2016-01-01

Airborne Collision Detection and Avoidance for Small UAS Sense and Avoid Systems

Laith Rasmi Sahawneh
Brigham Young University - Provo

Airborne Collision Detection and Avoidance for Small UAS Sense and Avoid Systems

Laith Rasmi Sahawneh

A dissertation submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Randal W. Beard, Chair
Timothy W. McLain
Dah J. Lee
Neal K. Bangerter
John L. Salmon

Department of Electrical and Computer Engineering

Brigham Young University

January 2016

ABSTRACT

Airborne Collision Detection and Avoidance for Small UAS Sense and Avoid Systems

Laith Rasmi Sahawneh
Department of Electrical and Computer Engineering, BYU
Doctor of Philosophy

The increasing demand to integrate unmanned aircraft systems (UAS) into the national airspace is motivated by the rapid growth of the UAS industry, especially small UAS weighing less than 55 pounds. Their use however has been limited by the Federal Aviation Administration regulations due to collision risk they pose, safety and regulatory concerns. Therefore, before civil aviation authorities can approve routine UAS flight operations, UAS must be equipped with sense-and-avoid technology comparable to the see-and-avoid requirements for manned aircraft.

The sense-and-avoid problem includes several important aspects including regulatory and system-level requirements, design specifications and performance standards, intruder detecting and tracking, collision risk assessment, and finally path planning and collision avoidance. In this dissertation, our primary focus is on developing an collision detection, risk assessment and avoidance framework that is computationally affordable and suitable to run on-board small UAS. To begin with, we address the minimum sensing range for the sense-and-avoid (SAA) system. We present an approximate close form analytical solution to compute the minimum sensing range to safely avoid an imminent collision. The approach is then demonstrated using a radar sensor prototype that achieves the required minimum sensing range.

In the area of collision risk assessment and collision prediction, we present two approaches to estimate the collision risk of an encounter scenario. The first is a deterministic approach similar to those been developed for Traffic Alert and Collision Avoidance (TCAS) in manned aviation. We extend the approach to account for uncertainties of state estimates by deriving an analytic expression to propagate the error variance using Taylor series approximation. To address unanticipated intruders maneuvers, we propose an innovative probabilistic approach to quantify likely intruder trajectories and estimate the probability of collision risk using the uncorrelated encounter model (UEM) developed by MIT Lincoln Laboratory. We evaluate the proposed approach using Monte Carlo simulations and compare the performance with linearly extrapolated collision detection logic.

For the path planning and collision avoidance part, we present multiple reactive path planning algorithms. We first propose a collision avoidance algorithm based on a simulated chain that responds to a virtual force field produced by encountering intruders. The key feature of the proposed approach is to model the future motion of both the intruder and the ownship using a chain of waypoints that are equally spaced in time. This timing information is used to continuously re-plan paths that minimize the probability of collision. Second, we present an innovative collision avoidance logic using an ownship centered coordinate system. The technique builds a graph in the local-level frame and uses the Dijkstra's algorithm to find

the least cost path. An advantage of this approach is that collision avoidance is inherently a local phenomenon and can be more naturally represented in the local coordinates than the global coordinates. Finally, we propose a two step path planner for ground-based SAA systems. In the first step, an initial suboptimal path is generated using A* search. In the second step, using the A* solution as an initial condition, a chain of unit masses connected by springs and dampers evolves in a simulated force field. The chain is described by a set of ordinary differential equations that is driven by virtual forces to find the steady-state equilibrium. The simulation results show that the proposed approach produces collision-free plans while minimizing the path length.

To move towards a deployable system, we apply collision detection and avoidance techniques to a variety of simulation and sensor modalities including camera, radar and ADS-B along with suitable tracking schemes.

Keywords: unmanned aircraft system, small UAS, sense and avoid, minimum sensing range, airborne collision detection and avoidance, collision detection, collision risk assessment, collision avoidance, conflict detection, conflict avoidance, path planning.

ACKNOWLEDGMENTS

First, I would like to take this opportunity to express my deepest thanks and gratitude to my advisor, Professor Randal Beard, for his excellent guidance, funding and mentor to me over the last four years. His feedback and insights have been greatly beneficial to my research and to my academic career, and more importantly, my development as a student and researcher. I also sincerely thank Professor Tim McLain for the attentive mentorship he has provided me in all aspects of the SAA project. I especially appreciate all of his help with writing in our ATCQ paper. I am also very grateful to Professor Dah Jye Lee, Professor Neal Bangerter and Dr. John Salmon for being on my committee.

I feel privileged to have worked in the MAGICC Lab and the Center for Unmanned Aircraft Systems (C-UAS) during the past four years. They are great environment that fosters learning and innovation. Special thanks goes to the former and present team members in the sense-and-avoid project: Jared Wikle, Matthew Duffield, Robert Klaus, and Paul Nyholm. I have also greatly enjoyed and benefitted from the white-board discussions with Matthew Argyle, David Wheeler, Peter Niedfeldt, Rajnikant Sharma, and Eric Quist. Thanks to all other professors and students with whom I have directly worked on various papers: Professor Karl Warnick, Jonathan Spencer, James Mackie, Kaleo Roberts and Michael Boren.

I owe more than I can say to my parents, who have always been there to guide and aid me, and who have taught me so much through example. A very special thanks goes to our second family in Provo, Lamar and Eileen Newton for their love, friendship, Christmas and New Year's Eves. I am extremely grateful to my wonderful and supportive wife, Safaa for her tireless support, patience, invaluable companionship along this journey; I could not have done any of this without her. To Brigham Young University, the place where you research at technology's edge, where faith, reason and science have no conflict, and where family thrive. I will always be grateful for the formative years I have spent here.

Table of Contents

List of Tables

List of Figures

Chapter 1. Introduction

Flying alone! Nothing gives such a sense of mastery over time over mechanism, mastery indeed over space, time, and life itself, as this.

-Cecil Day Lewis

In the CBS's "60 Minutes" prime-time interview on December 2013, Amazon CEO Jeff Bezos unveiled an ambitious future delivery service called Amazon Prime Air. The service will allow customers to choose small unmanned rotorcraft to deliver their packages. It would be able to carry objects of up to 5 lb within a 10 mile radius of an Amazon distribution center. *"I know this looks like science fiction, it is not,"* Bezos said. Bezos added that the "Prime Air" project that Amazon is working on in its research and development labs will take years to advance the technology, yet the service also have to comply with the aviation rules for unmanned aircraft, which the aviation regulatory body is planning to have in the near future.

Daily we hear similar stories of people, scholars, companies and manufacturers who are building or operating a wide-range of flying robots and trying to use them in various applications. The increasing demand to integrate unmanned aircraft systems (UAS) into the national airspace system (NAS) is motivated by the rapid growth of the UAS industry, especially small UAS weighing less than 55 pounds. The majority of the efforts have focused on integrating medium or larger UAS into the controlled airspace. However, most of the potential UAS applications, few examples are shown in Figure 1.1, including recreational activities, goods delivery, agriculture, wildfire monitoring, and infrastructure surveillance, are well suited to small UAS. Small UAS are particularly attractive, and their use is likely to grow more quickly in civil and commercial operations because of their versatility and relatively low initial cost and operating expense.

(a) Raven monitors crops. Photograph courtesy of © AeroVironment, Inc.

(b) Prime Air is a conceptual UAS-based delivery system currently under development by Amazon. Photograph courtesy of © Amazon.com, Inc, used with permission.

(c) Puma AE supports fire-monitoring. Photograph courtesy of © AeroVironment, Inc.

Figure 1.1: Examples of a potential small UAS applications.

A report compiled by the US Department of Transportation on UAS service demands estimates that by the year 2035 there will be approximately 70,000 UAS operated by federal, state, and local departments and agencies [5]. In the private sector, the ever growing number of UAS applications includes a wide variety of industries and tasks such as smoke stack inspection, cinematography, crop dusting, oil exploration, and news and traffic reporting. The demand for UAS operations is manifest by the over 1800 exemptions granted in one year to allow UAS operations under Section 333 of the H.R. 658 Federal Aviation Administration (FAA) Modernization and Reform Act of 2012 [4].

While UAS operations have increased as a result of the Section 333 exemptions approved since September of 2014, the overall realized benefit of UAS operations is still a small fraction of the demand. Additionally Section 333 exemptions are not a long-term solution to supporting UAS in the NAS. In laying the foundation for a long-term solution for UAS in the NAS, the FAA has mandated that UAS be capable of an equivalent level of safety (ELOS) to the see-and-avoid mandate for manned aircraft [44, 55]. As a result, similar to

a pilot's ability to visually scan the surrounding airspace for possible intruding aircraft and take action to avoid a collision, a UAS must be capable of monitoring and avoiding other manned or unmanned aircraft in its vicinity. Currently, UAS face limitations on their access into the national airspace because they do not have the ability to sense and avoid other air traffic. This sense-and-avoid (SAA) mandate is the capability of a UAS to remain well clear and avoid collisions with other air traffic [49]. The need for a robust SAA system is evident from the expected congested future airspace and is considerably crucial to assure safe airspace operations and minimize the impact to current airspace users.

On the other hand, while manned aviation is well established, regulations for unmanned aviation are still being developed since UAS technology is still immature and the FAA is adopting a cautious approach. Efforts to regulate and certify UAS have been driven by a rapid growth in technology and also potential UAS markets in military, civilian and public domains [28, 36]. Until acceptable SAA systems are available, the FAA has limited UAS access to the NAS by requiring a certificate of authorization (COA) under Section 333 exemption to fly. This lengthy and cumbersome process is further complicated in that UAS range across a much broader spectrum of sizes and performance capabilities [35]. In an effort to improve the FAA's rules regarding UAS in the NAS, multiple frameworks for the regulation and certification of UAS have been developed [28]. To date the FAA has not adopted a framework, but with the passage of H.R. 658 the FAA is taking the steps to establish clear guidelines to allow UAS access to the NAS in near future [4, 6].

As the design of any SAA system must meet the FAA requirements and regulations, an extensive amount of work has been done to study and understand the current manned aviation regulations as a guide to aid in the development and design of SAA for unmanned aircraft. In the United States the aviation regulations, collectively known as Federal Aviation Regulation (FAR), are codified in the Code of Federal Regulations (CFR), Title 14, Chapter I. The CFR along with supplementary material like advisory circulares, technical standards orders and manuals such as the Aeronautical Information Manual (AIM) issued by the FAA, define appropriate standards, procedures, and practices to ensure that operators and manufactures are able to establish a minimum level of safety and reliability required for civil aviation [12, 36]. Foremost among aviation regulations are parts for airspace categories

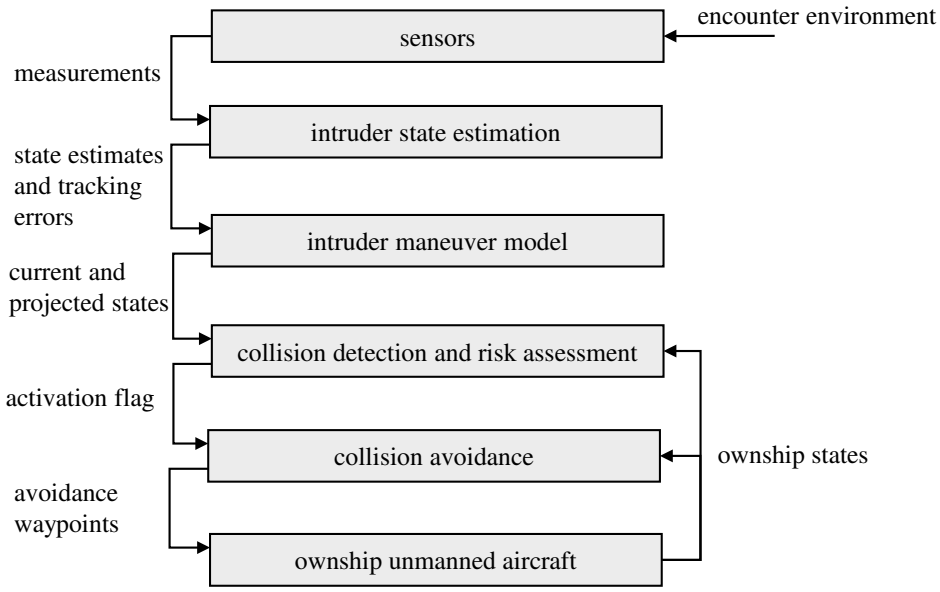


Figure 1.2: The sense-and-avoid system architecture that is assumed throughout this dissertation.

and classes [CFR § 71 − 73] and general operating and flight rules [CFR § 91], the right-of-way rules operations [CFR §91.113], aircraft speed [CFR §91.117], minimum safe altitudes [CFR §91.119] and visual flight rules (VFR) [CFR §91.151 − 161] to name a few, have a crucial impact on the design and development of SAA system. Appendix A provides more specific details about manned and up-to-date unmanned aviation regulations.

Typically, a complete functional sense and avoid system is comprised of sensors and associated trackers, collision detection, and collision avoidance as depicted in Figure 1.2. In this work, however, our main focus is on collision detection and avoidance algorithms. The main role of the first sub-function is to detect any intruders and track their motion. The block labeled *sensors* in Figure 1.2 process the raw signal and passes it to the intruder estimator. In the block labeled *intruders state estimation* the intruder state measurements are processed to generate the intruder position and velocity estimates. The block labeled *intruder maneuver model* rely on the intruder state estimates and a kinematic model of an aircraft to project the motion of the detected intruder forward in time to identify possible conflicts or collisions. Not every aircraft that is observed by the sensing system, however, presents a conflict or collision threat. Therefore the block labeled *collision detection and risk assessment* refers to the collision detection algorithm that determines whether or not

an approaching intruder aircraft is on a collision course. If a collision threat is detected, the intruder position and velocity estimates and an activation flag are passed into the *collision avoidance* block in Figure 1.2. Once the avoidance logic has been activated a new collision-free path is generated. In any encounter scenario that occurs, the UAS's avoidance maneuver must conform to the standard flight rules followed by manned aircraft. If UAS are to be integrated seamlessly alongside piloted aircraft, they must react to collision threats in the same way as a human pilot. Additionally, a collision avoidance system should only maneuver to avoid other aircraft when a collision threat exists. Unnecessary course changes limit flight efficiency and may also confuse other airspace users. It might in addition result in conflict with other air traffic. The output of the *collision avoidance* block is a revised set of waypoints that is free of collision risks. The new set of waypoints is then passed to the *ownship unmanned aircraft* block. The *ownship unmanned aircraft* block in this work refers to a generic system architecture of an unmanned aircraft that is typically composed of a path planner, path manager, path follower, and autopilot. A path planner produces straight-line or Dubins paths between waypoints, and a path manager is required to switch between waypoints. Then, the path follower produces commands to the low-level autopilot, which controls the ownship. For further analysis on *ownship unmanned aircraft* we refer the interested reader to [16].

The primary focus of this work is to develop a collision detection, risk assessment and avoidance framework for unmanned aircraft. The design, however, will be specifically tailored for small UAS operating in class G airspace. The proposed approach presented in this work will consider encounter scenarios such as the one depicted in Figure. 1.3, where the ownship encounters multiple intruders. We assume that there exists a sensing system that provides an estimate of the intruder's track. In such situations, the collision detection algorithm is responsible to determine the likelihood of future collisions with the detected intruders. It provides an alert threshold above which the avoidance is initiated to plan a proper evasive maneuver.

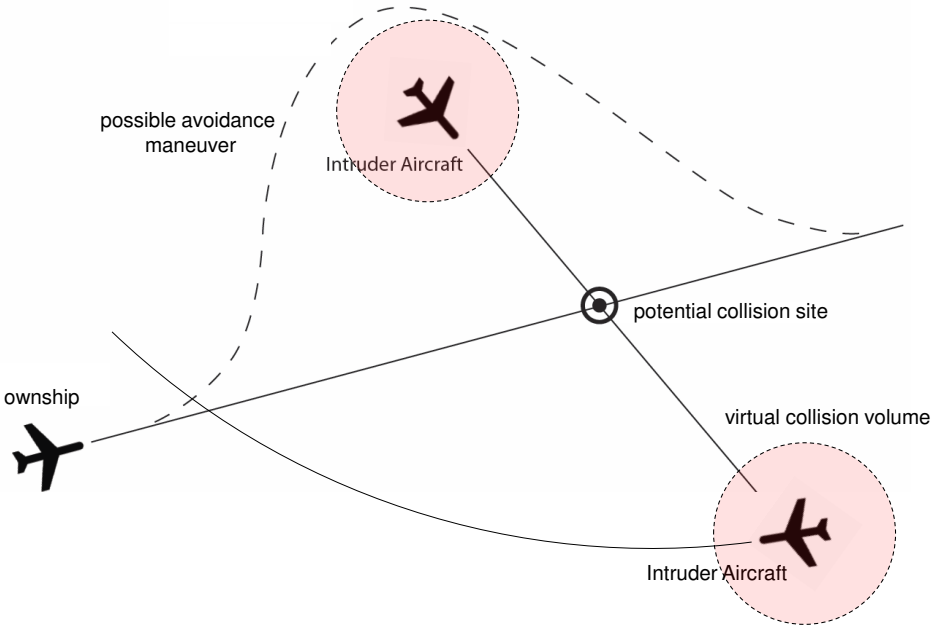


Figure 1.3: Encounter geometry scenario.

1.1 System Level Requirements

The main concern of the air traffic management system for manned aviation is safety, which is typically measured by the number of incidents that happen when the distance between aircraft becomes closer than a predefined safe distance to one another. This safety distance is quantified by means of a minimum allowed horizontal and vertical spacing [105]. The design of an SAA system for UAS should also address regulatory requirements, and performance and reliability standards. Initial efforts to address performance, design, construction, and reliability requirements of an SAA system for UAS are all discussed in the TR F2411-07, standard document produced by ASTM international [3]. An excellent review of existing regulations, standards, recommended practices, along with suggestions and recommendation for SAA requirements to facilitate the UAS integration into the NAS system are discussed in [11,36,106]. Geyer et al. address specific design parameters required by the SAA system such as sensor angular resolution, field of view, minimum time, and sensing range needed to prevent a collision assuming a 2D head-on encounter geometry [51]. The formulas proposed by Geyer et al. are based on a head-on collision scenario between two aircraft and are an approximation that is suitable for use as a heuristic to choose the right sensor and

its minimum detection range. This approximation is not suitable for short distances and close velocities [51]. Boskovic et al. derive radar-based sensor and tracking requirements combining worst-case 2D flight head-on collision scenarios and exhaustive Monte Carlo Simulations [20]. Melnyk et al. propose a framework that consists of a target level of safety (TLS) approach using an event tree format to develop specific SAA effectiveness standards linking UAS characteristics and operating environments to midair collision risk quantified by a fatality rate. [90].

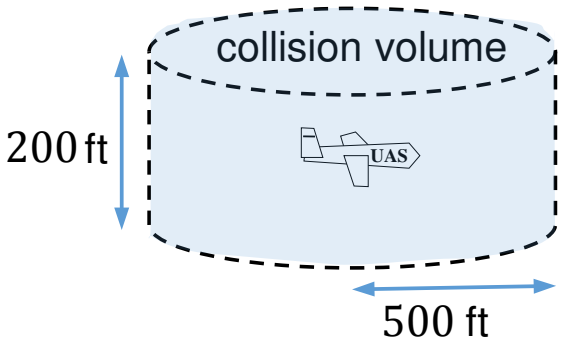


Figure 1.4: Collision volume.

In this dissertation our focus, however, is on determining the minimum required sensing range to help users to choose the right sensor for the UAS or to design the sensor to achieve the required minimum detection range if intended to be used for SAA systems. Another important requirement of the SAA system is the definition of a collision. Generally, a collision occurs when two aircraft or more come within the minimum allowed distance between each other. The current manned aviation regulations has no explicit values for the safe distance, however it is generally understood that the minimum safe distance is required to be at least 500 ft to 0.5 nautical miles (nmi) [3, 11]. For example, the near midair collision (NMAC) defined in the aeronautical information manual as proximity of less than 500 ft between two or more aircraft, or if the pilot decides to report the encounter as an NMAC [126]. Since the potential UAS and intruder aircraft cover a wide range of vehicle sizes, airframes, weights, designs, etc, the choice of a fixed volume is a substitute for the actual dimensions of the intruder. As shown in Figure 1.4, the collision volume or the protection zone is a virtual fixed volume boundary around the aircraft and the general choice

for this volume is a cylinder of radius d_s and height h_s centered at the current location of the UAS. A common requirement includes a horizontal distance of 500 ft and a vertical range of 200 ft [31, 49, 77]. The collision is defined as an incident that occurs when two aircraft pass less than 500 ft horizontally and 100 ft vertically. As a result of this requirement, the detection and collision avoidance must be done at a range that is sufficient to allow the SAA system to initiate a track of the detected intruder, detect a collision, plan an avoidance path, and to actually execute a maneuver with sufficient time that results in the minimum required safe distance to the intruder. According to [1], the estimate of the time required for a manned aircraft to consistently avoid mid air collisions, ranges from 5 s to 12.5 s. This time duration does not include the actual time required to perform a maneuver initiated by the collision avoidance system. The estimate is for two jet aircraft with a closing speed of about 492 m/s [1].

1.2 Literature Review

We divide our survey of the literature into three main sections. In the first section we review the sensing technologies used for the SAA systems. In the second and third sections, we review the collision avoidance and detection approaches.

1.2.1 Sensing Technology

For small UAS weighing less than 55 lb, the algorithms and hardware necessary for SAA can make up a notable portion of the available size, weight, and power (SWaP) resources. As discussed earlier in this chapter, one means of compliance for UAS to gain routine access into airspace is the ability see and avoid. There are two main approaches to meet with this requirement: airborne or ground-based sensors and algorithms. Sensors carried on board the UAS sensors provide a dynamic surveillance capability for the UAS through its mission, and will enable a fully autonomous sense and avoid system. The Airborne is the only solution to provide wide and dynamic surveillance sense-and-avoid capability, which is not achievable by ground-based SAA. The Ground-based systems provide support for unmanned aircraft within limited geographic and airspace regions. Currently, radar, electrooptical/infrared

cameras, and light detection and ranging (Lidar) are examples of sensors used to detect noncooperative traffic. Noncooperative traffic means that no information about conflicting traffic is communicated or transmitted to the aircraft from the conflicting intruders or from air traffic control. Alternatively, the automatic dependent surveillance-broadcast (ADS-B), traffic collision-avoidance system (TCAS), and airborne collision avoidance system (ACAS X), are examples of systems for detecting cooperative intruders.

Radar is one sensor that is widely used for air-to-air detection in manned aircraft. Among many possible sensor modalities that can be used for UAS sense-and-avoid system, radar makes a reasonable choice [20, 23, 118], in particular to detect non-cooperative intruders. One of the primary strengths of radar is the ability to detect all objects regardless of cooperative sensor equipage or functionality. In applying radar to small UAS, SWaP constraints impose restrictions on the hardware that result in significant trade offs between radar range, bearing accuracy, and field of view. Recent progress in radar technology and advancement in integrated circuit fabrication makes small, lightweight, low power radar sensors feasible for small UAS sense and avoid systems. Long range radar sensors require higher transmit power and consequently become a greater drain on the UAS power budget, or higher antenna directivity, which means a larger antenna size and narrower angular field of view. At a set transmit power, improving the range requires a narrower beam, which also improves the bearing accuracy. Narrowing the beam, however, reduces the field of view and consequently requires additional antennas or a method to steer the beam. A limitation of radar is that the signal strength of the return is dependent on the material properties of the intruder aircraft, the frequency used, and the angle at which the aircraft is observed [33]. This is represented in a radar cross section (RCS) profile that is specific to each aircraft and varies as a function of angle. Demonstrated hardware that falls within the SWaP limitations of small UAS is not currently suited to support a feasible set of range, bearing accuracy, and field of view requirements [84].

Optical sensors such as cameras are also candidate sensors for SAA on small UAS. Similar to radar, vision-based intruder detection methods do not require cooperative communication from intruders. Flight testing of visual methods has achieved intruder detection at 0.54 nmi from a small UAS [71]. Ground-based testing has resulted in detection up 4.3

nmi [37]. The flight tested range of 0.54 nmi is promising, but not sufficient to provide enough avoidance time for high-speed intruders. Even with sufficient range, visual methods inherently have low range accuracy. Adverse weather conditions such as fog, clouds, precipitation, and sun glare can reduce overall visibility and significantly limit visual intruder detection. While recent developments have improved visual intruder detection, such methods are not yet suitable for SAA implementation on small UAS.

Infrared (IR) cameras have the added advantage that they can be used at night [33]. However, both optical and IR sensors are subject to signal degradation due to weather, and both have difficulty determining range and velocity in high speed collision detection scenarios. Lidar is an extremely powerful imaging modality and is used extensively for mapping because it gives very precise range and angle measurements. Unfortunately, robust Lidar systems are generally too large to be integrated on small UAS and also suffer some of the same weather degradation as optical sensors [33].

Automatic Dependent Surveillance-Broadcast (ADS-B) is a cooperative sensor that is a promising option for SAA on small UAS. It has been demonstrated in small UAS flight testing to have an omni-directional range of 20 nmi [95], and due to the fact that the cooperative information is shared over radio waves it is relatively unaffected by adverse weather conditions. An omni-directional antenna and low-power requirements for both transmit and receive hardware contribute to the promising characteristics of ADS-B. Two drawbacks of ADS-B are its dependence on global positioning system (GPS) information and its fundamentally cooperative nature. While GPS coverage of the national airspace is very good, there are areas where GPS information can become degraded such as narrow valleys or urban canyons. Furthermore, the cooperative aspect of ADS-B requires widespread adoption of ADS-B technology to ensure detect-and-avoid reliability. While the Federal Aviation Administration does not yet require all aircraft to be equipped with ADS-B transponders, the 2020 mandate requiring all aircraft in A, B, C, and some E class airspace to equip with ADS-B [43] is a significant step. ADS-B is rapidly becoming a major tool in the air traffic management system. In 2010 the FAA issued a final rule for the implementation of ADS-B on manned aircraft [43]. This ruling mandated ADS-B Out in key parts of the NAS. The FAA Modernization and Reform Act of 2012 further directed the FAA to make

plans for the adoption of ADS-B In technology [4]. Valuable research has been done in developing and testing ADS-B technology. The MITRE Corporation provided initial development and testing of research-grade ADS-B hardware and demonstrated its ability to transmit up to 20 nmi [95]. The University of North Dakota (UND) extended this research by demonstrating an ADS-B-based collision avoidance system for a small-to-medium sized UAS with a large, commercial-grade ADS-B transceiver [86, 87]. R3 Engineering developed a commercially-available ADS-B-based collision avoidance system and tested it on a Tiger Shark UAS [32]. These research contributions paved the way for the application of small, commercially-available ADS-B on small UAS. As a result of the level of adoption, prior foundational research, and capability of ADS-B technology, ADS-B is an attractive sensor for detect and avoid efforts on UAS.

TCAS and ACAS X are aircraft collision avoidance systems that are based on a secondary surveillance radar transponders to provide advice to the pilot on proximity traffic and potential conflicts. Unlike primary radar sensors that measure range and azimuth angle of objects by passively processing the signal reflected from the aircraft body, the secondary surveillance radar interrogates aircraft equipped with transponders, that replies to each interrogation by transmitting a response containing encoded data including identification and altitude. Applying TCAS and ACAS X to small UAS is discussed in section 1.2.3.

1.2.2 Collision Detection and Risk Assessment Methods

Airborne collision detection is a difficult problem due to inherent noise, errors in prediction and modeling the dynamics of intruder aircraft. Moreover, on-board limited computational resources, fast closing speeds and unanticipated maneuvers make it very challenging to detect collision without creating too many false alarms. A number of conflict and collision detection methods have been suggested in the context of air traffic management, mobile robotics, and autonomous control. Recent surveys on this topic include Kuchar & Yang [68], Albaker & Rahim [10], and Angelov [11]. As shown in Figure 1.5, these different methods can be classified under four fundamental approaches: deterministic or straight line, worst case, probabilistic, and flight plan sharing. Many of these methods stress the deterministic approach, where a single trajectory of the intruder is predicted using straight

line extrapolation. This is a reasonable approach when there is a perfect knowledge of the states of the detected intruder. However in practice, the uncertainty free model could lead to erroneous prediction of collision threat particulary when solving the problem over a short time horizon. The term *"conflict"* is normally used in the context of manned aviation air traffic management and alerting systems. Conflict detection usually implies a multi-layer approach to detect loss of well clear, or minimum separation, defined as an event where two or more aircraft come within 3-5 nautical miles (nmi) over time horizons on the order of half a minute or more [56, 102]. On the other hand, *"collision"* is used when detection is required for close proximity encounters over time horizons of seconds [11]. While many of these techniques may be applicable to either conflict or collision detection, an appropriate scaling in design parameters, assumptions, and thresholds is required.

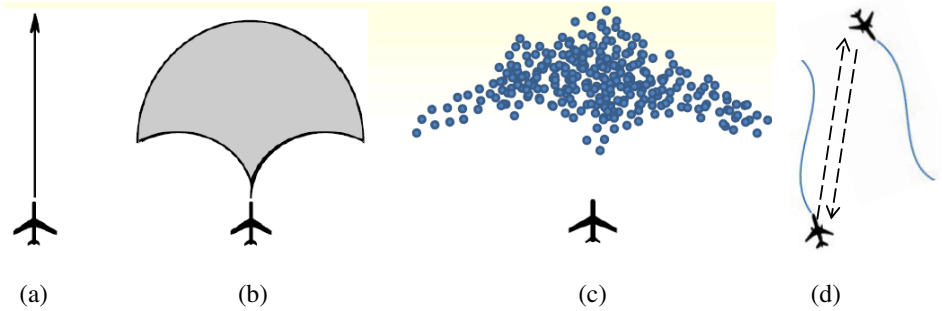


Figure 1.5: Most approaches to collision prediction stress the straight-line (nominal) approach (a), others use worst case (b), probabilistic (c) or flight plan sharing if aircraft are equipped to communicate with each other (d).

Less research has been dedicated to collision detection for UAS, mainly because this problem is relatively new in comparison with manned aviation air traffic management. Regardless, there is a growing body of research in sense and avoid for UAS, motivated by the need to integrate UAS into the national airspace, and in particular to establish separation standards, risk assessment, and collision detection metrics [7, 24, 51, 123, 129]. The work by Temizer et al. proposed a comprehensive collision alerting and avoidance scheme for different sensor modalities, including radar. The problem is formulated as a Markov decision process (MDP) and a partially observable Markov decision process (POMDP) to account for sensor uncertainties. A generic MDP/POMDP solver can be used to generate avoidance

strategies that optimize a cost function that includes minimizing probability of an air collision event [123]. Our work is similar in providing analysis of multiple components of an aircraft alerting system, however we focus on the collision detection functionality. The current paper contributes to this body of work by introducing a feasible framework for sensing, tracking, and collision detection that can be integrated on-board small UAS. A continuous focus of this dissertation is to develop an innovative approach to quantify likely intruder future trajectories and evaluate the collision risk for a pair of aircraft flying at the same altitude in close proximity. The proposed approach builds upon the reachability set concept and the statistical data contained in the uncorrelated encounter model (UEM) developed by MIT Lincoln Laboratory [61]. The main role of encounter models is to generate statistically representative traffic encounters to explore and evaluate the robustness and performance of a collision avoidance system. However, other ways to utilize these data sets have been suggested in the literature. For example, Jackson et al. used sampling methods to generate trajectories up to a future time of interest from the MIT LL UEM data set to increase the fidelity of the reachability set boundary of the intruder [58]. Reachable sets of all possible positions of a moving object have been extensively studied, due to their importance in mobile robotics [29, 40, 46, 93].

The purpose of computing the collision risk is to have an alert threshold value above which the collision avoidance system is triggered to initiate an evasive maneuver to avoid an imminent collision with the detected intruding aircraft. There are a number of approaches to evaluate the future collision risk of an encounter situation. Most of these approaches can either be classified as geometric or probabilistic, where each approach has different techniques to deal with errors. In the geometric approach, the collision risk is described based on the geometric relationship between aircraft. Aircraft trajectory predictions are based on linear projections of current aircraft states such that the uncertainty of the predicted trajectory is translated into areas around the predicted trajectory referred as to safe zones. Linear projections can be computationally efficient and prediction errors are negligible over short time horizons [25, 50] or assumed known when flight plans are communicated [98]. On the other hand, probabilistic methods estimate the probability of collision based on a probabilistic model of future intruder dynamics. This event probability is then compared to a

certain threshold above which the aircraft is deemed to be in collision. These probabilities can be estimated using approximate analytical solution [102], numerical approximation [73, 127], or Monte Carlo methods [58, 78, 81]. The expected utility is another approach used to develop a risk alerting system that accounts for future changes in alerts and the responses to them [26]. In general, the probabilistic approaches are computationally intensive but suffer less from false alarms than geometric approaches.

1.2.3 Collision Avoidance Methods

Once a collision threat has been detected, the collision avoidance system must select the proper evasive maneuver observing the manned aviation right-of-way rules. The essential requirement for the collision avoidance system is to perform the avoidance maneuver in such a way that the distance at the closest point of approach (CPA) to the intruder is equal or greater than a minimum required miss distance. Additionally, the initiated maneuver must be compatible with the performance of the ownship and once the conflict is resolved the collision avoidance/path planner system should return the ownship to the original flight path or to a newly assigned flight plan.

The subject of path planning is very broad and has been the focus of a significant body of research especially in the field of autonomous planning and robotics. Essentially there are two different approaches to path planning: deliberative motion planning, where explicit paths are computed based on a priori knowledge of the environment, and reactive path planning, which uses behavioral approaches to react to sensor information [16]. Reactive path planing is well suited for dynamic environments, particulary airborne collision avoidance, where sensor information is incomplete and uncertain. Additionally, the collision event occurs over a relatively short time horizon which requires a planing method that promptly plan an avoidance maneuver using limited computation power.

Collision avoidance is a fundamental part of path planning that involves the computation of an optimal collision-free path from a start point to a goal point while optimizing an objective function or performance metrics. A robust collision avoidance logic considers the kinematics constraints of the host vehicle, the dynamics of the intruders motion, and the uncertainty in the intruder's states estimate. Similar to collision detection, airborne

collision avoidance has gained considerable attention, and various methods and approaches have been suggested in the literature. Kuchar and Yang [68] conducted a detailed survey of conflict detection and resolution methods. Albaker and Rahim [9] provide a thorough survey of collision avoidance approaches for UAS. In addition, Angelov et al. presents a literature survey of collision avoidance approaches for sense and avoid for UAS applications [11]. The most common collision avoidance methods are geometric-based guidance methods [17, 21, 45, 48, 57, 110, 121, 130], potential field methods [72, 120], sampling-based methods [59, 74] and cell decomposition techniques.

Geometric approacher to collision avoidance are straightforward and intuitive. They lend themselves to fast analytical solutions based on the kinematics of the aircraft and the geometry of the encounter scenario. The approach utilizes the geometric relationship between the ownship and the intruder along with intuitive reasoning [11,130]. Generally, this approach assumes a straight-line projection to determine whether the ownship will penetrate a circular protected zone of an intruder such that the collision avoidance can be achieved by changing the velocity vector assuming a constant speed model. An optimal geometric based avoidance algorithm is proposed in [17]. It is optimal in the sense that it minimizes the velocity vector changes resulting in minimal deviation from nominal trajectory. Typically, geometric approaches do not account for uncertainty in intruder flight plans and noisy sensor information.

The potential fields method is another popular approach for collision avoidance in robotics. A typical potential field works by exerting virtual forces on the aircraft, usually an attractive force from the goal and repelling forces from nearby air traffic. Argyle et al. present a path planner based on a simulated chain of unit masses placed in a force field. This planner tries to find paths that go through maxima of an underlying bounded differentiable reward function [13]. Bortoff presents a method for modeling a UAS path using a series of point masses connected by springs and dampers. This algorithm generates a stealthy path through a set of enemy radar sites of known locations [19]. McLain and Beard present a trajectory planning strategy suitable for coordinated timing for multiple UAS. The paths to the target are modeled using a physical analogy of a chain [88]. Generally, the approach is very simple to describe and easy to implement. However, this approach has some

fundamental issues [64]. One of these issues is that it is a greedy strategy that is subject to local minima. However, heuristic developments to escape the local minima traps is also proposed in the literature [75]. Another problem is that typical potential fields approaches do not account for obstacle dynamics variability or uncertainly in observation or control.

