathbf{C}(\gamma_1, 2)\mathbf{C}(\chi_1, 3)(\mathbf{r}_{2,CPA} - \mathbf{r}_{1,CPA}) = [\tilde{x}_p \ \tilde{y}_p \ \tilde{z}_p]^T \quad (4.15)$$

The projected directional angle of the relative position, η_p , is defined

as:

$$\eta_P \triangleq \tan^{-1} \left(\frac{-\tilde{z}_P}{\tilde{y}_P} \right) \quad (4.16)$$

Note that directions related to $\eta_P + \pi/2$ or $\eta_P - \pi/2$ may not solve the collision situation. In the case of the head on collision, the two UAVs may perform the same climbing maneuvers or the same descending maneuvers using these directions, $\eta_P \pm \pi/2$, which is the case of the contradiction on the algorithm. To avoid this contradiction, the path angles for the collision avoidance are determined as

$$\begin{bmatrix} \chi_{cmd} \\ \gamma_{cmd} \end{bmatrix} = \begin{bmatrix} \chi_{2,CPA} + \Delta_{safe} \cos(\eta_P + \pi) \\ \gamma_{2,CPA} + \Delta_{safe} \sin(\eta_P + \pi) \end{bmatrix} \quad (4.17)$$

Again, the direction cannot be determined in the case where $\tilde{y}_P = 0$ and $\tilde{z}_P = 0$ which occurs when \mathbf{V}_1 and \mathbf{V}_2 lie on the same line. In this case, a predesignated collision avoidance maneuver is required to resolve the collision situation, which is similar to the case of the CPA singularity.

In summary, the collision avoidance direction is determined by Eq. (4.14) if η_P exists, or given Eq. (4.17) otherwise. The specific singularities are dealt with by the predesignated collision avoidance

maneuver. If the intruder is noninteractive, the collision avoidance direction is safe and robust to the motion of the intruder because the UAV avoids the intruder's path. For an interactive intruder, the following theorem states that there are no contradictions between the algorithms of the two aircraft at the exact collision situations.

Theorem 4.1. If the two aircraft perform the same avoidance maneuvers, according to Eqs. (4.9)-(4.14), within a bounded vertical flight path angle range $\left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$, then they avoid each other without crossing the other's path.

Proof. Consider the exact collision situations and Eq. (4.9) for each aircraft:

$$\begin{aligned} & \mathbf{C}(\gamma_1, 2)\mathbf{C}(\chi_1, 3)(\mathbf{V}_2 - \mathbf{V}_1) \\ &= \begin{bmatrix} V_2 \cos \gamma_2 \cos \gamma_1 \cos(\chi_2 - \chi_1) + V_2 \sin \gamma_2 \sin \gamma_1 - V_1 \\ V_2 \cos \gamma_2 \sin(\chi_2 - \chi_1) \\ V_2 \cos \gamma_2 \sin \gamma_1 \cos(\chi_2 - \chi_1) - V_2 \sin \gamma_2 \cos \gamma_1 \end{bmatrix} \end{aligned} \quad (4.18)$$

$$\begin{aligned} & \mathbf{C}(\gamma_2, 2)\mathbf{C}(\chi_2, 3)(\mathbf{V}_1 - \mathbf{V}_2) \\ &= \begin{bmatrix} V_1 \cos \gamma_1 \cos \gamma_2 \cos(\chi_1 - \chi_2) + V_1 \sin \gamma_1 \sin \gamma_2 - V_2 \\ V_1 \cos \gamma_1 \sin(\chi_1 - \chi_2) \\ V_1 \cos \gamma_1 \sin \gamma_2 \cos(\chi_1 - \chi_2) - V_1 \sin \gamma_1 \cos \gamma_2 \end{bmatrix} \end{aligned} \quad (4.19)$$

Note that the y- and z-axis components in Eqs. (4.18) and (4.19)

constitute η_v in Eq. (4.10), and the y-axis components of the two relative velocity vectors always have opposite signs when the components are non-zeros ($\chi_1 \neq \chi_2$) because V_1 , V_2 , $\cos \gamma_1$, and $\cos \gamma_2$ are always positive when γ_1 and γ_2 are within the range $\left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$. The z-axis components of the two vectors will have the same signs when $|\chi_1 - \chi_2| > \frac{\pi}{2}$, and opposite signs when $|\chi_1 - \chi_2| < \frac{\pi}{2}$. If $|\chi_1 - \chi_2| > \frac{\pi}{2}$, the two aircraft will face each other. Then, the avoidance directions of the two aircraft are opposite each other according to Eqs. (4.12) or (4.13) because the signs of the y-axis components are opposite and those of the z-axis components are the same. Conversely, if $|\chi_1 - \chi_2| < \frac{\pi}{2}$, the two aircraft will head in the same direction. In this case, the collision avoidance directions are also opposite. Eventually, the two aircraft do not meet at any point according to the collision avoidance direction. (Q.E.D.)

Figure 4.5 illustrates the proof of Theorem 4.1, which denotes the avoidance directions of the two aircraft.

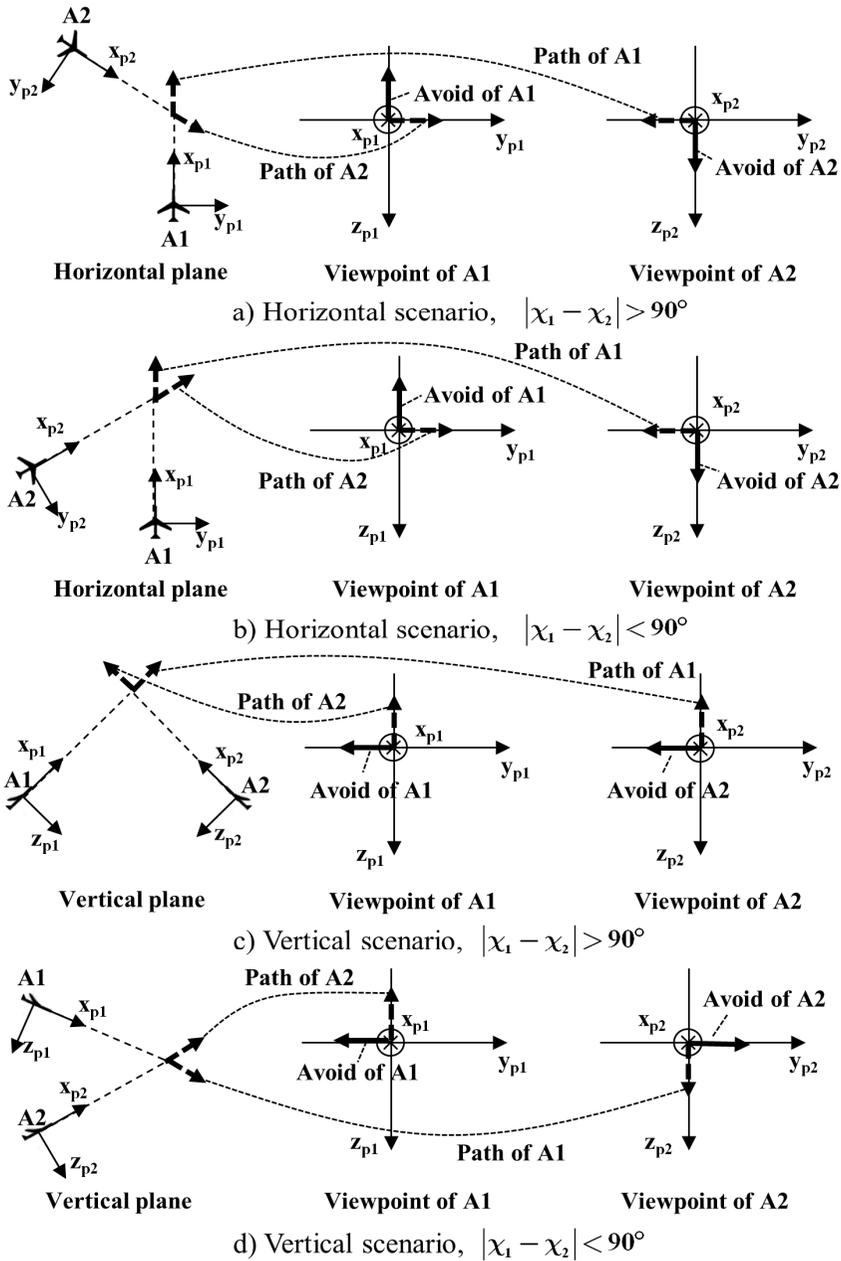


Figure 4.5 Avoidance directions according to approaching configurations

4.3.2 Collision Avoidance Time

If R_0 is set in accordance with Assumption 4.2, the UAV maneuvers more than required due to the characteristics mentioned in Lemma 4.2 and Corollary 4.4. This conservative effect is undesirable because it makes the UAV deviate too far from its path. To solve this weakness, a time condition is considered. Because t_{CPA} is independent of R_0 as stated in Lemma 4.1, time bounds $[t_L, t_U]$ and t_{CPA} can be considered to activate the algorithm. That is, the collision avoidance algorithm can be adjusted to be activated only when t_{CPA} is within the bounds. According to Corollary 4.3, the lower bound t_L can be set to be a negative value, whereas the upper bound t_U can be set using the independent characteristics of $\mathbf{r}_{1,CPA}$. Since $\mathbf{r}_{1,CPA}$ is also independent of R_0 as stated in Corollary 4.1, the UAV can maneuver to avoid the position, $\mathbf{r}_{1,CPA}$ as shown in Fig. 4.6, when the avoidance direction is given. Let us define a required time t_c for the UAV to avoid the location $\mathbf{r}_{1,CPA}$ by R_{safe} . Then, t_U can be set to be larger than t_c . The computation of the t_c value and the effect of the time condition are stated in the following lemma and theorem.