Sampling-based methods like Probability Road Maps (PRM) [59] and Rapidly-exploring Random Trees (RRTs) [74] have shown considerable success for obstacle avoidance and path planning, especially for ground robots. They often require significant computation time for replanning paths making them unsuitable for reactive avoidance. Recent extensions to the basic RRT algorithm, however show promising results for uncertain environments and nontrivial dynamics [65, 82, 83]. Cell decomposition is another widely used path planning approach that partitions the free area of the configuration space into cells, which are then connected to generate a graph [92]. Generally, cell decomposition techniques are considered to be global path planners that require a priori knowledge of the environment. A feasible path is found from the start node to the goal node by searching the connectivity graph using search algorithms, like A^* or Dijkstra's algorithm [38].

Traffic Alert and Collision Avoidance System (TCAS) is the primary certified and mandated cooperative collision avoidance system and is used by many manned aircraft with operating transponders. TCAS system was designed for pilot-in-the-loop control and assumes a delay to allow the pilot to respond. In addition to traffic alerts, TCAS issues vertical rate resolution advisories to pilots who are then responsible for maneuvering the aircraft. TCAS is currently limited to large aircraft capable of supporting its hardware, power requirements, and maneuverability demands required by the avoidance advisory logic [62]. In applying TCAS to UAS, the Multi-Sensor Integrated Conflict Avoidance (MuSICA) system demonstrated promising results by using a comprehensive sensor suite that utilized TCAS for UAS SAA [23]. The proposed MuSICA system was tested using Simulink desktop simulations and hardware-in-the-loop simulation. Additionally, a limited number of flight tests were conducted using a Learjet as a UAS surrogate. These efforts, however, were focused on the sensor data integration portion of the SAA system and concluded that more testing is required for further confirmation. Other studies uncovered issues and areas of concern when applying TCAS to UAS including inherited delay in TCAS logic, limited vertical maneu-

verability, and coordination with existing TCAS-equipped manned aircraft [136]. Although TCAS remains a candidate solution for larger-size UAS like the RQ-4A Global Hawk, it is not presently suitable for small UAS because of the complexity of SAA and the limited SWaP capabilities of small UAS [70, 123].

Another important advancement in SAA is the development of Airborne Collision Avoidance System X (ACAS X). The ACAS X system is currently undergoing testing and evaluation is designed to replace TCAS and support the future air traffic system. It addresses many of the design limitations of TCAS and leverage additional surveillance sources. ACAS Xu is envisioned for UAS implementation and proof-of-concept trials have already taken place. It is designed to use ADS-B messages for passive surveillance and coordination with TCAS-equipped aircraft. ACAS Xu is designed to accept additional surveillance sources such as radar or electro-optical (EO) sensors to track noncooperative traffic. Current research efforts focus on large aircraft and general aviation, while UAS applications remain longer term research. Similar to TCAS, the ACAS X system is aimed at close-range conflicts between 15 and 48 seconds with vertical advisory resolution. It is obviously desirable to have long-range conflict detection and resolution so that close-range conflicts can be prevented. Moreover, climbing or descending may not be the best collision avoidance approach for small UAS operating at lower altitudes [15].

1.3 Summary of Contributions

In this dissertation we make several contributions to the sense-and-avoid problem as summarized below:

- We derive a close form analytical solution to compute the minimum sensing range required for the SAA system. The existing approaches require extensive Monte Carlo Simulations. To the best of the author's knowledge, there is only one other approach that derives a formula that is suitable for use as a heuristic to determine the minimum sensing range [51]. Our proposed approach provides convenient equations that can be used by both sense-and-avoid logic or sensors processing algorithm. We demonstrate

the feasibility of this approach by describing a radar sensor prototype that achieves the required minimum sensing range.

- We develop two collision detection and risk assessment approaches based on deterministic and probabilistic framework. In the deterministic-based approach, we derive several evaluation metrics including time to closest point of approach and distance at closest point of approach and horizontal and vertical entry and exit time into the collision volume. We used these metric to design a collision detection algorithm to evaluate an encounter scenario and decide whether or not there exists a future collision. To account for uncertainties in state estimates of the intruder we derive an analytic expressions to propagate the error variance using a first-order Taylor series approximation. A contribution of this dissertation is the derivation of the most used and cited formula for the error propagation, $\sigma_f^2 \approx \sum(\frac{\partial f(x)}{\partial x})^2 \sigma_x^2$. We did not find the full derivation of this formula in the literature. Therefore, to better understand the assumptions and limitations of this equation we derive it. The derivation is presented in Appendix B. To account for maneuvers other than a straight-line flight path, we develop a probabilistic-based framework using the reachable set concept and the statistical data contained in the uncorrelated encounter model developed by MIT Lincoln Laboratory. The proposed collision detection approach has the advantage that the locus of states within the reachable sets with respect to the detected intruder are only computed once, which makes it attractive and feasible for real-time SAA on-board small UAS. The computational burden is expected to be much less than existing algorithms that require running Monte Carlo approximations in near real time or performing an extensive sampling from the probability distribution of intruder estimated states or probabilistic encounter models.

- We develop two reactive collision avoidance algorithms suitable for airborne-based sense-and-avoid systems. The first techniques is chain-based collision avoidance, which is an application of the potential fields methods. The basic idea is to model the future motion of the aircraft using a chain of waypoints, where the waypoints can dynamically move as if they were particles in a force field. We further explore the chain-based approach using the analogy of a spring-damper system. The algorithm

is further extended to account for multiple intruders in various encounter scenarios. It is also extended to account for uncertainties in the state estimates of the detected intruders. The approach is extended to 3D path planning, where the chain consists of unit masses connected by springs and dampers evolves in a simulated force field. The chain is initialized first with a suboptimal path generated using A* search. One main advantage of this approach is that it can trade-off a collision free path versus path length. We also develop a collision avoidance logic using a translated coordinate system. The technique builds graph in the local-level frame and use Dijkstra's algorithm to find the path with the least cost. An advantage of this approach is that collision avoidance is inherently a local phenomenon and can be more naturally represented in local coordinates than global coordinates. This technique eliminates the need to translate the sensor's measurements from local coordinates to global coordinates, which saves the computation cost and removes the error introduced by the transformation.

- To further evaluate the performance and explore the limitations of the proposed collision detection and avoidance approaches, we apply the proposed approaches to several senors modalities using different tracking and estimation schemes. First, we evaluate the close-loop system of the chain-based collision avoidance system using camera sensor model and passive ranging tracker and collision detection system developed by Utopia-Compression. The performance is examined using Monte Carlo simulations as detailed in Chapter 5. We demonstrate a complete, proof-of-concept sense-and-avoid solution for small unmanned aircraft including ADS-B-based multi-target tracking and estimation, deterministic-based collision detection approach and the local-level avoidance strategy assuming that both the intruder and ownship are equipped with ADS-B. Taking advantage of the long-range detection offered by the ADS-B system, the proposed approach is extended used to detect conflict events over the range of 5 to 10 nmi, and consequently provide self-separation of the host ownship. Additionally, as part of the C-UAS sense-and-avoid project we collaborate with Communications and Sensors Lab BYU. We have added a realistic radar model along with an estimator whose process and measurement equations are formulated in the so-called modified polar coordinates

(MPC) that appropriately handles polar measurements. We analyze the radar sensor, the tracker and collision detection by comparing the proposed collision detection approach with a linear extrapolation method using simulated encounters generated from the MIT LL UEM. The performance is quantified in terms of the probability of detection and the probability of false alarm using Monte Carlo based simulations.

- To address ground-based SAA system, we develop a two-step path planner approach based on the A* search and simulated chain of waypoints that is connected by springs and dampers. A unique strength of this approach is the ability to trade-off a collision-free path versus path length. We evaluate the performance of the proposed approach in typical encounter scenarios that include multiple intruders flying at different altitudes using simulated radar data and R-RANSAC tracking algorithm. We demonstrate the capability of the proposed collision detection and avoidance approach with flight tests using X8 multicopter in near real time.

The content of this dissertation is heavily based on our publications. Chapter 2 is based on [119]. Parts of Chapter 3 have appeared in [116]. Similarly, parts of Chapter 4 and 5 have appeared in [120]. Chapters 6, 7 and 8 is based on [118], [117], and [115].

1.4 Manuscript Organization

The outline of the remainder of this dissertation is as follows. In Chapter 2, we discuss the minimum sensing range for the SAA system. The developed collision detection and avoidance techniques are presented in Chapters 3 and 4. Applications of the collision detection and avoidance approaches using vision, radar and ADS-B sensors are discussed in Chapters 6, 7 and 8, respectively. Finally, in Chapter 9 we summarize the conclusions of this dissertation and discuss the possible future directions of this research

Chapter 2. Minimum Required Sensing Range

A small correction early is better than a large correction late.

-Aviation proverb

In this chapter we present an approximate close form analytical solution to compute the minimum sensing range required for the SAA system to safely avoid an imminent collision. The framework is based on worst-case collision encounter geometries for aircraft flying roughly at the same altitude. We demonstrate that this is a feasible result by describing a radar sensor prototype that achieves the required minimum sensing range.

2.1 Geometric-Based Approach

The minimum required sensing range arises from the time required for SAA operations. The minimum time for the SAA system to be able to track the intruder, detect a collision, plan an avoidance maneuver, and actually fly the maneuver determines the distance at which the UAS must detect the intruder. In other words, the detection of a collision threat must be accomplished at a minimum range allowing the ownship to execute the maneuver with sufficient time that results in the minimum required safe distance from the intruder. A time sequence for the SAA system, similar to [51], is shown in Figure 2.1.

To compute the sensing range, we first assume an approaching head-on collision encounter scenario similar to the 2D encounter geometry depicted in Figure 2.2(a). In this encounter scenario the intruder is approaching the ownship in a perfect head-on collision. We also assume that the intruder maintains its direction and speed the entire time of the encounter and that it does not maneuver. Such a situation may exist when encountering a non-cooperative intruder. As shown in Figure 2.2(a), the required sensing distance can be

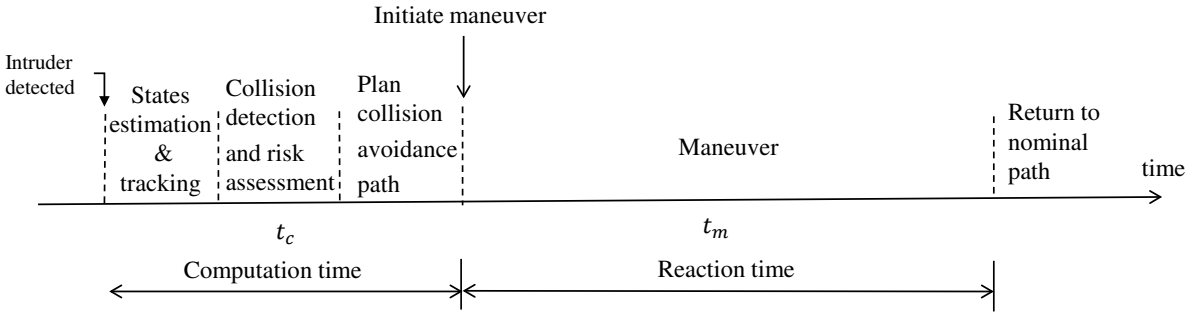


Figure 2.1: Proposed time sequence of the sense and avoid system.

expressed as

$$d_{\mathrm{r}} = v_c t_c + d_m, \tag{2.1}$$

where $v_c = v_o + v_i$ is the closing speed, t_c is the computation time required by the SAA system algorithms to track the intruder, detect a collision, plan an avoidance maneuver, and v_o and v_i are the speed of the ownship and the intruder, respectively. From Figure 2.2(a), d_m can be expressed as $d_m = d_O + d_I$, where $d_O = \sqrt{d_s^2 + 2d_s R_{min}}$, and d_I is the distance traveled by the intruder from the time instant the ownship initiated the maneuver until the time to CPA. d_I can be expressed as $d_I = \frac{v_i}{v_o}\theta R_{min}$, where we have used the fact that the length of the avoidance path traversed by the ownship must equal the distance traveled by the intruder to reach the CPA. In other words, $t_{cpa} = \frac{d_I}{v_i} = \frac{L}{v_o}$, where t_{cpa} is the time to closest point of approach, L is the length of the ownship's avoidance path, and R_{min} is the minimum turning radius of the ownship. Solving for the d_m gives

$$d_m = \sqrt{d_s^2 + 2d_s R_{min}} + \frac{v_i}{v_o}\theta R_{min}. \tag{2.2}$$

Then, substituting Eq. (2.2) in Eq. (2.1) gives the minimum required sensing range

$$d_{\mathrm{r}}^{(h)} = v_c t_c + \sqrt{d_s^2 + 2d_s R_{min}} + \frac{v_i}{v_o}\theta R_{min}, \tag{2.3}$$

where the superscript (h) indicates the assumption of the head-on collision encounter scenario. Using θ and R_{min} we can express $d_{\mathrm{r}}^{(h)}$ in terms of known parameters d_s, t_c, v_o, v_i, and

maximum banking angle of the ownship. We use the relationship [16]

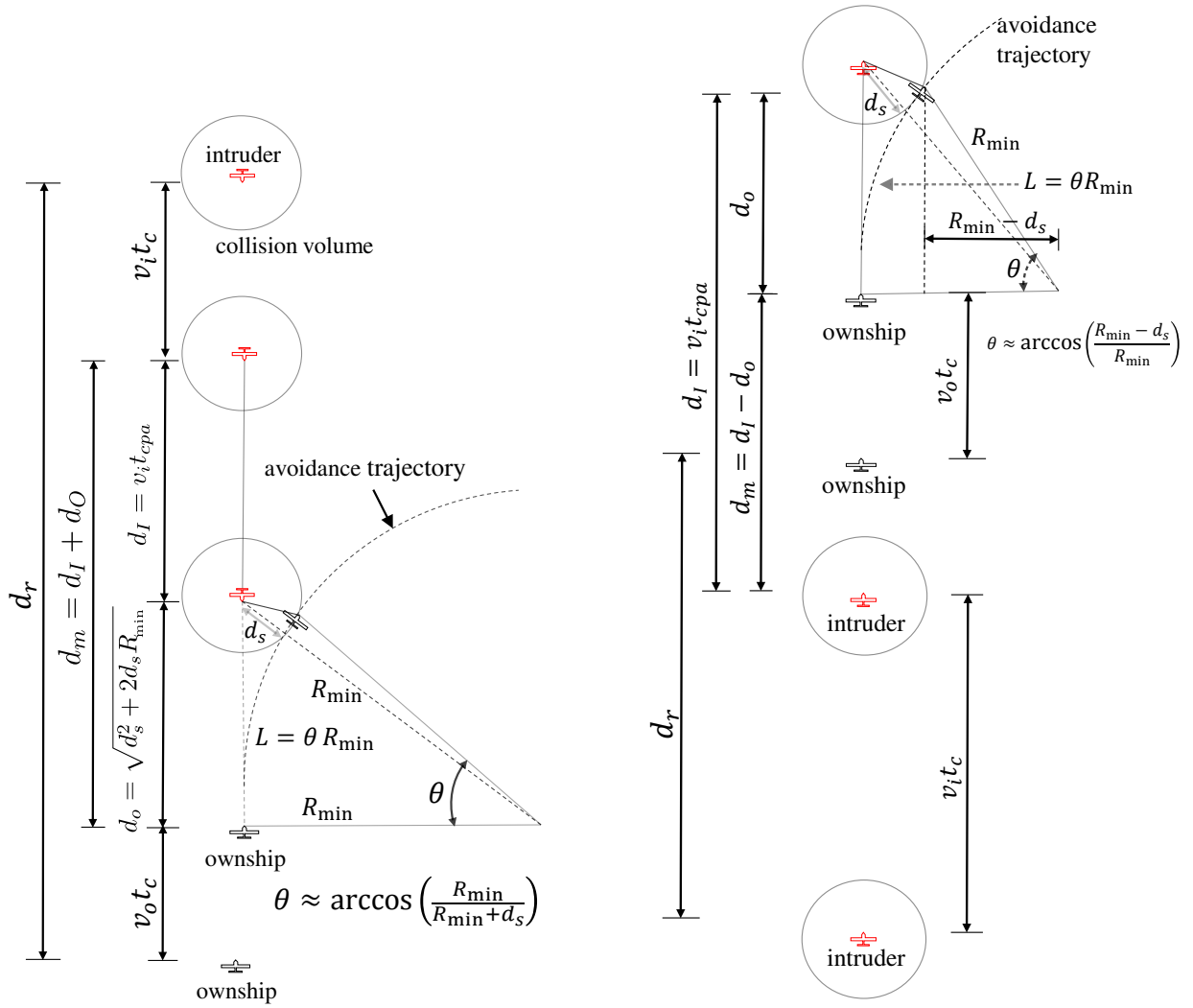
$$R_{min} = \frac{v_o}{\dot{\psi}_{max}} = \frac{v_o^2}{g \tan \phi_{max}}, \tag{2.4}$$

where $\dot{\psi}_{max}$ is the heading rate, ϕ_{max} is the maximum banking angle, and g is the gravitational constant. Then, the required sensing range becomes

$$d_{\mathrm{r}}^{(h)} \approx v_c t_c + \sqrt{d_s^2 + 2 d_s \frac{v_o^2}{g \tan(\phi_{max})}} + \frac{v_i v_o}{g \tan(\phi_{max})} \arccos\left(\frac{v_o^2}{v_o^2 d_s + g \tan(\phi_{max})}\right). \tag{2.5}$$

The approximation in Eq. (2.5) is due to the fact that the value of θ is under approximated by the relationship $\theta \approx \frac{R_{min}}{R_{min}+d_s}$. Eq. (2.5) shows that the closing airspeed of the encountering aircraft, and the maximum bank angle of the ownship affect the required sensing range. In addition, the allowed minimum safe distance that the ownship is required to maintain to the intruder, and the computation time also have an equal contribution in the required sensing range equation.

Another important collision scenario an aircraft may encounter in the airspace is an overtaking collision scenario. The overtaking geometry is shown in Figure 2.2(b). Similar to the head-on collision scenario discussed earlier in this section, we assume that the ownship and the intruder are on a perfect collision course, and the intruder does not alter course but maintains its direction and speed during the time of the encounter. Although in manned aviation the overtaken aircraft has the right-of-way as stated in the Code of Federal Regulations (CFR) title 14, (14CFR,§91.113(f)), in the following analysis we will assume a worst-case scenario in which the overtaking aircraft does not alter course possibly because the pilot does not see the UAS. Such situations may exist when operating under visual flight rules where a transponder or a states-reporting device is not required. For instance, a general aviation aircraft flying in G-class airspace at a speed of 100 m/s encounters a slower speed small UAS such as a Raven that has a wingspan of 4.5 ft and flies at a maximum airspeed of 22 m/s. This analysis also implies that the UAS has the sensing capability to detect an overtaking aircraft. Under this assumption, and in similar situations, it is more convenient,



(a) The geometry associated with approaching head-on scenario to estimate the minimum sensing range.

(b) The geometry associated with overtaking scenario to estimate the minimum sensing range.

Figure 2.2: Encounter geometry to estimate the minimum sensing range.

and possibly safer for the smaller UAS to alter course and give the right-of-way to a larger and faster aircraft.

As shown in Figure 2.2(b), the ownship is required to execute a maneuver that results in a minimum safe distance to the intruder by the time both aircraft reach the closest point of approach. In other words, the length of the avoidance path $L = v_o t_{cpa}$ should equal the

distance traversed by the intruder $d_I = v_i t_{cpa}$. Then, solving for d_I gives

$$d_I = \frac{v_i}{v_o} L, \tag{2.6}$$

where $L = \theta R_{min}$. From the geometry shown in Figure 2.2(b), d_O can be expressed as $d_O = \sqrt{R_{min}^2 + (R_{min} - d_s)^2}$, then solving for d_m gives

$$d_m = d_I - d_O = \frac{v_i}{v_o}\theta R_{min} - \sqrt{2 d_s R_{min} - d_s^2}. \tag{2.7}$$

Note that, Eq. (2.7) is only valid for $R_{min} \geq \frac{d_s}{2}$. However, if $R_{min} = d_s$ then $d_O = d_s$, and $\theta = \arccos(\frac{R_{min} - d_s}{R_{min}})$ equals to 90 deg as shown in Figure 2.3. And, if $R_{min} = \frac{d_s}{2}$ then θ equals 180 deg as shown in Figure 2.3. That means the ownship is required to execute a circular arc path that defines a semi-circle of radius R_{min} to result in the minimum safe distance required by the ownship to maintain the intruder at the closest point of approach. Additionally, when $R_{min} = \frac{d_s}{2}$, d_O equals zero which is an unreasonable solution. Therefore, in this case d_O must be defined to equal d_s. However, in practice when $R_{min} = \frac{d_s}{2}$ the ownship should execute a circular arc followed by a straight line path as shown in Figure 2.3. In addition, $R_{min} < \frac{d_s}{2}$ suggests that the ownship is too slow to avoid an overtaking scenario. Possible solutions to that are either to increase the ownship speed or to reduce the minimum required safe distance.

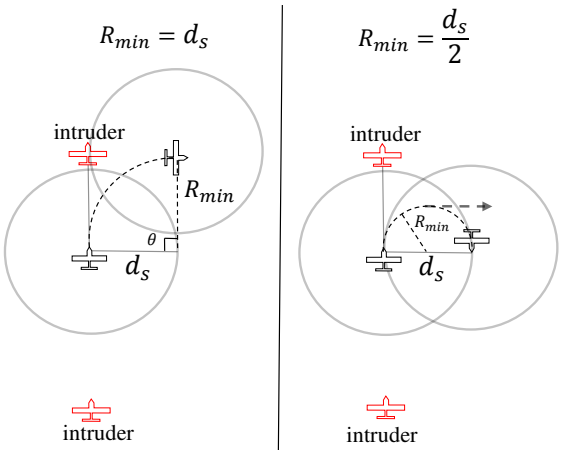


Figure 2.3: Overtaking scenario, $R_{min} = d_s$, and $R_{min} = \frac{d_s}{2}$.

From the geometry depicted in Figure 2.2(b), the sensing range is then given by

$$\begin{aligned} d_{\mathrm{r}}^{(o)} &= v_i t_c + d_m - v_o t_c, \\ &= \bar{v}_c t_c + \frac{v_i}{v_o}\theta R_{min} - \sqrt{2 d_s R_{\min} - d_s^2}, \end{aligned} \tag{2.8}$$

where the superscript (o) indicates an overtaking collision scenario, and $\bar{v}_c$ is defined as $\bar{v}_c = v_i - v_o$. Using the minimum turning radius relationship in Eq. (2.4) and $\theta \approx \frac{R_{min} - d_s}{R_{min}}$, the minimum required sensing range for an overtaking scenario becomes

$$d_{\mathrm{r}}^{(o)} \approx v_c t_c + \frac{v_i}{v_o} g \tan(\phi_{max}) \arccos\left(\frac{v_o^2}{v_o^2 d_s + g \tan(\phi_{max})}\right) - \sqrt{2 d_s \left(\frac{v_o^2}{g \tan(\phi_{max})}\right) - d_s^2}. \tag{2.9}$$

Since $d_r^{(h)} > d_r^{(o)}$ because the closing speed in the head-on scenario is always larger than the closing speed in the overtaking scenario i.e. $v_c > \bar{v}_c$, then we have that $d_{r_{min}} = d_r^{(h)}$ is the minimum required sensing range given that the ownship avoids a collision by initiating a turn maneuver. We still, however, need to know the minimum required sensing range for the overtaking scenario in the case when the ownship uses a separate rear looking sensor. Since Eqs 2.5 and 2.9 under approximate the minimum required sensing range, compensation can be made by selecting a non zero design parameter $\delta_r > 0$ such that the minimum required sensing range becomes

$$\bar{d}_{r_{min}} = (1 + \delta_r) d_{r_{min}}. \tag{2.10}$$

2.2 Results

In the following sections we first provide a numerical analysis of the proposed approach to design the minimum required sensing range using various UAS and manned aircraft models, and second we demonstrate the feasibility of using radar sensor for SAA systems by briefly describing a prototype radar sensor that achieve the required minimum sensing range.

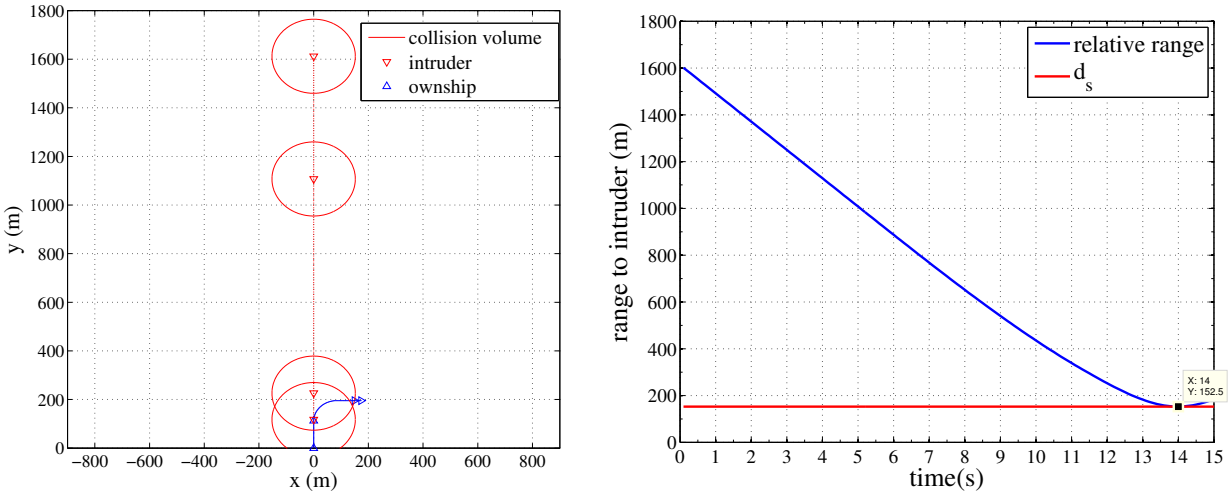
2.2.1 Numerical Analysis

The analysis results shown in Table 2.1 illustrate the validity of the geometry-based analytical solution to design the minimum required sensing range for a sense and avoid system. We consider Eqs. (2.5), and (2.9) for the worst-case horizontal head-on and overtaking collision scenarios, respectively. We have assumed that the computation time t_c is 5 s, d_s = 152.4 m (500 ft), and the performance characteristics of aircraft are given in Table 2.1.

Table 2.1: Sensing range required for SAA system. The symbols H and O are abbreviations for head-on and overtaking collision scenarios, respectively. The numerals marked with asterisk are our best estimate.

Intruder			Raven RQ-11B		ScanEagle		Altus II		Cessna SkyHawk		Airtractor AT-802F	
Ownship			Sensing range (m)									
	v m/s	ψ_{max} deg	H	O	H	O	H	O	H	O	H	O
Raven RQ-11B	22	30*	545	-	729	435	825	581	961	786	1,290	1,282
ScanEagle	41	30*	787	-	999	-	1,111	183	1,267	361	1,646	792
Altus II	51	30*	912	-	1,130	-	1,245	-	1,406	219	1,797	646
Cessna SkyHawk	65	-										
Air Tractor AT-802F	99	-										

It is an obvious conclusion from these results, that an SAA-equipped UAS will require a sensor to provide higher sensing range with increasing speed of encountering aircraft. For instance, if the Raven RQ-11B is required to fly in a specific portion of the airspace in which it expects to encounter a similar type of Raven, ScanEagle and Altus II, then the Raven is required to be equipped with a sensor that is capable of providing a minimum sensing range of 825 m. If a slack parameter of $\delta_r = 0.25$ is used, then the required sensing range becomes $\bar{d}_{r_{min}} = 1.031$ km. However, if any of the aircraft shown in Table 2.1 might be encountered, then a sensor with a minimum range of 1.29 km is required. If a slack parameter of $\delta_r = 0.25$ is used, then the required sensing range becomes $\bar{d}_{r_{min}} = 1.613$ km. To demonstrate the validation of the minimum required sensing range computed in Table 2.1, we simulate an encounter scenario similar to the encounter geometry shown in Figure 2.4. The ownship with dynamic characteristics similar to Raven RQ-11B starts at position (0,0) with initial heading of 0 degrees to the y-axis and moving with constant speed of 22 m/s. The intruder starts at



(a) Overhead view of SAA operation based on minimum sensing range of $\bar{d}_{r_{min}} = 1.613$ km.

(b) Range between ownship and intruder.

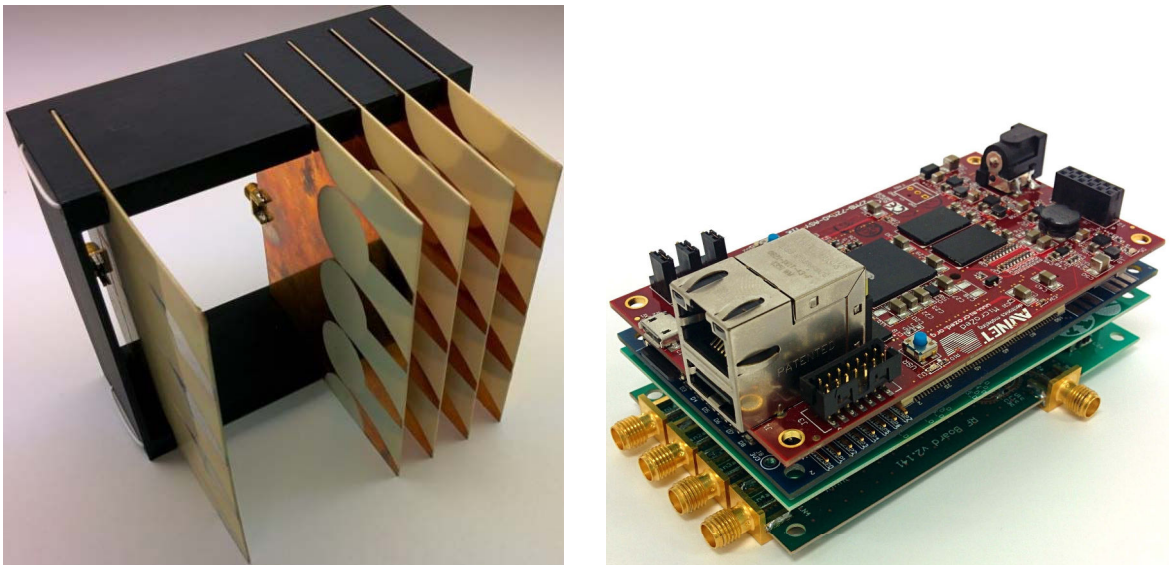
Figure 2.4: Simulation results using ownship dynamic model similar to Raven RQ-11B and intruder similar to Airtractor AT-802F.

position (0, 1613) with 180 deg heading with respect to the y-axis and moving at a speed of 99 m/s. The ownship keeps moves on the same path for 5 s, and then a turn maneuver followed by a straight path to avoid the collision as shown in Figure 2.4(a). Figure 2.4(b) shows that the ownship avoids the intruder and maintains the required minimum safe distance d_s to the intruder given that it detected the intruder at 1.613 km.

2.2.2 Validation Using Prototype Radar Sensor

The proposed radar sensor[1] is an integrated system comprised of a radio frequency transceiver, analog to digital converters, and signal processing. The radar operates in frequency modulated continuous wave (FMCW) mode, which allows it to operate on much less power than traditional pulsed radar systems. The major system specifications are listed in Table 8.3. The processing is accomplished using a microZed processor board that features an on-board field programmable gate array (FPGA) in addition to the central processing unit (CPU). The FPGA executes the computationally intense radar processing, while the

[1]The section is jointly written with Jonathan Spencer, Dr. Randal W. Beard and Dr. Karl F. Warnick as part of a manuscript published in the AIAA Infotech @ Aerospace 2016 [119].



(a) Four channel phased array antenna. (b) Radar system processing boards.

Figure 2.5: Portable radar system designed for use on-board small UAS. The photos are courtesy of the BYU Communications and Sensors Lab©.

operations of collision risk assessment and avoidance path planning are performed on the CPU, which communicates way-points to an external autopilot unit.

Objects are only visible to the naked eye if they can reflect visible light. In a similar manner, objects are only visible to a radar if they can reflect radio waves at the frequency of operation. This ability to reflect radio waves is called the radar cross section (RCS) of the object and arises from the material and geometric properties of the target. The ability to detect an intruder depends as much on the properties of the radar system, such as transmitted power and antenna gain, as it does on the properties of the intruder. The direct dependence on the system properties and intruder properties is evident in the standard radar equation for received power P_{rec} given by

$$P_{rec} = \frac{P_{trans} G^2 \sigma_{\text{RCS}} \lambda^2}{(4\pi)^3 r^4}, \tag{2.11}$$

where P_{trans} is the transmitted power, G is the gain of both the transmitter and receiver respectively, σ_{RCS} is the radar cross section of the target, λ is the wavelength and r is the range to the target [111]. RCS is a strong function of both the frequency of operation and the angle of observation to the target. However, when the size of the target is much

larger than the wavelength of operation, the calculation can be simplified using geometrical approximations.

Table 2.2: Radar sensor parameters [119]

Parameter	**Value**	**Parameter**	**Value**
Antenna size	2.25in x 4in x 1in	Weight	120 g (0.26 lbs)
Consumed Power	8 watts	Approx. Cost of Materials	US$1000
Transmitted Power (P_{trans})	5 miliwatts	Center Frequency	10.25 GHz
Chirp Period (T_c)	1 ms	Chirp Bandwidth	500 MHz
System Noise Figure (F)	8 dB	ADC sample rate	1 Msamp/s

The radar system shown in Fig. 2.5 was able to detect human targets with an RCS of 1 m^2 at a range of 150 m using antennas with a measured gain of $G = 20$. The range at which the radar system can observe a target can be increased by transmitting more power or by using antennas with a higher gain. This relationship is evident in Eq. (2.11), however both options have trade-offs. Transmitting more power will always extend the range of a radar system, but there are both legal and practical limits to the amount of transmitted power. At 10 GHz, where this system operates, the legal transmission limit in the USA is 5 watts of radiated power. Even though this is considerably lower than what many military radars transmit, it still imposes practical limitations for operation on UAS. Power amplifiers generally have a power added efficiency of approximately 20%, so in order to transmit 5 watts of power, the system would require at least 25 watts of battery power, which would severely limit the flight time of smaller aircraft.

Manipulating the antenna configuration of a radar system can also provide the needed gain to detect far away targets, but this also comes at a cost. Higher gain antennas have, by definition, smaller beamwidths and are larger in size. When a set of low gain antennas are used, their beams cover a broad field of view. This can be quite advantageous in a phased array antenna configuration where the total field of view is dictated by the pattern of a single element. A phased array system requires no mechanical scanning and can track multiple targets simultaneously.

Table 2.3: Antenna gain requirements for detecting intruders of specified RCS. An asterisk marks intruders for which geometrical approximations were used to estimate RCS. All other values were found courtesy of [133]

Intruder				Raven RQ-11B	ScanEagle	Altus II	Cessna SkyHawk	Airtractor AT-802F
Ownship	RCS m^2	Payload kg	P_{trans} watt	Required Antenna Gain (G)				
Raven RQ-11B	0.032*	0.36	0.4	166	167	68	52	53
ScanEagle	0.1	6	5	99	89	35	25.5	24
Altus II	1*	150	5	132	114	44	31.5	29
Cessna SkyHawk	3.16	-	-					
Air TractorAT-802F	10*	-	-					

The data in Table 2.1 can also be used to analyze the requirements on the sensor as well, however the dependence of the sensor on RCS adds another dimension to consider. Although RCS data is not available for every type of aircraft, the data that is available correlates quite closely with geometrical approximations for those aircraft. This correlation justifies the use of geometrical approximations for rough RCS estimates when published data is not available, and is therefore the approach taken in this work. The specifications listed in Section 2.2.2 and Eq. (2.11) predict that in order for this radar system to detect a target, it must receive at least $P_{rec} = 1.7\times10^{-15}$ watts of power. Using the RCS of each aircraft, we solve Eq. (2.11) for the antenna gain necessary for each aircraft to detect an intruder at the defined sensing range. If the calculated values for required antenna gain fall within a realistic range, we can conclude that the system is feasible. We assume that the ScanEagle and Altus II operate at the maximum transmittable power level of 5 watts because their payloads are large enough to store the necessary batteries. In like manner, we assume the Raven transmits only 0.4 watts of power, in an effort to maintain low power requirements and not limit flight time. The results are shown in Table 2.3.

From these results it is evident that detecting the aircraft with small RCS is far more difficult than detecting the larger aircraft, despite the fact that they require a much smaller sensing range. The larger payload sizes of the Altus and the ScanEagle would make it easier to implement an antenna system, gimballed or otherwise, that would permit the safe detection of other aircraft. On the other hand, the Raven has an extremely small payload capacity, and the required values of gain are reasonable enough that a phased array

system could successfully be implemented on the Raven if antennas were used that could be attached to the wings. For a four channel phased array system such as the one described in this chapter, the resulting field of view for the gain specified would be approximately 90 deg in azimuth by 15 deg in elevation.

2.3 Conclusions

The potential to integrate UAS into the national airspace is highly dependent on their ability to sense and avoid other air traffic. Based on a worst-case collision encounter geometry, the minimum sensing distance required to safely execute a collision avoidance maneuver was calculated. This minimum sensing distance took into account the computation time involved in target tracking, risk assessment, and path planning in addition to the actual reaction time required to execute the collision avoidance maneuver. The results listed in Table 2.1 demonstrate the need for a sensor with a moderate sensing range of 1 to 2 kilometers and high range accuracy. Radar was selected as the primary sensor because it offers the best performance in non-cooperative scenarios where high range resolution is needed. An analysis was performed to determine whether this could be accomplished on-board small UAS, similar to a Raven RQ-11B. A portable radar sensor under development was used as a benchmark to judge its feasibility for SAA. Our analysis determined that the antenna properties required to detect intruders at a safe distance were quite reasonable, especially for the larger UAS. Although the system appeared feasible for a UAS similar to the Raven, the implementation of a radar system on any aircraft with a smaller payload than the Raven would be impractical. It was found that the antenna requirements are affected more by RCS than they are by range. Any radar-based SAA system designed to detect small RCS intruders at a close range will be powerful enough to detect large RCS intruders from far away.

Chapter 3. Collision Detection and Risk Estimation

It is possible to fly without motors, but not without knowledge and skill.

-Wilbur Wright

In this chapter we develop two approaches to estimate the collision risk of an encounter scenario. The first is a deterministic-based approach similar to those been developed for TCAS in manned aviation. The second is a probabilistic-based approach to address unanticipated intruder maneuvers.

3.1 Deterministic-Based Collision Detection Approach

In this section, we present a collision detection method based on a deterministic approach, where a single trajectory of the intruder is predicted using straight line extrapolation. Then, we estimate the collision risk using the time to the CPA and the distance at the CPA metrics.

3.1.1 Collision Detection Metrics and Logic

We formulate the problem by decoupling the horizontal and vertical state equations. In the horizontal coordinates, we assume the ownship's position and velocity are known and given at time t as $\mathbf{p}_{own}(t)$ and $\mathbf{v}_{own}(t)$, respectively. Let the states of the intruder's position and velocity be $\mathbf{p}_{int}(t)$ and $\mathbf{v}_{int}(t)$. The horizontal positions and velocities of the ownship and intruder starting at time $t = t_0$ are given by

$$\mathbf{p}_{own} = \mathbf{p}_{own}(t_0) + \mathbf{v}_{own}(t_0)t, \tag{3.1}$$

$$\mathbf{p}_{int} = \mathbf{p}_{int}(t_0) + \mathbf{v}_{int}(t_0)t. \tag{3.2}$$

Then, the relative position and velocity between aircraft are given by

$$\mathbf{p}_r = \mathbf{p}_{int}(t) - \mathbf{p}_{own}(t), \tag{3.3}$$

$$\mathbf{v}_r = \mathbf{v}_{int}(t) - \mathbf{v}_{own}(t). \tag{3.4}$$

In this relative coordinate system the ownship is stationary at the center of the frame. The relative position of the intruder with respect to the ownship is $\mathbf{p}_r$ and it moves at a relative velocity of $\mathbf{v}_r$. Assuming a constant velocity model, the time evolution of the horizontal relative position of the intruder with respect to the ownship at time t is given by

$$\mathbf{p}_r = \mathbf{p}_r(t_0) + \mathbf{v}_r(t_0)t. \tag{3.5}$$

Using the Euclidean norm property $\|\mathbf{a} \pm \mathbf{b}\|^2 = \|\mathbf{a}\|^2 + \|\mathbf{b}\|^2 + 2\mathbf{a}^\top\mathbf{b}$, the range to the intruder at any time instant t is given by

$$\begin{aligned} r(t) &= \|\mathbf{p}_r(t_0) + \mathbf{v}_r(t_0)t\|, \\ &= \sqrt{\mathbf{p}_r^\top\mathbf{p}_r + t^2\mathbf{v}_r^\top\mathbf{v}_r + 2t\mathbf{p}_r^\top\mathbf{v}_r}. \end{aligned} \tag{3.6}$$

The rate change of range $r(t)$ with respect to time is given by

$$\begin{aligned} \dot{r}(t) &= \frac{d}{dt}\left(\sqrt{\|\mathbf{p}_r\|^2 + t^2\|\mathbf{v}_r\|^2 + 2t\mathbf{p}_r^\top\mathbf{v}_r}\right), \\ &= \frac{t\|\mathbf{v}_r\|^2 + \mathbf{p}_r^\top\mathbf{v}_r}{\|\mathbf{p}_r(t)\|}. \end{aligned} \tag{3.7}$$

A metric that is used to predict the criticality of an encounter situation is the time remaining to the horizontal closest point of approach between the ownship and the intruder denoted as t_{cpa}. It is defined as the future time instant t at which the range between aircraft is at a minimum i.e. when $\dot{r}(t) = 0$. Assuming both aircraft will continue to fly in a straight line without acceleration then t_{cpa} is given by

$$t_{cpa} = -\frac{\mathbf{p}_r^\top\mathbf{v}_r}{\|\mathbf{v}_r\|^2}. \tag{3.8}$$

Note that t_{cpa} in Eq. 3.8 is undefined when $\|\mathbf{v}_r\| = 0$. This occurs when the velocity vectors of the ownship and the intruder are identical. Hence, to modify Eq. (3.8), t_{cpa} is defined as

$$t_{cpa} \triangleq \begin{cases} \frac{-\mathbf{p}_r^\top \mathbf{v}_r}{\|\mathbf{v}_r\|^2} & \text{if } \|\mathbf{v}_r\| \neq 0, \\ 0 & \text{Otherwise.} \end{cases} \tag{3.9}$$

Note also that t_{cpa} will be zero when the relative position vector $\mathbf{p}_r$ is perpendicular to the relative velocity vector $\mathbf{v}_r$. That means if the relative velocity is not in the direction of the relative position then the two aircraft cannot get any closer. The product $\mathbf{p}_r^\top \mathbf{v}_r$ characterizes whether or not the intruder is horizontally converging, i.e. $\mathbf{p}_r^\top \mathbf{v}_r < 0$, or horizontally diverging, i.e. $\mathbf{p}_r^\top \mathbf{v}_r > 0$. A negative time t_{cpa} occurs when $\mathbf{p}_r^\top \mathbf{v}_r > 0$ meaning that the aircraft are diverging from each other or that the closest point of approach is past and the paths are diverging. The horizontal distance between the ownship and the intruder at the closest point of approach is computed by evaluating the range to the intruder at t_{cpa}

$$\begin{aligned} d_{cpa} &= r(t_{cpa}) = \sqrt{\|\mathbf{p}_r\|^2 + 2t_{cpa}\mathbf{p}_r^\top \mathbf{v}_r + t_{cpa}^2 \|\mathbf{v}_r\|^2} \\ &= \sqrt{\|\mathbf{p}_r\|^2 + 2t_{cpa}\mathbf{p}_r^\top \mathbf{v}_r - t_{cpa}\mathbf{p}_r^\top \mathbf{v}_r}, \end{aligned} \tag{3.10}$$

$$d_{cpa} = \sqrt{\|\mathbf{p}_r\|^2 + t_{cpa}\mathbf{p}_r^\top \mathbf{v}_r}. \tag{3.11}$$

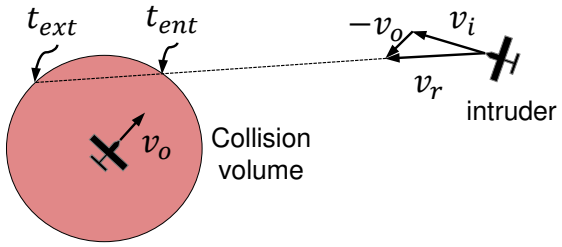


Figure 3.1: Entry and exit times into the collision volume in the horizontal plane.

We know that the protected zone is penetrated when the $r(t) \leq d_s$. As shown in Figure 3.1, when $r(t) \leq d_s$ then there exist time instants to enter and exit the collision

volume horizontally. Using Eq. (3.6) we can solve for these times as follows

$$\begin{aligned} \|\mathbf{p}_r\|^2 + t^2 \|\mathbf{v}_r\|^2 + 2t\mathbf{p}_r^\top \mathbf{v}_r = d_s^2, \\ t^2 \|\mathbf{v}_r\|^2 + 2t\mathbf{p}_r^\top \mathbf{v}_r + \|\mathbf{p}_r\|^2 - d_s^2 = 0. \end{aligned} \tag{3.12}$$

Eq. (3.12) is a quadratic in t, and the roots of this equation gives us the entry and exit times into the collision volume given by

$$t_{ext}, t_{ent} = \begin{cases} \frac{-\mathbf{p}_r^\top \mathbf{v}_r \pm \sqrt{\Delta}}{\|\mathbf{v}_r\|^2} & \text{if } \|\mathbf{v}_r\| \neq 0 \,\text{and}\, \Delta \geq 0, \\ \infty & \text{Otherwise,} \end{cases} \tag{3.13}$$

where $\Delta = (\mathbf{p}_r^\top \mathbf{v}_r)^2 - \|\mathbf{v}_r\|^2 (\|\mathbf{p}_r\|^2 - d_s^2)$ is the discriminant of the quadratic equation. The discriminant term Δ characterizes whether or not the horizontal component of the relative velocity vector penetrates the collision volume i.e. $\Delta > 0$ or outside the collision volume i.e. $\Delta < 0$. The codintion $\Delta > 0$ suggests that there exist two distinct times, that is the entry time to the collision volume in the horizontal plane t_{ent}, and the exit time t_{ext}. When $\Delta = 0$ then there exists only one solution to Eq (3.13) i.e., the entry time equals the exit time $t_{ent} = t_{ext}$. That occurs when the horizontal component of the relative velocity vector is tangent to the collision volume.

In the vertical dimension, we also assume that the vertical component of the position and velocity of the ownship pz_{own}, vz_{own} are known and the states of the vertical component of the position and velocity of the intruder be pz_{int} and vz_{int}, respectively. The vertical positions and velocities of the ownship and intruder at time t is given by

$$pz_{own} = pz_{own}(t_0) + vz_{own}(t_0)t, \tag{3.14}$$

$$pz_{int} = pz_{int}(t_0) + vz_{int}(t_0)t. \tag{3.15}$$

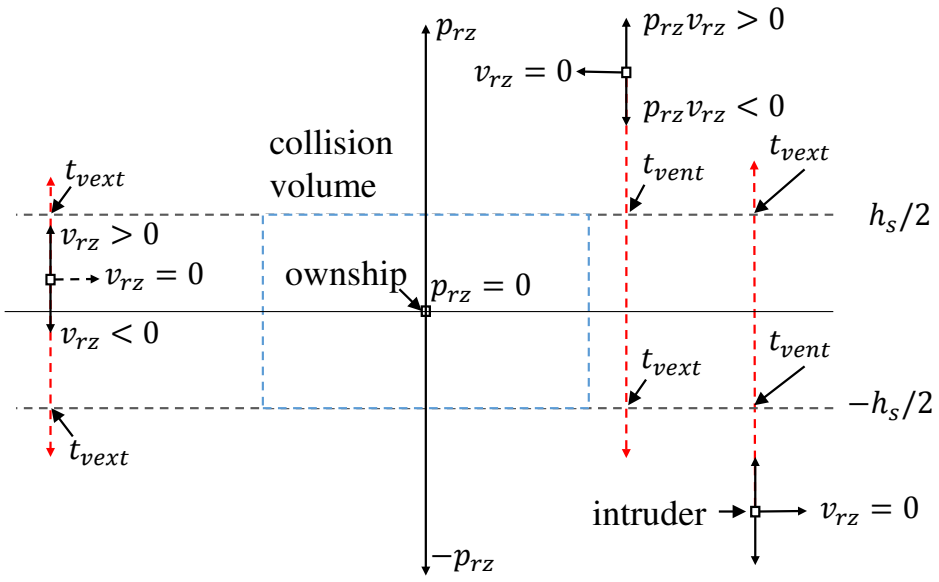


Figure 3.2: Entry and exit times into the collision volume in the vertical plane.

Then, the relative vertical position and velocity between aircraft at any time instant t is given by

$$p_{r_z} = pz_{int}(t) - pz_{own}(t), \tag{3.16}$$

$$v_{r_z} = vz_{int}(t) - vz_{own}(t). \tag{3.17}$$

Assuming a constant vertical speed, the time evolution of the relative vertical position to the intruder at time t is given by

$$p_{r_z} = p_{r_z}(t_0) + v_{r_z}(t_0)t. \tag{3.18}$$

The relative altitude to the intruder at time t is defined as

$$h_r \triangleq \left| p_{r_z} \right|. \tag{3.19}$$

The time that satisfies $p_{r_z} = 0$ in Eq (3.18) is defined as the time to co-altitude [98, 125]. However, since in the vertical plane the protected zone ranges from $-h_s/2$ to $h_s/2$, we are interested in the time when the relative altitude between aircraft is equal to $h_s/2$. The

collision volume is penetrated vertically when $p_{r_z} = \pm h_s/2$. Using Eq. (3.18) the entry and the exit times into the vertical plane are defined as

$$t_{vent} \triangleq \begin{cases} \frac{\epsilon_1 \frac{h_s}{2} - p_{r_z}}{v_{r_z}} & \text{if } p_{r_z} > \frac{h_s}{2}, \text{ or } p_{r_z} < \frac{-h_s}{2} \text{ and } p_{r_z} v_{r_z} < 0, \\ 0 & \text{if } p_{r_z} \in \left[\frac{-h_s}{2}, \frac{h_s}{2}\right], \\ \infty & \text{Otherwise,} \end{cases} \tag{3.20}$$

where

$$\epsilon_1 = \begin{cases} 1 & \text{if } p_{r_z} > \frac{h_s}{2}, \\ -1 & \text{if } p_{r_z} < \frac{-h_s}{2}, \end{cases} \tag{3.21}$$

and

$$t_{vext} \triangleq \begin{cases} \frac{\epsilon_2 \frac{h_s}{2} - p_{r_z}}{v_{r_z}} & \text{if } p_{r_z} > \frac{h_s}{2}, \text{ or } p_{r_z} < \frac{-h_s}{2} \text{ and } p_{r_z} v_{r_z} < 0, \\ 0 & \text{if } p_{r_z} > \frac{h_s}{2}, \text{ or } p_{r_z} < \frac{-h_s}{2} \text{ and } p_{r_z} v_{r_z} \geq 0, \\ \frac{\epsilon_3 \frac{h_s}{2} - p_{r_z}}{v_{r_z}} & \text{if } p_{r_z} \in \left[\frac{-h_s}{2}, \frac{h_s}{2}\right], \end{cases} \tag{3.22}$$

where

$$\epsilon_2 = \begin{cases} -1 & \text{if } p_{r_z} > \frac{h_s}{2}, \\ 1 & \text{if } p_{r_z} < \frac{-h_s}{2}, \end{cases} \tag{3.23}$$

and

$$\epsilon_3 = \begin{cases} 1 & \text{if } v_{r_z} > 0, \\ -1 & \text{if } v_{r_z} < 0, \\ \infty & \text{if } v_{r_z} = 0. \end{cases} \tag{3.24}$$

The definition of the entry and the exit times into the vertical plane given in Eqs. (3.20) and (3.22) cover all the possible cases shown in Figure 3.2. In Figure 3.2 there exist two main cases, the first is defined when the relative vertical position is outside the collision volume cylinder or $p_{r_z} > \frac{h_s}{2}$ or $p_{r_z} < \frac{-h_s}{2}$. In the first case the intruder is maintaining the relative

altitude i.e., $v_{r_z} = 0$, diverging or converging towards the top or bottom of the collision volume cylinder. Similar to the horizontal case, the product $p_{r_z} v_{r_z}$ determines whether the intruder is vertically converging i.e. $p_{r_z} v_{r_z} < 0$, or diverging i.e. $p_{r_z} v_{r_z} > 0$. The second when the relative vertical position of the intruder is within $\pm h_s/2$. In the second case the relative vertical position is inside the collision volume cylinder. The intruder is maintaining the relative altitude or diverging outside the cylinder. In the later case we have assumed that the intruder is in or entered the collision volume cylinder at $t_{vent} = 0$. Additionally, the definitions of the exit and entry time in Eqs. (3.20) and (3.22) avoids a negative sign. Similar to the horizontal plane, a negative entry time occurs when $p_{r_z} v_{r_z} > 0$ meaning that the aircraft are vertically diverging from each other or have already vertically entered and exited the collision volume cylinder. Also, a negative exit time occurs when the relative vertical position is inside the collision volume cylinder and $v_{r_z} \neq 0$ meaning that the intruder has already entered the collision volume cylinder.

Algorithm 1: Collision detection algorithm

```
for each detected intruder do
    if ‖p_r‖ < d_th or (p_r^⊤ v_r < 0 and t_cpa ≤ τ_th) then
        if h_r < h_th/2 or (p_rz v_rz < 0 and t_vent ≤ τ_th) then
            if d_cpa ≤ d_s and t_vext ≥ t_ent then
                Collision is detected.
            end if
        end if
    else
        No collision is detected.
    end if
end for
```

To predict whether or not the collision will occur with detected intruders, we use the collision detection logic given in Algorithm 1. In Algorithm 1, the horizontal distance threshold d_{th}, the vertical distance threshold $h_{th}/2$, and the time threshold τ_{th}, are design parameters that define the collision avoidance threshold. The collision avoidance threshold is a variable boundary around the ownship, larger than the collision volume and depends not only on distance, but time and other factors [30]. It is designed to give a practical separation

such that if a collision is predicted then the ownship has adequate time to plan an avoidance maneuver and to actually fly the maneuver. The design of the collision avoidance threshold is beyond the scope of this dissertation, we refer the interested reader to [30, 34]. Line 2 checks whether the horizontal range $\|\mathbf{p}_r\|$ to the intruder is less than a horizontal threshold distance d_{th}, or the intruder is horizontally converging and the time to CPA is below a time threshold τ_{th}. Line 3 checks whether the relative altitude h_r is less than the vertical distance threshold $h_{th}/2$, or the intruder is vertically converging and the horizontal entry time is below a time threshold τ_{th}. Line 4 checks to see if d_{cpa} is less than or equal the horizontal minimum safe distances d_s and the vertical exit time t_{vext} is less than or equal the horizontal entry time t_{ent}. If so, then line 5 declares a collision.

3.1.2 Accounting for Uncertainty

Since there will always be an uncertainty associated with these estimates, an additional provision should be made to account for that. Sampling-based methods such as Monte Carlo simulations and importance sampling are existing uncertainty propagation approaches. However, instead of using extensive simulations, an analytic expression can be also used to propagate the error variance using a Taylor series approximation. This approach to variance estimation, mostly used in statistical applications has several names in the literature, including the linearization method, the delta method, and propagation of variance [76]. Let $X \in \mathbb{R}^n$ be an $n \times 1$ vector of jointly-Gaussian random variable (r.v.) with mean $\boldsymbol{\mu}_X$, and covariance matrix $\mathbf{Q} \in \mathbb{R}^{n \times n}$. Let Y be a new r.v. that is not measured directly but determined by $Y = g(X)$, where $g(\cdot) : \mathbb{R}^n \to \mathbb{R}$ be a function of X. If $g(X)$ is differentiable, then up to the first-order Taylor series approximation, $Y \approx g(\boldsymbol{\mu}_X) + \nabla g(X - \boldsymbol{\mu}_X)$, where $\nabla g(X) \in \mathbb{R}^{n \times 1}$ is the gradient vector $\frac{\partial g}{\partial X}$ evaluated at $\boldsymbol{\mu}_X$. Then, the r.v. Y has mean $\mu_Y \approx g(\boldsymbol{\mu}_X)$ and variance given by

$$\sigma_Y^2 \approx (\nabla g)^\top \mathbf{Q} \nabla g. \tag{3.25}$$

If the r.v. in X are uncorrelated, Eq (B.1) simplifies further to

$$\sigma_{_Y}^2 \approx (\nabla g)^\top \mathbf{Q}_d \nabla g, \tag{3.26}$$

where $\mathbf{Q}_d = diag([\sigma_{X_1}^2, \sigma_{X_2}^2, \cdots, \sigma_{X_n}^2])$. The approximation given by Eq. (B.2) is the most widely used expression to propagate the statistical error. The full derivation of both Eqs. (B.1) and (B.2) is presented in Appendix B.

Typically, an estimator such as the Kalman filter is used to provide the state estimates of the position and velocity of the ownship and the intruder and also an estimate of the accuracy of these state estimates is also provided and is quantified in an error covariance matrix. In Algorithm 1 Line 4, we used three quantities $d_{_{cpa}}$, $t_{_{ent}}$, and $t_{_{vext}}$. Each of these variables is a function of either horizontal or vertical relative position and velocity states. Using Eq (B.2) we estimate the error uncertainty associated with these variables. Assume that the error uncertainty of the horizontal position and velocity of the intruder are expressed by the variances $\sigma_{\mathbf{p}_{_{int}}}^2$ and $\sigma_{\mathbf{v}_{_{int}}}^2$, respectively. Assume that the error uncertainty of the horizontal position and velocity states of the ownship is negligible. To be more specific, we assume that the uncertainty error of position and velocity components in the NE-plane of the inertial NED reference frame be $\sigma_{n_{_{own}}}^2 = \sigma_{e_{_{own}}}^2 \approx 0$ and $\sigma_{vn_{_{own}}}^2 = \sigma_{ve_{_{own}}}^2 \approx 0$. Assume for convenience, the north-east components of the horizontal position, $(n_{_{int}}, e_{_{int}})^\top$ and velocity, $(vn_{_{int}}, ve_{_{int}})^\top$ of the intruder are independent and identically distributed random variables. Then, the uncertainty error of relative position and velocity can be quantified by the error covariance matrix $\mathbf{Q}_r = diag([\sigma_{n_{_{int}}}^2, \sigma_{e_{_{int}}}^2, \sigma_{vn_{_{int}}}^2, \sigma_{ve_{_{int}}}^2])$. Using Eq. (B.2) the error uncertainty in $t_{_{cpa}}$ is given by

$$\sigma_{_{tcpa}}^2 \approx (\nabla t_{_{cpa}})^\top \mathbf{Q}_r \nabla t_{_{cpa}}, \tag{3.27}$$

where $\nabla t_{cpa} = [\frac{\partial t_{cpa}}{\partial p_{rx}}, \frac{\partial t_{cpa}}{\partial p_{ry}}, \frac{\partial t_{cpa}}{\partial v_{rx}}, \frac{\partial t_{cpa}}{\partial v_{ry}}]^\top$, and where

$$\begin{aligned}
\frac{\partial t_{cpa}}{\partial p_{rx}} &= -\frac{v_{rx}}{\|\mathbf{v}_r\|^2}, \\
\frac{\partial t_{cpa}}{\partial p_{ry}} &= -\frac{v_{ry}}{\|\mathbf{v}_r\|^2}, \\
\frac{\partial t_{cpa}}{\partial v_{rx}} &= -\frac{p_{rx}\|\mathbf{v}_r\| - 2v_{rx}(\mathbf{p}_r^\top \mathbf{v}_r)}{\|\mathbf{v}_r\|^4}, \\
\frac{\partial t_{cpa}}{\partial v_{ry}} &= -\frac{p_{ry}\|\mathbf{v}_r\| - 2v_{ry}(\mathbf{p}_r^\top \mathbf{v}_r)}{\|\mathbf{v}_r\|^4}.
\end{aligned}$$

The error variance in d_{cpa} is given by

$$\sigma^2_{dcpa} \approx (\nabla d_{cpa})^\top \mathbf{Q}_{rd} \nabla d_{cpa}, \tag{3.28}$$

where $\mathbf{Q}_{rd} = diag([\sigma^2_{n_{int}}, \sigma^2_{e_{int}}, \sigma^2_{vn_{int}}, \sigma^2_{ve_{int}}, \sigma^2_{tcpa}])$, and $\nabla d_{cpa} = [\frac{\partial d_{cpa}}{\partial p_{rx}}, \frac{\partial d_{cpa}}{\partial d_{ry}}, \frac{\partial d_{cpa}}{\partial v_{rx}}, \frac{\partial d_{cpa}}{\partial v_{ry}}, \frac{\partial d_{cpa}}{\partial t_{cpa}}]^\top$, where

$$\begin{aligned}
\frac{\partial d_{cpa}}{\partial p_{rx}} &= \frac{2p_{rx} + v_{rx}t_{cpa}}{2\sqrt{d_{cpa}}} \\
\frac{\partial d_{cpa}}{\partial p_{ry}} &= \frac{2p_{ry} + v_{ry}t_{cpa}}{2\sqrt{d_{cpa}}}, \\
\frac{\partial d_{cpa}}{\partial v_{rx}} &= \frac{p_{rx}t_{cpa}}{2\sqrt{d_{cpa}}}, \\
\frac{\partial d_{cpa}}{\partial v_{ry}} &= \frac{p_{ry}t_{cpa}}{2\sqrt{d_{cpa}}}, \\
\frac{\partial d_{cpa}}{\partial t_{cpa}} &= \frac{\mathbf{p}_r^\top \mathbf{v}_r}{2\sqrt{d_{cpa}}}.
\end{aligned}$$

The error variance associated with the horizontal entry time, t_{ent} is given by

$$\sigma^2_{ent} \approx (\nabla t_{ent})^\top \mathbf{Q}_r \nabla t_{ent}, \tag{3.29}$$

where $\nabla t_{ent} = [\frac{\partial t_{ent}}{\partial p_{rx}}, \frac{\partial t_{ent}}{\partial d_{ry}}, \frac{\partial t_{ent}}{\partial v_{rx}}, \frac{\partial t_{ent}}{\partial v_{ry}}]^\top$, where

$$\begin{aligned}
\frac{\partial t_{ent}}{\partial p_{rx}} &= \frac{-v_{rx}}{\|\mathbf{v}_r\|^2} + \frac{v_{rx}\mathbf{p}_r^\top \mathbf{v}_r - p_{rx}\|\mathbf{v}_r\|^2}{\|\mathbf{v}_r\|^2 \sqrt{\Delta}}, \\
\frac{\partial t_{ent}}{\partial p_{ry}} &= \frac{-v_{ry}}{\|\mathbf{v}_r\|^2} - \frac{v_{ry}\mathbf{p}_r^\top \mathbf{v}_r - p_{ry}\|\mathbf{v}_r\|^2}{\|\mathbf{v}_r\|^2 \sqrt{\Delta}}, \\
\frac{\partial t_{ent}}{\partial v_{rx}} &= \frac{-p_{rx}\|\mathbf{v}_r\|^2 + 2v_{rx}\mathbf{p}_r^\top \mathbf{v}_r}{\|\mathbf{v}_r\|^4} + \frac{\left(p_{rx}\mathbf{p}_r^\top \mathbf{v}_r - v_{rx}\left(\|\mathbf{p}_r\|^2 - d_s^2\right)\right)\|\mathbf{v}_r\|^2 - 2v_{rx}\Delta}{\|\mathbf{v}_r\|^4 \sqrt{\Delta}}, \\
\frac{\partial t_{ent}}{\partial v_{ry}} &= \frac{-p_{ry}\|\mathbf{v}_r\|^2 + 2v_{ry}\mathbf{p}_r^\top \mathbf{v}_r}{\|\mathbf{v}_r\|^4} + \frac{\left(-p_{ry}\mathbf{p}_r^\top \mathbf{v}_r - v_{ry}\left(\|\mathbf{p}_r\|^2 - d_s^2\right)\right)\|\mathbf{v}_r\|^2 - 2v_{ry}\Delta}{\|\mathbf{v}_r\|^4 \sqrt{\Delta}},
\end{aligned}$$

where $\Delta = (\mathbf{p}_r^\top \mathbf{v}_r)^2 - \|\mathbf{v}_r\|^2 (\|\mathbf{p}_r\|^2 - d_s^2)$ is the discriminant of the quadratic equation (3.12).

Similarly, under the assumption that the error uncertainty of the relative altitude and vertical speed is determined by the error variance of the vertical position and speed estimates of the intruder. Assume the error variance in the vertical component (i.e. the down component in the NED reference frame) of the position and speed estimates of the intruder be $\sigma_{d_{int}}^2$ and $\sigma_{vd_{int}}^2$, respectively, and that the vertical position and velocity of the intruder are independent and identically distributed random variables, the, the error variance in t_{vext} is given by

$$\sigma_{vext}^2 \approx \left(\frac{\partial t_{vext}}{\partial p_{rz}}\right)^2 \sigma_{d_{int}}^2 + \left(\frac{\partial t_{vext}}{\partial v_{rz}}\right)^2 \sigma_{vd_{int}}^2, \tag{3.30}$$

where

$$\begin{aligned}
\frac{\partial t_{vext}}{\partial p_{rz}} &= \frac{-1}{v_{rz}}, \\
\frac{\partial t_{vext}}{\partial v_{rz}} &= \frac{p_{rz} - v_{rz} - \epsilon h_s/2}{v_{rz}^2},
\end{aligned}$$

and where $\epsilon = \{-1, 1\}$ can be determined from Eq. (3.22).

Therefore, Algorithm 1 is modified by Algorithm 2 to account for the uncertainties associated with d_{cpa}, t_{ent}, and t_{vext}. In Algorithm 2, the parameters m_1, m_2 are positive integers.

Algorithm 2: Collision detection algorithm with uncertainty

for each detected intruder **do**
  if $\|\mathbf{p}_r\| < d_{th}$ or ($\mathbf{p}_r^\top \mathbf{v}_r < 0$ and $t_{cpa} \leq \tau_{th}$) **then**
    if $h_r < h_s/2$ or ($p_{rz} v_{rz} < 0$ and $t_{vent} \leq \tau_{th}$) **then**
      if $d_{cpa} - m_1 \sigma_{dcpa} \leq d_s$ and $(t_{vext} + m_2 \sigma_{vext}) \geq (t_{ent} - m_2 \sigma_{ent})$ **then**
        Collision is detected.
      end if
    end if
  else
    No collision is detected.
  end if
end for

The parameters σ_{dcpa}, σ_{vext}, and σ_{ent} are the standard deviation of the distance at closest point of approach, vertical exit time and horizontal entry time derived from Eqs. (3.28), (3.29), and (3.30).

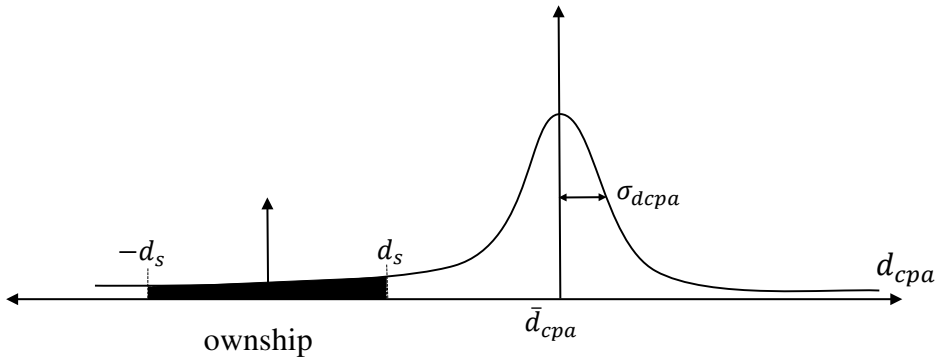


Figure 3.3: The probability distribution function (PDF) of the distance at closest point of approach. The probability of collision is the shaded area under the PDF curve.

The above computation of the variance of distance at closest point of approach, exit and entry time can be used to predict the probability of collision. Given the mean and variance of the distance at closest point of approach $\bar{d}_{cpa}$ and σ^2_{dcpa}, respectively. The probability of a collision can be determined by integrating the area under the probability distribution function of d_{cpa} between the the minimum required safety distance $-d_s$ and d_s.

As shown in Figure 3.3, the probability of a collision $Pr(Col)$ is given by

$$Pr(Col) = \frac{1}{\sigma_{_{dcpa}}\sqrt{2\pi}} \int_{-d_s}^{d_s} \exp(-\frac{(u - \bar{d}_{_{cpa}})^2}{2\sigma_{_{dcpa}}})du. \tag{3.31}$$

Using the definition of the error function

$$erf(w) = \frac{2}{\sqrt{\pi}} \int_0^u \exp(-w^2)dw, \tag{3.32}$$

then, the probability of a collision $Pr(Col)$ is given by

$$Pr(Col) = \frac{1}{2}erf(\frac{d_s + d_{_{cpa}}}{\sqrt{2}\sigma_{_{dcpa}}}) + \frac{1}{2}erf(\frac{d_s - d_{_{cpa}}}{\sqrt{2}\sigma_{_{dcpa}}}). \tag{3.33}$$

3.2 Probabilistic-Based Collision Detection Approach

In this section, we introduce the probabilistic-based approach to detect a collision and estimate the collision risk. We have obtained an analytical discrete-time approximation of the reachable sets over look-ahead time window using Dubins kinematic model of an aircraft flying at a constant altitude. Similar to [58], we have used reachable sets to predict the intruder's trajectory rather than predicting the ownship's ability to determine the collision free space. Alternatively, rather than sampling, we infer a probability distribution from the transition probabilities contained in the MIT LL UEM and project it over a finite look-ahead time horizon. Then we infer a probability distribution over the discretized reachability region to quantify likely intruder trajectories to compute collision risk.

3.2.1 Maneuver Prediction Model

For ease of exposition, we will restrict the attention to the 2D case since the generalization to 3D is straight forward. Let $\mathbf{p}_{_{int}} = (x, y)^\top \in \mathbb{R}^2$ denotes the position of the intruder. The motion of the intruder is described by the Dubins kinematic model of the aircraft flying in the 2D plane with constant speed and subject to an upper bound on the

curvature of its path [39]. The kinematics motion model is given by

$$\dot{x} = v\cos\psi(t), \tag{3.34}$$

$$\dot{y} = v\sin\psi(t), \tag{3.35}$$

$$\dot{\psi} = u, \tag{3.36}$$

where v is the intruder speed and $u \in \mathcal{S}_u$ where $\mathcal{S}_u \triangleq \{u : |u| \leq \bar{u}\}$ is the set of possible turn rate inputs. The system of differential equations (3.34)-(3.36) can be represented as

$$\dot{\mathbf{w}} = f(\mathbf{w}, u),\ |u| \leq \bar{u} \tag{3.37}$$

where $\mathbf{w} \triangleq (x, y, \psi)^\top$ is the system states and u is the system input. The reachable set is the set of states that are accessible at time τ from the initial state $\mathbf{w}_0$.

Definition 1. Let $\hat{w}(\tau, t_0, w_0, u(\cdot))$ be the solution of the system $\dot{w} = f(\mathbf{w}, u)$ at $\tau \geq t_0$ given the initial condition $\mathbf{w}(t_0) = \mathbf{w}_0$, with input $u(\cdot)$ where $u(\tau) \in \mathcal{S}_u$. The reachable set is therefore defined as

$$\mathcal{R}(\tau, t_0, \mathbf{w}_0) \triangleq \{\xi \in \mathbb{R}^{\mathbf{n}} | \exists \text{ input } u(\cdot) \text{ such that } \hat{w}(\tau, t_0, w_0, u(\cdot)) = \xi\}.$$

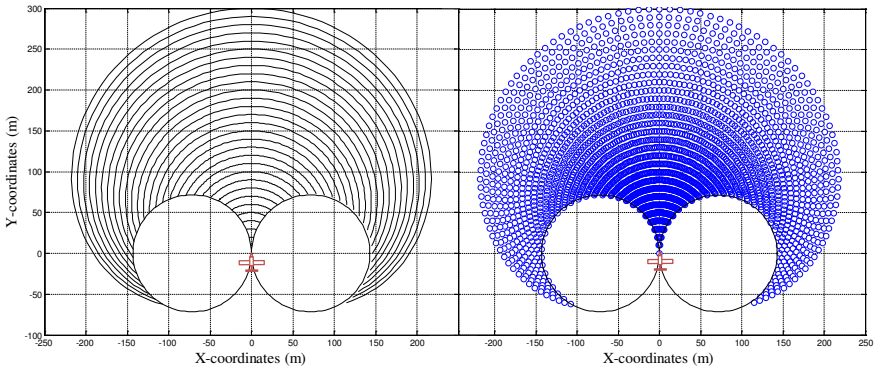


Figure 3.4: Representations of reachable sets level curves in a)continuous time; and b) a discrete-time approximation.