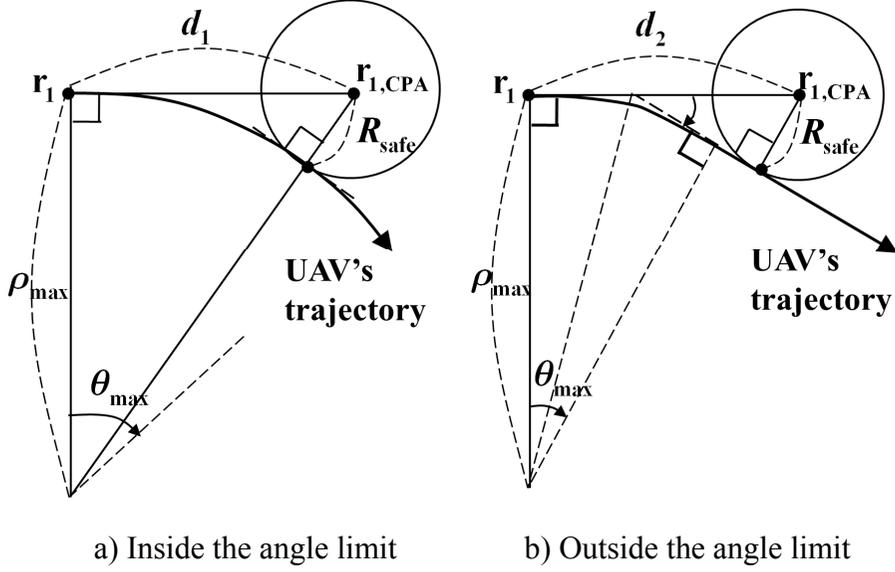


Figure 4.6 Geometry of the single point avoidance

Lemma 4.3. Consider the UAV with speed V , the path angles of (χ, γ) , and UAV's maximum turn rates $(\dot{\chi}_{\max}, \dot{\gamma}_{\max})$. The required time t_c to avoid the location of $\mathbf{r}_{1,CPA}$ by R_{safe} is obtained as follows:

$$t_c = \begin{cases} d_1/V & \text{if } d_1 \leq \rho_{\max} \tan \theta_{\max} \\ d_2/V & \text{otherwise} \end{cases} \quad (4.20)$$

where $d_1 = \sqrt{2\rho_{\max} R_{safe} + R_{safe}^2}$, $d_2 = \rho_{\max} \tan \frac{\theta_{\max}}{2} + \frac{R_{safe}}{\sin \theta_{\max}}$,

$\rho_{\max} = \max\left(\frac{V}{\dot{\chi}_{\max} \cos \gamma}, \frac{V}{\dot{\gamma}_{\max}}\right)$, and θ_{\max} is the maximum flight path angle due to vehicle's angular rate constraints.

Proof. The curvature radius ρ of the UAV's trajectory can be described as follows:

$$\rho = \frac{\|\dot{\mathbf{r}}\|^3}{\|\dot{\mathbf{r}} \times \ddot{\mathbf{r}}\|} = \frac{V}{\sqrt{\dot{\gamma}^2 + \dot{\chi}^2 \cos^2 \gamma}} \quad (4.21)$$

If the UAV maneuvers with the maximum turn rates, then the curvature radii of the lateral, longitudinal, and lateral/longitudinal maneuvers are $\rho_{lat} = \frac{V}{\dot{\chi}_{\max} \cos \gamma}$, $\rho_{long} = \frac{V}{\dot{\gamma}_{\max}}$, and

$$\rho_{latlong} = \frac{V}{\sqrt{\dot{\gamma}_{\max}^2 + \dot{\chi}_{\max}^2 \cos^2 \gamma}}, \text{ respectively.}$$

To encompass each maneuver, the maximum curvature radius ρ_{\max} among the maneuvers is considered, which is used to determine t_c . Considering $\rho_{latlong} < \rho_{lat}$ and $\rho_{latlong} < \rho_{long}$, ρ_{\max} can be chosen from either ρ_{lat} or ρ_{long} . As shown in Fig. 4.6, the required distances d_1 and d_2 to avoid the location of $\mathbf{r}_{1,CPA}$ are obtained as:

$$d_1 = \sqrt{(\rho_{\max} + R_{safe})^2 - \rho_{\max}^2} = \sqrt{2\rho_{\max}R_{safe} + R_{safe}^2} \quad (4.22)$$

$$d_2 = \rho_{\max} \tan \frac{\theta_{\max}}{2} + \frac{R_{safe}}{\sin \theta_{\max}} \quad (4.23)$$

From Eqs. (4.20)-(4.23), t_c can be obtained. (Q.E.D.)

Theorem 4.2. If the collision avoidance algorithm is activated only when t_{CPA} is within the bounds $[t_L, t_U]$ and the UAV performs an avoidance maneuver below its maximum turn rates, then the maneuver is independent of the intruder's size.

Proof. Let us consider the UAV's trajectories as shown in Fig. 4.6. Because t_{CPA} and $\mathbf{r}_{1,CPA}$ are independent of the intruder's size, t_L and t_U are also independent of the intruder's size. Therefore, the maneuvers avoiding the location of $\mathbf{r}_{1,CPA}$ are only dependent on the UAV's performance, regardless of the intruder's size. (Q.E.D.)

Remark 4.2. For a moving intruder, the sensing range of the vision sensor r_{sensing} should be longer than $(V_1 + V_2)t_{CPA}$ when $t_{CPA} \geq 0$, and longer than d_1 and d_2 for safety. That is, the sensing

range of the vision sensor should be enough to observe distant intruders.

In summary, data flow of the guidance and vision-based collision avoidance system is shown in Fig. 4.7.

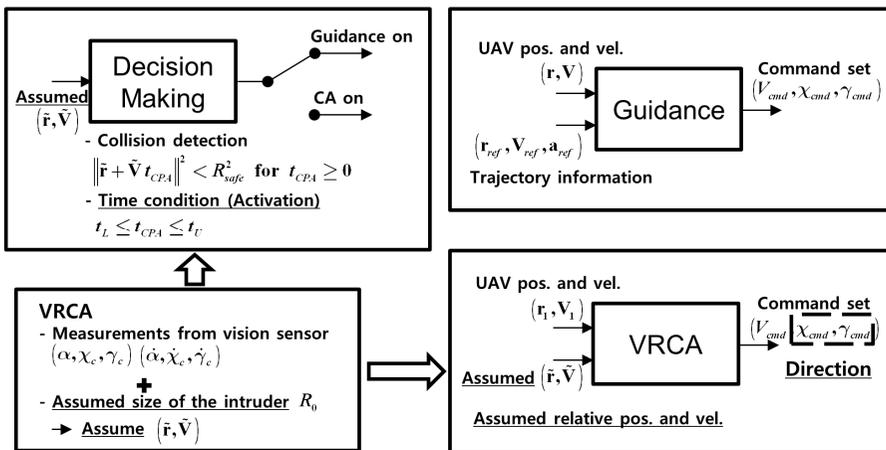


Figure 4.7 Data flow of Guidance/VRCA system

Chapter 5

Simulations on Collision Avoidance

To demonstrate the performance of the proposed collision avoidance algorithm, numerical simulations are performed. The simulations consider the reactive collision avoidance algorithm for the case of ideal measurements and the vision-based reactive collision avoidance algorithm using a single vision sensor.

5.1 Simulations on Reactive Collision Avoidance

For the numerical simulations of the reactive collision avoidance (RCA) algorithm, ideal one-to-one collision situations are considered. There exist a UAV and an intruder in the collision course, and it is assumed that the position and velocity information of the intruder are provided to the UAV. The aircraft are flying at an altitude of 3000m, which is Class E airspace as shown in Table 4.1. The parameters used in the simulations are summarized in Table 5.1.

Table 5.1 Parameters used in the RCA algorithm

Reactive collision avoidance algorithm	
Speed of aircraft (UAV and intruder)	$V = 30 \text{ m/s}$
Sensing range	1500 m
Safety radius	$R_{safe} = 150 \text{ m}$
Command filter	$\zeta = 1, \omega_n = 2 \text{ rad/s}$

In the simulations, the heading of the UAV is set to northeast (NE) and the intruder is approaching from four different directions, (i) east (E), (ii) southeast (SE), (iii) south (S), and (iv) southwest (SW). All cases yield a collision when the UAV does not perform any evasive maneuvers, and the collision situations are defined in the 2-dimensional plane.

The conventional geometric collision avoidance algorithms consider 2-dimensional collision avoidance, and therefore these algorithms provide 2-dimensional collision avoidance solutions in the collision situations. On the other hand, the RCA algorithm considers 3-dimensional collision avoidance. To evaluate performance of the RCA algorithm, comparison with a conventional collision avoidance algorithm, the geometric optimization (GO) approach [24], is

performed. Note that the GO approach is similar to the collision cone [20] and velocity obstacle [21] algorithms when the speed is constant. Also, the solution of the GO approach is identical to the RCA solution using Eq. (3.35), $\mu = \mu_0$, in these simulations.

Figures 5.1-5.4 show the trajectory results of the GO approach. In the cases of the GO approach, the UAV performs 2-dimensional avoidance maneuvers. Figures 5.5-5.8 show the trajectory results of the RCA algorithm. In the case of the RCA algorithm, the UAV can perform 3-dimensional avoidance maneuvers by determining μ in the range of $[0, 2\pi)$. In these simulations, μ is sampled at intervals of $\pi/8$, and a solution which has the minimum difference with the direction of the UAV is chosen. As the results, the UAV performs a vertical maneuver to avoid the intruder for some cases as shown in Figs. 5.5-5.7.

Miss distances between the UAV and the intruder are summarized in Table 5.2. The miss distance is the smallest relative distance between the two aircraft after the avoidance maneuvers. Since the simulations consider ideal measurements, the miss distances of the most cases are close to the value of the safety radius R_{safe} .

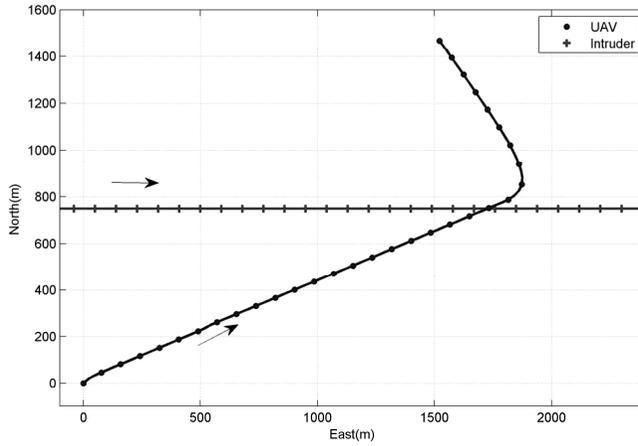


Figure 5.1 Trajectory of the GO approach (Intruder's heading: E)

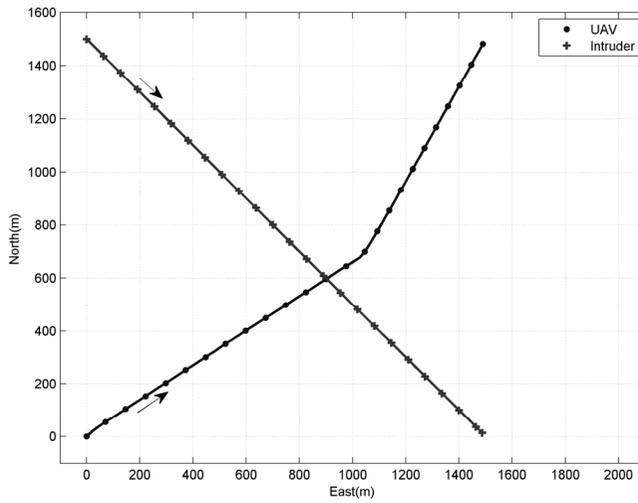


Figure 5.2 Trajectory of the GO approach (Intruder's heading: SE)