The reachable sets $R(\tau, t_0, \mathbf{w}_0)$ for increasing values of $\tau \geq t_0$ are shown in Figure 3.4(a), where level curves of the reachable sets $\mathcal{R}_{rd}(\tau, t_0, \mathbf{w}_0)$ are generated using Eqs. (3.34)-(3.36) with $\mathbf{w}_0 = (0, 0, 0)^\top$, $\mathcal{S}_u = [-8, 8]$ deg/s, and $v = 10$ m/s over $\tau = [0, 30]$ s. To obtain a discrete-time approximation, we quantize the input as $\mathcal{S}_{u_d} = \{-\bar{u}, -\bar{u}+\delta_u, \cdots, -\bar{u}+m\delta_u, \bar{u}\}$ where $\delta_u = \frac{2\bar{u}}{m}$ and m is the quantization level. Integrating Eq. (3.36) gives

$$\psi(t) = \psi(t_0) + \int_{t_0}^{t} u(\tau) d\tau. \tag{3.38}$$

Letting $t = (k+1)T_s$ and $t_0 = kT_s$, where T_s is the sampling time and assuming that $u(\cdot)$ is constant between sample times, we get

$$\psi((k+1)T_s) = \psi(kT_s) + u(kT_s) \int_{kT_s}^{(k+1)T_s} d\tau. \tag{3.39}$$

Adopting the notation $\xi_k \triangleq \xi(kT_s)$, the discrete-time solution of Eq. (3) is given by

$$\psi_{k+1} = \psi_k + T_s u_k. \tag{3.40}$$

Then, integrating Eq. (3.34) assuming that v is a constant gives

$$\begin{aligned} x_{k+1} &= x_k + v \int_{kT_s}^{(k+1)T_s} \cos\Big(\psi_k + u_k \int_{kT_s}^{\tau} d\tau_1\Big) d\tau \\ &= x_k + v \int_{kT_s}^{(k+1)T_s} \cos\Big(\psi_k + u_k(\tau - kT_s)\Big) d\tau. \end{aligned}$$

Letting $\lambda = \psi_k + u_k(\tau - kT_s)$, then $d\tau = d\lambda/u_k$. Making the change of variables in the integral gives

$$x_{k+1} = x_k + \frac{v}{u_k} \left(\sin(\psi_k + T_s u_k) - \sin(\psi_k)\right).$$

Using the trigonometric identity $2\cos(a)\sin(b) = \sin(a+b) - \sin(a-b)$, then discrete-time solution of Eq. (3.34) becomes

$$x_{k+1} = x_k + \frac{2v}{u_k} \sin(\frac{T_s u_k}{2}) \cos(\psi_k + \frac{T_s u_k}{2}). \tag{3.41}$$

Following similar steps, the discrete-time solution of Eq. (3.35) is given as

$$y_{k+1} = y_k + \frac{2v}{u_k}\sin(\frac{T_s u_k}{2})\sin(\psi_k + \frac{T_s u_k}{2}). \tag{3.42}$$

However, implementing both Equations (3.42) and (3.41) requires a divide by zero when $u_k = 0$. Using $\text{sinc}(x) \triangleq \frac{\sin(\pi x)}{\pi x}$, the discrete-time solution of Eqs. (3.34)-(3.36) is given as

$$x_{k+1} = x_k + T_s v\,\text{sinc}(\frac{T_s u_k}{2\pi})\cos(\psi_k + \frac{T_s u_k}{2}), \tag{3.43}$$

$$y_{k+1} = y_k + T_s v\,\text{sinc}(\frac{T_s u_k}{2\pi})\sin(\psi_k + \frac{T_s u_k}{2}), \tag{3.44}$$

$$\psi_{k+1} = \psi_k + T_s u_k. \tag{3.45}$$

The reachable set for the discrete dynamics with quantized input is defined as follows.

Definition 2. Let $\hat{w}_d(\ell, t_0, w_0, u_d(\cdot))$ be the solution of the system $w_{k+1} = f_d(w_k, u_k)$ at time $t = t_0 + \ell T_s$, where $\ell = 1, 2, \cdots, L$ is the discrete look-ahead index, starting from the initial condition $\mathbf{w}(t_0) = \mathbf{w}_0$ and with input $u_d(\cdot)$ where $u_d(t_0 + kT_s) \in \mathcal{S}_{u_d}$. The definition of the reachable set at time ℓ, given the initial state $\mathbf{w}_0$ at the initial time t_0 is

$$\mathcal{R}(\ell, t_0, \mathbf{w}_0) \triangleq \{\xi \in \mathbb{R}^{\mathbf{n}} |\ \exists \text{ input } u_d(\cdot) \text{ such that } \hat{w}_d(\ell, t_0, w_0, u_d(\cdot)) = \xi\}.$$

Note that the size of the discretized input set $\mathcal{S}_{u_d}$ is $m + 1$, which implies that the cardinality of the discrete-time reachable set $\mathcal{R}_d(\ell, t_0, \mathbf{w}_0)$ is $(m + 1)^\ell$ states. To manage the exponential growth of the size of $\mathcal{R}_d(\ell, t_0, \mathbf{w}_0)$ observe that aircraft tend to maneuver slowly, and that the most likely future trajectory, at least over the near term, is to hold the current airspeed and turn rate. Therefore, we will limit the reachable set to those states that can be reached with constant input trajectories $u_d(\ell_1) = u_d(\ell_2)$ for all $\ell_1, \ell_2 = 1, \ldots, L$. We denote this set as $\mathcal{R}_{rd}(\ell, t_0, \mathbf{w}_0)$ and note that its cardinality is $(m + 1)$. Figure 3.4(b) shows the reduced discrete reachable set with a fixed number of configurations along each level curve, where the discrete-time approximation, $\mathcal{R}_{rd}(\ell, t_0, \mathbf{w}_0)$ is generated with $\delta_u = \frac{16}{m}$ deg/s, $m = 100$, and $T_s = 1$ s for $\ell = 1, 2, \cdots, 30$.

Having determined the region of accessibility of the intruder over a look-ahead window, the question becomes: What is the probability that the intruder will reach a specific configuration point within the reachability region? To answer this question we use the transition probabilities contained in the MIT LL UEM. The MIT LL UEM is a high fidelity probabilistic airspace encounter model based on actual radar data collected from more than 120 radar sites across the United States and is statistically representative of the behavior of aircraft maneuvering in close proximity to each other. The MIT LL UEM is based on the use of Bayesian networks to represent relationships between the airspace class A, altitude layer L, speed v, acceleration $\dot{v}$, turn rate $\dot{\psi}$, and climb rate $\dot{h}$. The airspace class A is divided into four values (class B, C, D and O) where O represent other airspace class such as class G. Altitude layer is also divided into four layers, 500-1200, $1200-3000$, 3000-5000 and 5000-18000 feet above the ground level. The continuous variables, such as the turn rate $\dot{\psi}$, is quantized into seven bins symmetric about zero, [-8, -6], [-6, -4.5], [-4.5, -1.5], [-1.5, 1.5], [1.5, 4.5], [4.5, 6] and [6, 8] degree/s from which a specific value is then sampled uniformly $[61, 63]$. Consider a collision encounter where the ownship and the intruder are

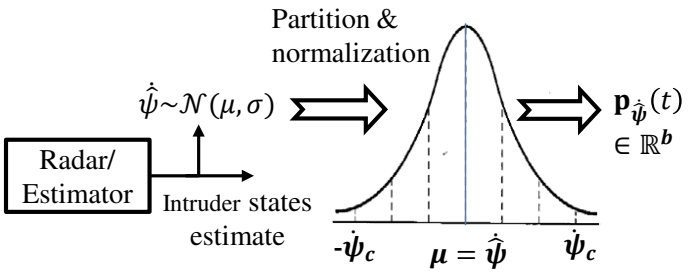


Figure 3.5: The state probability vector, $\mathbf{p}_{\dot{\hat{\psi}}}(t)$ can be obtained through partitioning and normalizing $p_{\dot{\hat{\psi}}}(\dot{\hat{\psi}})$ over the discrete bins of $\dot{\psi}$

flying in a specific airspace class and altitude layer. Let $\hat{\mathbf{r}}(t)$, $\hat{v}(t)$, and $\dot{\hat{\psi}}(t)$ be the ownship's estimates of the intruder's position, speed and turn rate, respectively. Then these estimates determine where to look in the transition probability tables of the MIT LL UEM. We can extract from these tables the probabilities of each possible candidate of $\dot{\psi}(t + T_s)$ jointly conditioned on A, L, $\hat{v}(t)$ and $\dot{\hat{\psi}}(t)$. These transition probabilities can be represented by a one-time step transition probability matrix $M_{\text{trans}} \in \mathbb{R}^{b \times b}$, where b is the number of the turn

rate discrete bins defined in the MIT LL UEM. Assuming that the estimated turn rate, $\dot{\hat{\psi}}(t)$ is normally distributed with $\dot{\hat{\psi}} \sim \mathcal{N}(\mu, \sigma)$ and the associated probability density function (pdf) is $p_{\dot{\hat{\psi}}}(\dot{\hat{\psi}})$. Suppose that the pdf of the turn rate estimate can be represented by a discrete state probability vector $\mathbf{p}_{\dot{\hat{\psi}}}(t) \in \mathbb{R}^{1\times b}$. As shown in Figure 3.5, the state probability vector $\mathbf{p}_{\dot{\hat{\psi}}}(t)$ at the current time t can be obtained by partitioning and normalizing $p_{\dot{\hat{\psi}}}(\dot{\hat{\psi}})$ over the turn rate range $\dot{\psi} \in [-\dot{\psi}_c, \dot{\psi}_c]$. Now, we would like to propagate $p_{\dot{\hat{\psi}}}(\dot{\hat{\psi}})$ over the time horizon $[t, t + \ell T_s]$ using the one-step transition probability M_{trans}. However, the MIT LL UEM transition distribution table projects the transition probabilities one time step into future while our approach requires the knowledge of the probability distribution over the entire time horizon. A possible solution is to employ the concept of the n-steps transition probability matrix of a discrete Markov random process. The probability distribution of a discrete Markov process after n$\geq$ 1 transitions is completely determined by the one-step transition probability matrix and the initial state probability vector. Similarly, the probability distribution of $\dot{\hat{\psi}}$ or the state probability vector $\mathbf{p}_{\dot{\hat{\psi}}}(t)$ propagated ℓ time steps into the future can be given by

$$\mathbf{p}_{\dot{\hat{\psi}}}(t + \ell T_s) = \mathbf{p}_{\dot{\hat{\psi}}}(t) M^{\ell}_{\text{trans}}. \tag{3.46}$$

Let $\mathbf{P}_{\text{tran}} \in \mathbb{R}^{b\times L}$ be the matrix of all the state probability vectors $\mathbf{p}_{\dot{\hat{\psi}}}(\tau_d)$, defined as

$$\mathbf{P}_{\text{tran}} \triangleq [\mathbf{p}^{\top}_{\dot{\hat{\psi}}}(t + T_s), \mathbf{p}^{\top}_{\dot{\hat{\psi}}}(t + 2T_s), \ldots, \mathbf{p}^{\top}_{\dot{\hat{\psi}}}(t + LT_s)]^{\top}. \tag{3.47}$$

The b quantization intervals for $\dot{\psi}$ define b wedges in the reachability set. As shown in Figure 3.6(a) we will denote these wedges by $W_i, i = 1, \ldots, b$. The current model does not account for the uncertainty associated with the intruder's position, velocity and heading estimates. However, we have assumed that the reachable sets are determined based on the current estimate of position, velocity and heading estimates of the intruder provided by the radar/tracker which represent the mean of their respective distribution.

3.2.2 Collision Risk Estimation

The purpose of computing the collision risk is to have an alert threshold value above which the collision avoidance system is triggered to initiate an evasive maneuver to avoid an imminent collision with the detected intruding aircraft. There are a number of approaches to evaluate the future collision risk of an encounter situation. Most of these approaches can either be classified as geometric or probabilistic, where each approach has different techniques to deal with errors. In the geometric approach, the collision risk is described based on the geometric relationship between aircraft. Aircraft trajectory predictions are based on linear projections of current aircraft states such that the uncertainty of the predicted trajectory is translated into areas around the predicted trajectory referred as to safe zones. Linear projections can be computationally efficient and prediction errors are negligible over short time horizons [25, 50] or assumed known when flight plans are communicated [98]. On the other hand, the probabilistic methods estimate the probability of collision based on a probabilistic model of future intruder dynamics. This event probability is then compared to a certain threshold above which the aircraft is deemed to be in collision. These probabilities can be estimated using approximate analytical solution [102], numerical approximation [73, 127], or Monte Carlo methods [58, 78, 81]. The expected utility is another approach used to develop a risk alerting system that accounts for future changes in alerts and the responses to them [26]. In general, the probabilistic approaches are computationally intensive but suffer less from false alarms than geometric approaches. Our work is similar to the threshold-based probabilistic methods. However, rather than making simplifying assumptions in order to derive an analytical solution or performing a computationally expensive sampling to account for all possible states, our approach presents a tradeoff that accounts for likely intruder trajectories with a computational scheme that can be handled by limited resources available on small UAS.

Let $\mathbf{p}_{own}(t)$ be the position of the ownship and let $\hat{\mathbf{p}}_{int}(t)$ be the estimated position of the intruder at the current time t. The protected zone of the ownship is a virtual region

centered on the ownship, usually represented by a circular disk with radius d_{safe}, defined as

$$\mathcal{B}_{d_s}(\mathbf{p}_{own}(t)) = \left\{\mathbf{d} \in \mathbb{R}^2 : \left\|\mathbf{p}_{own}(t) - \mathbf{d}\right\| \leq d_s\right\}.$$

The encounter situation is said to generate a collision if the intruder enters $\mathcal{B}_{d_s}$. The collision risk is the largest probability of a collision event over the look ahead window $\tau \in [t, t + \ell T_s]$. The probability of collision at step $\ell = 1, \ldots, L$ is computed as

$$P_{\text{col}}(\ell, t) = \bigcup_{w \in R_d(\ell, t, \hat{w}(t))} \mathcal{I}\left[w \in \mathcal{B}_{d_s}(\mathbf{p}_{own}(t + \ell T_s))\right] P\left(\left\{\hat{\mathbf{p}}_{int}(t + \ell T_s) = w\right\}\right), \tag{3.48}$$

where $\mathcal{I}[\cdot]$ is the indicator function, and $P\left(\left\{\hat{\mathbf{p}}_{int}(t + \ell T_s) = w\right\}\right)$ is the probability of the event that the actual future state of the intruder at time $t + \ell T_s$ will be w. In this section, the future predicted position of the ownship is given given by

$$\mathbf{p}_{own}(t + \ell T_s) = \mathbf{p}_{own}(t) + \ell T_s v_{own}(t)(\cos\psi_{own}(t), \sin\psi_{own}(t))^\top,$$

where $\mathbf{p}_{own}(t)$, $v_{own}(t)$, and $\psi_{own}(t)$ are the current position, speed, and course of the ownship.

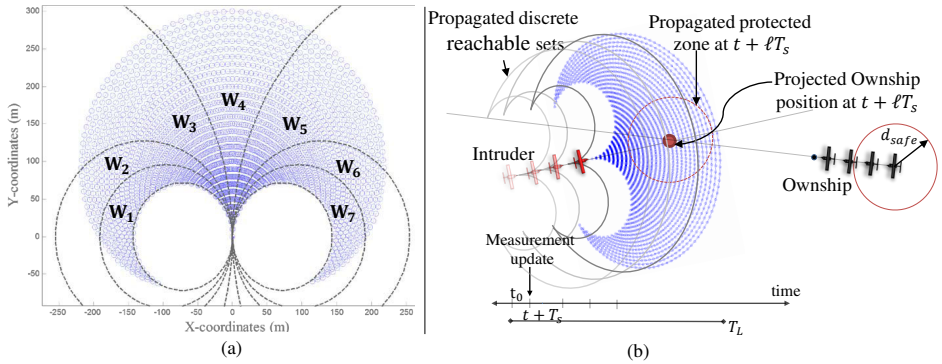


Figure 3.6: a) There are b-wedges in the reachability set; and b) collision risk is continuously updated after every new measurement.

As indicated in Figure 3.6, the probability of collision at step ℓ given in Eq. (3.48) is the number of states belonging to the intersection between the reachable set $\mathcal{R}_d(\ell, t, \hat{\mathbf{p}}_{int}(t))$

and the safe zone $\mathcal{B}_{d_s}(\mathbf{p}_{own}(t+\ell T_s))$ relative to the total number of states in the relative wedges, weighted by the corresponding transition probabilities in the ℓ^{th} column of $\mathbf{P}_{\text{tran}}$. If we use the notation $|\mathcal{A}|$ to denote the cardinality of the discrete set $\mathcal{A}$, then we have

$$P_{\text{col}}(\ell,t)=\sum_{i=1}^{b}\frac{\left|\mathcal{R}_{rd}(\ell,t,\hat{\mathbf{p}}_{int}(t))\bigcap\mathcal{B}_{d_s}(\mathbf{p}_{own}(t+\ell T_s))\bigcap W_i\right|}{\left|\mathcal{R}_{rd}(\ell,t,\hat{\mathbf{p}}_{int}(t))\bigcap W_i\right|}\mathbf{P}_{\text{tran}}(\ell,i). \tag{3.49}$$

To account for the fact that there are many paths to each location, and that the input trajectory for the intruder may not have been constant, we modify Equation (3.49) to include several fronts of the reachable set as

$$P_{\text{col}}(\ell,t)=\frac{1}{2\mathcal{L}+1}\sum_{m=\ell-\mathcal{L}}^{\ell+\mathcal{L}}\sum_{i=1}^{b}\frac{\left|\mathcal{R}_{rd}(m,t,\hat{\mathbf{p}}_{int}(t))\bigcap\mathcal{B}_{d_{\text{safe}}}(\mathbf{p}_{own}(t+\ell T_s))\bigcap W_i\right|}{\left|\mathcal{R}_{rd}(m,t,\hat{\mathbf{p}}_{int}(t))\bigcap W_i\right|}\mathbf{P}_{\text{tran}}(m,i), \tag{3.50}$$

where $\mathcal{L}$ is the size of the window of level curves inside the safe zone at time step ℓ. To maximize the safety of the system, the maximum collision risk estimate is evaluated over the look ahead window $\ell\in[1,L]$ as

$$P_{\text{col}}(t)=\max_{\ell\in[1,L]}P_{\text{col}}(\ell,t). \tag{3.51}$$

Generally, the collision risk involves extracting some measure of how the current encounter situation is critical compared to a threshold $\mathcal{C}_D$. If the threshold is exceeded i.e. $P_{\text{col}}(t)\geq\mathcal{C}_D$, then the collision detection logic issues an alert to declare that the detected intruder is on a collision course. As shown in Figure 3.6(b), the collision risk is evaluated regularly, that is, after every new measurement the state estimates of the intruder and the locus of reachable configurations is updated with respect to the new estimated location and orientation of the intruder. In addition, the transition probability assignment is updated over the course of the collision encounter depending on the estimated air class, altitude level, speed, acceleration, and heading rate variations of the intruder. The collision risk is continuously evaluated after every measurement update using Eq. (3.51).

3.3 Assessment and Validation Measure

The objective of this assessment and validation is to provide a quantifiable performance measure in terms of the traditional detection metrics of probability of detection P_d and probability of false alarm P_{fa}. A Monte Carlo simulation is used to generate the system operating characteristic (SOC) curve which quantifies false alarm rates and correct detection rates. If $\mathcal{N}$ is the number of performed simulations, among which there are $\mathcal{E}$ true collision encounters for a specific decision threshold $\mathcal{C}_D$, and the proposed collision detection algorithm detects $\mathcal{M}$ collision encounters, among which $\tilde{\mathcal{E}}$ encounters are the true collision events, then the correct detection rate P_{cd} and false alarm rate P_{fa} are given by

$$P_{cd} = \frac{\tilde{\mathcal{E}}}{\mathcal{E}}, \tag{3.52}$$

$$P_{fa} = \frac{\mathcal{M} - \tilde{\mathcal{E}}}{\mathcal{N} - \mathcal{E}}. \tag{3.53}$$

By varying $\mathcal{C}_D$ for different sizes of intruder aircraft, we obtain P_{cd} and P_{fa} as a function of $\mathcal{C}_D$. Plotting P_{cd} versus P_{fa} as $\mathcal{C}_D$ varies gives us the SOC that determines the performance of the detection logic and also can be used to select a decision threshold that yields the optimal P_{cd} and P_{fa}

3.4 Performance Evaluation Using Monte Carlo Simulations

To validate the collision detection algorithm, we create a simulation environment with 4 degree-of-freedom aircraft models for both the ownship and intruder. We assume that both the ownship and intruder are flying at the same altitude. To generate realistic encounter scenarios that are representative of what is observed in the national airspace, we sample flight trajectories from the MIT LL UEM. The encounter scenarios are assumed to take place in class G airspace within 500-1200 ft altitude layer. The initial airspeed of the intruder varies over (0, 31] m/s based on the type of the intruder used in the simulation. In addition, the control variables $\dot{v}$ and $\dot{\psi}$ vary from -2 to 2 m/s^2 and -8 to 8 degree/s, respectively while the climb rate $\dot{h}$ is set to 0 m/s. To construct the encounter geometry we follow an approach suggested by M.J. Kochenderfer et al. [61]. We assume a virtual encounter circle centered

on the ownship. The radius of the encounter circle is set to $d_{enc} = 500$ m. The ownship is initialized at the center of the encounter circle $\mathbf{p}_{own}(t_0) = (0, 0)$ and follows a straight line path. However, at each Monte Carlo run the intruder is randomly initialized on the perimeter of the encounter circle. The bearing angle of the intruder relative to the ownship is drawn from a uniform distribution over $[0, 2\pi]$ and the heading of the intruder is randomly drawn from a uniform distribution over $[0, 2\pi]$ while the ownship heading is set to 0 degree. Monte Carlo based simulations are used to test the performance of the detection algorithm. Additive zero mean white noise is introduced on the intruder states to model what a sensor and detection/tracking system would actually deliver to the sense-and-avoid system. The following 1σ values are assumed for the states of interest:

$$\sigma_x = \sigma_y = 5 \text{ m}, \qquad \sigma_v = 1.5 \text{ m/s}, \qquad \sigma_\psi = 1.2 \text{ deg/s}.$$

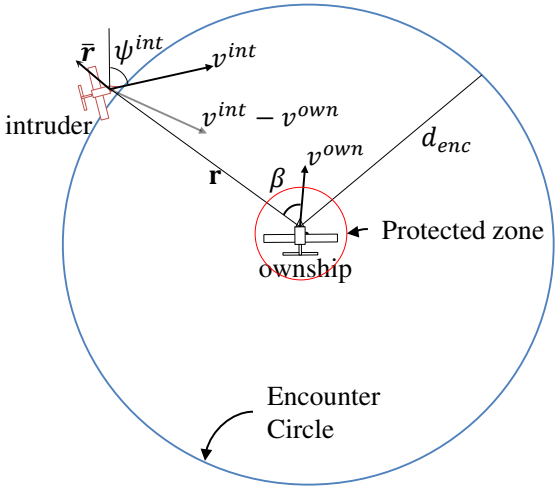


Figure 3.7: The encounter geometry used for Monte Carlo simulations.

The simulated encounters are similar to the planner geometry shown in Figure 3.7. Monte Carlo based simulations are used to test the performance of the detection algorithm.The procedure of the Monte Carlo simulations consists of the following steps.

1. The trajectories of both the ownship and intruder are generated over a 30 s time horizon where the sample time is set to 0.1 s.

2. At each run, the intruder is initialized on the perimeter of the encounter circle, with a random heading that ensures that the intruder is penetrating the encounter region at the first time instant.

3. For each Monte Carlo simulation trial, execute the collision detection algorithm. A collision threat is declared as soon as the collision risk measure $P_{\text{col}}(t)$ exceeds the threshold $\mathcal{C}_D$. The definition of collision threat detection accounts for the fact that a collision may be detected before it is too late to take an action. In these simulation, a collision threat alert is considered valid if it is declared at least 10 s before the collision would actually occur.

4. We compute the probability of false alarm and the probability of correct detection for each collision threshold value and plot the system operating characteristic curve, P_{cd} versus P_{fa}. A thousand Monte Carlo simulation runs are conducted for each collision threshold value.

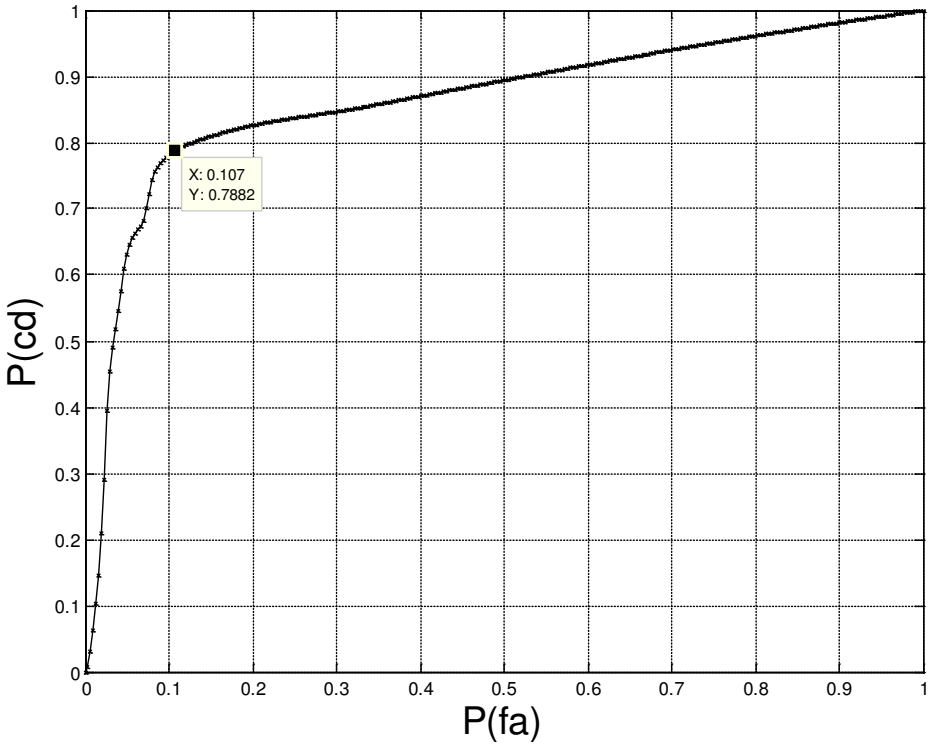


Figure 3.8: The SOC curve parametrized by the threshold. The data tip (0.107, 0.7882) corresponds to the *"optimal"* threshold point.

The estimates of the probability of false alarm and the probability of correct detection are obtained by 1000 Monte Carlo simulations for each threshold $\mathcal{C}_D$ value. The system operating characteristic (SOC) curve for the proposed collision detection and risk estimation is shown in Figure 3.8. An ideal collision detection algorithm would operate at the point $P_{fa} = 0$ and $P_{cd} = 1$ i.e. all collision threats are detected and there are no false alarms. Unfortunately, any real scheme cannot operate at such a point due to the uncertainty in the intruder state estimates. However, the closer the SOC curve is to the point $(0, 1)$ the better the performance will be. Therefore, the *"optimal"* compromise between correct detection and false alarm rates corresponds to the point on the SOC curve that has a minimum distance to the ideal operating point $(0, 1)$.

3.5 Conclusions

We have developed in this chapter a deterministic-based collision detection approach using the time and distance to closest point of approach metrics. We extend the approach to account for uncertainties in the intruder state estimates by deriving analytic expressions to propagate the error variance using a Taylor series approximation.

To address unanticipated intruder maneuvers, we have also developed a probabilistic based collision detection and risk estimation approach that is computationally feasible for small UAS. The computational burden is expected to be much less than existing methods that require running Monte Carlo approximations in near real time or performing extensive sampling from a probabilistic encounter model. That makes the proposed approach a tractable solution in particular for small and mini UAS. The proposed collision approach has the advantage that the locus of states within the reachable sets with respect to the detected intruder are only computed once, which makes it attractive and feasible for real-time SAA on board small UAS. However, the orientation and translation of the constructed reachable sets with respect to the intruder may need to be updated every time a new measurement is received.

Chapter 4. Collision Avoidance System

All the calculations show it can't work. There's only one thing to do: make it work.

-Pierre Georges Latécoère, early French aviation entrepreneur.

In this chapter, we present multiple avoidance path techniques suitable for small UAS sense-and-avoid systems. We first propose a feasible collision avoidance based on a virtual chain placed in a virtual force field. Second, we present an innovative collision avoidance logic using a local coordinate system centered at the ownship.

4.1 Chain-Based Collision Avoidance

The chain-based concept was initially presented in [13] to plan a path using a series of connected waypoints. The waypoints serve as the links of a simulated chain. Since the path is represented using waypoints that are a fixed distance apart, it is easy to determine roughly where the UAS will be at any given time. This timing information can be used to prevent collisions and spread out paths when creating plans to avoid multiple UAS [13].

The method for finding a path that safely avoids intruder aircraft is based on the work of [13, 22, 88]. Similarly, we model the chain as a collection of unit-mass points constrained to the 2-D plane. Letting $z_i = (x_i, y_i)^\top \in \mathbb{R}^2$, $i = 1, 2, ..., N$ be the position of the ith point or node in the chain, the N-link chain is represented by

$$\mathbf{c} = \left[\mathbf{z}_1^\top, \mathbf{z}_2^\top, \ldots, \mathbf{z}_N^\top\right]^\top.$$

If the point masses are unconstrained, then by Newton's second law of motion, the unconstrained dynamics equation describing the entire chain motion is given by

$$\ddot{\mathbf{c}} = \mathbf{F}, \tag{4.1}$$

where $\mathbf{F}$ is the sum of applied forces, which consists of constrained and unconstrained forces. The unconstrained forces applied to the chain are composed of two forces: the first pushes the chain towards the ownship target, while the second repels the ownship from the intruder observing the right-of-way rules.

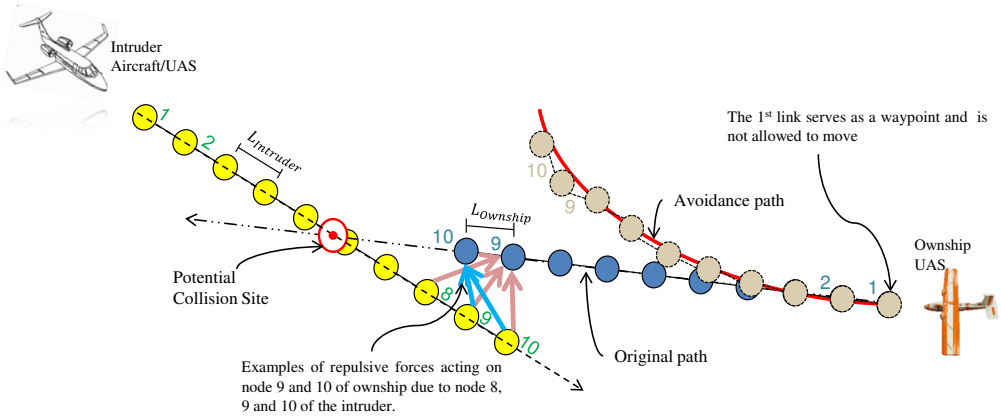


Figure 4.1: Representation of repulsive forces between chains.

To push the chain in the desired direction it is placed in a force-field that is generated using the gradient of a bounded differential reward function. To create a force that causes the chain of the ownship to avoid the intruder, first we assume that the position of the ith element in the intruder chain is known. As depicted in Figure 4.1 let $\mathbf{c}_O = \left[\mathbf{z}_{O1}^\top,\ \mathbf{z}_{O2}^\top, \ldots, \mathbf{z}_{ON}^\top\right]^\top$ be the N-link chain associated with the ownship, and let $\mathbf{c}_I = \left[\mathbf{z}_{I1}^\top,\ \mathbf{z}_{I2}^\top, \ldots, \mathbf{z}_{IN}^\top\right]^\top$ be the N-link chain associated with the intruder. Note that while both chains have the same number of links the desired distance between each link does not need to be the same for both the ownship and the intruder but should be proportional to speed, so that both chains are associated with a fixed look ahead time T. This implies that each UAS will fly through the ith node in its chain at approximately the same time. Let $\mathbf{d}_{OI}(m, n)$ be the vector from the

nth node of the intruder aircraft to the mth node of the ownship UAS, defined as

$$\mathbf{d}_{OI}(m,n) = \mathrm{z}_{Om} - \mathrm{z}_{In} \tag{4.2}$$

and

$$\hat{\mathbf{d}}_{OI}(m,n) = \frac{\mathbf{d}_{OI}(m,n)}{\|\mathbf{d}_{OI}(m,n)\|.} \tag{4.3}$$

Where $\|\cdot\|$ is the standard Euclidean norm. As shown in Figure 4.1, the repulsive force acting on node m of the ownship UAS due to node n of the intruder aircraft is defined as

$$f_{rp}(m,n) = \begin{cases} \hat{\mathbf{d}}_{OI}(m,n)\gamma_{rp1}e^{(-\gamma_{rp2}\|\mathbf{d}_{OI}(m,n)\|)} & if\ |m-n| < k\ and \\ & \|\ \mathbf{d}_{OI}(m,n)\ \| < d_{max} \\ 0 & Otherwise, \end{cases} \tag{4.4}$$

where $\gamma_{rp1}, \gamma_{rp2}$ are nonnegative constants, d_{max} is the greatest distance over which nodes can influence each other, and k is a positive integer that determines the time dependence of the interactions. As we have no control over the intruder's chain, the ownship's path is pushed away from the intruder's path while the intruder's path is not influenced by the ownship's path.

Merging all of the repulsive forces acting on the chain of the ownship by the chain of the intruder, we obtain

$$\mathbf{F}_{rpOI} = \left[F_{rpOI}^{\top}(1),\ F_{rpOI}^{\top}(2), \cdots, F_{rpOI}^{\top}(N)\right]^{\top}. \tag{4.5}$$

Where

$$F_{rpOI}(m) = \sum_{n=1}^{N} f_{rp}(m,n), \tag{4.6}$$

is the sum of all the forces acting on the mth node of the ownship by all the nodes of the intruder.

To be consistent with the kinematics of fixed wing aircraft, we introduce a straightening force that prevents the chain from violating the minimum turn radius of the UAS. Similar to [13], the applied force on any link that causes a turn is designed to ensure that

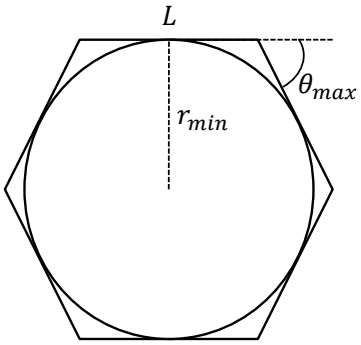


Figure 4.2: θ_{max}, the maximum allowable turn angle to approximate a circle using a discrete chain.

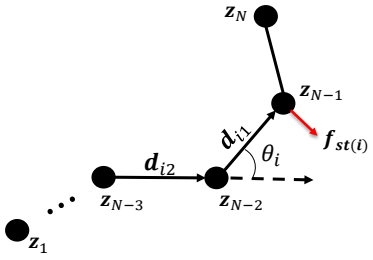


Figure 4.3: The straightening force applied to link N is designed to ensure that $|\theta_N| < \theta_{max}$.