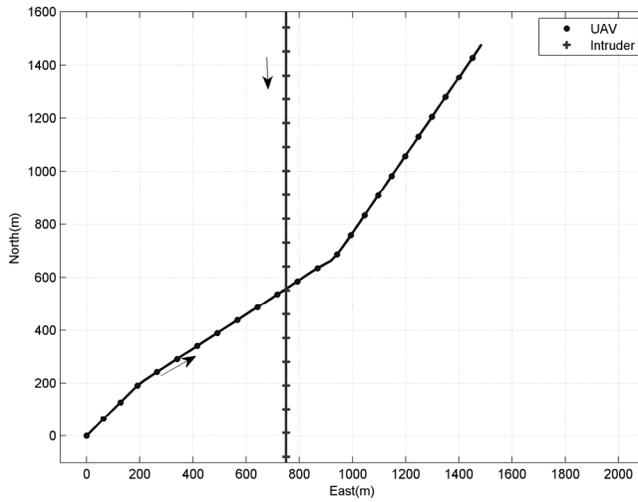


Figure 5.3 Trajectory of the GO approach (Intruder's heading: S)

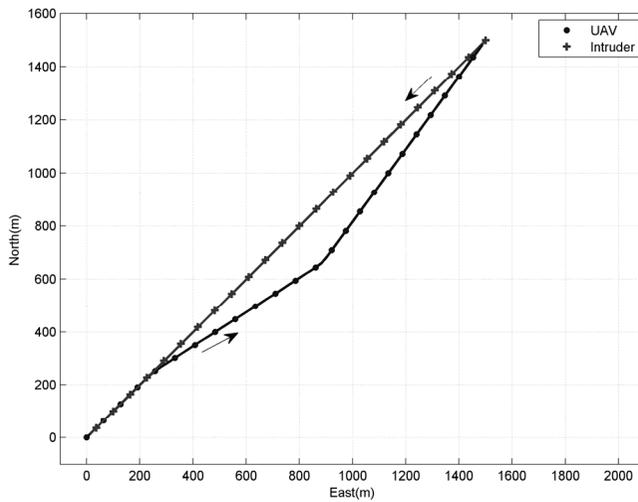


Figure 5.4 Trajectory of the GO approach (Intruder's heading: SW)

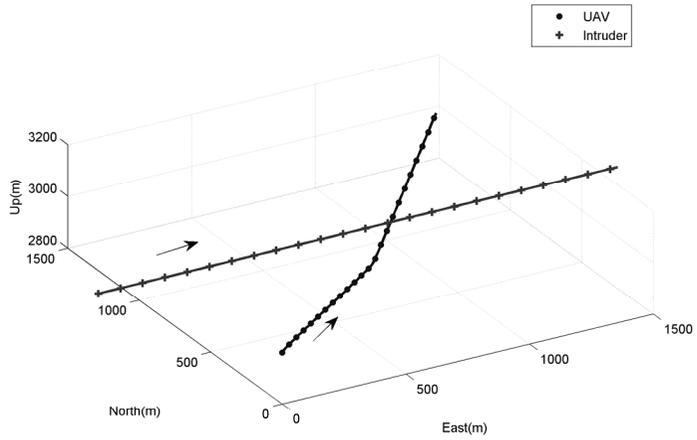


Figure 5.5 Trajectory of the 3-D RCA (Intruder’s heading: E)

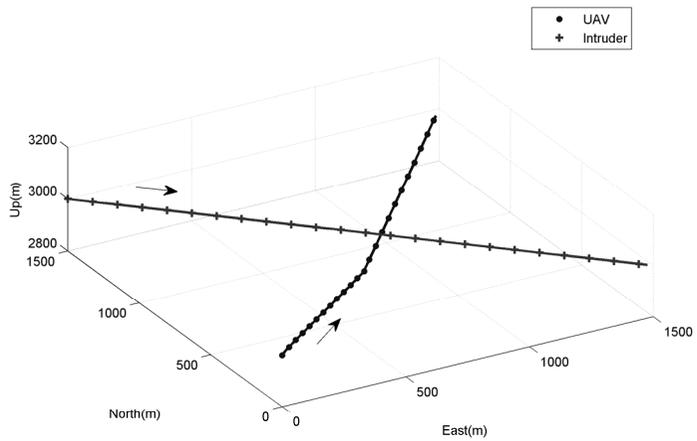


Figure 5.6 Trajectory of the 3-D RCA (Intruder’s heading: SE)

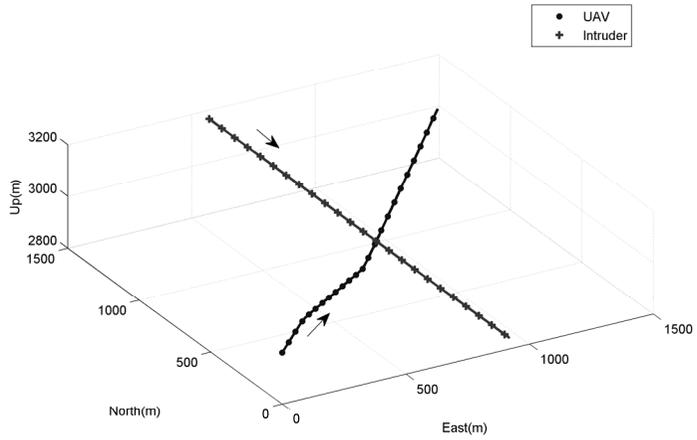


Figure 5.7 Trajectory of the 3-D RCA (Intruder's heading: S)

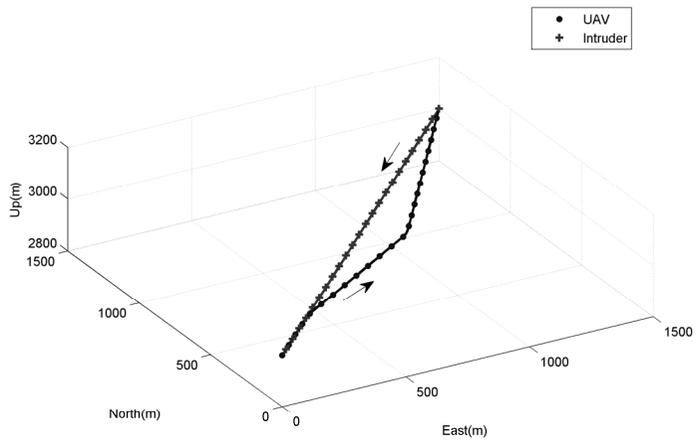


Figure 5.8 Trajectory of the 3-D RCA (Intruder's heading: SW)

Table 5.2 Miss distance results

Intruder's heading	E	SE	S	SW
GO approach	150.00 m	150.00 m	149.78 m	150.03 m
3-D RCA	150.00 m	150.01 m	150.02 m	150.03 m

Using the command switching scheme proposed in Chapter 3.3, the overall guidance and collision avoidance system of the UAV can be considered. Numerical results are shown in Figs. 5.9-5.11. When the UAV encounters an obstacle during the line tracing maneuver, it can avoid the obstacle and return to its original path as shown in Fig. 5.9. As shown in Fig. 5.10, the position error between the UAV and the path is increased when the UAV performs the collision avoidance. After the collision avoidance maneuver, the error is decreased due to the UAV guidance. Figure 5.11 shows time histories of the heading angle command and filtered heading angle command. It can be seen that the command filter mitigates the abrupt command changes from the collision avoidance to the guidance.

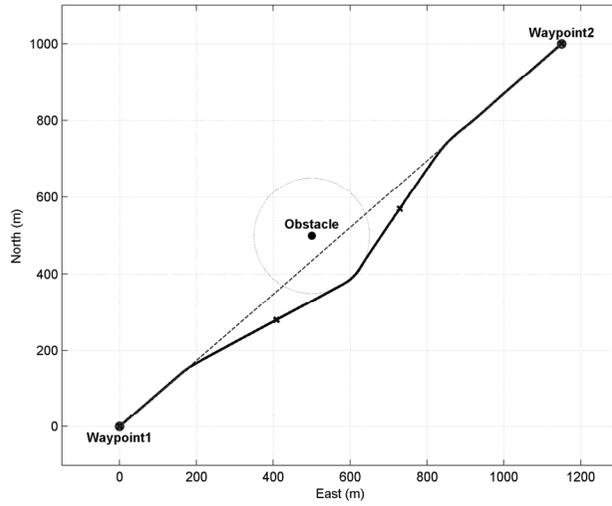


Figure 5.9 Trajectory of the UAV guidance with collision avoidance

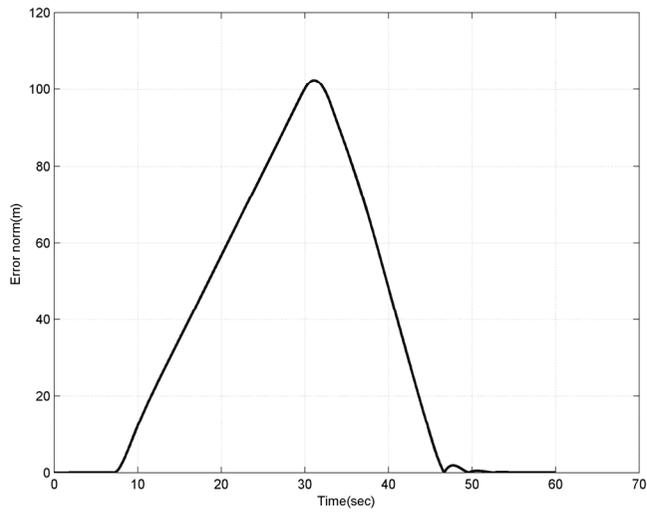


Figure 5.10 Position error between the UAV and its path

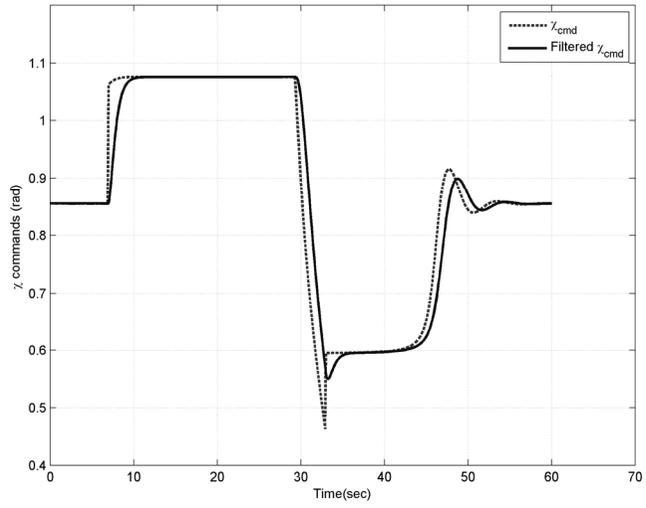


Figure 5.11 Time histories of the heading angle command and filtered heading angle command

5.2 Simulations on Vision-Based Reactive Collision Avoidance

In the cases of the proposed vision-based reactive collision avoidance (VRCA) algorithm, simulation parameters are summarized in Table 5.3.

Table 5.3 Parameters used in the VRCA algorithm

Vision-based reactive collision avoidance algorithm	
Speed of aircraft (UAV and intruder)	$V = 30 \text{ m/s}$
Sensing range	1500 m
Safety radius	$R_{safe} = 150 \text{ m}$
Command filter	$\zeta = 1, \omega_n = 2 \text{ rad/s}$
Size of the intruder	$R_r = 2 \text{ m}, [R_{min}, R_{max}] = [1, 4] \text{ m}$
Time condition	$t_L = -1 \text{ s}, t_U = t_c + 1.5 \text{ s}$

Because the proposed collision avoidance algorithm is conceptually one of the geometric approaches, comparison with the geometric optimization (GO) approach [24] is also performed. To implement the GO approach to the UAV using a single vision sensor,

the relative states of the intruder are constructed by using the assumed size of the intruder R_0 . Then, the GO approach is implemented.

Simulation scenarios are the same as the cases of the reactive collision avoidance in Chapter 5.1. In the simulations, the intruder is assumed to be noninteractive, or the intruder does not interact with the UAV. R_0 is assumed as $R_0 = 0.5R_r$. In the GO approach, the 2-dimensional collision avoidance maneuver will be obtained because the collision situations are defined in the 2-dimensional plane. Figures 5.12-5.15 shows the trajectories of the GO approach with $R_0 = 0.5R_r$. As shown in Figs. 5.12-5.15, the trajectory of the UAV deviates much from the original path because the avoidance direction is in the direction of the intruder's path. For this reason, the avoidance results are sensitive to the approach direction of the intruder.

Note that there exists a failure case that the UAV cannot resolve the conflict as shown in Fig. 5.12. Figure 5.12 shows that the UAV is dragged by the intruder when it determines a right-turn maneuver. This is because the determined direction has an uncertainty related to the loss-of-depth problem of the vision sensor. Therefore, it can be stated that the 2-dimensional solution may not guarantee the perfect collision avoidance for the vision-based UAV.

On the other hand, the proposed collision avoidance algorithm considering 3-dimensional avoidance is to mitigate the effect of the intruder's moving directions. Figures 5.16-5.19 show the trajectories of the VRCA algorithm. In the simulations, the time condition of the algorithm is not considered to check the effectiveness of the size assumption and the collision avoidance direction. As shown in Figs. 5.16-5.18, each UAV performs a climb maneuver to avoid the intruder. The velocity vectors of the two aircraft do not meet at any point during the simulations.

Let us consider a maximum deviation as well as the miss distance. The maximum deviation is defined as the UAV's maximum deviated distance from its original path, which is composed of line segments between the UAV's waypoints. Figure 5.20 shows the results of the two algorithms for the case of a noninteractive intruder when $R_0 = 0.5R_r$. It can be seen that the miss distances and maximum deviations of the proposed algorithm are consistent regardless of the approaching direction of the intruder, whereas those of the GO approach are not. Moreover, there exists a failure case for the GO approach when the intruder's heading is E.

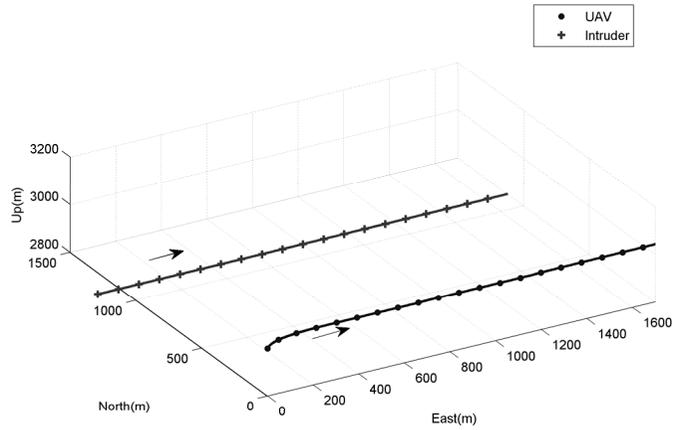


Figure 5.12 Trajectory of the GO approach (Noninteractive intruder, $R_0 = 0.5R_r$, intruder's heading: E, failure case)

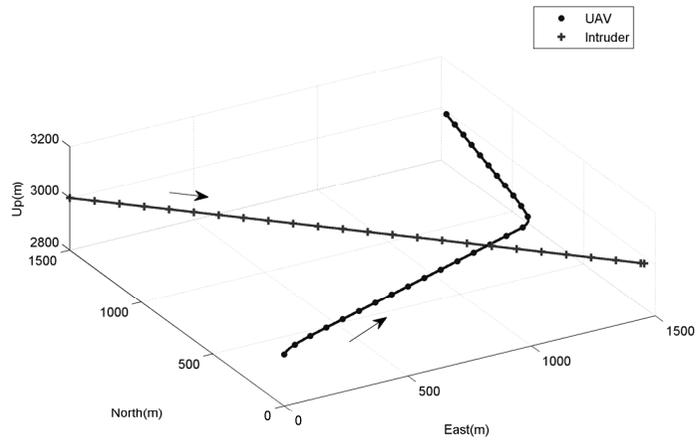


Figure 5.13 Trajectory of the GO approach (Noninteractive intruder, $R_0 = 0.5R_r$, intruder's heading: SE)

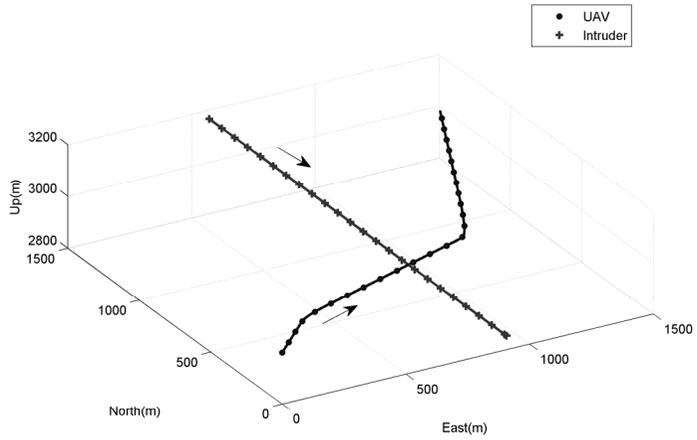


Figure 5.14 Trajectory of the GO approach (Noninteractive intruder, $R_0 = 0.5R_r$, intruder's heading: S)

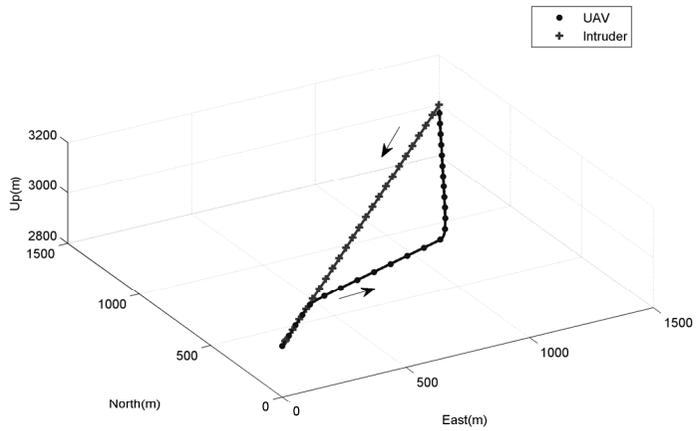


Figure 5.15 Trajectory of the GO approach (Noninteractive intruder, $R_0 = 0.5R_r$, intruder's heading: SW)

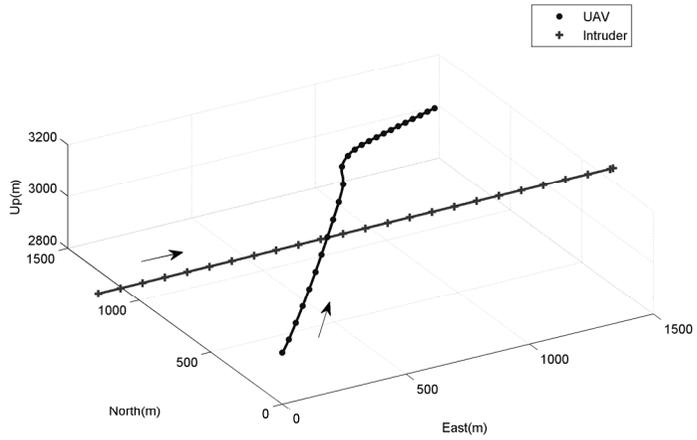


Figure 5.16 Trajectory of the VRCA (Noninteractive intruder, $R_0 = 0.5R_r$, intruder's heading: E)

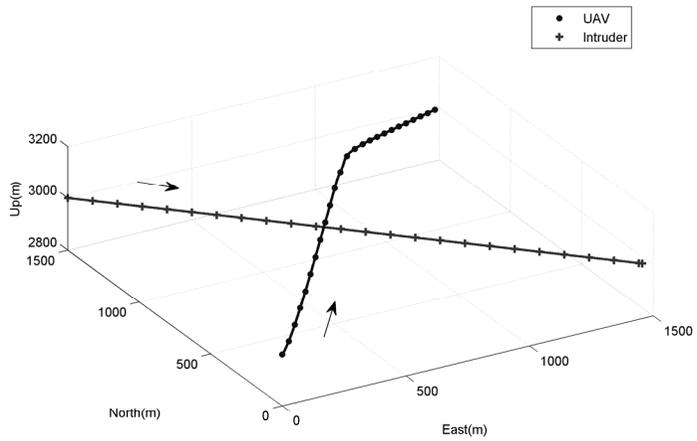


Figure 5.17 Trajectory of the VRCA (Noninteractive intruder, $R_0 = 0.5R_r$, intruder's heading: SE)