$|\theta_i| < \theta_{max}$. As shown in Figure 4.2, θ_{max} is selected to ensure that

$$\theta_{max} < \frac{L}{r_{min}}, \tag{4.7}$$

and as Figure 4.3 shows, θ_i is the angle between $\mathbf{d}_{i1}$ and $\mathbf{d}_{i2}$, defined as

$$\theta_i = \arccos(\mathbf{d}_{i1}.\mathbf{d}_{i2}),$$

where $\mathbf{d}_{i1} = z_{N-1} - z_{N-2}$ and $\mathbf{d}_{i2} = z_{N-2} - z_{N-3}$, then the straightening force is defined as

$$\mathbf{f}_{st}(i) = \frac{\mu_i(\mathbf{d}_{i1})^{\perp}}{1 + \exp(k(\theta_{max} - \theta_i))}, \tag{4.8}$$

where $i = 1, 2, \ldots, N$, $\mu_i = (N + 1 - i)$ is the upper limit of the straightening force for node i, k is a positive constant that define how closely the logistic function approximates a step function and $(\mathbf{d}_{i1})^{\perp}$ is the orthogonal complement of $\mathbf{d}_{i1}$.

Define $\mathbf{F}_{st} \triangleq \left[\mathbf{f}_{st}^{\top}(1), \mathbf{f}_{st}^{\top}(2), \ldots, \mathbf{f}_{st}^{\top}(N)\right]^{\top}$ to be the vector of all straightening forces applied to the chain and let $\mathbf{F}_{rw}$ be the reward force that is responsible for pushing the chain in the desired direction. Then the total unconstrained force $\mathbf{F}$ of system (4.1) is given by

$$\mathbf{F} = \mathbf{F}_{rw} + \mathbf{F}_{rpOI} + \mathbf{F}_{st}. \tag{4.9}$$

In addition, we want the motion of the link to be constrained by the kinematics of the chain, so that the distance between adjacent links is fixed as shown by Figure 4.4. Let L be the desired length of each link in the chain. These constraints can be written as

$$\begin{aligned} \| \mathbf{z}_2 - \mathbf{z}_1 \|^2 &= L^2 \\ \| \mathbf{z}_3 - \mathbf{z}_2 \|^2 &= L^2 \\ &\vdots \\ \| \mathbf{z}_N - \mathbf{z}_{N-1} \|^2 &= L^2, \end{aligned}$$

or alternatively as

$$\phi(\mathbf{c}) \triangleq \begin{bmatrix} \| \mathbf{z}_2 - \mathbf{z}_1 \|^2 - L^2 \\ \| \mathbf{z}_3 - \mathbf{z}_2 \|^2 - L^2 \\ \vdots \\ \| \mathbf{z}_N - \mathbf{z}_{N-1} \|^2 - L^2 \end{bmatrix} = 0. \tag{4.10}$$

Differentiating the constraints once with respect to time results in the velocity constraint

$$\psi(\mathbf{c}) \triangleq \begin{bmatrix} 2(\mathbf{z}_2 - \mathbf{z}_1)^{\top}(\dot{\mathbf{z}}_2 - \dot{\mathbf{z}}_1) \\ 2(\mathbf{z}_3 - \mathbf{z}_2)^{\top}(\dot{\mathbf{z}}_3 - \dot{\mathbf{z}}_2) \\ \vdots \\ 2(\mathbf{z}_N - \mathbf{z}_{N-1})^{\top}(\dot{\mathbf{z}}_N - \dot{\mathbf{z}}_{N-1}) \end{bmatrix} = 0. \tag{4.11}$$

Differentiating once more results in the acceleration constraint, which can be written in matrix notation as

$$\mathbf{A}(\mathbf{c})\ddot{\mathbf{c}} = \mathbf{b}(\dot{\mathbf{c}}), \tag{4.12}$$

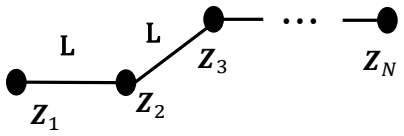


Figure 4.4: N-links chain.

where

$$\mathbf{A}(\mathbf{c}) \triangleq \begin{bmatrix} (\mathbf{z}_2 - \mathbf{z}_1)^\top & 0 & \cdots & 0 \\ -(\mathbf{z}_3 - \mathbf{z}_2)^\top & (\mathbf{z}_3 - \mathbf{z}_2)^\top & \cdots & 0 \\ \vdots & \ddots & & \vdots \\ 0 & \cdots & -(\mathbf{z}_{N-2} - \mathbf{z}_{N-1})^\top & (\mathbf{z}_{N-2} - \mathbf{z}_{N-1})^\top \\ 0 & \cdots & 0 & -(\mathbf{z}_N - \mathbf{z}_{N-1})^\top \end{bmatrix}, \tag{4.13}$$

and

$$\mathbf{b}(\dot{\mathbf{c}}) \triangleq - \begin{bmatrix} (\dot{\mathbf{z}}_2 - \dot{\mathbf{z}}_1)^\top (\dot{\mathbf{z}}_2 - \dot{\mathbf{z}}_1) \\ (\dot{\mathbf{z}}_3 - \dot{\mathbf{z}}_2)^\top (\dot{\mathbf{z}}_3 - \dot{\mathbf{z}}_2) \\ \vdots \\ (\dot{\mathbf{z}}_N - \dot{\mathbf{z}}_{N-1})^\top (\dot{\mathbf{z}}_N - \dot{\mathbf{z}}_{N-1}) \end{bmatrix}. \tag{4.14}$$

Using *Gauss's principal* developed in [124], the equation of motion of system (4.1) subject to constraints (4.12) is given by

$$\ddot{\mathbf{c}} = \mathbf{F} + \mathbf{A}^+(\mathbf{c})(\mathbf{b}(\dot{\mathbf{c}}) - \mathbf{A}(\mathbf{c})\mathbf{F}), \tag{4.15}$$

where $\mathbf{A}^+$ is the pseudo-inverse of $\mathbf{A}$ [124]. As shown in [88], when solving equations of this type numerical error may cause the constraints $\phi(\mathbf{c})$ and $\psi(\mathbf{c})$ to drift from zero. Therefore, adding two additional terms forces the constrained accelerations to descend along the gradient of the constraints until they are no longer violated. Accordingly, Eq. (4.15) is modified as

$$\ddot{\mathbf{c}} = \mathbf{F} + \mathbf{A}^+(\mathbf{c})(\mathbf{b}(\dot{\mathbf{c}}) - \mathbf{A}(\mathbf{c})\mathbf{F}) - \gamma_p \frac{\partial \phi^\top}{\partial \mathbf{c}} \phi - \gamma_v \frac{\partial \psi^\top}{\partial \mathbf{c}} \psi, \tag{4.16}$$

where γ_p and γ_v are positive constants. A damping term is added to reduce oscillations and by combining the unconstrained force terms of eq. (4.9), eq. (4.15) becomes

$$\begin{aligned}\ddot{\mathbf{c}} = \mathbf{F}_{rw} + \mathbf{F}_{rpOI} + \mathbf{F}_{st} + \mathbf{A}^{+}(\mathbf{c})(\mathbf{b}(\dot{\mathbf{c}}) - \mathbf{A}(\mathbf{c})\mathbf{F}) \\ - \gamma_p \frac{\partial \phi^{\top}}{\partial \mathbf{c}} \phi - \gamma_v \frac{\partial \psi^{\top}}{\partial \mathbf{c}} \psi + \gamma_d \dot{\mathbf{c}}.\end{aligned} \tag{4.17}$$

Similar to [13], when using the chain to avoid collisions in real-time, the first link serves as waypoint for the UAS to follow, and is not allowed to move. When the UAS nears the end of the first link, the link is removed from the chain and a new link is added to the end with the same direction as the last link in the chain. The first link that then comprise the beginning of the chain is fixed as the new waypoint, so that the UAS always has an unchanging waypoint to follow while the remainder of the chain continuously adapts to changing virtual forces.

This discontinuity in motion can be eliminated by further introducing additional constraints. In the following, we develop dynamic equations for a one-link chain and then generalize our findings for an arbitrary N-links. Figure 4.5 depicts the one-link chain system. The one-link chain is modeled as two point masses m_1 and m_2, with unit mass, connected to each other by a weightless rod. Let the two masses whose position in the 2-dimensional space specified by their Cartesian coordinate $\mathbf{z}_i = (x_i, y_i)^{\top} \in \mathbb{R}^2$, $i = 1, 2$. The system is subject to the following constraints:

1. The motion of the one-link chain to be constrained by the kinematics of a chain such that the distance between the two masses, m_1 and m_2 is constrained to be a fixed quantity. Let us say that the distance between the pair of masses of constant length, L. Now, the two masses can not move independently of each other; for at any instant of time the distance between them is fixed and equal to L. This constraint can be written as

$$\|\mathbf{z}_2 - \mathbf{z}_1\|^2 = L^2. \tag{4.18}$$

2. Let $\mathbf{V}_{m_1} = (V_{m1_x}, V_{m1_y})^{\top} \in \mathbb{R}^2$ and $\mathbf{V}_{m_2} = (V_{m2_x}, V_{m2_y})^{\top} \in \mathbb{R}^2$ be the velocity of mass m_1 and m_2, respectively. And let $\mathbf{r}_{2/1} = \mathbf{z}_2 - \mathbf{z}_1 \in \mathbb{R}^2$ be the position vector directed

from m_1 to m_2, or equivalently the position of m_2 with respect to m_1. The velocity $\mathbf{V}_{m_1}$ is constrained to be always directed towards m_2, then the constraint equation would be expressed as

$$\mathbf{V}_{m_1} \times \mathbf{r}_{2/1} = \mathbf{0} \Leftrightarrow \dot{\mathbf{z}}_1 \times \mathbf{r}_{2/1} = \mathbf{0}, \tag{4.19}$$

where $\times$ denotes the cross product operator of two vectors.

3. The weightless rod is pined at m_1 such that the rotated motion of the rod is constrained by

$$|\theta_1| \leq \theta_{max}, \tag{4.20}$$

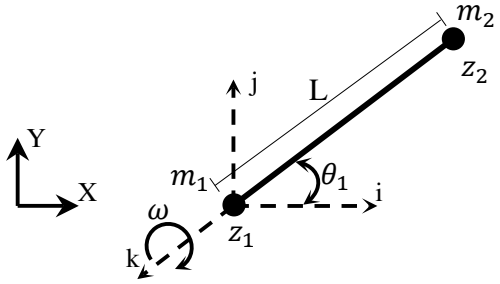


Figure 4.5: One-link chain system.

If the one-link chain is unconstrained, then the dynamic equations describing its motion are

$$\ddot{\mathbf{z}}_1 = \mathbf{f}_1,$$
$$\ddot{\mathbf{z}}_2 = \mathbf{f}_2,$$

where $\mathbf{f}_1, \mathbf{f}_2 \in \mathbb{R}^2$ are the applied or the impressed forces acting on m_1 and m_2, respectively. Alternatively, defining $\mathbf{c} \triangleq (\mathbf{z}_1^\top, \mathbf{z}_2^\top)^\top \in \mathbb{R}^2$ and $\mathbf{F}^a \triangleq (\mathbf{f}_1^\top, \mathbf{f}_2^\top)^\top \in \mathbb{R}^2$ gives

$$\ddot{\mathbf{c}} = \mathbf{F}^a. \tag{4.21}$$

However, the dynamic equation defined in system (4.21) is subject to the set of constraints described by Eqs. (4.18)-(4.20). Thus, the constrained equation of motion for the one-link chain system becomes

$$\ddot{\mathbf{c}} = \mathbf{F}^a + \mathbf{F}^c, \tag{4.22}$$

where the additional term $\mathbf{F}^c = \sum_{i=1}^{n} \mathbf{F}_i^c, n = 3$ is the sum of the additional forces on the one-link system caused by the fact that they have to satisfy the constraint Eqs. (4.18)-(4.20), and $\mathbf{F}_i^c$ is the constrained force due to the constraint i. The main objective at this step is to determine $\ddot{\mathbf{c}}$ of the constrained system at each instant of time t, in the presence of the specified constraints, and to compute the constrained forces vector $\mathbf{F}^c$ that engender motion that is satisfied with the constraints. Before to proceed further, we will assume that the applied force $\mathbf{F}^a$ acting on the system is known and that the position and the velocities of the two masses at the initial time instant satisfy the constraints (4.18)- (4.20). We will assume also that the constrained force, $\mathbf{F}_3^c$ caused by constraint described in Eq. (4.20) can be designed beforehand to be compatible with this constraint. Both unit masses, m_1 and m_2 are common to the same body, so their velocities are related by

$$\mathbf{V}_{m_1} = \mathbf{V}_{m_2} - \omega \times \mathbf{r}_{2/1}, \tag{4.23}$$

where ω is the angular velocity of the weightless rod about m_1. The position vector $\mathbf{r}_{2/1}$ can be expressed in terms of the generalized coordinates as

$$\mathbf{r}_{2/1} = L\cos\theta_1\hat{i} + L\sin\theta_1\hat{j}. \tag{4.24}$$

Substitution of Eq. (4.24) into (4.23) results

$$\dot{\mathbf{z}}_1 = (\dot{x}_2 + \dot{\theta_1}L\sin\theta)\hat{i} + (\dot{y}_2 + \dot{\theta_1}L\cos\theta_1)\hat{j}. \tag{4.25}$$

Also, substitution of Eqs. (4.24) and (4.25) into (4.19) leads to an equivalent expression of constraint (4.19)

$$\sin\theta_1\dot{x}_2 - \cos\theta_1\dot{y}_2 + L\dot{\theta_1} = 0. \tag{4.26}$$

Differentiating both constraints Eqs. (4.18) and (4.26) twice with respect to time results

$$(\mathbf{z}_2 - \mathbf{z}_1)^\top (\ddot{\mathbf{z}}_2 - \ddot{\mathbf{z}}_1) = -(\dot{\mathbf{z}}_2 - \dot{\mathbf{z}}_1)^\top (\dot{\mathbf{z}}_2 - \dot{\mathbf{z}}_1) \tag{4.27}$$

and

$$\sin\theta_1 \ddot{x}_2 - \cos\theta_1 \ddot{y}_2 + L\ddot{\theta}_1 = -(\cos\theta_1 \dot{x}_2 \dot{\theta}_1 + \sin\theta_1 \dot{y}_2 \dot{\theta}_1). \tag{4.28}$$

Combine Eqs. (4.27) and (4.28), involving the accelerations, in matrix form $\mathbf{A}(\mathbf{c})\ddot{\mathbf{c}} = \mathbf{b}(\mathbf{c}, \dot{\mathbf{c}})$ as

$$\begin{bmatrix} -(\mathbf{z}_2 - \mathbf{z}_1)^\top & (\mathbf{z}_2 - \mathbf{z}_1)^\top & 0 \\ 0 & \mathbf{R}_{\theta_1} & L \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{z}}_1 \\ \ddot{\mathbf{z}}_2 \\ \ddot{\theta} \end{bmatrix} = - \begin{bmatrix} (\dot{\mathbf{z}}_2 - \dot{\mathbf{z}}_1)^\top (\dot{\mathbf{z}}_2 - \dot{\mathbf{z}}_1) \\ \cos\theta_1 \dot{x}_2 \dot{\theta}_1 + \sin\theta_1 \dot{y}_2 \dot{\theta}_1 \end{bmatrix}, \tag{4.29}$$

where $\mathbf{R}_{\theta_1} \triangleq \begin{bmatrix} \sin\theta_1 & -\cos\theta_1 \end{bmatrix}$.

The constraints Eq. 4.29 can be generalized for N-links as

$$\mathbf{A}(\mathbf{c}) \triangleq \begin{bmatrix} -(\mathbf{z}_2 - \mathbf{z}_1)^\top & (\mathbf{z}_2 - \mathbf{z}_1)^\top & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & -(\mathbf{z}_3 - \mathbf{z}_2)^\top & (\mathbf{z}_3 - \mathbf{z}_2)^\top & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & & \cdots & \vdots & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & -(\mathbf{z}_N - \mathbf{z}_{N-1})^\top & (\mathbf{z}_N - \mathbf{z}_{N-1})^\top & 0 & 0 & \cdots & 0 \\ 0 & \mathbf{R}_{\theta_1} & 0 & \cdots & 0 & L & 0 & \cdots & 0 \\ 0 & 0 & \mathbf{R}_{\theta_2} & \cdots & 0 & 0 & L & \cdots & 0 \\ \vdots & \ddots & & \cdots & \vdots & 0 & 0 & \cdots & 0 \\ 0 & 0 & & \cdots & \mathbf{R}_{\theta_{N-1}} & 0 & 0 & \cdots & L \end{bmatrix}, \tag{4.30}$$

and

$$\mathbf{b}(\dot{\mathbf{c}}) \triangleq - \begin{bmatrix} (\dot{\mathbf{z}}_2 - \dot{\mathbf{z}}_1)^\top (\dot{\mathbf{z}}_2 - \dot{\mathbf{z}}_1) \\ (\dot{\mathbf{z}}_3 - \dot{\mathbf{z}}_2)^\top (\dot{\mathbf{z}}_3 - \dot{\mathbf{z}}_2) \\ \vdots \\ (\dot{\mathbf{z}}_N - \dot{\mathbf{z}}_{N-1})^\top (\dot{\mathbf{z}}_N - \dot{\mathbf{z}}_{N-1}) \\ (\cos\theta_1 \dot{x}_2 \dot{\theta}_1 + \sin\theta_1 \dot{y}_2 \dot{\theta}_1) \\ (\cos\theta_1 \dot{x}_3 \dot{\theta}_2 + \sin\theta_2 \dot{y}_3 \dot{\theta}_2) \\ \vdots \\ (\cos\theta_{N-1} \dot{x}_N \dot{\theta}_{N-1} + \sin\theta_{N-1} \dot{y}_N \dot{\theta}_{N-1}) \end{bmatrix}, \tag{4.31}$$

Then, using *Gauss's principal* the equation of motion of system (4.22) subject to constraints $\mathbf{A}(\mathbf{c})\ddot{\mathbf{c}} = \mathbf{b}(\mathbf{c}, \dot{\mathbf{c}})$ is given by

$$\ddot{\mathbf{x}} = \mathbf{F} + \mathbf{A}^{+}(\mathbf{x})(\mathbf{b}(\dot{\mathbf{x}}) - \mathbf{A}(\mathbf{x})\mathbf{F}), \tag{4.32}$$

where $\mathbf{A}^{+}$ is the pseudo-inverse of $\mathbf{A}$. Similar to Eq. (4.9) $\mathbf{F} = \mathbf{F}_{rw} + \mathbf{F}_{rpOI} + \mathbf{F}_{st}$. Unlike Eq. (4.4), the equation of motion of system (4.32) solves the problem discontinuity in motion.

4.1.1 Chain-Based Collision Avoidance with Uncertainties

The collision avoidance algorithm presented in section 4.1 does not consider uncertainties associated with estimates of the intruder's states. In this section we develop a collision avoidance algorithm that takes into account the uncertainties of the estimates. We will assume that the estimated states of the intruder aircraft include both position and velocity. We also assume that an estimate of the accuracy of these state estimates is also provided and is quantified in an error covariance matrix. To take into account the associated uncertainty with these estimates, a statistical distance is defined in lieu of the normal Euclidean distance computed between the intruder's chain nodes and the ownship's chain nodes. A Kalman Filter (KF) stage is introduced to the collision avoidance algorithm to receive the states estimate (position & velocity) and covariance at each time step from the 'sense' stage. The KF then propagates the state estimates and covariance N-steps into future, where N is the number of chain nodes, shown with solid red line in Figure 4.6. The intruder's chain nodes at $(t = T_1, t = T_2 \ldots t = T_N)$ represents an N-step position prediction of the intruder aircraft. At the next time step $(t = T_1)$, collision avoidance algorithm receives a new state estimate and covariance to update the the predicted intruder's location at $(t = T_1)$ and propagates the intruder's chain nodes at $(t = T_2, t = T_3 \ldots t = T_{N+1})$, shown with dashed green line. The N-link intruder chain is represented as

$$\mathbf{c}_I(k) = [\mathbf{z}_{I1}^{\top}(k+1|k),\ \mathbf{z}_{I2}^{\top}(k+2|k), \ldots, \mathbf{z}_{IN}^{\top}(k+N|k)]^{\top},$$

where k is the time index, $(\cdot)(k)$ indicates the value of the state at time k and $(\cdot)(k+N|k)$ is the predicted value of the state at time $k+N$ given observations at time k. The associated covariance that represents the prediction uncertainty of each node is given by

$$\mathbf{Q}_I(k) = [Q_{I1}(k+1|k),\ Q_{I2}(k+2|k), \ldots, Q_{IN}(k+N|k)]^\top,$$

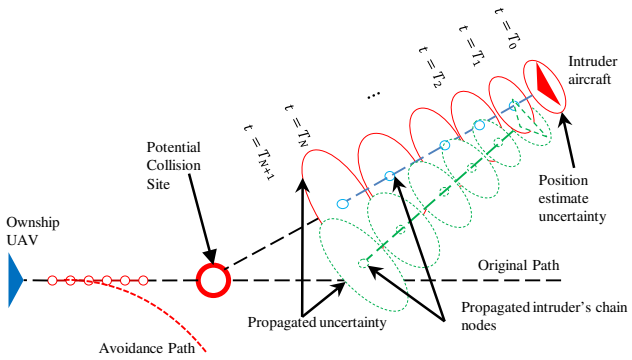


Figure 4.6: Intruder's chain Propagation.

Definition. The statistical distance or Mahalanobis distance between two points $x = (x_1, \ldots, x_p)^T$ and $y = (y_1, \ldots, y_p)^T$ in the p-dimensional space $\mathbb{R}_p$ is defined as

$$d_{\mathrm{M}}(x, y) = \sqrt{(x-y)^\top Q^{-1}(x-y)}, \tag{4.33}$$

where Q is the covariance matrix that represents the measurement uncertainty of both variables x and y [52, 85]. $d_{\mathrm{M}}(x, 0) = \|x\|_{\mathrm{M}} = \sqrt{x^\top Q^{-1} x}$ is the Mahalanobis-norm (M-norm) of x.

The distance $\|\mathbf{d}_{OI}(m,n)\|_{\mathrm{M}}$ is then defined as

$$\|\mathbf{d}_{OI}(m,n)\|_{\mathrm{M}} = \sqrt{\Big(\mathbf{z}_{Om} - \mathbf{z}_{In}(k+n|k)\Big)^\top Q_{In}^{-1}(k+n|k)\Big(\mathbf{z}_{Om} - \mathbf{z}_{In}(k+n|k)\Big)}. \tag{4.34}$$

To account for uncertainty in the estimate of the intruder's states, the repelling force given by Eq. (4.4) is replaced by

$$f_{rp}(m,n) = \begin{cases} \hat{\mathbf{d}}_{OI}(m,n)\gamma_{rp1}e^{(-\gamma_{rp2}\|\mathbf{d}_{OI}(m,n)\|_{\mathrm{M}})} & if\ |m-n| < k\ and \\ & \|\mathbf{d}_{OI}(m,n)\| < d_{max} \\ 0 & Otherwise. \end{cases} \tag{4.35}$$

4.2 Local-Level Collision Avoidance

If the collision-detection metrics are derived from the relative position and velocity vectors between conflicting aircraft, then it is convenient to develop collision avoidance logic using a relative coordinate system. In this relative coordinate system, the ownship is fixed at the center of the coordinate system, and the intruder is located at a relative position $\mathbf{p}_r$ and moves with a relative velocity $\mathbf{v}_r$ with respect to the ownship. An advantage of this approach is that collision avoidance is inherently a local phenomenon and can be more naturally represented in the local coordinates than the global coordinates.

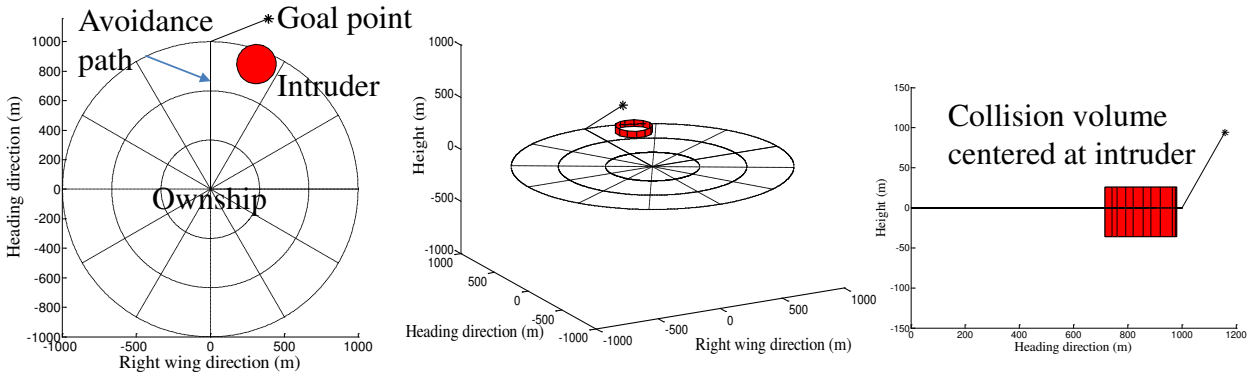


Figure 4.7: Local-level reference frame.

We call this translated coordinate system the local-level frame because the environment is mapped to the unrolled, unpitched local coordinates where the ownship is stationary at the center. The local-level reference frame is depicted in Figure 4.7, where the origin of the local-level frame is the current position of the ownship. In this configuration the x-axis

points out the nose of the unpitched airframe, the y-axis points points out the right wing of the unrolled airframe, and the z-axis points down forming a right handed coordinate system. The local-level coordinates also provides a convenient method for defining a collision between two aircraft. In the following discussion, we assume that the collision volume is a cylinder of radius d_s and height h_s centered at the current location of the intruder. If the origin of the local-level frame penetrates the collision volume around the intruder, a collision is said to have occurred.

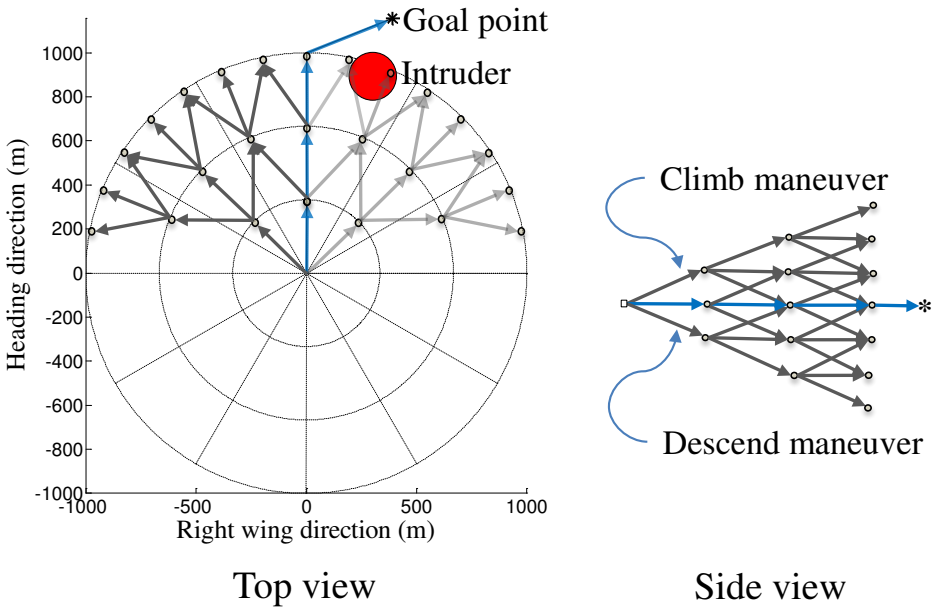


Figure 4.8: Discretized local-level reference workspace. The three concentric circles represent three maneuvers points.

As shown in Figure 4.8, the detection region is divided into concentric circles that represent maneuvers points at increasing range from the ownship. The radius of the outmost circle represents the sensor detection range. Let the region in the space covered by the sensor be called the workspace. Then, we discretize the workspace and construct a weighted graph where the edges represent potential maneuvers, and the weights represent the collision risk and maneuver cost. The workspace is discretized by using a cylindrical grid, and the ownship is commanded to move along the edges of the grid. The result is a directed weighted graph that can be described by the tuple $\mathcal{G}(\mathcal{N}, \mathcal{E}, \mathcal{C})$, where $\mathcal{N}$ is a finite nonempty set of nodes,

and $\mathcal{E}$ is a collection of ordered pairs of distinct nodes from $\mathcal{N}$ such that each pair of nodes in $\mathcal{E}$ is called a directed edge or link, and $\mathcal{C}$ is the cost associated with traversing each edge.

The path is then constructed from a sequence of non repeated nodes $(\mathbf{n}_1, \mathbf{n}_2, \cdots, \mathbf{n}_N)$ such that each consecutive pair $(\mathbf{n}_i, \mathbf{n}_{i+1})$ is an edges in $\mathcal{G}$. As depicted in Figure 4.9, we let $\mathcal{L}_l$, $l = 1, 2, \cdots, m$ be the lth level curve of the concentric circles. Let r be the radius of the innermost circle and the detection range, d_r be the radius of the outermost circle such that $d_r = mr$. Let us assume that the level curves are equally partitioned or discretized by a number of nodes such that any node on the lth level curve, $\mathcal{L}_l$ connects to k nodes in the next level curve in the forward direction, i.e. along the positive direction of the heading axis as shown in Figure 4.8. Additionally, we assume that only nodes along the positive direction of the heading axis $x = 0$ connects to nodes in the vertical plane. This assumption allows the aircraft to climb or descend by connecting to nodes in the vertical plane as shown in Figure 4.8. Using the notation $|\mathcal{A}|$ to denote the cardinality of the discrete set $\mathcal{A}$, let the first level curve of the innermost circle be discretized into $|\mathcal{L}_1| = k+2$ nodes including nodes in the vertical plane. Then, number of nodes in the lth level curve is given by

$$|\mathcal{L}_l| = \begin{cases} k+2 & \text{if } l = 1, \\ 2\,|\mathcal{L}_{l-1}| + 2l + 1 & \text{if } l = 2, 3, \cdots, m, \end{cases}$$

where the total number of nodes is $|\mathcal{N}| = \sum_{l=1}^{m} |\mathcal{L}_l|$. Assuming that the start node is located at the origin of the local-level coordinates and given that $k = 3$, i.e. allowing the ownship to move straight, right or left, then the total number of nodes in the graph including the start and destination node is given by

$$|\mathcal{N}| = \left(\sum_{l=1}^{m+1} 2^l + 2l - 3 \right) + 1.$$

Assuming that the ownship travels between the nodes with constant velocity and climb rate, the location of the ith node at the lth level curve, $\mathbf{n}_{i,l}$ in the horizontal plane of the graph is given by

$$\mathbf{n}_{i,l} = [lr \sin \psi_j^{\mathcal{L}_l}, lr \cos \psi_j^{\mathcal{L}_l}, 0]^\top,$$

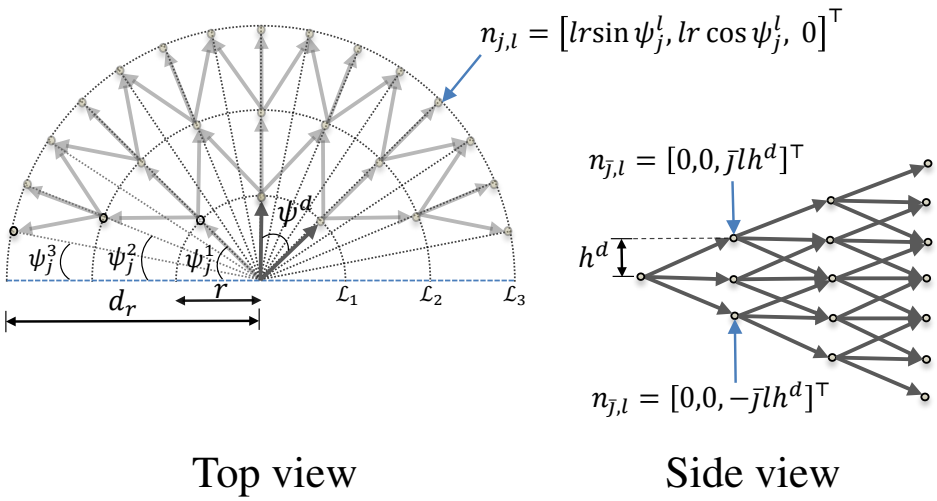


Figure 4.9: Nodes location in the local-level reference frame.

where $\psi_j^l = \frac{j\psi^d}{2^{(l-1)}}$, and $j = \{-\frac{|\mathcal{L}_l|-1}{2}, -\frac{|\mathcal{L}_l|-1}{2}+1, \cdots, \frac{|\mathcal{L}_l|-1}{2}-1, \frac{|\mathcal{L}_l|-1}{2}\}$. In the vertical plane the location of nodes is $\mathbf{n}_{\bar{j},l} = [0, 0, \pm\bar{j}lh^d]^\top$, where $\bar{j} = \{1, 2, \cdots, l\}$.

As shown in Figure 4.9, ψ^d, and h^d are the allowed heading, and altitude change at each step. For example, if $\psi^d = \pi/4$, $h^d = 50$ m, $r = 500$ m, $k = 3$ and $|\mathcal{L}_1| = 5$, then we have $j = \{-1, 0, 1\}$, $\bar{j} = \{-1, 1\}$, $\psi_j^1 = \{-\pi/4, 0, -\pi/4\}$, and the locations of nodes at $\mathcal{L}_1$ in the horizontal plane are $\{(-500\sin\pi/4, 500\cos\pi/4, 0)^\top, (0, 500, 0)^\top, (500\sin\pi/4, 500\cos\pi/4, 0)^\top\}$, and in the vertical plane are $\{(0, 0, 50)^\top, (0, 0, -50)^\top\}$. The nodes on the graph can be thought of as predicted locations of the ownship over a look-ahead time window.

In assigning a cost to each edge in the resulting graph, it is important to note that the main priority of an ownship under distress is to maneuver to avoid predicted collisions. The cost associated with traveling along an edge is a function of the edge length and the collision risk. The cost associated with the length of the edge $e_{i,i+1}$ that connects between the consecutive pair nodes $(\mathbf{n}_i, \mathbf{n}_{i+1})$ is simply the Euclidean distance between the nodes $\mathbf{n}_i$ and $\mathbf{n}_{i+1}$, expressed as

$$C_L(e_{i,i+1}) = \|\mathbf{n}_{i+1} - \mathbf{n}_i\|. \tag{4.36}$$

The collision cost for traveling along an edge is determined if at any future time instant the future position of the ownship along that edge is inside the collision volume. An exact collision cost computation would involve the integration of the cost along the edge. A simpler approach involves calculating the collision risk cost at several locations

along the edge taking into account the predicated locations of the intruder over look-ahead time window. Assuming a constant velocity model, a linear extrapolation of the current position and velocity of the detected intruders is computed at evenly spaced time instants over the look-ahead time window. The look-ahead time interval is then discretized into several discrete time instants. At each discrete time instant all candidate locations of the ownship along each edge is checked to determine whether it is or will be inside the collision volumes that are centered around the propagated locations of the intruders. For this work, the collision risk cost is calculated at three points along each edge in $\mathcal{G}$. If v_o is the speed of the ownship, then the distance along an edge is given by $v_o T$ where $T = r/v_o$. The three points are computed as

$$\mathbf{p}_1 = \mathbf{n}_i + v_o T_s \frac{\mathbf{n}_{i+1} - \mathbf{n}_i}{\|\mathbf{n}_{i+1} - \mathbf{n}_i\|}, \tag{4.37}$$

$$\mathbf{p}_2 = \mathbf{p}_1 + v_o T_s \frac{\mathbf{n}_{i+1} - \mathbf{n}_i}{\|\mathbf{n}_{i+1} - \mathbf{n}_i\|}, \tag{4.38}$$

$$\mathbf{p}_3 = \mathbf{p}_2 + v_o T_s \frac{\mathbf{n}_{i+1} - \mathbf{n}_i}{\|\mathbf{n}_{i+1} - \mathbf{n}_i\|}, \tag{4.39}$$

where $T_s = T/3$. Let the relative horizontal and vertical position of the intruder with respect to the ownship at the current time t be $\mathbf{p}_r(t)$, and $p_{r_z}(t)$, respectively. Define the collision volume as

$$\mathcal{C}(\mathbf{p}_r(t)) = \{d \in \mathbb{R}^2 : \|\mathbf{p}_r(t) - d\| \leq d_s \,\text{and}\, h \in \mathbb{R} : \left|p_{r_z} - h\right| \leq h_s/2\}.$$

For the ease of exposition, we let $\mathbf{p}_{r_{3D}}(t)$, $\mathbf{v}_{r_{3D}}(t)$ be the 3D relative position and velocity of the intruder with respect to the ownship in the relative coordinate system. The risk of a collision event is computed over the look-ahead time window $\tau \in [t, t + mT]$. The predicted locations of each detected intruder over time horizon T at three discrete time

samples T_s are

$$\mathbf{p}_{r_{3D}}(t+(1+3(l-1))T_s) = \mathbf{p}_{r_{3D}}(t) + \mathbf{v}_{r_{3D}}(t)(1+3(l-1))T_s, \tag{4.40}$$

$$\mathbf{p}_{r_{3D}}(t+(2+3(l-1))T_s) = \mathbf{p}_{r_{3D}}(t) + \mathbf{v}_{r_{3D}}(t)(2+3(l-1))T_s, \tag{4.41}$$

$$\mathbf{p}_{r_{3D}}(t+(3+3(l-1))T_s) = \mathbf{p}_{r_{3D}}(t) + \mathbf{v}_{r_{3D}}(t)(3+3(l-1))T_s. \tag{4.42}$$

In Eqs. (4.40)- (4.42), if $e_{i,i+1}$ is the current edge being evaluated then the node $\mathbf{n}_{i+1}$ determines the value of l. In other words, if $\mathbf{n}_{i+1} \in \mathcal{L}_1$ then we have $l = 1$. For example, if we are to compute the three points along the edge $e_{1,2}$ in Eqs (4.37)- (4.39) then $\mathbf{n}_2 \in \mathcal{L}_1$ and $l = 1$. Using the definition of the binary cost function the collision risk cost associated with the $e_{i,i+1}$ edge with respect to each detected intruder is given by the expression

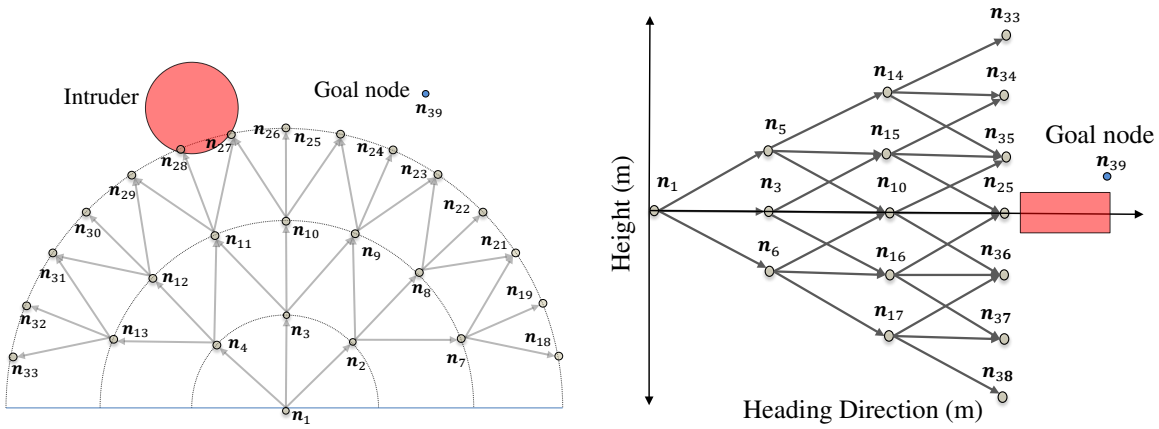
$$C_{col}(\text{int}, e_{i,i+1}) = \begin{cases} \infty & \text{if any of } \mathbf{p}_1, \mathbf{p}_2, \text{ or } \mathbf{p}_3 \in \mathcal{C}(\mathbf{p}_{r_{3D}}(t+(\ell+3(l-1))T_s)), \\ 0 & \text{otherwise,} \end{cases} \tag{4.43}$$

where $\ell = \{1, 2, 3\}$. The collision risk assigned to any edge that leads to a collision in Eq. (4.43) is ∞ or the maximum allowable cost, thus basically eliminating that edge and the path passing through it. The total collision risk associated with the ith edge with regards to all intruders is given by

$$C_{col}(e_{i,i+1}) = \sum_{int=1}^{M} C_{col}(\text{int}, e_{i,i+1}), \tag{4.44}$$

where M is the number of detected intruders.

Figure 4.11 shows snap shots of the propagated locations of a detected intruder over a look-ahead time window. Figures 4.11(a) and 4.11(b) show the candidate locations of the ownship at three points along each edge over the first time interval T both in horizontal and vertical plane. Clearly, there is no intersection between these points and the predicted locations of the intruder over the same interval. The cost assigned to these edges is zero according to Eq (4.44). Next, all candidate locations of the ownship along each edge over the second time interval $2T$ are investigated. As shown in Figure 4.11(c), edges $e_{2,7}$, $e_{2,8}$ and



(a) Top view: location and index of nodes.

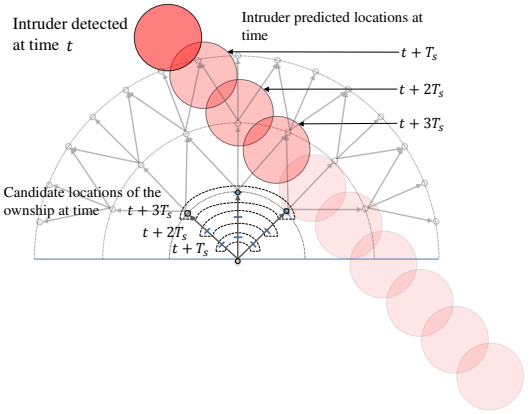
(b) Side view: location and index of nodes.

Figure 4.10: Local-level map. In this example we have $k = 3$ and $l = 3$. The total number of nodes is 39.

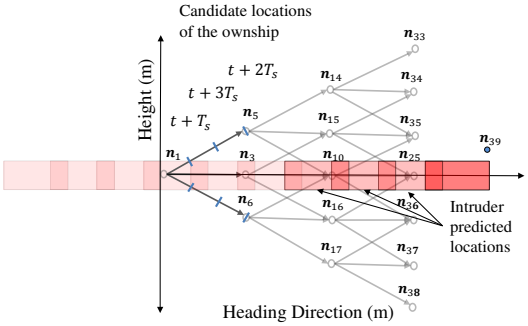
$e_{2,9}$ intersect with the predicted intruder location at time $t + 4T_S$ and $t + 5T_S$, respectively. Similarly, edges $e_{3,15}$ and $e_{3,16}$ in the horizontal plane intersect with the predicted intruder location at time $t+4T_S$ as shown in Figure 4.11(d). Accordingly, the maximum allowable cost will be assigned to these edges, hence eliminating these edges and the path passing through them. As shown in Figure 4.11(e) and 4.11(f), all the candidate locations of the ownship over the time interval $3T$ do not intersect with the predicted locations of the intruder. Therefore, by the time the ownship will reach these edges the intruder will be leaving the map and consequently a cost of zero is assigned to edges belongs to the third level curve $\mathcal{L}_3$.

To provide an increased level of robustness, an additional cost is used to penalize edges close to the collision volume even if they are not within the collision volume. At each discrete time instant the distances from the propagated locations of the intruders to all candidate locations of the ownship are computed at that time instant. The cost of collision threat along each edge is then the sum of the reciprocal of the associated distances to each intruder.

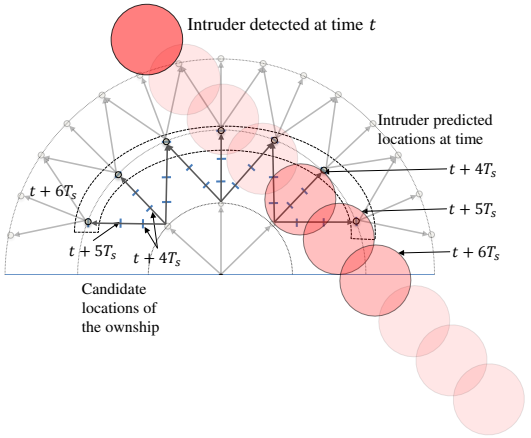
$$C_{th}(\text{int}, e_{i,i+1}) = \frac{1}{d_1} + \frac{1}{d_2} + \frac{1}{d_3}. \tag{4.45}$$



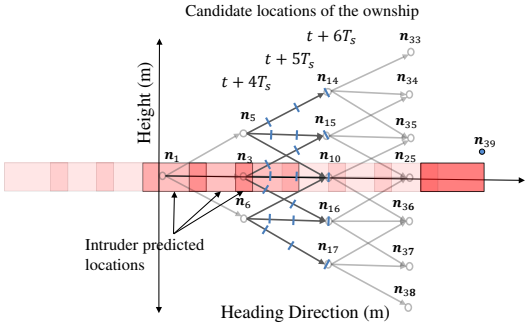
(a) Top view: Predicted locations of intruder (less transparent circles), and candidate locations of ownship.



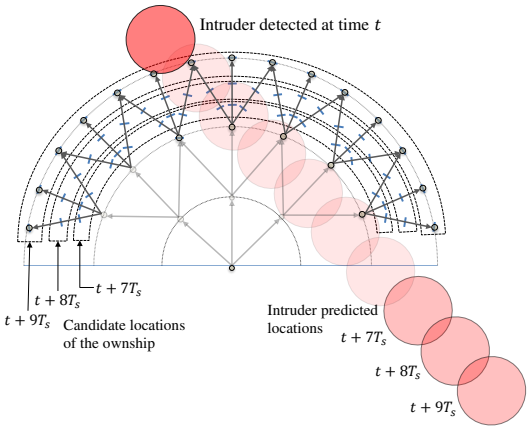
(b) Side view: Predicted locations of intruder (less transparent rectangles), and candidate locations of ownship.



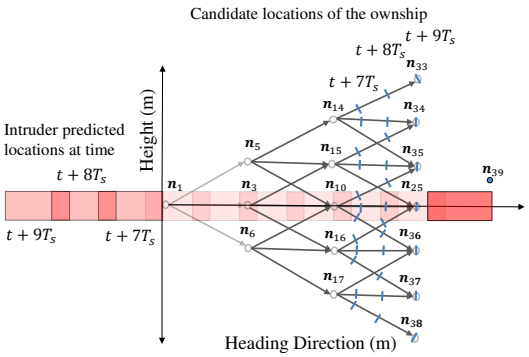
(c) Predicted locations of intruder and candidate locations of ownship over time window $(t + 4T_s, t + 6T_s)$.



(d) Time window $(t + 4T_s, t + 6T_s)$.



(e) Time window $(t + 7T_s, t + 9T_s)$.



(f) Time window $(t + 7T_s, t + 9T_s)$.

Figure 4.11: Example illustrating the steps to compute the collision risk. In this example we have $k = 3$ and $l = 3$.

where d_1, d_2, and d_3 are given by

$$d_1 = \left\| p_1 - \mathbf{p}_{r_{3D}}(t + (1 + 3(l-1))T_s) \right\|,$$
$$d_2 = \left\| p_2 - \mathbf{p}_{r_{3D}}(t + (2 + 3(l-1))T_s) \right\|,$$
$$d_3 = \left\| p_3 - \mathbf{p}_{r_{3D}}(t + (3 + 3(l-1))T_s) \right\|,$$

and the total collision risk cost associated with the ith edge with regards to all intruders is given by

$$C_{th}(e_{i,i+1}) = \sum_{int=1}^{M} C_{th}(\text{int}, e_{i,i+1}). \tag{4.46}$$

For example, the edges $e_{1,2}$, $e_{1,3}$, $e_{1,4}$, $e_{1,5}$ and $e_{1,6}$ shown in in Figure 4.11(a) are not intersecting with the predicated locations of the intruder over the first time interval, yet they will be penalized based on their distances to the predicated locations of the intruder according to Eq (4.46). Note that edge $e_{1,2}$ will have greater cost as it is the closest to the intruder among other candidate edges.

Another objective of a path planning algorithm is to minimize the deviation from the original path i.e. the path the ownship was following before it detected a collision. Generally, the path is defined as an ordered sequence of waypoints $\mathcal{W} = \mathbf{W}_1, \mathbf{W}_2, \cdots . \mathbf{W}_f$, where $\mathbf{W}_i = (w_{n,i}, w_{e,i}, w_{d,i})^\top \in \mathbb{R}^3$ is the north-east-down location of the ith waypoint in a globally known NED reference frame. The transformation from the global frame to the local-level frame is given by

$$\mathbf{W}_i^b = \mathcal{R}_g^b(\psi_o)\mathbf{W}_i,$$

where

$$\mathcal{R}_g^b(\psi_o) = \begin{pmatrix} \cos\psi_o & \sin\psi_o & 0 \\ -\sin\psi_o & \cos\psi_o & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

where ψ_o is the heading angle of the ownship. Let $\mathbf{W}_s$ be the location of the ownship at the current time t and $\mathbf{W}_f \in \mathcal{W}$ be the next waypoint in the local-level frame. Assuming a straight-line segment between the waypoints $\mathbf{W}_s$ and $\mathbf{W}_f$, then any point on this segment can be described as $\mathfrak{L}(\varrho) = (1 - \varrho)\mathbf{W}_s + \varrho\mathbf{W}_f$ where $\varrho \in [0, 1]$, and the minimum distance

between an arbitrary node $\mathbf{n}_i$ in $\mathcal{G}$ can be expressed by [16]

$$D(\mathbf{W}_s, \mathbf{W}_f, n_i) \triangleq \begin{cases} D(\varrho^*), & \text{if } \varrho^* \in [0,1], \\ \|\mathbf{n}_i - \mathbf{W}_s\|, & \text{if } \varrho^* < 0, \\ \|\mathbf{n}_i - \mathbf{W}_f\|, & \text{if } \varrho^* > 1, \end{cases} \tag{4.47}$$

where

$$D(\varrho^*) = \sqrt{\|\mathbf{n}_i - \mathbf{W}_s\|^2 - \frac{\Big((\mathbf{W}_s - \mathbf{n}_i)^\top (\mathbf{W}_s - \mathbf{W}_f)\Big)^2}{\|\mathbf{W}_s - \mathbf{W}_f\|^2}},$$

and

$$\varrho^* = \frac{(\mathbf{W}_s - \mathbf{n}_i)^\top (\mathbf{W}_s - \mathbf{W}_f)}{\|\mathbf{W}_s - \mathbf{W}_f\|^2}.$$

Then, the cost that penalize the deviation of an edge in $\mathcal{G}$ from the nominal path is given by

$$C_{dev}(e_{i,i+1}) = D(\mathbf{W}_s, \mathbf{W}_f, n_i). \tag{4.48}$$

An additional cost can be also added to favor edges in the horizontal plane over those in the vertical plane. Since the positive direction of the y-axis in the local-level frame is the right-wing direction, it is convenient to define right and left maneuvers as the positive and the negative directions along the right-wing direction axis, respectively. Then, a cost can be added to penalize edges that violate right-of-way rules. Let $\vec{\mathbf{e}}_i \triangleq \mathbf{n}_{i+1} - \mathbf{n}_i$ be the direction vector of the edge $e_{i,i+1}$ in $\mathcal{G}$, where $\mathbf{n}_i \triangleq (x_i, y_i, z_i)^\top \in \mathbb{R}^3$ is the location of ith node in the local-level reference frame. Let the direction vector $\vec{\mathbf{e}}_i$ be expressed as $\vec{\mathbf{e}}_i = (e_{i_x}, e_{i_y} e_{i_z})^\top \in \mathbb{R}^3$. We define $\mathbf{E} \triangleq (e_{i_x}, L, R, e_{i_z})^\top \in \mathbb{R}^4$, where e_{i_x} and e_{i_z} are the x and the z components of $\vec{\mathbf{e}}_i$. The y-component of $\vec{\mathbf{e}}_i$ is decomposed into two components left L and right R which are defined by

$$L, R \triangleq \begin{cases} L = e_{i_y}, R = 0 & \text{if } e_{i_y} \leq 0, \\ L = 0, R = e_{i_y} & \text{if } e_{i_y} > 0. \end{cases}$$

We define the maneuvering design matrix to be $\mathbf{J} = diag([0, c_L, c_R, c_z])$, then the maneuvering cost associated with each edge $\mathcal{Q}$ is given by

$$C_m(e_{i,i+1}) = \sqrt{\mathbf{E}^\top \mathbf{J} \mathbf{E}}, \tag{4.49}$$

The cost c_z allows the designer to penalize vertical maneuvering. Similarly, c_L, and c_R allows the designer to place more or less cost on the left or right edges. Multiple values of these cost parameters may be saved in look-up table, and the collision avoidance algorithm picks the appropriate value based on the geometry of the encounter.

The total cost for traveling along an edge comes from the weighted sum of all different costs given as

$$C(e_{i,i+1}) = C_L(e_{i,i+1}) + C_{col}(e_{i,i+1}) + k_1 C_{th}(e_{i,i+1}) + k_2 C_{dev}(e_{i,i+1}) + k_3 C_m(e_{i,i+1}), \tag{4.50}$$

where k_1, k_2 and k_3 are positive real numbers and design parameters. These parameters allow the designer to place weight on collision risk or deviation from path or maneuvering preferences depending on the encounter scenario. Once the cost is assigned to each edge in $\mathcal{G}$, then a graph-search method such as Dijkstra's algorithm can be used to find the least cost path from the start point to the destination point. Dijkstra's algorithm solves the problem of shortest path in a directed graph in polynomial time given that there are not any negative weights assigned to the edges. The main idea in Dijkstra's algorithm is to generate the nodes in order of increasing value of the cost to reach them. It starts by assigning some initial values for the distances from the start node and to every other node in the graph. It operates in steps, where at each step the algorithm updates the cost values of the edges. At each step, the least cost from one node to another node is determined and saved such that all nodes that can be reached from the start node are labeled with cost from the start node. The algorithm stops either when the node set is empty or when every node is examined exactly once. A naive implementation of Dijkstra's algorithm runs in a total time complexity of $O(|\mathcal{N}|^2)$. However, with suitable data structure implementation, the overall time complexity can be reduced to $O(|\mathcal{E}| + |\mathcal{N}| \log_2 |\mathcal{N}|)$ [75].

The output of the local-level collision avoidance algorithm generates a waypoint avoidance path that consists of an ordered sequence of waypoints $\mathcal{W}_c = \mathbf{W}_{c1}, \mathbf{W}_{c2}, \cdots, \mathbf{W}_{ci}$. These waypoints are basically nodes in the discretized local-level graph selected by the Dijkstra search. Then, the avoidance waypoint are transformed from the local-level frame to the the global reference frame and added to the original waypoints path $\mathcal{W}$. Next, a path manager is required to follow the waypoints path. One possible approach is to transition when the ownship enters a ball around the waypoint $\mathbf{W}_i$, or a better strategy is to use the half-plane switching criteria that is not sensitive to tracking error [16]. In addition, a smooth transition between the waypoints is required to make the generated path flyable by the ownship. This can be achieved by implementing the fillet maneuver or using Dubins paths. For further analysis on path manager, switching between waypoints and smoothing paths, we refer the interested reader to [16].

A key feature of the proposed approach is that the future motion of the ownship is constrained to follow nodes on the map that are spaced by a constant time. Since the path is represented using waypoints that are at fixed time instants, it is easy to determine roughly where the ownship will be at any given time. This timing information is used when assigning cost to edges to better plan paths and prevent collisions.

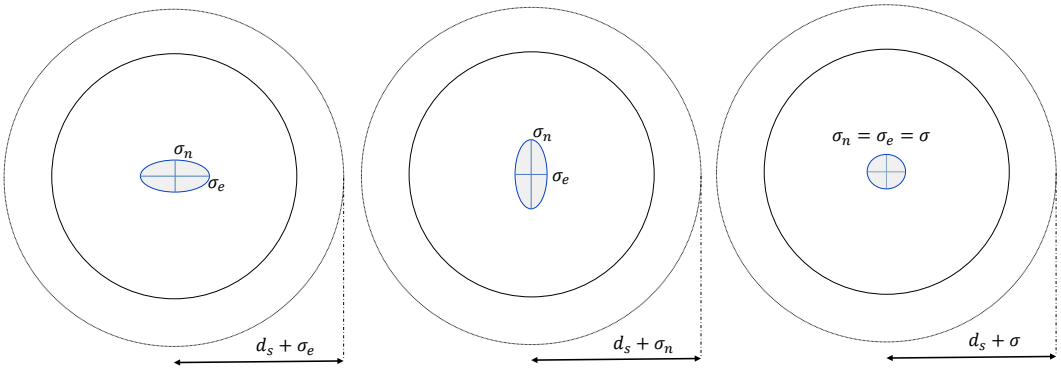


Figure 4.12: Top view: the radius of the collision volume cylinder is increased to account for the uncertainty in the 2D plane.

To account for the uncertainty error in the relative position estimates, we consider the following simple and intuitive solution. We assume that an estimate of the accuracy of the states is provided and is quantified as σ_n, σ_e and σ_d. As illustrated in examples in

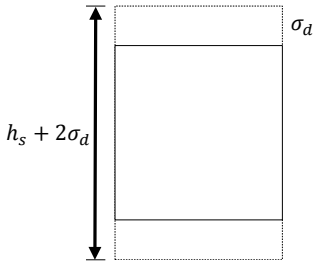


Figure 4.13: Side view: the height of the collision volume cylinder is increased to account for the uncertainty of the down component of the relative position.

Figures 4.12 and 4.13, we enlarge the radius and the height of the collision volume as follows

$$\hat{d}_s = d_s + \sigma,$$
$$\hat{h}_s = h_s + 2\sigma_d,$$

where $\sigma = \max(\sigma_n, \sigma_e)$. Then, when cost is assigned to edges we consider the following collision volume

$$\mathcal{C}(\mathbf{p}_r(t)) = \{d \in \mathbb{R}^2 : \|\mathbf{p}_r(t) - d\| \leq \hat{d}_s \,\text{and}\, h \in \mathbb{R} : \left|p_{r_z} - h\right| \leq \hat{h}_s/2\}.$$

4.3 Conclusions

In this chapter, we have presented two reactive collision avoidance algorithms suitable for unmanned aircraft SAA systems. We have proposed a collision avoidance algorithm based on a simulated chain that responds to a virtual force field produced by encountering intruders. The key feature of the proposed approach is to model the future motion of both the intruder and the ownship using a chain of waypoints that are equally spaced in time. We have also developed a collision avoidance logic using an ownship centered coordinate system. The technique builds graph in the local-level frame and use DijkstraŠs algorithm to find the path with the least cost. Both algorithms account for multiple intruders in various encounter scenarios, and uncertainties in the state estimates of the detected intruders.

Chapter 5. Airborne Vision-Based Sense-and-Avoid System

Never fly the A model of anything.

-World War II Pilot Officer Edward Thompson of 433 (RCAF) Squadron.

In this chapter, we examine the closed-loop performance of the chain-based collision avoidance approach presented in Chapter 4 Section 4.1, and the passive ranging tracker and collision detection system developed by UtopiaCompression. The tracker and the collision detection system system is based on UtopiaCompression's unique technology – Monocular Maneuverless PAssive Ranging System (M2PARS) for detecting imminent collisions using passive imaging sensors alone.

5.1 Airborne Camera Sensor Model

Range estimation of the intruder using only a passive sensor can be formulated as a bearing-only tracking problem. The conditions of unobservability of the intruder aircraft in a bearing-only tracking problem have been extensively studied since the late 1970's [8, 54, 80, 99, 104]. This body of research has established that the intruder aircraft state is observable, in general, only if the order of the ownship dynamics is greater than the intruder dynamics. For an intruder moving with constant velocity, this implies that the sensor dynamics must involve an acceleration component. With the growing use of UAS in recent years, such a "maneuver-based" approach has been proposed as a solution to the EO based SAA problem for UAS: upon detecting an intruder, the UAS maneuvers in order to triangulate and resolve the position of the intruder [101]. However, a maneuver-based approach is undesirable in many ways. It may lead to waste of fuel, loss in mission performance, and is in general bad airmanship.

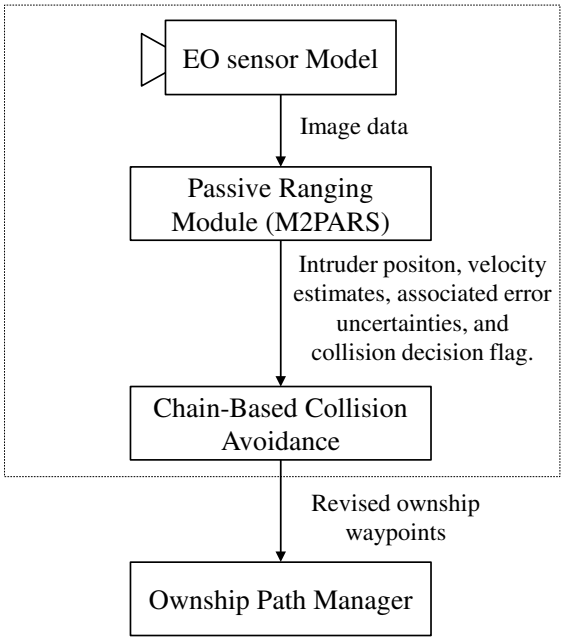


Figure 5.1: Airborne vision based sense and avoid structure diagram.

As depicted in Figure 5.1, the sense and avoid system consists of a camera sensor model, passive ranging module, and the chain-based collision avoidance algorithm. The sensing model assumes 640×480 camera resolution, 48 deg camera field-of-view and frame rate at 30 frame per second. The M2PARS estimates intruder's range and range rate with monocular passive sensors and without ownship maneuver. It makes use of multiple image features to estimate the intruder's state, including position and velocity. An estimate of the accuracy of these state estimates is also generated and is quantified in an error covariance matrix. Using the state estimate information and the current position and velocity state of the ownship, the M2PARS estimates the time to the closest point of approach and distance at closest point of approach. If a collision threat is detected, the intruder position and velocity estimates and an activation flag are passed to the collision avoidance algorithm. The avoidance logic plans an evasive maneuver and passes revised waypoints that are free from collision risk to the ownship's path manager.

5.2 Collision Detection Logic

In this section[1], we discuss a simple collision detection logic similar to that presented in [47]. To be more specific, let the estimates of the intruder's position and velocity at time t be $\mathbf{p}_{int}(t)$ and $\mathbf{v}_{int}(t)$, respectively. Assume that the ownship's position and velocity from inertial navigation system are known as $\mathbf{p}_{own}(t)$ and $\mathbf{v}_{own}(t)$. Using Eq (3.3), we further calculate the relative position and velocity between intruder and ownship as $\mathbf{p}_r(t)$ and $\mathbf{v}_r(t)$. Assume that the intruder moves with a constant velocity. Using $\mathbf{p}_r(t)$ and $\mathbf{v}_r(t)$, we predict the distance between the intruder and the ownship at time $t + \tau$ as

$$d(t+\tau) = \|\mathbf{p}_r(t) + \mathbf{v}_r(t)\tau\| . \tag{5.1}$$

Define d_s as the safe distance between intruder and ownship. That is, a collision occurs whenever the true range between intruder and ownship is less than d_s. A non-collision encounter means that the true range between intruder and ownship is always greater than d_s. To predict whether the collision will occur, we use the simple collision detection logic given in Algorithm 3.

Algorithm 3: Collision detection algorithm

1: **if** $\exists \tau \in (0, T]$ such that $d(t+\tau) < (1+\delta) \times d_s$ **then**
2: collision detected
3: **else**
4: no collision detected at time t
5: **end if**

In Algorithm 3, the parameters δ and T are positive real numbers and design parameters for the collision decision rule. The parameter T is the length of the prediction horizon. At any given time t, we check whether the intruder range is below the safe distance scaled by a factor of δ anytime within the next T seconds. Since there is always uncertainty associated with the estimates and the future maneuver of the intruder, we choose a non-zero δ to compensate for this uncertainty. We tune δ such that a desired correct detection rate and false alarm rate are achieved in

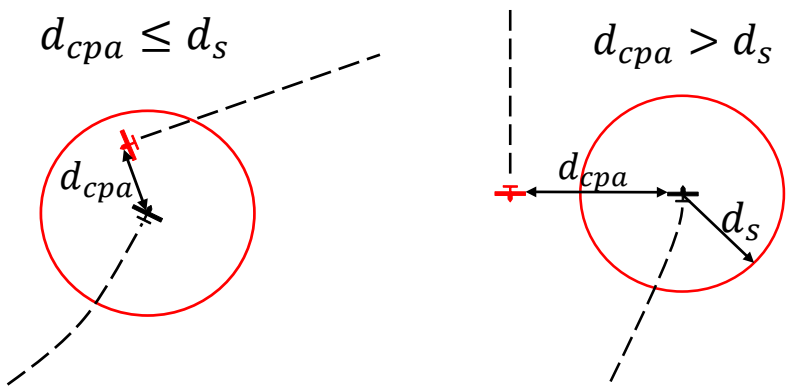
[1]The section is jointly written with Sharath Avadhanam, He Bai, Randal W. Beard as part of a paper published in the AIAA Guidance, Navigation and Control Conference [120].

simulation. To facilitate such tuning, we can perform Monte-Carlo simulations and plot the system operative characteristics curve by varying δ. By varying δ, we generate different P_{cd} and P_{fa} and plot them as the SOC curve. One can expect that as δ increases, both P_{cd} and P_{fa} will increase. Therefore, there exists a tradeoff between high P_{cd} and low P_{fa}. In the aircraft collision detection, we would like to detect all potential collisions, if possible. Thus, some false alarms will be expected. Monte Carlo simulations with constant velocity intruders have shown that we can achieve 95% P_{cd} with 2% P_{fa} using the M2PARS and Algorithm 3. [1]

5.3 Simulation Results

In order to test the performance of the chain-based collision avoidance algorithm, we developed a simulation environment with four-degree-of-freedom aircraft model for both the ownship and the intruder. We integrate the collision avoidance logic with the M2PARS module as shown in Figure 5.1. The minimum safe distance is set to 30 meters. A 10-link chain is chosen for the chain-based collision avoidance logic. As shown in Figure 5.2, a collision incident occurs when the the distance at closet point of approach is less than or equal the minimum safe distance i.e $d_{cpa} \leq d_s$. We first test two typical collision encounter scenarios namely, approaching head-on and converging scenario. The initial encounter geometry of both scenarios are shown in Figure 5.3. As shown in Figure 5.3(a), the ownship starts at the origin of North-East coordinates system and follows a straight-line path, which is composed of line segments between predefined waypoints. The speed and the initial heading of the ownship are set to 13 m/s and 0 deg, respectively. The heading angle is measured from the North axis. The intruder is initialized at location (500, 0) with speed and heading of 15 m/s and 180 deg, respectively. If collision avoidance is not planned the d_{cpa} is 0 m. Ideally both the ownship and the intruder should alter course to the right according the the right-of-way rules. However, we have implemented the worst case scenario where the intruder keeps moving in the same direction and does not plan an avoidance maneuver. Therefore, once the potential collision is detected by the M2PARS algorithm, the ownship plans a collision avoidance path as shown in Figure 5.3(c).

[1] We refer interested readers to `http://www.utopiacompression.com/technologies/imminent_collision_detection.php`



(a) A collision encounter occurs when $d_{cpa} \leq d_s$.

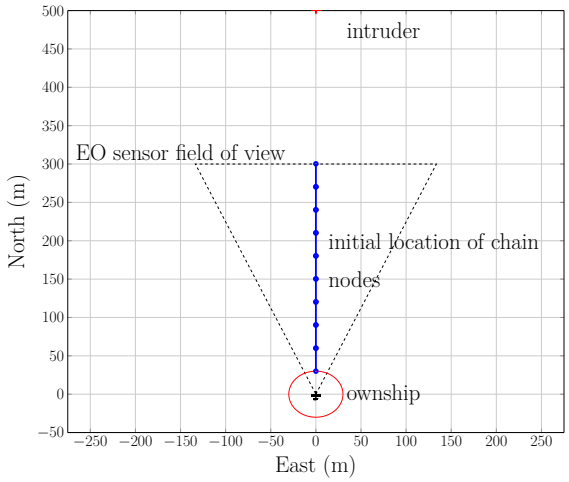
(b) A non-collision encounter occurs when $d_{cpa} > d_s$.

Figure 5.2: Collision volume definition.

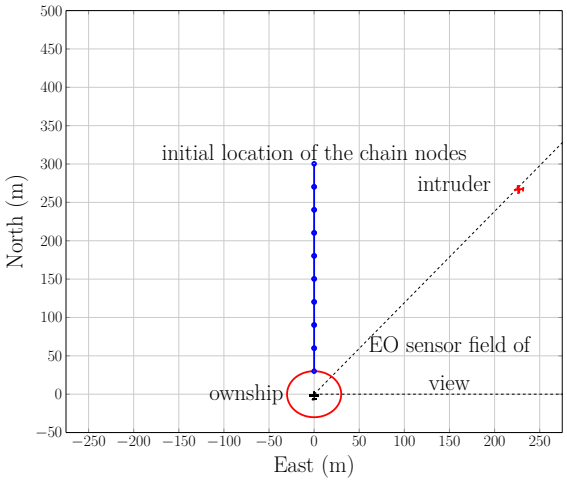
The converging scenario is shown in Figure 5.3(b). Similar to the head-on scenario, the ownship initially starts from the origin in the inertial frame and follows a straight-line path with speed of 13 m/s and initial heading of 0 deg. The intruder is located at (260, 225) and follows a straight-line path. The speed and heading of the intruder are set to 15 m/s and -94 deg, respectively. It is also assumed that the camera is rotated towards the right covering approximately 50 deg in azimuth in the inertial frame. The d_{cpa}, if the collision avoidance algorithm is not initiated, is 8.20 m. The avoidance path is shown in Figure 5.3(d).

We consider the miss distance or the distance at the closest point of approach and the maximum deviation from the original path to measure the performance of the collision avoidance algorithm. As shown in Figures 5.3(e) and 5.3(f) the miss distance is 32.2 m and 58.11 m in the head-on and converging scenarios, respectively. Both are greater than the minimum safe distance. The maximum deviation is defined as the ownship's maximum deviated distance from its original path. As shown in Figures 5.3(e) and 5.3(f), the maximum deviation distance is 38.01 m and 22.32 m in the head-on and converging scenarios, respectively. In both scenarios once the collision is cleared the collision avoidance logic returns the ownship back to its nominal path.

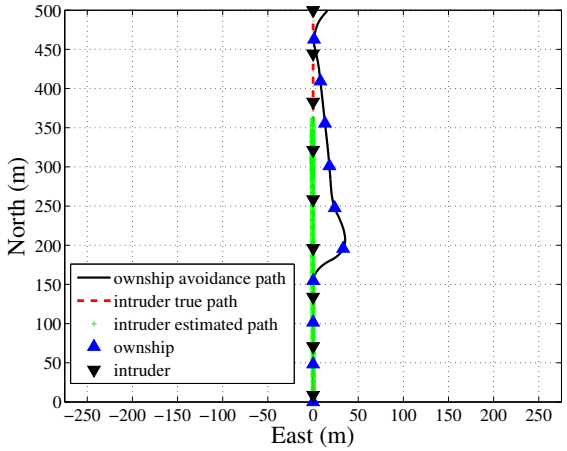
We next use Monte Carlo simulation to investigate the level of performance that can be gained by using the chain based collision avoidance algorithm as the initial range to the intruder, initial bearing to the intruder and speed of the intruder varies provided that



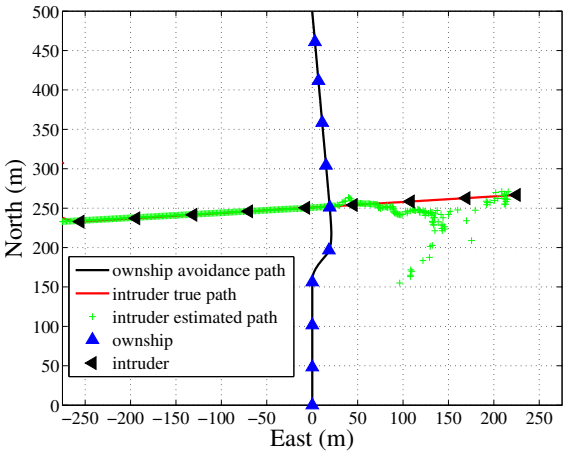
(a) Encounter geometry of the head-on scenario.