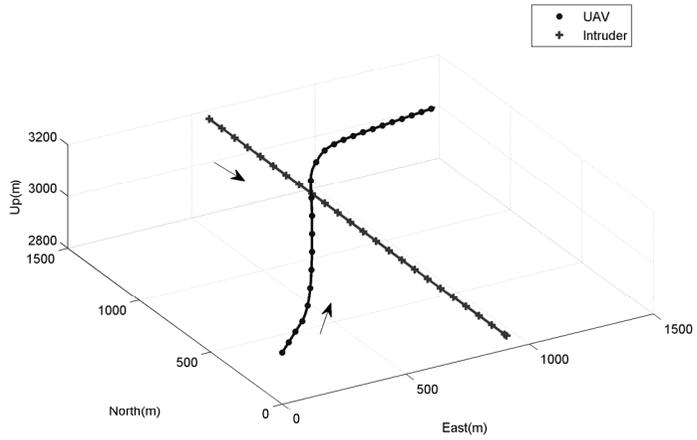


Figure 5.18 Trajectory of the VRCA (Noninteractive intruder, $R_0 = 0.5R_r$, intruder's heading: S)

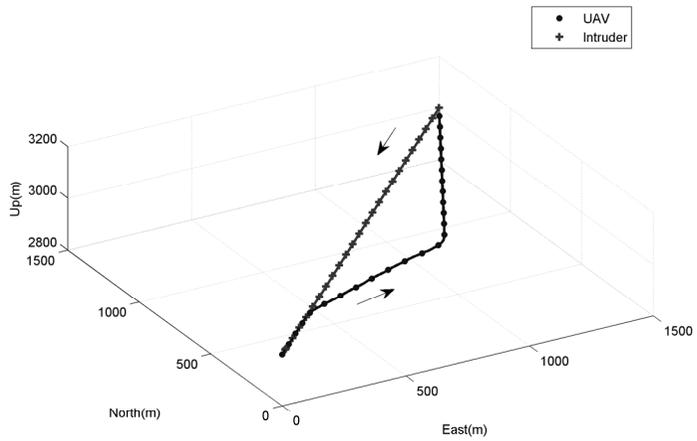
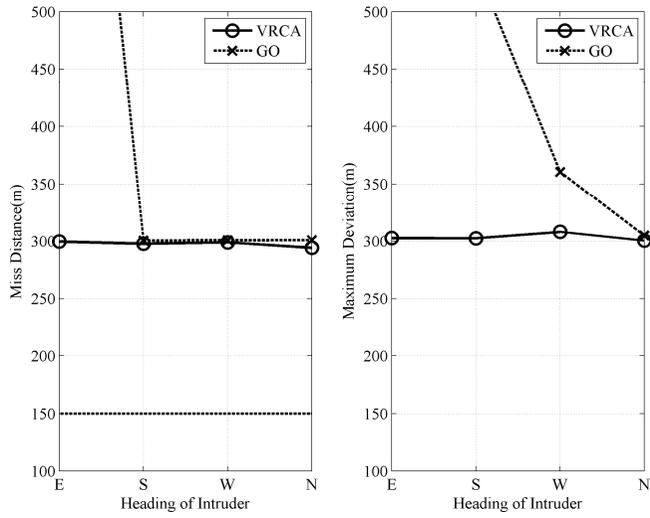


Figure 5.19 Trajectory of the VRCA (Noninteractive intruder, $R_0 = 0.5R_r$, intruder's heading: SW)



**Figure 5.20 Miss distance and maximum deviation results
(Noninteractive intruder, $R_0 = 0.5R_r$)**

Now, let us consider the cases of an interactive intruder, where the intruder has the same collision avoidance algorithm as the UAV. Figures 5.21-5.24 show the trajectories of the GO approach. As shown in Fig. 5.21, the UAV and intruder fail to avoid each other, which is related to the conflict of the algorithm as well as the uncertainty in the measurement. If the two aircraft performed the right-turn maneuvers at the same time, then the situation of the conflict could be resolved. However, the intruder did not determine the right-turn maneuver in the simulation because of the measurement uncertainty. As a result, the two aircraft using the GO approach cannot effectively resolve the conflict situation.

Figures 5.25-5.28 show the trajectories of the proposed collision avoidance algorithm. In the figures, the two aircraft are separated effectively without any conflicts because the avoidance directions of the two aircraft are opposite each other as mentioned in Theorem 4.1. Figure 5.29 shows the miss distance and maximum deviation for the GO approach and the proposed algorithm. According to Fig. 5.29, the overall features of the performance are very similar to those of Fig. 5.20, which are the cases of the noninteractive intruder.

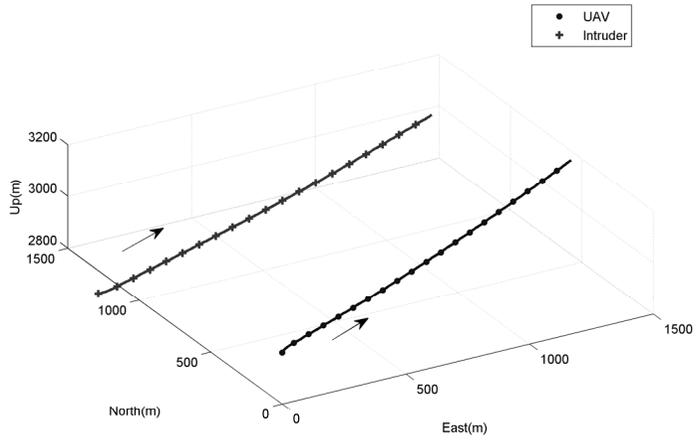


Figure 5.21 Trajectory of the GO approach (Interactive intruder, $R_0 = 0.5R_r$, intruder's heading: E, failure case)

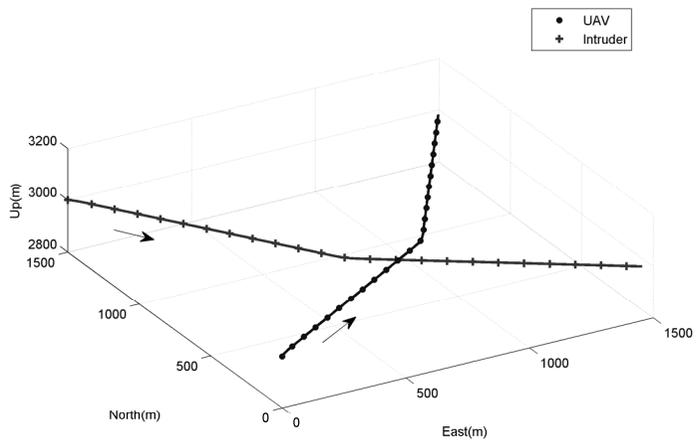


Figure 5.22 Trajectory of the GO approach (Interactive intruder, $R_0 = 0.5R_r$, intruder's heading: SE)

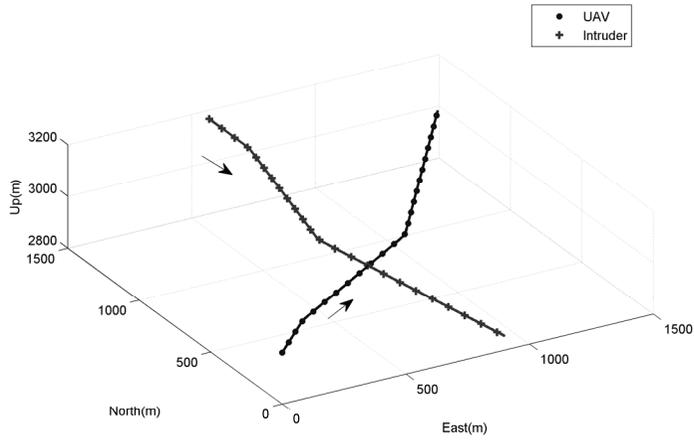


Figure 5.23 Trajectory of the GO approach (Interactive intruder, $R_0 = 0.5R_r$, intruder's heading: S)

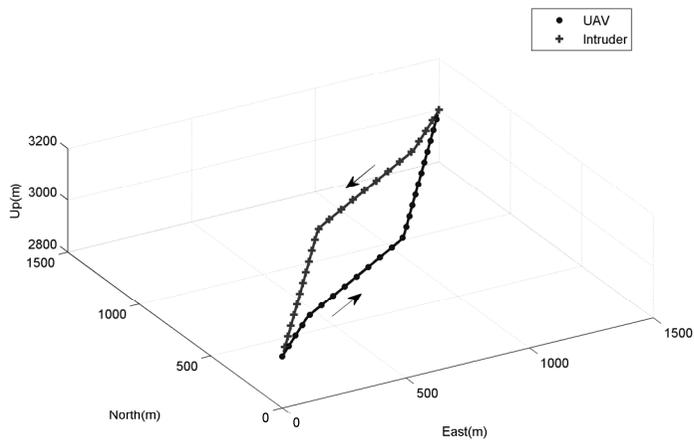


Figure 5.24 Trajectory of the GO approach (Interactive intruder, $R_0 = 0.5R_r$, intruder's heading: SW)

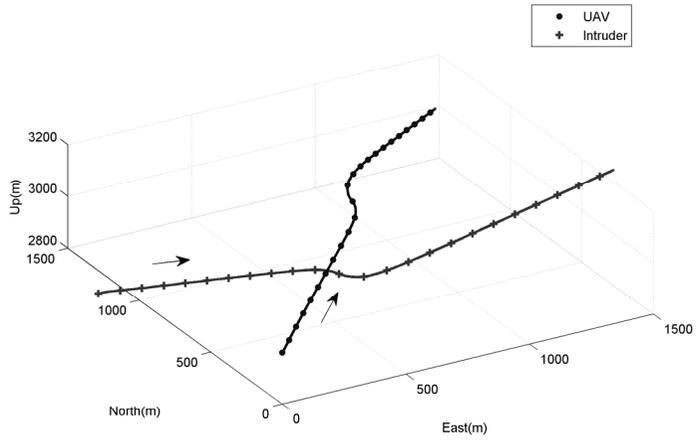


Figure 5.25 Trajectory of the VRCA (Interactive intruder, $R_0 = 0.5R_r$, intruder's heading: E)

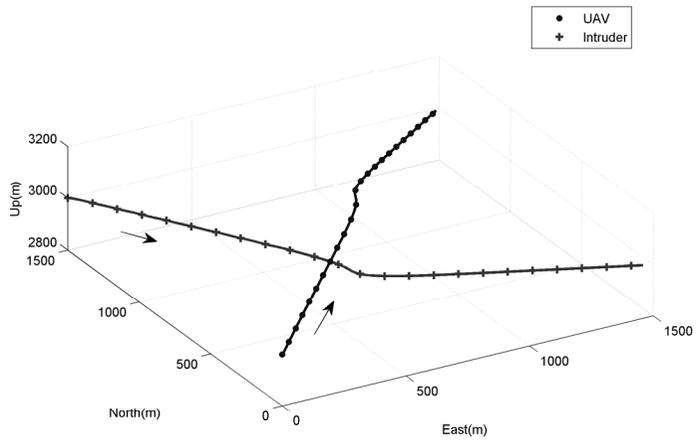


Figure 5.26 Trajectory of the VRCA (Interactive intruder, $R_0 = 0.5R_r$, intruder's heading: SE)

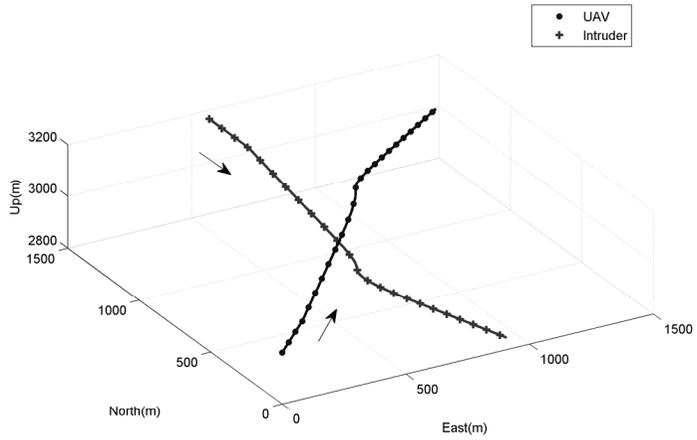


Figure 5.27 Trajectory of the VRCA (Interactive intruder, $R_0 = 0.5R_r$, intruder's heading: S)

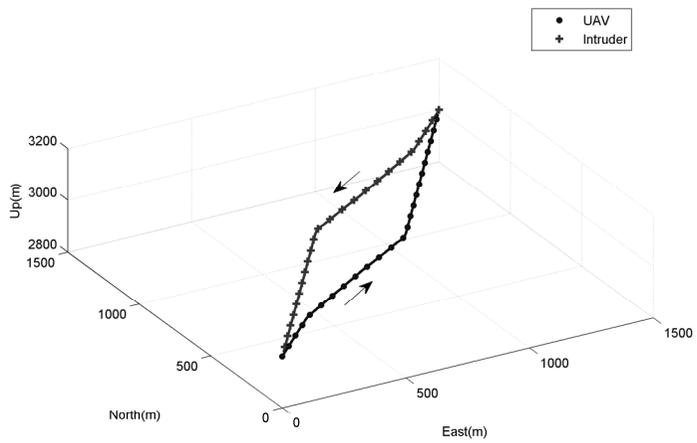
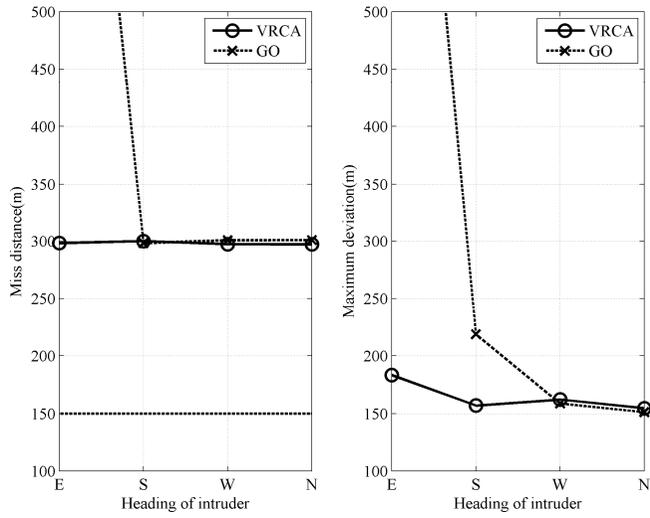


Figure 5.28 Trajectory of the VRCA (Interactive intruder, $R_0 = 0.5R_r$, intruder's heading: SW)



**Figure 5.29 Miss distance and maximum deviation results
(Interactive intruder, $R_0 = 0.5R_r$)**

The time condition prevents the miss distance and maximum deviation from becoming too large in accordance with Theorem 4.2. Figures 5.30 and 5.31 show the trajectories of the proposed collision avoidance algorithm with the time condition. As shown in Figs. 5.30 and 5.31, the UAV performs the collision avoidance maneuver when the intruder is close, which is different from the cases of Fig. 5.17 and Fig. 5.19. The time condition makes the UAV perform restricted evasive maneuvers by delaying the activation time of the algorithm. Because the states of the UAV are bounded through the command filter, it can resolve the collision situation safely.

Figure 5.32 shows the effect of the time condition on the miss distance for the avoidance of a noninteractive intruder. It can be seen that the miss distances are regulated by the time condition even though the ratio of R_0/R_r is too small. Similarly, the maximum deviation can be also regulated by the time condition as shown in Fig. 5.33.

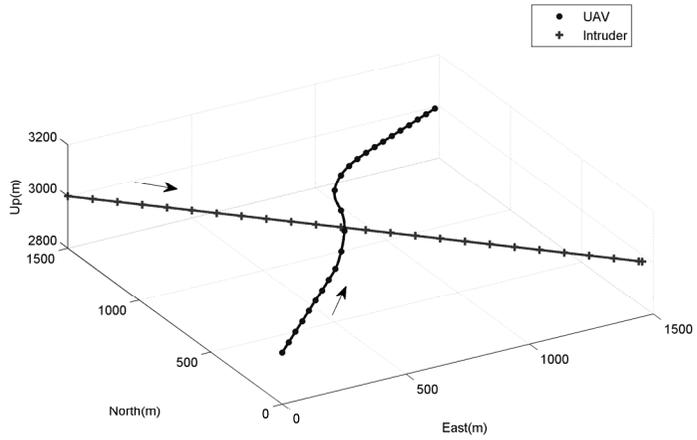


Figure 5.30 Trajectory of the VRCA with time condition (Noninteractive intruder, $R_0 = 0.5R_r$, intruder's heading: SE)

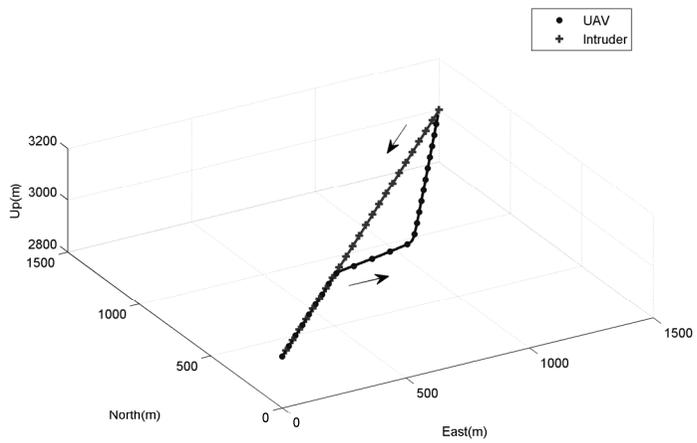
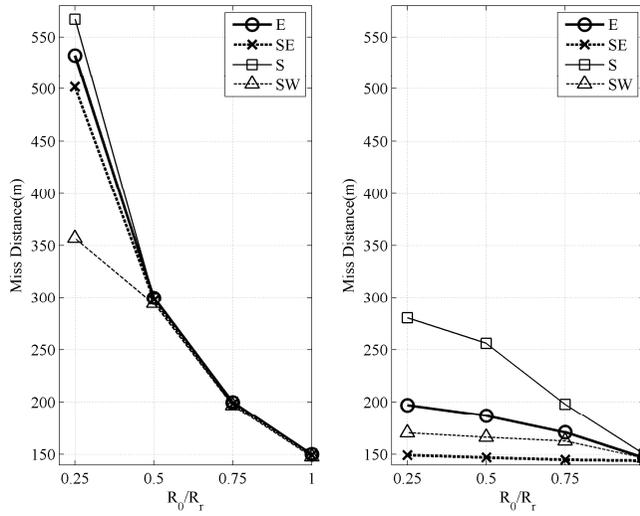
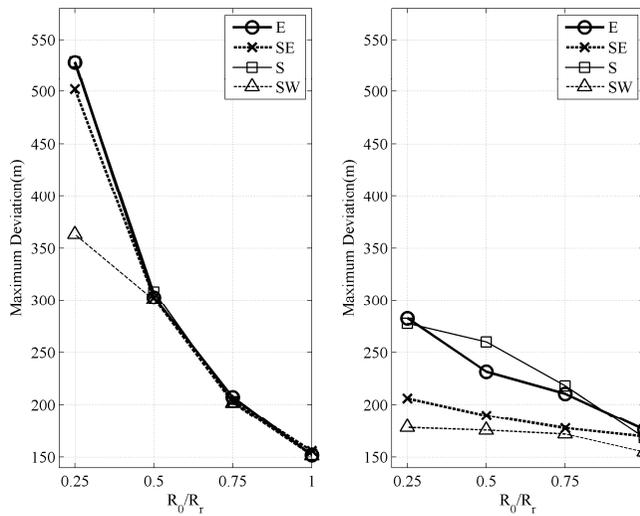


Figure 5.31 Trajectory of the VRCA with time condition (Noninteractive intruder, $R_0 = 0.5R_r$, intruder's heading: SW)



a) Without time condition b) With time condition

Figure 5.32 Miss distance results for noninteractive intruder



a) Without time condition b) With time condition

Figure 5.33 Maximum deviation results for noninteractive intruder

Chapter 6

Conclusions

6.1 Concluding Remarks

In this dissertation, a vision-based reactive collision avoidance algorithm has been proposed for the “Sense-and-Avoid” of UAV. In particular, a single vision sensor, which has perspective vision characteristics, is considered as a sensor of the UAV. The vision sensor has the loss-of-depth problem, and it is difficult to be utilized in collision avoidance situations. The problem has been dealt with by adopting a measurement model of a vision sensor and the geometry of the CPA. Using the characteristics of the CPA, a vision-based reactive collision avoidance algorithm has been proposed, which is based on the conservative assumption of the intruder’s size to make sure of safety. The proposed algorithm generates a collision avoidance direction, which is robust with respect to the motion of the intruder. Thus, the algorithm provides a safe and robust solution with respect to the interactive intruder as well as the noninteractive intruder. A time

condition is also proposed to restrict the unexpected trajectory extension caused by the conservative assumption. Thus, the proposed collision avoidance algorithm makes it possible to avoid a moving intruder by using a single vision sensor only. Numerical simulations have demonstrated the effectiveness of the proposed algorithm.

A guidance law which guarantees convergence with a desired trajectory has been also proposed. The guidance law is designed by considering stable internal dynamics, and therefore the corresponding guidance commands, which consist of speed, heading angle, and flight path angle, guarantee exact trajectory tracking. In addition, bounds of the commands related to the kinematic constraints of the UAV are considered. The commands are compatible with the commands of the collision avoidance algorithm. Thus, a guidance and collision avoidance system of the UAV based on the command switching scheme and command filter is proposed.