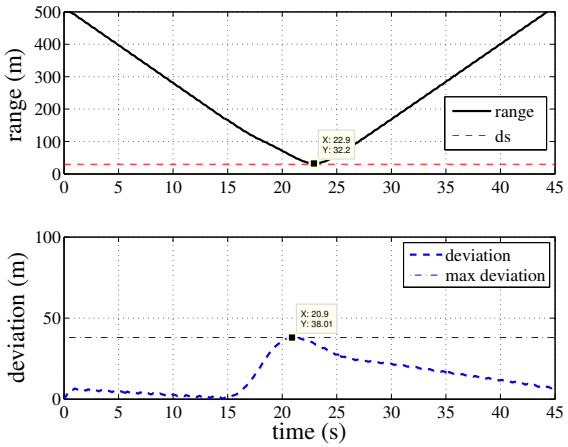
(b) Encounter geometry of the converging scenario.



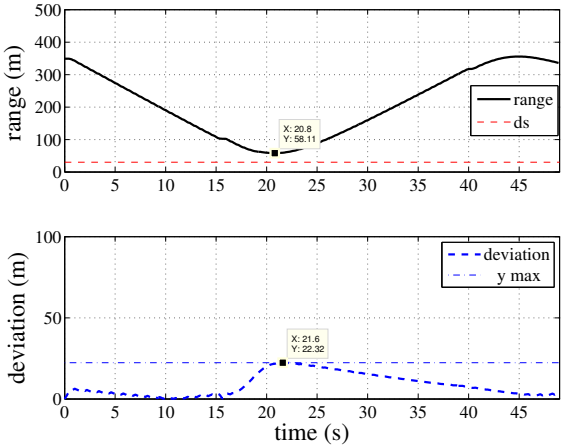
(c) Ownship's avoidance path.



(d) Ownship's avoidance path.



(e) Range to the intruder and deviation from original path.



(f) Range to the intruder and deviation from original path.

Figure 5.3: Results of the chain based collision avoidance algorithm.

the uncertainty of the intruder position and velocity estimates also change. Generally, any performance metric needs to take into account the competing objectives of collision avoidance logic, for instance preventing an imminent collision and minimizing deviation from flight-plan.

Table 5.1: Set values of the Monte Carlo simulations parameters.

Parameter	Set value
Initial range to the intruder	$[30, 150, 270, 390, 510, 630]$
Speed of intruder	$[13, 16, 19, 22, 25, 28]$
Bearing angles, converging scenario	$[20, 30, 40, 50, 60, 70]$
Bearing angles, head-on scenario	$[65, 75, 85, 95, 105, 115]$

In all Monte Carlo simulations, we have assumed that the ownship starts from the origin of the North-East coordinates system and follows a straight-line path, with a 0 deg initial heading angle and a speed of 13 m/s. As shown in Table 5.1 , we defined a set of values for each one of the three chosen encounter geometry parameters, namely initial range to the intruder, initial bearing to the intruder and speed of the intruder. In these simulations bearing angle is measured counter-clockwise from the inertial East axis. The Monte Carlo simulation starts with one of the three aforementioned parameters fixed for each value from the set, while other parameters are drawn from a uniform distribution derived from their nominal moments. At each given value from the set a hundred Monte Carlo trials is repeated for each pair of the uncertainty where each trial runs the simulated collision avoidance scenario for 45 seconds. The four pairs of uncertainty associated with position and velocity estimates of the intruder are shown in Table 5.2. The covariance matrix used in the Monte Carlo simulation using each pair of position and velocity is derived as

$$\mathbf{Q} = \begin{pmatrix} \mathbf{Q}_p & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_v \end{pmatrix}, \qquad \mathbf{Q}_p = \begin{pmatrix} \sigma_{pn}^2 & \sigma_{pn}\sigma_{pe} \\ \sigma_{pn}\sigma_{pe} & \sigma_{pe}^2 \end{pmatrix} \qquad \mathbf{Q}_v = \begin{pmatrix} \sigma_{vn}^2 & \sigma_{vn}\sigma_{ve} \\ \sigma_{vn}\sigma_{ve} & \sigma_{ve}^2 \end{pmatrix},$$

The Monte Carlo routine simulate a 1000 trials for each value in the set. Similarly, the Monte Carlo simulation is then repeated for each value of the next parameter's set having

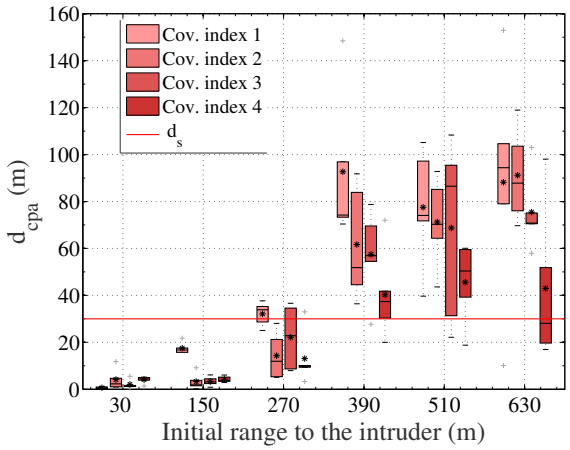
the other parameters drawn from a uniform distribution. Both the approaching head-on and converging scenarios were simulated the same way.

Table 5.2: 1σ values assumed for position and velocity estimates of the intruder.

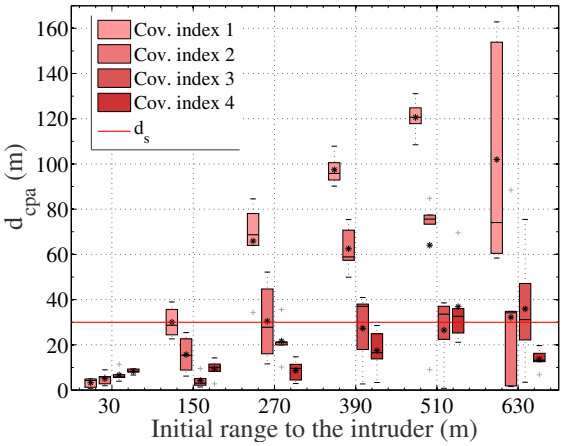
Covariance Pairs Index	$[\sigma_{pn},\ \sigma_{pe}]$	$[\sigma_{vn},\ \sigma_{ve}]$
1	$[0, 0]$	$[0, 0]$
2	$[17.32, 1.71]$	$[2.45, 0.18]$
3	$[34.66, 3.42]$	$[4.91, 0.35]$
4	$[51.99, 5.12]$	$[7.36, 0.53]$

Figures 5.4 shows the results of the distance at the closest point of approach averaged over 1000 Monte Carlo trials. The results are measured for various values of initial range, speed and bearing of the intruder given that intruder states estimates are defined according to uncertainty levels shown in Table 5.2. Clearly as shown in Figures 5.4(a) and 5.4(b), the ownship is able to avoid collisions and maintain the minimum safe distance as initial range to the intruder increases. This result reinforces our discussion in Chapter 2, where the ownship requires a minimum distance to the intruder to be able to avoid a collision. In the head-on scenario the chances of avoiding a collision decreases as the speed of the encounter intruder increases as shown in Figure 5.4(c). In the converging scenario, it is almost the opposite, as the speed of the intruder increases the chances of avoiding the collision increases. This is because the specific nature of the encounter geometry. Figures 5.4(e) and 5.4(f) shows that relative angle to the intruder is not as critical of a parameter as the range and speed of the intruder. In almost all initial bearing angles the ownship was able to avoid collisions. Additionally, in all cases the collision avoidance performs better when the error levels of the position and velocity estimates decrease.

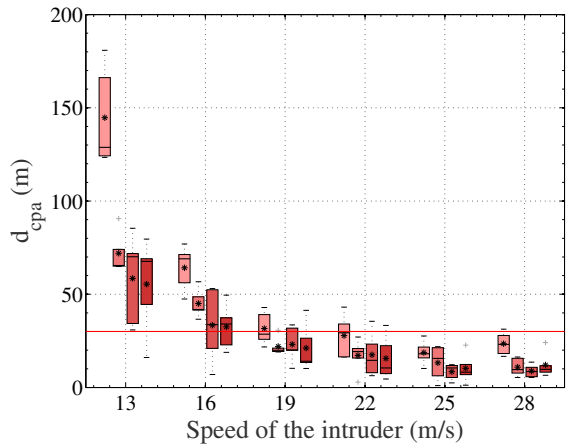
Another important aspect of evaluating the performance of the proposed approach is its ability to run in real time. To assess this capability, we have measured the time it requires to execute one cycle of the collision avoidance algorithm. Figure 5.5 shows that in all experiments the run time of the avoidance algorithm does not exceed 0.2 s. In Chapter 2, we have assumed the computation time of the sense and avoid system is 5 s. Therefore,



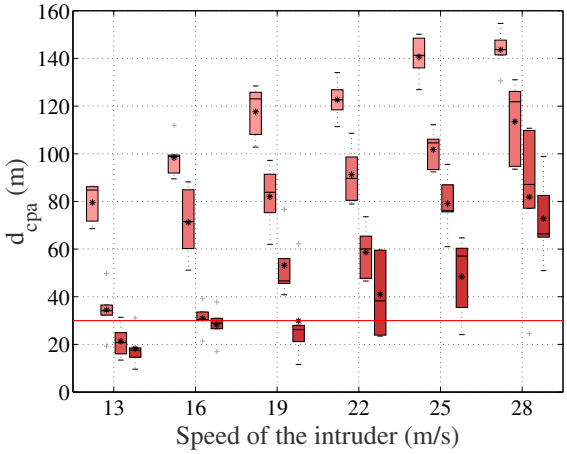
(a) Head-on scenario: d_{cpa} as range to the intruder varies.



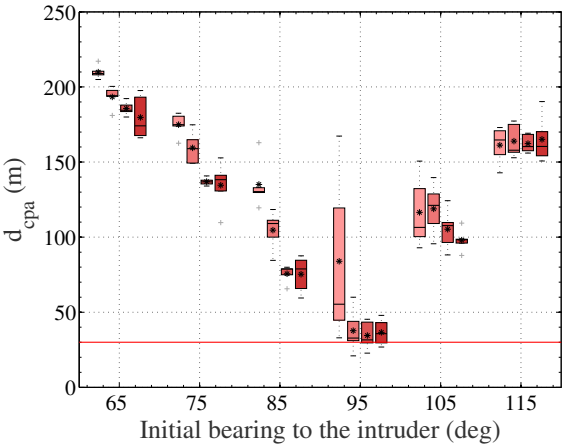
(b) Converging scenario: d_{cpa} as range to the intruder varies.



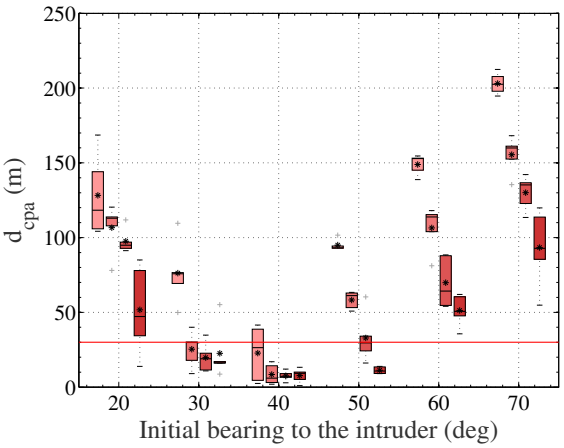
(c) Head-on scenario: d_{cpa} as speed of the intruder varies.



(d) Converging scenario: d_{cpa} as speed of the intruder varies.

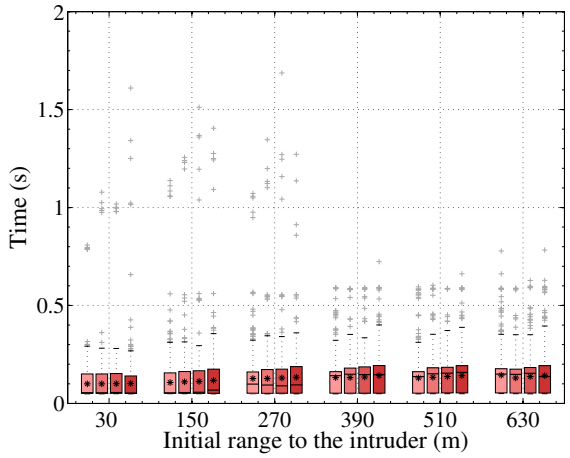


(e) Head-on scenario: d_{cpa} as bearing angle to the intruder varies.

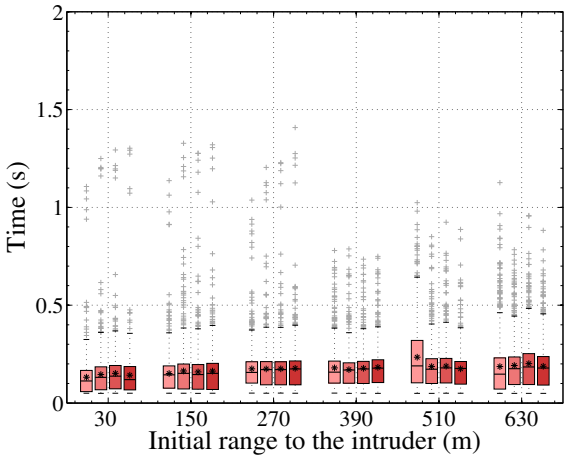


(f) Converging scenario: d_{cpa} as bearing angle to the intruder varies.

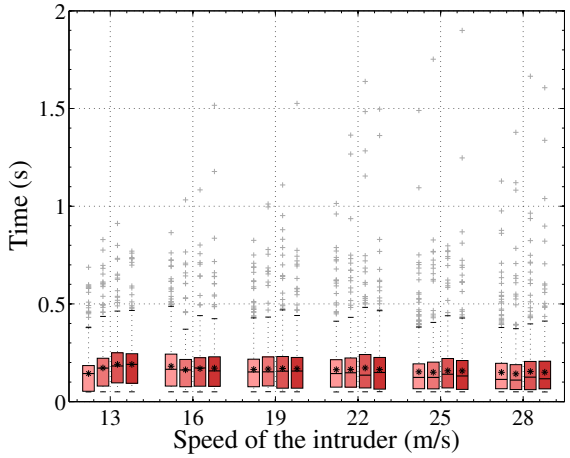
Figure 5.4: Monte Carlo simulations to evaluate the chain-based collision avoidance algorithm. The mean is marked by the asterisk, the median is marked by the line in the box, while the box defines the 25th and 75th percentiles



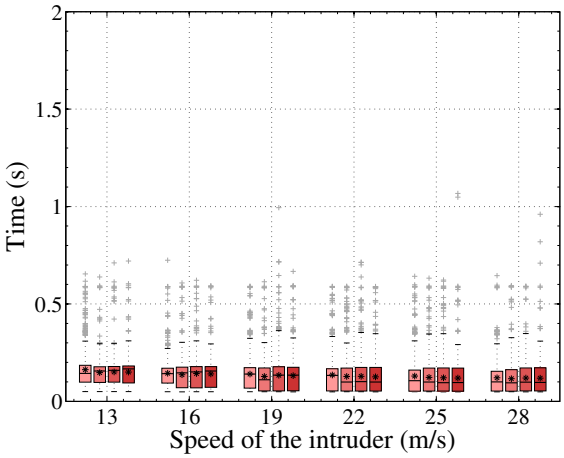
(a) Head-on scenario: run time as range to the intruder varies.



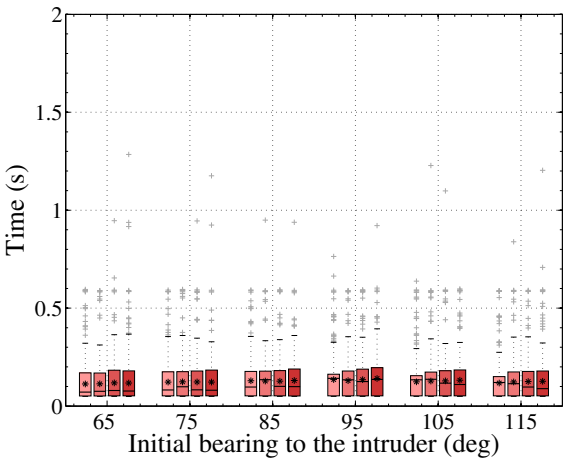
(b) Converging scenario: run time as range to the intruder varies.



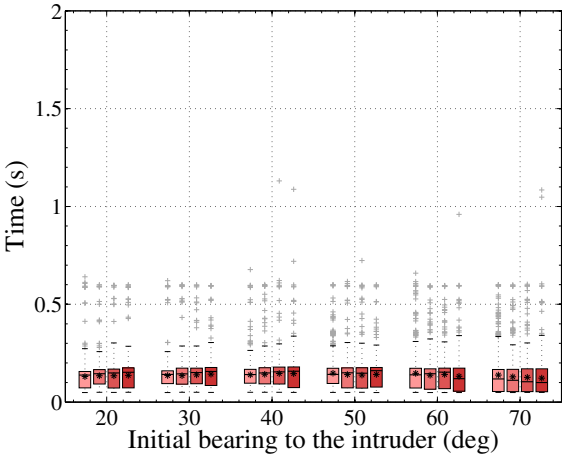
(c) Head-on scenario: run time as speed of the intruder varies.



(d) Converging scenario: run time as speed of the intruder varies.



(e) Head-on scenario: run time as speed of the intruder varies.



(f) Converging scenario: run time as speed of the intruder varies.

Figure 5.5: Monte Carlo simulations to evaluate execution time for one cycle of the chain-based algorithm. The mean is marked by the asterisk, while the box defines the 25th and 75th percentiles.

running the collision avoidance system with 0.2 s results in a time window of 4.8 s to detect, track and evaluate the collision encounter. The 4.8 s may be more than what is needed. In that case we may relax the assumption on the computation time. Requiring less time for the SAA computation decreases the minimum required sensing range, and thus allows additional sensors with lower detection range.

5.4 Conclusions

To examine the proposed chain-based collision avoidance algorithm we combined the electro-optical (EO) based Monocular Maneuverless PAssive Ranging System (M2PARS) technology developed by UtopiaCompression Corporation for detecting imminent collisions using passive imaging sensors and simulated the integrated model to predesigned encounter geometry scenarios in the Matlab/Simulink environment. The Monte Carlo simulation results shows an acceptable performance level for the integrated system measured by miss distance and run time. The algorithm was implemented using Matlab on an Intel i7 processor. We expect that implementing these algorithms in a compiled language, such as C or C++, will show that real-time execution is feasible using low-cost computational hardware.

Chapter 6. Airborne Radar-Based Collision Detection and Risk Estimation

One learns by doing a thing; for though you think you know it, you have no certainty until you try.

-Sophocles

In Chapter 3 we evaluate the probabilistic-based collision detection method using additive zero mean white noise on the intruder states to model what actually a sensor and tracker system would deliver to the sense-and-avoid system. In this chapter, however, rather than simply adding artificial measurement noise, we develop a sensor model along with an estimator whose process and measurement equations are formulated in the so-called modified polar coordinates (MPC) that appropriately handles polar measurements [8,41]. The sensor model incorporates a realistic model for a radar that closely matches the specifications and performance of a physical radar systems that is currently under development. We analyze the radar sensor, the tracker and collision detection by comparing the proposed collision detection approach with a linear extrapolation method using simulated encounters generated from the MIT LL UEM.

6.1 Airborne Radar Sensor Model

The proposed radar model[1] implements a monopulse configuration with a single transmit antenna and two receive antennas fixed on a rotating gimbal. All antennas are modeled with a sinc radiation pattern. The two receive antennas have a 3 dB beamwidth of 20 deg and

[1]The section is jointly written with James Mackie, Jonathan Spencer, Dr. Randal W. Beard and Dr. Karl F. Warnick as part of a journal manuscript published in the Journal of Aerospace Information Systems (JAIS) [118].

are offset ±10 deg from the transmit antenna. The transmit antenna has a 3 dB beamwidth of 40 deg and is angled at the monopulse system boresight, or the gimbal angle θ_{gim}. A monopulse system can be gimbaled in order to scan a broader area than the beamwidth of the antennas. In order to optimally cover the entire 360 deg field of regard (FOR), our system used two monopulse radar configurations with one at each end of the aircraft. This configuration is shown in Figure 6.1 (a) and (b). With radar, there is generally a trade-off

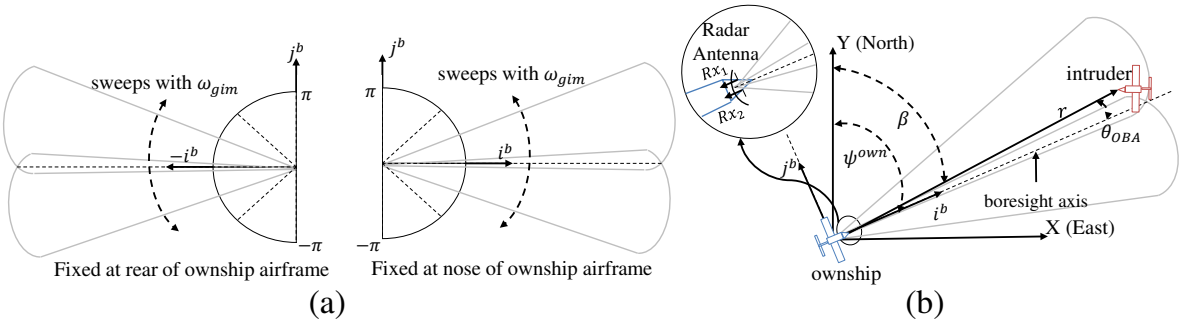


Figure 6.1: a) Two radar units on-board the ownship, each is sweeping back and forth to cover 180 deg; and b) Radar configuration geometry.

between tracking and searching capabilities. The nature of the problem being solved in this paper requires a search mode radar that will scan the entire FOR. However, when monopulse tracking methods are applied, an angular accuracy much finer than the antenna beamwidth can be obtained [111]. In order to integrate monopulse into the radar search mechanism, the radar system scans across the entire FOR at a series of discrete look angles and averages several radar returns at each angle. When a target is detected at one of the look angles, the monopulse method is applied to fine tune the initial detection angle, θ_{gim}. The angular correction, or off-boresight angle θ_{OBA}, is determined by comparing the sum and the difference of the two receive antenna amplitudes, Rx_1 and Rx_2, and is given by

$$\theta_{OBA} \approx \frac{\theta_{3dB}(Rx_1 - Rx_2)}{k_m(Rx_1 + Rx_2)}, \tag{6.1}$$

Table 6.1: Simulated FMCW radar parameters

Parameter	Value	Parameter	Value
Transmitted Power (P_{trans})	1 W	Center Frequency	10.25 GHz
Chirp Period (T_c)	1 ms	Chirp Bandwidth	50 MHz
Transmitter Antenna Gain (G^t)	14.83 dB	TX Antenna Beamwidth	40 deg Azimuth
Receiver Antenna Gain (G^r)	17.57 dB	RX Antenna Beamwidth	20 deg Azimuth
Gimbal rotation rate	360 deg/s	TX and RX Beamwidth	40 deg Elevation
System Noise Figure (F)	6.02 dB	FFT Noise Bandwidth B_{noise}	1 kHz

where θ_{3dB} is the 3 dB beamwidth of the antenna, and k_m is a linearity factor with a value between 1 and 2 [111]. The bearing from the North to the intruder is given by

$$\beta = \theta_{gim} + \theta_{OBA} + \psi_{own} \tag{6.2}$$

where ψ^{own} is the heading of the ownship.

The monopulse method drastically improves the initial angle estimate provided by the gimbal angle in a way that is computationally inexpensive and relatively cheap to implement in hardware. The angular resolution can be further improved by switching the radar to track mode and pointing the gimbal at the predicted angle and iterating through successive monopulse approximations and gimbal adjustments until the off-boresight angle is zero and the boresight angle of the gimbal is equal to the angle of the target. However, since this problem necessitates a search mode radar, the gimbal continues rotating and the slight angular error introduced by using a single monopulse iteration is accounted for in the Kalman Filter used to track the radar returns. Using Eq (2.11), the power received by each antenna as a function of angle and range is given by

$$P_{rec}(\theta, r) = \frac{P_{trans} G^t(\theta) G^r(\theta) \sigma_{\text{RCS}} \lambda^2}{(4\pi)^3 r^4}, \tag{6.3}$$

where $G^t(\theta)$ and $G^r(\theta)$ are the gains of the transmitter and receiver respectively. The angle θ used in Eq. (6.3) is the off-boresight angle from each antenna individually and should not be confused with the off-boresight angle of the combined monopulse system. The signal-to-noise

ratio (SNR) for our model is given by

$$SNR = \frac{P_{rec}}{P_{noise,thermal} + P_{noise,system} + P_{noise,clutter}} \approx \frac{P_{rec}}{P_{noise,thermal}F}. \tag{6.4}$$

Our model uses the standard approximation where system, thermal, and clutter noise sources are lumped into a single value that multiplies the thermal noise with a noise figure F for the system that approximates the contribution from system noise and clutter. We use the standard equation for thermal receiver noise given by

$$P_{noise,thermal} = k_B T_0 B_{noise}, \tag{6.5}$$

where k_B is the Boltzmann constant, T_0 is standard temperature 290 K, and B_{noise} is the bandwidth of the Fast Fourier Transform (FFT) bin over which the noise is observed. To produce the simulated radar returns we create a time-domain signal with amplitude given by Eq. (6.3) as a function of the true range and bearing to the intruder with respect to the gimbal pointing angle. Noise is added as given by Eq. (6.4) and the resulting signal is passed through an FFT to produce realistic radar range compressed data that undergoes monopulse processing to generate a bearing estimate. Using Eq. (6.4), the radar parameters listed in Tabel 6.1 result in approximately 10 dB of SNR for 0.1 m^2 RCS at a range of 800 m.

6.2 States Estimation

The problem of estimating the time evolution of the position, velocity and acceleration of an intruder from sequential sensor measurements has been previously addressed in the literature [14, 122]. The measurements provided by the radar sensor i.e. range, bearing (azimuth), elevation, and possibly the range rate are polar in nature; while the intruder dynamics are best expressed in rectangular coordinates. Since practical estimators, such as least squares and Kalman filters, operate under assumptions of linearity, there is a problem in directly applying the Kalman filter to measurements that are nonlinear in the state. The approximation residuals which result from linearization can accumulate and, unless preventive measures are taken, may cause the estimator to diverge from the intruder's true

state [18, 91]. On the other hand, the EKF is one of the most popular approaches to handle nonlinear measurements. It performs implicit coordinate transformation by linearizing the measurement model through a Taylor series expansion. However, it was shown that transforming a zero-mean measurement error from polar to rectangular coordinates generates a bias error [112]. As a consequence, debiased filtering techniques were proposed and validity limits on the applicability of the EKF were derived [79, 134]. In this paper, we adopted an alternative approach in which the state and measurement equations are formulated in the so-called modified polar coordinates (MPC). The resulting EKF has been shown to be stable and an asymptotically unbiased estimator [8, 41]. Consider the scenario shown in Figure 6.1 (b), where both the intruder and the ownship are flying at the same fixed altitude. Let the state vector $\mathbf{x}$ represent the Cartesian coordinates of the relative position and velocity between the intruder and the ownship at time t where

$$\mathbf{x} = \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix} \triangleq \begin{bmatrix} p_x(t) \\ p_y(t) \\ v_x(t) \\ v_y(t) \end{bmatrix} \triangleq \begin{bmatrix} n_{int}(t) - n_{own}(t) \\ e_{int}(t) - e_{own}(t) \\ vn_{int}(t) - vn_{own}(t) \\ ve_{int}(t) - ve_{own}(t) \end{bmatrix}. \tag{6.6}$$

Let $\mathbf{y}$ denote the modified polar coordinate state vector, given as

$$\mathbf{y} \triangleq [\dot{\beta}(t), \dot{r}(t)/r(t), \beta(t), 1/r(t)]^\top, \tag{6.7}$$

where $r(t) = \sqrt{p_x^2(t) + p_y^2(t)}$ and $\beta(t) = \text{atan2}(p_x(t), p_y(t))$ are the range and bearing to the intruder provided by the radar sensor at time t, respectively. It is shown in [8, 41], that the time evolution of the state vector $\mathbf{y}$ is given by

$$\dot{\mathbf{y}} = \begin{bmatrix} \dot{y}_1(t) \\ \dot{y}_2(t) \\ \dot{y}_3(t) \\ \dot{y}_4(t) \end{bmatrix} = \begin{bmatrix} -2y_1(t)y_2(t) + y_4(t)[\dot{x}_1(t)\cos(y_3(t)) - \dot{x}_2(t)\sin(y_3(t))] \\ y_1^2(t) - y_2^2(t) + y_4(t)[\dot{x}_1(t)\sin(y_3(t)) + \dot{x}_2(t)\cos(y_3(t))] \\ y_1(t) \\ -y_2(t)y_4(t) \end{bmatrix}. \tag{6.8}$$

Since in modified polar coordinates the state propagation model is nonlinear and the measurement update model is linear, the continuous-discrete EKF is a good choice for this application [16]. The continuous-discrete EKF assumes that the process evolution and measurement equations have the form

$$\dot{\mathbf{y}} = f(\mathbf{y}(t)) + \xi, \tag{6.9}$$

$$\mathbf{z}[n] = C\mathbf{y}[n] + \eta[n] \tag{6.10}$$

where $\mathbf{z}[n] = \mathbf{z}(t_n)$ is the nth sample of $\mathbf{z}$. In this specific implementation we have $C = [0_{2\times 2} \vdots I_{2\times 2}]$. The signals ξ and η are the process and measurement noise, respectively and are zero-mean Guassian random processes with known covariance matrices. While the process noise covariance is a tuning parameter, the covariance matrix of the measurement noise are determined by the noise properties of the radar [14, 53, 135]. A one-to-one nonlinear transformation which maps the MPC state vector $\mathbf{y}(t)$ into its rectangular counterpart $\mathbf{x}(t)$ is given by

$$\mathbf{x}(t) = f_x[\mathbf{y}(t)] = \frac{1}{y_4}\begin{bmatrix} \sin(y_3) \\ \cos(y_3) \\ y_2\sin(y_3) + y_1\cos(y_3) \\ y_2\cos(y_3) - y_1\sin(y_3) \end{bmatrix}. \tag{6.11}$$

Using Eq. (6.6), and given that the states of the ownship are known, the position, speed and heading of the intruder can be estimated as

$$n_{int}(t) = x_1(t) + n_{own}(t), \tag{6.12}$$

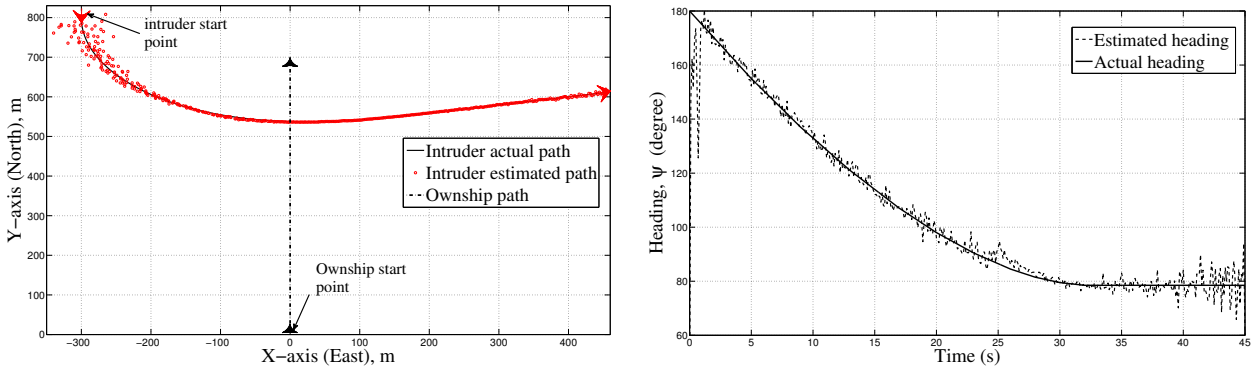
$$e_{int}(t) = x_2(t) + e_{int}(t), \tag{6.13}$$

$$v_{int}(t) = \sqrt{\Big(x_3(t) + vn_{own}(t)\Big)^2 + \Big(x_4(t) + ve_{own}(t)\Big)^2}, \tag{6.14}$$

$$\psi_{int}(t) = \text{atan2}\Big(x_3(t) + vn_{own}(t), x_4(t) + ve_{own}(t)\Big). \tag{6.15}$$

6.3 Simulation Results

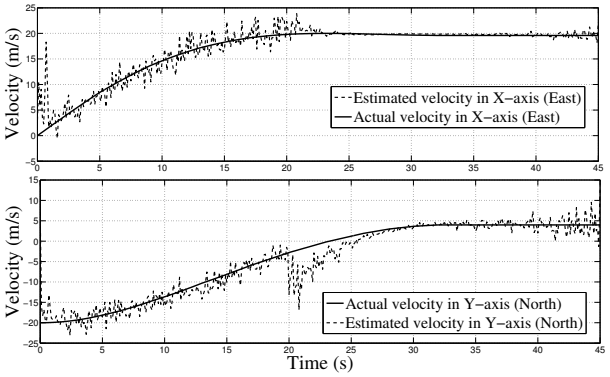
To show the performance of the sensor and estimator, we simulate a simple case scenario similar to the planner geometry shown in Figure 6.1(b) The ownship starts at position (0,0) with initial heading of 0 deg and moving with constant speed of 15 m/s. The intruder starts at position (-300, 800) with 180 deg with respect to the Y-axis (North) and moving at speed of 20 m/s. The path of the ownship, and the actual and estimated path of the intruder are shown in Figure 6.2 (a). The noise variance of the simulated range and bearing measurements are $\sigma_r = 0.87$ m and $\sigma_\beta = 2.72$ deg, when RCS=1 m^2. The aircraft dynamics are simulated using a simplified model that captures the flight characteristics of an autopilot controlled UAS. The actual and estimated heading, velocity components, and speed states of the intruder are shown in Figure 6.2(b), Figure 6.3(a) and (b), respectively. For further analysis on initialization, design and performance of an MPC EKF, we refer the interested reader to [8, 41].



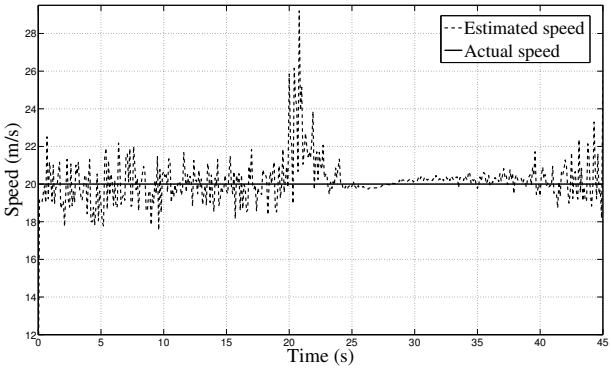
(a) (b)

Figure 6.2: Actual and estimated track (a) and (b) heading.

To validate the performance of the radar sensor model, the estimation scheme, and the collision detection algorithm presented in Chapter 3, Section 3.2, we create a simulation environment with 4 degree-of-freedom aircraft models for both the ownship and intruder. We use the assessment and validation metrics and follow the Monte Carlo simulations scheme discussed in Chapter 3, Section 3.3. Similarly, we sample flight trajectories from the MIT LL UEM. The encounter scenarios are assumed to take place in class G airspace flying within



(a) Actual and estimated velocity components of the intruder.



(b) Actual and estimated speed of the intruder.

Figure 6.3: Actual and estimated velocity components (a) and (b) speed.

an altitude layer of 500-1200 ft. The initial airspeed of the intruder varies over (0, 65] m/s based on the type of the intruder used in the simulation. In addition, the control variables $\dot{v}$ and $\dot{\psi}$ vary from -2 to 2 m/s^2 and -8 to 8 deg/s, respectively while the climb rate $\dot{h}$ is set to 0 m/s. We assume a virtual encounter circle centered on the ownship. The radius of the encounter circle is set to $d_{enc} = 800$ m. The ownship is initialized at the center of the encounter circle $\mathbf{p}_{own}(t_0) = (0, 0)$ and follows a straight line path. However, for each Monte Carlo run the intruder is randomly initialized on the perimeter of the encounter circle. The bearing angle of the intruder relative to the ownship is drawn from a uniform distribution over $[0, 2\pi]$ and the heading of the intruder is randomly drawn from a uniform distribution over $[0, 2\pi]$ while the ownship heading is set to 0 deg. The simulated encounters are similar to the planner geometry shown in Figure 3.7. The radar parameters used in simulation are listed in Table 6.1. The trajectories of both the ownship and intruder are generated over a 45 s time horizon where the sample time is set to 0.1 s.