6.2 Future Works

The proposed collision avoidance algorithm requires reliable measurement data related to the image processing of the vision sensor, which is an important problem of the vision-based navigation. Moreover, the vision sensor should detect distant intruders to avoid them, and therefore a high-definition vision sensor is required. Also, noise of the sensor could degrade the collision avoidance performance. These problems could be limitations of the proposed collision avoidance algorithm. The performance could be also degraded for rapidly maneuvering intruders or unidentified types of intruders because these intruders may nullify the assumptions in the collision avoidance algorithm. Therefore, studies to overcome these limitations remain as future works.

In relation to the limitations of the vision sensor, experimental demonstrations are required to show the effectiveness of the proposed algorithm in the real environment. These experimental demonstrations include sensor selection, application of image processing techniques, noise filtering, and various tests including ground and flight tests.

In addition, autopilot design using the UAV dynamic model should

be done before implementing the vision sensor because the autopilot, which was considered as a first-order dynamic system in this study, does not exactly correspond with the actual UAV system. The autopilot design also requires experimental demonstrations through flight tests. Therefore, these issues also remain as future works.

Bibliography

- [1] Fahlstorm, P. G. and Gleason, T. J. *Introduction to UAV Systems*, UAV Systems Inc., Columbia, MD, 1994.
- [2] Austin, R., *Unmanned Aircraft Systems: UAVS Design, Development and Deployment*, John Wiley & Sons, New York, NY, 2010.
- [3] Beard, R. W. and McLain, T. W., *Small Unmanned Aircraft: Theory and Practice*, Princeton University Press, Princeton, NJ, 2012.
- [4] Choset, H., Lynch, K. M., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L. E., and Thrun, S., *Principles of Robot Motion: Theory, Algorithm, and Implementations*, MIT Press, Cambridge, MA, 2005, pp. 77-268.
- [5] LaValle, S. M., *Planning Algorithm*, Cambridge University Press, Cambridge, UK, 2006.
- [6] Federal Aviation Administration, Federal Aviation Regulations Title 14 CFR part 91 - General Operating and Flight Rules, *U.S. Department of Transportation*, [online database],
URL: <http://www.gpoaccess.gov/cfr> [cited Oct. 2010].
- [7] Federal Aviation Administration, Advisory Circular 90-48C, Pilot's Role in Collision Avoidance, *U.S. Department of Transportation*, 1983.

- [8] Leveson, N. G., Heimdahl, M. P. E., Hildreth, H., and Reese, J. D., "Requirements Specification for Process-Control Systems," *IEEE Transactions on Software Engineering*, Vol. 20, No. 9, 1994, pp. 684-707.
- [9] Livadas, C., Lygeros, J., and Lynch, N. A., "High-Level Modeling and Analysis of the Traffic Alert and Collision Avoidance System," *Proceedings of the IEEE*, Vol. 88, No. 7, 2000, pp. 926-948.
- [10] Romli, F. I., King, J. D., Li, L. and Clarke, J. P., "Impact of Automatic Dependent Surveillance-Broadcast (ADS-B) on Traffic Alert and Collision Avoidance System (TCAS) Performance," *AIAA Guidance, Navigation and Control Conference*, Honolulu, HI, Aug. 2008.
- [11] Shakenia, O., Chen, W.-Z., Graham, S., Zvanya, J., White, A., Weingarten, N., and Raska, V. M., "Sense and Avoid (SAA) Flight Test and Lessons Learned," *AIAA Infotech@Aerospace Conference*, Rohnert Park, CA, May 2007.
- [12] Zeitlin, A. D., "Sense & Avoid Capability Development Challenges," *IEEE Aerospace and Electronic Systems Magazine*, Vol. 25, No. 10, 2010, pp. 27-32.
- [13] Barnhart, R. K., Hottman, S. B., Marshall, D. M., Shappee, E.,

Introduction to Unmanned Aircraft Systems, 1st ed., CRC press, Boca Raton, FL, 2011, pp. 137-154.

[14] Kephart, R. J., and Braasch, M. S., “See-and-Avoid Comparison of Performance in Manned and Remotely Piloted Aircraft,” *IEEE Aerospace and Electronic Systems Magazine*, Vol. 25, No. 5, 2010, pp. 36-42.

[15] Forsyth, D., and Ponce, J., *Computer Vision: A Modern Approach*, Prentice-Hall, NewYork, NY, 2003, pp. 3-19.

[16] Lee, S., Yoon, S., Kim, H. J., and Kim, Y., “Wireless Stereo Vision System Development for Rotary-wing UAV Guidance and Control,” *The 3rd International Conference on Sensing Technology*, Tainan, Taiwan, Nov.-Dec. 2008.

[17] Jankovic, M., and Ghosh, B. K., “Visually Guided Ranging from Observations of Points, Lines, and Curves via Identifier based Nonlinear Observer,” *Systems and Control Letter*, Vol. 25, No. 1, 1995, pp. 63-73.

[18] Borenstein, J., and Koren, Y., “The Vector Field Histogram – Fast Obstacle Avoidance for Mobile Robots,” *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 3, 1991, pp. 278-288.

[19] Minguez, J., and Montano, L., “Nearness Diagram (ND)

Navigation: Collision Avoidance in Troublesome Scenarios,” *IEEE Transactions on Robotics and Automation*, Vol. 20, No. 1, 2004, pp. 45-59.

[20] Chakravarthy, A., and Ghose, D., “Obstacle Avoidance in a Dynamic Environment: A Collision Cone Approach,” *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, Vol. 28, No. 5, 1998, pp. 562-574.

[21] Fiorini, P. and Shiller, Z., “Motion Planning in Dynamic Environments using Velocity Obstacles,” *International Journal of Robotics Research*, Vol. 17, No. 7, 1998, pp. 760-772.

[22] Kuchar, J. K., and Yang, L. C., “A Review of Conflict Detection and Resolution Modeling Methods,” *IEEE Transactions on Intelligent Transportation Systems*, Vol. 1, No. 4 2000, pp. 179-189.

[23] Zeghal, K., “A Comparison of Different Approaches based on Force Fields for Coordination among Multiple Mobiles,” *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Victoria, B.C., Canada, Oct. 1998.

[24] Bilimoria, K. D., “A Geometric Optimization Approach to Aircraft Conflict Resolution,” *AIAA Guidance, Navigation and Control Conference*, Denver, CO, Aug. 2000.

- [25] Hwang, I., Kim, J., and Tomlin, C., "Protocol-Based Conflict Resolution for Air Traffic Control," *Air Traffic Control Quarterly*, Vol. 15, No. 1, 2008, pp. 1-34.
- [26] Lalish, E., Morgansen, K. A., and Tsukamaki, T., "Decentralized Reactive Collision Avoidance for Multiple Unicycle-type Vehicles," *American Control Conference*, Seattle, WA, June 2008.
- [27] Han, S., Bang, H., and Yoo, C., "Proportional Navigation-Based Collision Avoidance for UAVs," *International Journal of Control, Automation, and Systems*, Vol. 7, No. 4, 2009, pp. 553-565.
- [28] Carbone, C., Ciniglio, U., Corraro, F., and Luongo, S., "A Novel 3D Geometric Algorithm for Aircraft Autonomous Collision Avoidance," *IEEE Conference on Decision and Control*, San Diego, CA, Dec. 2006.
- [29] Mujumdar, A., and Padhi, R., "Reactive Collision Avoidance Using Nonlinear Geometric and Differential Geometric Guidance," *Journal of Guidance, Control, and Dynamics*, Vol. 34, No. 1, 2011, pp. 303-310.
- [30] Gates, D. J., "Properties of a Real-Time Guidance Method for Preventing a Collision," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 3, 2009, pp. 705-716.

- [31] Gates, D. J., Gates, E. A., Westcott, M., and Fulton, N. L., "Stereo Projections of Miss Distance in Some New Cockpit Display Formats," *Journal of Aircraft*, Vol. 45, No. 5, 2008, pp. 1725-1735.
- [32] Choi, H., Kim, Y., Lee, Y., and Kim, E. T., "A Reactive Collision Avoidance Algorithm for Multiple Midair Unmanned Aerial Vehicles," *Transactions of the Japan Society for Aeronautical and Space Sciences*, Vol. 56, No. 1, 2013, pp.15-24.
- [33] Kim, H. J., Shim, D. H., and Sastry, S., "Nonlinear Model Predictive Tracking Control for Rotorcraft-based Unmanned Aerial Vehicles," *American Control Conference*, Anchorage, AK, May 2002.
- [34] Ha, J., Johnson, E., and Tannenbaum, A., "Real-time Visual Tracking Using Geometric Active Contours and Particle Filters," *Journal of Aerospace Computing, Information, and Communication*, Vol. 5, No. 10, 2008, pp. 361-379.
- [35] Vela, P., Bester, A., Malcolm, J., and Tannenbaum, A., "Vision-Based Range Regulation of a Leader-Follower Formation," *IEEE Transactions on Control Systems Technology*, Vol. 17, No. 2, 2009, pp. 442-448.
- [36] Oh, S., and Johnson, E. N., "Relative Motion Estimation for Vision-based Formation Flight using Unscented Kalman Filter," *AIAA*

Guidance, Navigation, and Control Conference, Hilton Head, SC, Aug. 2007.

[37] Calise, A. J., Johnson, E. N., Sattigeri, R., Watanabe, Y., and Madyastha, V., “Estimation and Guidance Strategies for Vision-based Target Tracking,” *American Control Conference*, Portland, OR, June 2005.

[38] Stepanyan, V., and Hovakimyan, N., “Visual Tracking of a Maneuvering Target,” *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 1, 2008, pp. 66-80.

[39] Choi, H., and Kim, Y., “UAV Guidance using a Monocular-Vision Sensor for Aerial Target Tracking,” *Control Engineering Practice*, Vol. 22, 2014, pp. 10-19.

[40] Watanabe, Y., Calise, A. J., and Johnson, E. N., “Vision-Based Obstacle Avoidance for UAVs,” *AIAA Guidance, Navigation and Control Conference*, Hilton Head, SC, Aug. 2007.

[41] Dippold, A., Ma, L., and Hovakimyan, N., “Vision-Based Obstacle Avoidance of Wheeled Robots Using Fast Estimation,” *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 6, 2009, pp. 1931-1937.

[42] Shakernia, O., Chen, W.-Z., and Raska, M. V. M., “Passive Ranging for UAV Sense and Avoid Applications,” *AIAA*

Infotech@Aerospace Conference, Arlington, VA, Sep. 2005.

[43] Green, W. E., and Oh, P. Y., “Optic-Flow-Based Collision Avoidance,” *IEEE Robotics and Automation Magazine*, Vol. 15, No. 1, 2008, pp. 96-103.

[44] Choi, H., Kim, Y., and Hwang, I., “Reactive Collision Avoidance of UAV using a Single Vision Sensor,” *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 4, 2013, pp.1234-1240.

[45] No, T. S., Min, B. M., Stone, R. H., and Wong, K. C., “Control and Simulation of Arbitrary Flight Trajectory-Tracking,” *Control Engineering Practice*, Vol. 13, No. 5, 2005, pp. 601-612.

[46] Stevens, B., and Lewis, F., *Aircraft Control and Simulation*, 2nd ed., Wiley, New York, NY, 1992, pp. 1-142.

[47] Kim, S., and Kim, Y., “Optimum Design of Three-dimensional Behavioural Decentralized Controller for UAV Formation Flight,” *Engineering Optimization*, Vol. 41, No. 3, 2009, pp. 199-224.

[48] Farrel, J., Sharma, M., and Plyarpou, M., “Backstepping-Based Flight Control with Adaptive Function Approximation,” *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 6, 2005, pp. 1089-1102.

[49] Sonneveldt, L., van Oort, E. R., Chu, Q. P., and Mulder, J. A., “Nonlinear Adaptive Trajectory Control Applied to an F-16 Model,”

Journal of Guidance, Control, and Dynamics, Vol. 32, No. 1, 2009, pp.25-39.

[50] Kanayama, Y., Kimura, Y., Miyazaki, F., and Noguchi, T., “A Stable Tracking Control Method for a Non-Holonomic Mobile Robot,” *IEEE/RSJ International Workshop on Intelligent Robots and Systems*, Osaka, Japan, Nov. 1991.

[51] Hogg, R. W., Rankin, A. L., Roumeliotis, S. I., McHenry, M. C., Helmick, D. M., Bergh, C. F., and Matthies, L., “Algorithms and Sensors for Small Robot Path Following,” *IEEE International Conference on Robotics & Automation*, Washington, DC, May 2002.

[52] Wit, J., Crane III, C. D., and Armstrong, D., “Autonomous Ground Vehicle Path Tracking,” *Journal of Robotic Systems*, Vol. 21, No. 8, 2004, pp. 439-449.

[53] Slotine, J.-J. E., and Li. W., *Applied Nonlinear Control*, Prentice Hall, Eaglewood Cliffs, NJ, 1991, pp. 40-99.

[54] Khalil, H. K., *Nonlinear Systems*, Prentice Hall, Upper Saddle River, NJ, 2002.

[55] Cambone, S. A., Krieg, K. J., Pace, P., and Wells II, L., Unmanned Aircraft Systems (UAS) Roadmap 2005-2030, *Office of the Secretary of Defense*, 2005, pp. 4-10.

Appendix A

Lyapunov Stability Theorem

These materials are borrowed from [54].

Consider an autonomous system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) \tag{A.1}$$

where $\mathbf{f} : D \rightarrow \mathbb{R}^n$ is a locally Lipschitz map from a domain $D \subset \mathbb{R}^n$ into \mathbb{R}^n . Suppose $\bar{\mathbf{x}} \in D$ is an equilibrium point of Eq. (A.1); that is, $\mathbf{f}(\bar{\mathbf{x}}) = \mathbf{0}$. Let us consider the following definition.

Definition A.1. The equilibrium point $\mathbf{x} = \bar{\mathbf{x}}$ of Eq. (A.1) is

- stable if, for each $\varepsilon > 0$, there is $\delta = \delta(\varepsilon) > 0$ such that

$$\|\mathbf{x}(0) - \bar{\mathbf{x}}\| < \delta \Rightarrow \|\mathbf{x}(t) - \bar{\mathbf{x}}\| < \varepsilon, \quad \forall t \geq 0$$

- unstable if it is not stable
- asymptotically stable if it is stable and δ can be chosen such

that

$$\|\mathbf{x}(0) - \bar{\mathbf{x}}\| < \delta \Rightarrow \lim_{t \rightarrow \infty} \mathbf{x}(t) = \bar{\mathbf{x}}$$

Considering a continuously differentiable function called as

Lyapunov candidate function V , the following Lyapunov stability theorem can be stated.

Theorem A.1. Let $\mathbf{x} = \bar{\mathbf{x}}$ be an equilibrium point for Eq. (A.1) and $D \subset \mathbb{R}^n$ be a domain containing $\mathbf{x} = \bar{\mathbf{x}}$, Let V be a continuously differentiable function such that

$$V(\bar{\mathbf{x}}) = 0 \text{ and } V(\mathbf{x}) > 0 \text{ in } D - \{\bar{\mathbf{x}}\} \quad (\text{A.2})$$

$$\dot{V}(\mathbf{x}) \leq 0 \text{ in } D \quad (\text{A.3})$$

Then, $\mathbf{x} = \bar{\mathbf{x}}$ is stable, Moreover, if

$$\dot{V}(\mathbf{x}) < 0 \text{ in } D - \{\bar{\mathbf{x}}\} \quad (\text{A.4})$$

Then $\mathbf{x} = \bar{\mathbf{x}}$ is asymptotically stable.

The proof is presented in [54].

Appendix B

Perspective Projection of Vision Sensor

These materials are partially borrowed from [15].

The pinhole camera model is a simplified vision sensor model, and it provides an approximation of the imaging process. According to the model, perspective projection of a real object is done onto an image plane as shown in Fig. B.1.

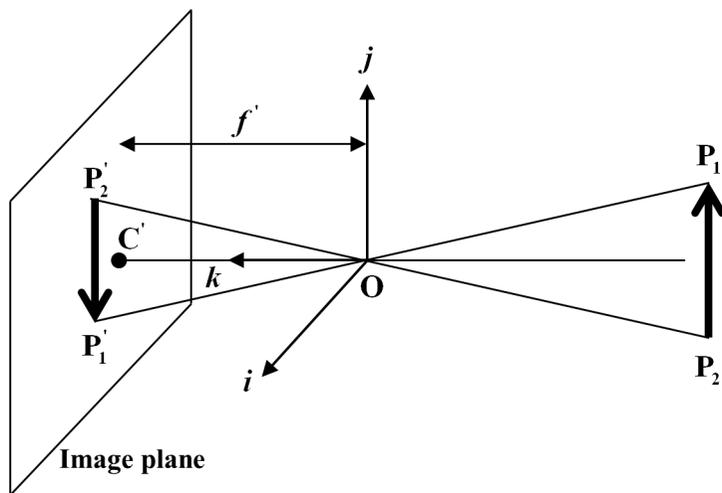


Figure B.1 Pinhole perspective projection model

Let us consider that P_1 has the coordinates (x_1, y_1, z_1) and P'_1

has the coordinates (x'_1, y'_1, z'_1) , respectively. Note that $z'_1 = f'$, which is a focal length of the sensor. Since the three points \mathbf{P}_1 , \mathbf{O} , and \mathbf{P}'_1 are collinear, $\mathbf{OP}'_1 = \lambda \mathbf{OP}_1$ for some number λ . Then, the following relation can be obtained.

$$\lambda = \frac{x'_1}{x_1} = \frac{y'_1}{y_1} = \frac{f'}{z_1} \quad (\text{B.1})$$

$$x'_1 = f' \frac{x_1}{z_1}, \quad y'_1 = f' \frac{y_1}{z_1} \quad (\text{B.2})$$

The points \mathbf{P}_2 and \mathbf{P}'_2 have the similar relations. Note that \mathbf{P}'_1 and \mathbf{P}'_2 can be obtained from the vision sensor. Then, the middle point of the image can be obtained, and relative direction angles of the real object can be obtained. Also, let us define the length of $\mathbf{P}'_1\mathbf{P}'_2$, \mathbf{OP}'_1 , and \mathbf{OP}'_2 as follows.

$$P'_1P'_2 \triangleq a, \quad OP'_1 \triangleq b, \quad OP'_2 \triangleq c \quad (\text{B.3})$$

Then, the angle $\angle \mathbf{P}'_1\mathbf{O}\mathbf{P}'_2 = \angle \mathbf{P}_1\mathbf{O}\mathbf{P}_2 \triangleq \theta$ can be obtained as follows.

$$\theta = \cos^{-1} \left(\frac{b^2 + c^2 - a^2}{2bc} \right) \quad (\text{B.4})$$

These measurements are utilized in Chapter 4.

초 록

영상센서는 수동적이고, 비협조적 특성으로 인해 무인항공기의 충돌탐지 및 충돌회피 문제를 해결할 수 있는 잠재력을 지닌다. 이에 따라 움직이는 침입기를 회피하기 위한 무인항공기의 “Sense-and-Avoid”에 적합한 센서로 고려된다. 일반적인 영상센서는 장애물과의 상대거리를 측정하지 못하기 때문에 장애물의 상태를 파악하는 데에 어려움을 준다. 침입기와 같은 이동 장애물에 대해서는 장애물의 움직임에 따른 불확정성 또한 존재한다. 따라서 기존의 충돌회피 알고리즘을 영상센서를 기반으로 하는 무인항공기에 직접 적용할 수 없는 문제가 발생한다. 본 논문에서는 이러한 문제점들을 해결하기 위해 무인항공기의 충돌회피 시스템에 영상센서를 적용한 영상기반 반응적 충돌회피 알고리즘을 제안하였다. 제안한 충돌회피 알고리즘은 단일 영상센서의 측정치 모델과 최근접점 기하학을 이용하여 충돌회피에 유용한 특징들을 도출하고, 도출된 특징을 바탕으로 침입기 크기를 가정하여 침입기 움직임에 강건하고 안전한 충돌회피 방향을 제공한다. 침입기의 크기에 대한 가정으로 인해 발생하는 무인항공기의 불필요한 기동들을 제한하기 위해 알고리즘의 실행 타이밍을 고려한 시간

조건을 추가로 제안하였다.

한편, 무인항공기는 목표점까지 또는 기준궤적을 따라 비행하기 위해 유도법칙이 필요하고, 이 유도법칙은 충돌회피 알고리즘과 호환되어야 한다. 이를 위해 본 논문에서는 경로점 추종, 경로점 및 경로선 추종, 궤적 추종 문제에 대한 무인항공기 유도법칙을 제안하였다. 유도법칙은 수렴성을 보장하는 속력, 방향각, 비행 경로각의 명령을 생성하는데, 이 명령들은 충돌회피 알고리즘에서 생성되는 명령과 동일한 형태를 지닌다. 따라서 유도법칙과 충돌회피 알고리즘을 결합한 시스템을 구성하였으며, 부드러운 명령전환을 위해 명령필터를 사용하도록 제안하였다. 본 논문에서 제안된 충돌회피와 유도 알고리즘의 유효성은 수치 시뮬레이션을 통해 검증하였다.

주요어: 무인항공기, 충돌회피, 단일 영상센서, 이동장애물, 최근접점, 궤적추종

학 번: 2010-30200