We repeated the Monte Carlo simulation procedure for small, medium, and large sized intruder aircraft. We define a large intruder aircraft as having a wingspan greater than 15 ft, similar to a Cessna aircraft. We define a medium size intruder as having a 10-15 ft wingspan similar to that of a ScanEagle, and a small intruder UAS as having a wingspan less than 10 ft, similar to a Zagi. The intruder aircraft are assumed to have RCS sizes of 0.1, 1, and 10 m^2 for small, medium, and large aircraft, respectively [96]. Although RCS can have variations in angle and time due to moving parts, we used constant values for

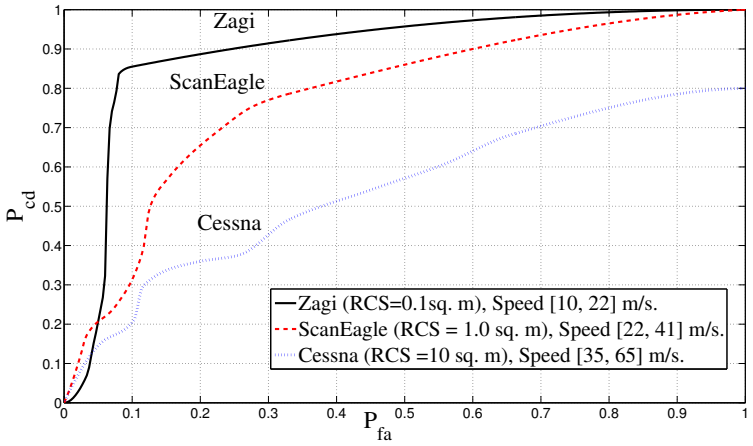
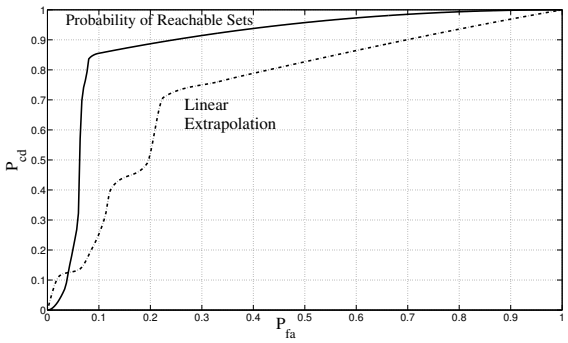


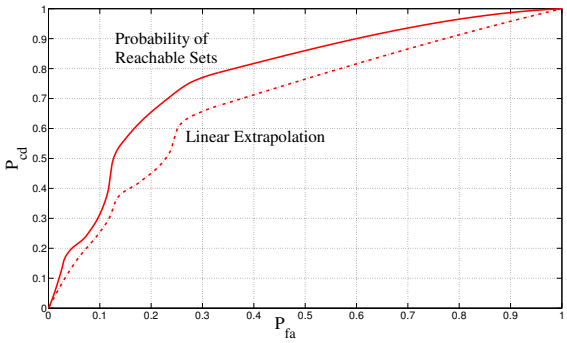
Figure 6.4: The P_{cd} and P_{fa} versus different threshold values examined for different intruder types.

simulated RCS. This approach is justified because of the temporal integration inherent in FMCW radar as well as significant averaging applied in post processing. Any small RCS variations in time will average to a relatively constant value. Thus, constant values are used for simulated RCS in order to simplify the model, yet maintain proof of concept. The authors note that while these RCS values are approximate, the performance of the system over such a wide range (20 dB) of RCS fluctuation demonstrates its efficacy. Commercial aircraft will generally have RCS values much larger than 10 m^2 and small UAS will have RCS close to 0.1 m^2 [96]. To be consistent we use identical path trajectory profiles sampled from MIT LL UEM for the different sized intruder aircraft. The SOC curves for the three different intruder aircraft obtained by Monte Carlo simulation are shown in Figure 6.4. Although smaller size intruders like the Zagi have a lower RCS value that causes the radar/tracker to produce poor estimates, the low speeds of this aircraft allow an adequate amount of time for the tracker to converge, thus helping the collision detection algorithm to perform better.

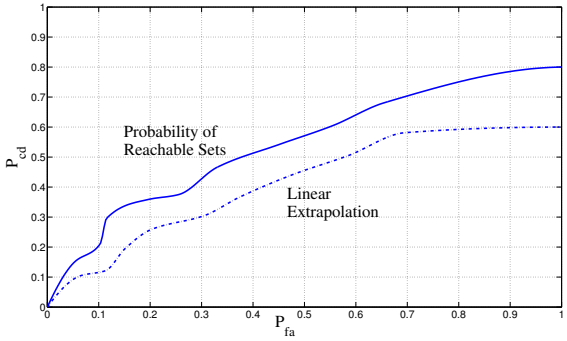
Finally, we compare the performance achieved by our proposed approach with a simplified scheme where the future position of the intruder is projected by propagating the current position along a straight in the direction of the current heading. This comparison is conducted by repeating the Monte Carlo procedure applying similar geometric and flight path profiles for the ownship and the intruder, and using the same parameters for the radar sensor and estimator. The SOC results shown in Figure 6.5 indicate that our algorithm



(a) Zagi size intruder.



(b) ScanEagle size intruder.



(c) Cessna size intruder.

Figure 6.5: SOC curves for the proposed approach (solid line) and simple linear extrapolation (dashed line).

outperforms the linear extrapolation approach. However, since the comparison results show that the difference in performance is not huge, the computationally simple approach of linear extrapolation might be efficiently used along with the probabilistic approach in a multi-tiered collision detection system.

6.4 Conclusions

We have introduced in this chapter a radar sensor model, the estimation scheme, and probabilistic based collision detection and risk estimation approach that is computationally feasible for small UAS. The computational burden is expected to be much less than existing algorithms that require running Monte Carlo approximations in near real time or performing an extensive sampling from the probability distribution of intruder estimated states or probabilistic encounter models. That makes the proposed approach a tractable solution in particular for small and mini UAS. The average simulation time required to execute one

cycle of our collision detection code is 0.00355 seconds using Matlab on an Intel i7 processor. This small run time motivates the application of this algorithm to multiple intruder scenarios. This small run time motivates the application of this algorithm to multiple intruder scenarios. We introduced a realistic radar model which could be built and implemented on-board small UAS. Although several standards have suggested that the necessary field of regard for a sense and avoid system should be 220 deg or 240 deg in the azimuthal plane, our simulations and analysis have extended this to a full 360 deg to account for overtaking scenarios.

Chapter 7. ADS-B Based Sense and Avoid Systems

What is chiefly needed is skill rather than machinery.

-Wilbur Wright

In this chapter[1] we present a complete, proof-of-concept sense-and-avoid solution for small unmanned aircraft including reliable intruder sensing, multi-target tracking and estimation, conflict/collision detection, and self-separation/collision avoidance using ADS-B as the primary detection mechanism. As shown in Figure 7.1, the ADS-B Out transmissions are received by a dual-link ADS-B In receiver. This receiver decodes the raw signal and passes it to the intruder tracker/estimator. In the estimator, the intruder state measurements are processed to have a coherent set of units and then passed through a Kalman filter. After Kalman filtering, the intruder position and velocity estimates are projected forward in time to identify possible conflicts or collisions. If either a conflict or collision threat is detected, the intruder position and velocity estimates and an activation flag are passed into the self-separation/collision avoidance algorithm. Once either the conflict or collision level of the avoidance logic has been activated a new conflict and collision-free path is generated. In the case of long-range intruders that pose a conflict risk, the ownship takes less aggressive action due to the longer allowable reaction time. For short-range collision risks the ownship plans a more aggressive action to quickly reduce the possibility of a collision. The ultimate output of the SAA system is a revised set of ownship waypoints that is free from conflict and collision risks. The system shown in Figure 7.1 is viable for both fixed-wing and multirotor aircraft, and could reasonably be extended for larger UAS outside of the small UAS definition.

[1]This chapter is jointly written with Matthew O. Duffield, Randal W. Beard, Timothy W. McLain as part of the journal manuscript accepted for publication in the Air Traffic Control Quarterly: An International Journal of Engineering and Operation [117].

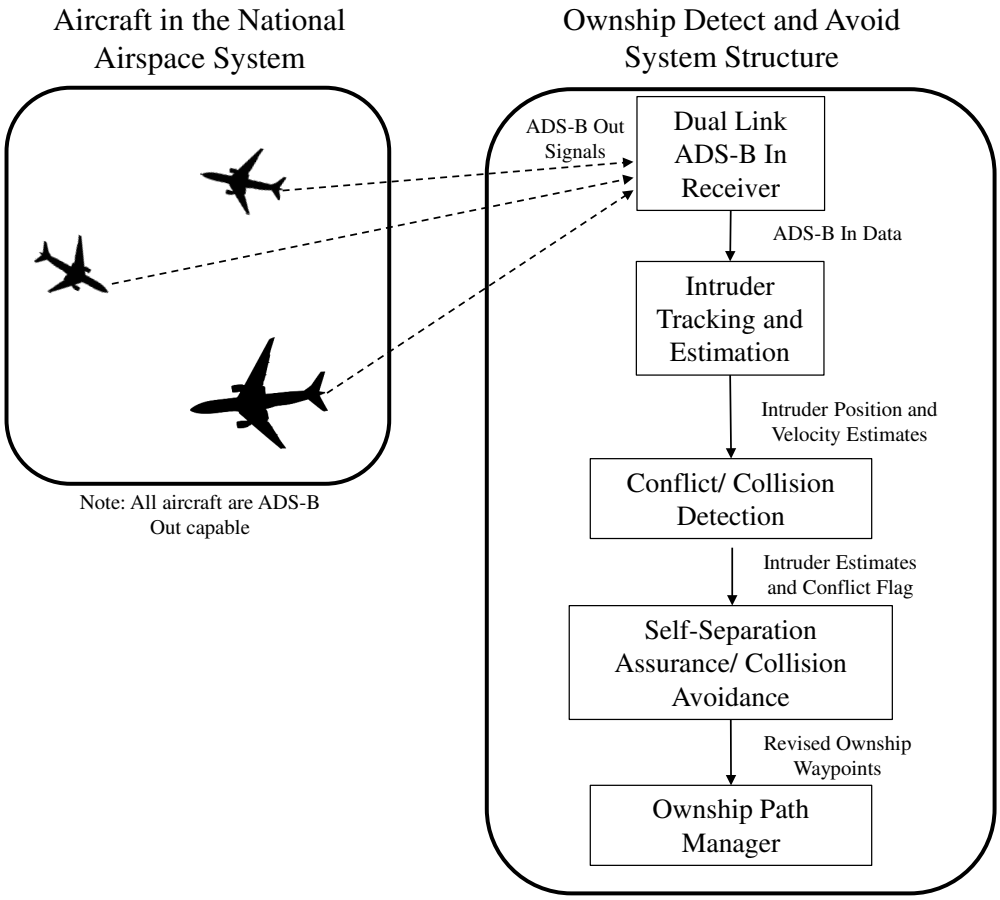


Figure 7.1: Proposed sense-and-avoid system structure diagram [117].

The purpose of this chapter is to explore the technical aspects of ADS-B as a detection mechanism for sense-and-avoid on small unmanned aircraft and to demonstrate conflict/collision detection and self-separation/collision avoidance methods that take advantage of ADS-B characteristics. It is important to note that prior to full implementation, ADS-B technologies and the proposed conflict/collision avoidance methods require additional validation in terms of expanded simulations and testing. Additional consideration must be given to the economic, safety, operational, and policy challenges that an ADS-B-based system poses. The discussion, simulations, and results presented in this chapter focus primarily on the technical aspects of ADS-B-based SAA. For the methods and simulations presented, we assume that the intruder aircraft are equipped with ADS-B Out, in other words the ability to transmit their cooperative information. The small UAS ownship is assumed to have ADS-B Out and dual-link ADS-B In. Thus it is capable of both transmitting its cooperative

information and receiving the cooperative information from all other aircraft. Consequently the responsibility of conflict detection, self-separation assurance, collision detection, and collision avoidance lies entirely on the small UAS ownship. Although these assumptions do not exactly match the requirements of the FAA 2020 mandate, they do represent a condition where full integration of UAS into the NAS would be possible. Thus in addition to presenting an SAA system for small UAS, we submit that a complete ADS-B equipage mandate could meet the technical requirements of the wide demand for significantly increased UAS operations in the NAS.

7.1 ADS-B Sensor

This section[1] provides a brief description of the ADS-B sensor, error characteristics, and its capability and limitations as an SAA sensor. Further analysis on the ADS-B associated regulations as they relate to sense and avoid, the statistical characterization of ADS-B error, message elements, airspace class and power requirements, and other related issues can be found in [117].

7.1.1 Error Characterization

ADS-B is a cooperative sensor that supports the exchange of a wide variety of information over long ranges. Typically, the information exchanged includes aircraft state information, state error estimates, aircraft identifiers, and aircraft operating indicators. This exchange occurs approximately once per second [27]. To exchange this information, two sets of hardware are necessary, ADS-B In and ADS-B Out. As the names suggest, ADS-B In allows for information to be received, and ADS-B Out supports the broadcasting of information. The hardware performing these two functions can be sold separately or as a single unit. In addition to the In or Out capability of ADS-B hardware, ADS-B transmissions can occur over two different frequencies, 1090 MHz and 978 MHz [43]. ADS-B Out hardware is specific to one of these two frequencies. The airspace class in which an aircraft will operate dictates

[1]This section is primarily written by Matthew O. Duffield and Timothy W. McLain as part of the journal manuscript published in the Air Traffic Control Quarterly: An International Journal of Engineering and Operation [117].

the required frequency. ADS-B In hardware also is specific to a particular frequency, but dual-link hardware that is capable of receiving transmissions on both frequencies is becoming increasingly common.

Table 7.1: Required set of message elements for ADS-B Out [117].

State Elements	Identification Elements	Error Elements	Other Elements
Latitude	Mode 3/A Transponder Code	NACp	Emitter Category
Longitude	Call Sign	NACv	Emergency Code
Barometric Altitude	IDENT	NIC	TCAS II equipped
Geometric Altitude	ICAO 24-bit address	SDA	TCAS II Advisory
Velocity	Length and Width	SIL	ADS-B In Equipped

Table 7.1 shows a list of the message elements exchanged by ADS-B. While a detailed explanation of each of the identification elements listed in Table 7.1 is beyond the scope of this dissertation, it is useful to note that each of these elements provides a unique identifier for the aircraft. The state elements transmitted are the latitude and longitude, barometric altitude, geometric altitude, and velocity. The error messages reported are Navigation Accuracy Category for Position (NACp), the Navigation Accuracy Category for Velocity (NACv), Navigation Integrity Category (NIC), Source Integrity Level (SIL), and System Design Assurance (SDA). In addition to the error metrics outlined in Table 7.1, ADS-B is subject to several additional sources of error namely latency error, resolution error, and message success rate (MSR) error. These additional sources of error, along with those previously defined in Table 7.1, play a role in defining an error characterization for ADS-B.

The NACp, NACv, NIC, SIL, SDA, latency error, resolution error, and MSR error provide a basis from which to derive an error characterization to model ADS-B. Given the NACp and NACv, the horizontal position and velocity can be modeled as a Rayleigh random process. From the Rayleigh process, the 95% bound on both the position and velocity error can be used to derive the variance for a Gauss-Markov process with zero-mean Gaussian noise for the North and East position and velocity [94, 103]. Based on values of NACp=303.8 ft and NACv=19.4 kn, the horizontal North and East position error can be modeled as a zero-mean Gaussian distribution with a standard deviation of 124 ft, and the North and East

velocity can be modeled as a zero-mean Gaussian distribution with a standard deviation of 8 kn. Correlation of errors in the position are accounted for by a Gauss-Markov model [16]. In the particular case of ADS-B where $\sigma_x = \sigma_y = 124$ ft, the variance of the Gaussian noise in the Gauss-Markov process can be estimated to be 5.28 ft [94].

FAA regulations require that ADS-B pressure altitude reporting equipment must report an altitude that is within 125 ft of the true altitude with 95% confidence [42, 44]. If the pressure altitude error, A_{pres}, is modeled as a zero-mean Gaussian random variable, it can then be shown that the standard deviation of this error is $\sigma_{Apres} = 75.9$ ft. For geometric altitude, the error is typically less than 147.6 ft with 95% certainty [108, 109]. Assuming that the geometric altitude error A_{geo} is a zero-mean Gaussian random variable, it can be shown that $\sigma_{Ageo} = 89.8$ ft. In addition to the noise of the pressure sensors, the encoding of barometric altitude has a resolution of 25 ft and geometric altitude has a resolution of 45 ft. This resolution introduces some additional error.

The error in the ADS-B reported vertical velocity varies with increasing vertical rate. For vertical rates between ± 500 ft/min, the vertical rate tolerance is ± 46 ft/min. For rates outside that range, the tolerance is 5% of the vertical rate [107, 114]. Given the assumption that these tolerances are 95% bounds, it can be shown that the standard deviation of the climb rate is 27.96 ft/min for vertical rates of ± 500 ft/min. Additionally the vertical rate error is effected by the resolution of the ADS-B message encoding which is 64 ft/min.

The loss of a valid ADS-B signal can be modeled using SIL, SDA, and MSR error. FAA regulations stipulate that position measurements outside the reported NIC can only be transmitted once per 10^7 transmissions. The SDA requirements permit values outside the NIC with a probability of 10^{-5}. MSR error requirements allow for a 10% or 15% message loss rate. These probabilities of erroneous or lost messages provide a method with which to model ADS-B signal dropout.

The error characteristics detailed above make it possible to model the error in ADS-B reported horizontal position, altitude, horizontal velocity, and vertical velocity. This results in a method capable of simulating ADS-B messages. It also provides a basis for estimating ADS-B messages and developing conflict detection, collision detection, separation assurance and collision avoidance methods.

7.1.2 Significance of ADS-B as SAA Sensor

The characteristics and requirements of ADS-B make it a capable sensor for SAA on small UAS in the National Airspace System. One key aspect of ADS-B that makes it feasible for use on small UAS is the availability of ADS-B receivers the meet the (SWaP) constraints of a small UAS. The Clarity ADS-B receiver provides a dual-link ADS-B receiver that is 2.5× 2.5× 1.5 in, weighs 0.344 lb, and consumes 2.4 Watts of power. Freeflight Systems has also recently introduced the RANGR RXD which is a dual-link ADS-B receiver. While slightly larger at 5× 5.75× 1.7 in, it still weighs less than one pound and consumes approximately 2.4 Watts of power. These hardware options both provide a suitable ADS-B In solution for small UAS.

Another key advantage of ADS-B is the long range at which information is available. While there is a significant amount of variation in the range of ADS-B signals, the shortest expected range is 10 nmi. Flight tests of ADS-B units suitable for small UAS have demonstrated reliable ranges of up to 80 nmi [95]. Additionally the long range of ADS-B is advantageous in that the quality of information transmitted over ADS-B does not degrade with range. Thus the accuracy of ADS-B is not dependent on the size, power, or range of the transmitter and receiver units. This is a significant advantage over radar and optical sensors, and makes path planning for conflict detection and separation assurance at long ranges.

ADS-B is a very capable sensor for SAA on small UAS, but it is not without limitations. One notable limitation of ADS-B is that it is a cooperative technology. This means that to have visibility of other aircraft they also must be equipped with ADS-B. Given the FAA mandate that only some aircraft need to be ADS-B compliant, there certainly will be aircraft in lower altitudes that are not ADS-B equipped. While these lower altitudes are prime locations for small UAS operations, the capability of ADS-B presented in this paper provides motivation to implement an ADS-B equipage requirement for all aircraft.

Another limitation of ADS-B is that it is heavily dependent on line-of-sight availability of GPS and ADS-B transmissions. Without GPS information, ADS-B transponders are unable to transmit usable position information. Air-to-air ADS-B transmissions also require line-of-sight visibility for reliable exchange of information. Bandwidth constraints of ADS-B

can also be a limitation. Due to the fact that all ADS-B Out-capable aircraft must transmit a message at least once per second on the same nominal frequency, the ADS-B protocol specifies a multiple access scheme. While the scheme is different for the 978 MHz frequency and 1090 MHz frequency, all multiple access schemes have a finite number of transmitters that they can support. Particularly applicable to small UAS are the limitations of the time-division multiple access (TDMA) scheme used for the 978 MHz frequency.

The cost of ADS-B equipage may pose a limitation. Certified ADS-B Out hardware costs typically range from $1,500 to $25,000 USD. ADS-B In hardware costs range from $400 to $3,000. While these costs are not necessarily prohibitive, they are significant especially for many of the small-to-medium-sized companies that plan to use UAS for commercial purposes. For ADS-B to be a fully viable, hardware costs need to decrease. As the FAA 2020 mandate approaches, an increasing number of companies are producing ADS-B hardware, and the cost of hardware is trending downward.

Ultimately the message elements, airspace and range requirements, hardware availability, and error characteristics of ADS-B make it a viable sensor for sense and avoid on small UAS in the NAS. While there are limitations to ADS-B sensors, development of promising solutions is reducing the impact of those limitations. As an SAA sensor, ADS-B offers all the information necessary to detect conflicts, maintain separation, and detect and prevent collisions.

7.2 Conflict/Collision Detection

The goal of conflict/collision detection is to identify intruder aircraft and determine the collision risk that they pose to the ownship. To do this, it is necessary to track and estimate the intruder states and extrapolate those states forward in time to identify possible future conflicts/collisions. In this section, we address the key components of a conflict/collision detection algorithm.

7.2.1 ADS-B Signal Processing

Estimation[1] using the ADS-B messages is capable of mitigating some of the error in the transmitted measurements. The primary goal of estimation is to account for missed measurements that result from signal drop out or frequency congestion. Additionally, by filtering and estimating ADS-B measurements, it is possible to account for grossly erroneous measurements such as would be occasionally permitted through the SIL and SDA probabilities, smooth measurement noise that is typical of any real sensor, and estimating the transmitting aircraft state at a rate greater than the 1 Hz measurement rate [67]. Due to the fact that ADS-B messages contain an aircraft identifier such as the call sign or International Civil Aviation Organization (ICAO) address, there is no need for data association methods. This greatly simplifies the tracking task.

We use a Kalman filter to process ADS-B In tracks. The Kalman filter is an optimal linear estimator that is computationally efficient. The prediction model in our implementation is a constant-jerk model capable of accounting for high maneuverability of the intruders [89]. While it is not expected that fixed-wing aircraft will maneuver aggressively, more aggressive maneuvers such as would be characteristic of a rotor-craft or small UAS must also be accounted for in the model. The states of the filter are position North, position East, altitude, velocity North, velocity East, climb rate, acceleration North, acceleration East, vertical acceleration, jerk North, jerk East, and vertical jerk. The measurements used to update the estimator states are the position North, position East, altitude, and climb rate. Recorded ADS-B data sets from the NAS have revealed that on rare occasions the North and East velocities are transmitted in reverse order resulting in an apparent velocity that is perpendicular to the actual direction of travel of the transmitting aircraft. Updating the Kalman filter with only a subset of measurements mitigates this problem and results in equally accurate estimation after a brief transient estimation period of several measurements.

Each transmitting aircraft broadcasts an ADS-B message approximately once per second; however, the broadcasts can occur at any point within a given second. Thus the

[1]This subsection is primarily written by Matthew O. Duffield, and Timothy W. McLain as part of the journal manuscript published in the Air Traffic Control Quarterly: An International Journal of Engineering and Operation [117].

Kalman filter must run at a higher rate than 1 Hz to account for the different times at which a transmission may be received. Our Kalman filter implementation runs at 10 Hz, where received ADS-B messages are assigned to the nearest discrete time-step.

A set of measurement gates is necessary to account for message dropout and grossly erroneous measurements. If at a given time step there is no measurement, only the Kalman filter prediction step is instantiated. The measurement update step occurs only when there is a valid measurement. The validity of the horizontal position and altitude measurements is determined separately due to the fact that in ADS-B messages the horizontal position and altitude can be updated at different times. A horizontal position is determined to be valid if it is confirmed to be a new position and if the innovation falls within a Mahalinobis distance bound of 5σ. An altitude/climb rate measurement is valid only if it falls with in a Mahalinobis distance bound of 5σ.

Each track is initialized using the first measurement from a given transmitting aircraft. The initial track covariance is initialized using the error levels given by the reported NACp and NACv and the error characterization described earlier. At each time step, the track covariance is monitored to ensure that the track is still valid. If the covariance of the track grows such that the position uncertainty in the track is greater than the NIC bound, then the track is determined to be invalid. Should another measurement from that aircraft be received, the track would be re-initialized.

The Kalman filter is capable of overcoming ADS-B message drop out and rejecting grossly erroneous measurements. Additionally it smooths the ADS-B signal and provides estimates of transmitting aircraft at a much faster rate than the 1 Hz measurement update rate. This ultimately allows for more accurate and more timely conflict/collision detection and resolution.

7.2.2 Conflict/Collision Risk Assessment

As discussed in Chapter 1, the collision volume or the protection zone is a virtual fixed-volume-based boundary. The common choice for this volume is a cylinder of 500 ft in radius and $\pm$100 ft. As depicted in Figure 7.2 the collision volume threshold or the well clear boundary is a variable boundary around the ownship, larger than the collision volume and

depends not only on distance, but time, and maneuverability [31, 49]. It is designed to give a practical buffer such that if a collision is predicted then the ownship has adequate time to plan an avoidance maneuver and to actually fly the maneuver. The design of the collision avoidance threshold is beyond the scope of this dissertation. We refer the interested reader to [30, 34]. A self-separation volume is added to the airspace volumes to provide a minimum practical separation distance between the UAS and any intruder, and to compensate for unexpected maneuvers by the intruders as shown in Figure 7.2 [30]. In the current Air Traffic Management System, a conflict is defined when an aircraft encounter happen with 3-5 nautical miles of one another horizontally and within 2000 ft above an altitude level of 29,000 ft and 1000 feet below the 29,000 ft level.

The self-separation volume is typically much larger than the collision volume but it may vary in size with operational area and airspace class. A conflict occurs when another aircraft penetrates the self-separation volume. The self-separation threshold is then defined as the threshold boundary at which the UAS performs a maneuver to prevent an intruder from penetrating the self-separation volume. Hence, the addition of the self-separation volume provides a performance goal that is analogous to the collision volume.

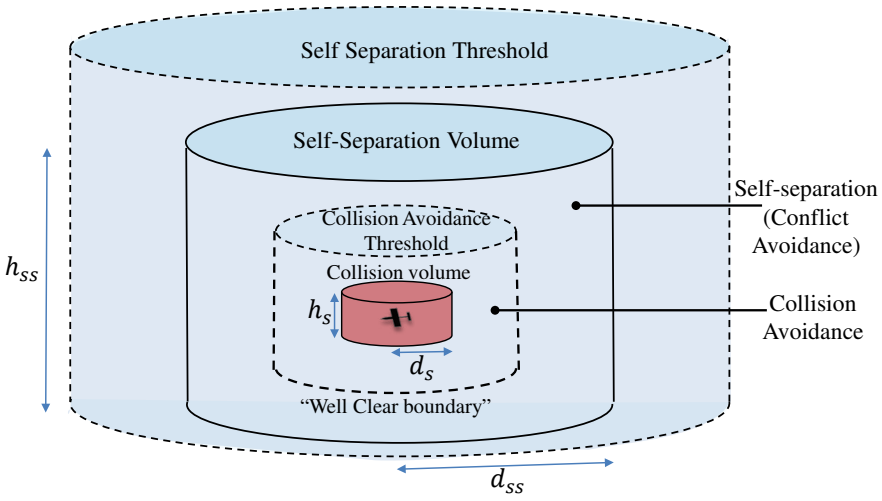


Figure 7.2: Definition of the SAA airspace volumes and thresholds.

The encounter geometry is evaluated in the relative coordinate frame where the relative motion of the aircraft is analyzed by investigating the dynamics of the intruder aircraft

with respect to the ownship. The proposed conflict/collision detection framework is based on evaluating both near-term threats that need immediate action and long-term conflicts that can be smoothly resolved so that they will not become a close proximity threat.

Algorithm 4: Conflict detection algorithm

```
for each detected intruder do
    if ‖p_r‖ < d_ss_th or (p_r^⊤ v_r < 0 and t_cpa ≤ τ_ss_th) then
        if h_r < h_ss_th/2 or (p_rz v_rz < 0 and t_vent ≤ τ_ss_th) then
            if d_cpa ≤ d_ss and t_ss_vext ≥ t_ss_ent then
                Conflict is detected.
            end if
        end if
    else
        No conflict is detected.
    end if
end for
```

To predict whether or not a collision will occur, we use the collision logic given in Algorithm 1. The proposed collision detection approach constructs a virtual volume surrounding the ownship UAS, that, when penetrated by the intruder, declares a collision. The logic in Algorithm 1 can be easily modified to detect conflicts by changing the horizontal and vertical safe distances to be the radius and the height of the self-separation volume to satisfy the well clear boundary. The conflict detection logic given in Algorithm 4. In detecting conflicts, we adopt the well clear candidate selected by the Sense and Avoid Research Panel (SARP) for recommendation to the sense and avoid stakeholders upon extensive analysis [34]. The candidate well clear boundary is defined by a truncated cylinder that consists of a modified τ value of 35 s with distance of 4000 ft in the horizontal plane, and a vertical distance from the ownship of 700 ft. The design parameters $d_{ss_{th}}$, $h_{ss_{th}}$, and $\tau_{ss_{th}}$ determine the boundaries of the self-separation threshold volume and how early an the system should be alerted. In computing the horizontal entry time, $t_{ss_{ent}}$ and the vertical exit time, $t_{ss_{vext}}$ into the self-separation volume in Algorithm 4, the variables d_s and $h_s/2$ in Eqs. (3.12) and (3.22) are substituted with the radius and height of the self-separation volume d_{ss} and h_{ss}, respectively.

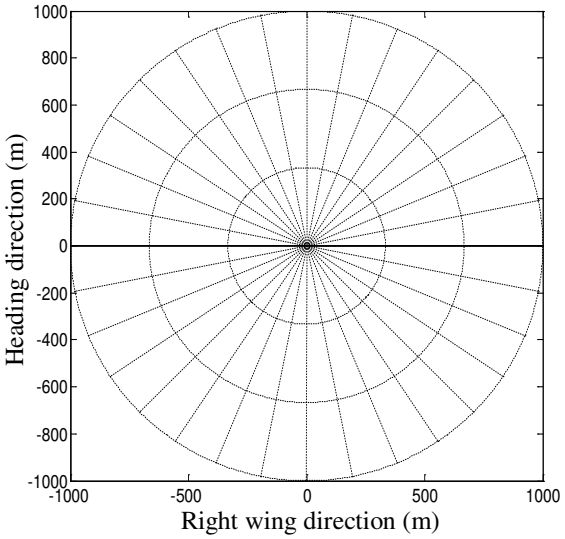
Algorithm 5: Conflict detection algorithm with uncertainty

```
for each detected intruder do
  if ‖p_r‖ < d_ss_th or (p_r^⊤ v_r < 0 and t_cpa ≤ τ_ss_th) then
    if h_r < h_ss_th/2 or (p_rz v_rz < 0 and t_vent ≤ τ_ss_th) then
      if d_cpa − m_1 σ_dcpa ≤ d_ss and (t_ss_vext + m_2 σ_ssvext) ≥ (t_ss_ent − m_2 σ_ssent) then
        Conflict is detected.
      end if
    end if
  else
    No conflict is detected.
  end if
end for
```

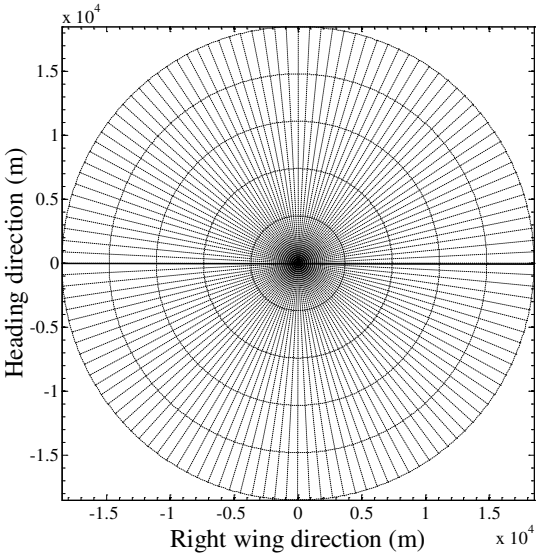
Since there will always be uncertainty associated with intruder state estimates, an additional provision should be made to account for it. Therefore, Algorithm 2 is used to detect collisions, and Algorithm 4 is modified by Algorithm 5 to account for the uncertainties in the state estimates. In Algorithm 5, the parameters m_1, m_2 are positive integers. The parameters σ_{dcpa}, σ_{ssvext}, and σ_{ssent} are the standard deviation of the distance at closest point of approach, vertical exit time and horizontal entry time into the self-separation volume. If we assume, for convenience, that aircraft in conflict use the same type of ADS-B sensor and that they are observing mostly the same GPS satellites, then both aircraft will experience the same position and velocity error accuracy. In addition, if we assume that the NE components of the GPS horizontal position and velocity measurements are independent random variables, then the error variances of the horizontal position and velocity measurements, denoted as σ_p and σ_v respectively, are the same for both aircraft. It can be shown that the variance of the distance at CPA is $\sigma^2_{dcpa} = 2\sigma^2_p + 2t_{cpa}\sigma^2_v$ [66].

7.3 Self-Separation/Collision Avoidance

Since the collision-detection metrics are derived from the relative position and velocity vectors between conflicting aircraft, it is convenient to develop collision avoidance logic using a local coordinate system. We have used the local-level collision avoidance approach discussed in Chapter 4 to plan avoidance maneuvers when encountering intruder at close proximity. To handle conflicts at long-range, the resolution algorithm should plan smooth maneuvers.



(a) Top view:Local-level frame for collision avoidance, with d_r =1000 m, three levels and 39 nodes including start and final nodes.



(b) Top view:Local-level frame for self-separation, with d_r =18.5 km, five levels and 151 nodes including start and final nodes.

Figure 7.3: Local-level reference frame in collision avoidance and self-separation.

This can be achieved by increasing the resolution of the discretized map or by using smooth path parameterizations like Dubins paths.

As shown in Figure 7.3(a), the increased resolution of the local-level reference frame is obtained at the detection range of 18.5 km offered by the ADS-B sensor. The map is discretized into five level curves with 151 nodes. Similar to the collision avoidance method, the cost of edges is assigned according to Eq (4.50) and Dijkstra's algorithm is also used to find least cost path from the start point to the destination point.

7.4 Monte Carlo Simulation Results

To validate the performance of the presented ADS-B sensor model, estimation scheme, conflict/collision detection, and self-separation/avoidance approaches, we conducted two separate sets of Monte Carlo simulations to address encounter scenarios over short and long ranges. We developed a simulation environment with a six-degree-of-freedom aircraft model for both the ownship and the intruders. The state estimates of the intruders are provided by the ADS-B sensor, while we assume a perfect knowledge of the ownship states. To avoid simulating encounters that are unlikely to result in a collision or loss of self-separation, we fo-

cus on encounters that occur in an encounter circle centered on the ownship. The encounter geometry is constructed using an approach similar to that suggested by [61]. As shown in Figure 7.4, the encounter circles used for simulating short and long range encounter scenarios have radii of 1.62 nmi and 10 nmi respectively. For both sets of Monte Carlo simulations, the ownship is initialized at the center of the encounter circle and follows a straight-line waypoint path. The ownship initial heading is zero, and the initial airspeed is set to 80 kn, which is similar to the maximum airspeed of the ScanEagle UAS. At the beginning of each simulation, the first intruder is initialized at one of 20 evenly spaced points on the perimeter of the encounter circle, while other intruders are initialized by sampling a uniform distribution over the remaining points. In addition, the intruders are initialized with random headings that are required to penetrate the encounter region at the initial time. The speed of the intruders is randomly drawn from a uniform distribution over $[39, 250]$ kn. The 39 kn lower bound is approximately the minimum airspeed of a small UAS, and the 250 kn upper bound is the maximum allowed airspeed of an aircraft flying below 10,000 ft above MSL. Most potential applications of small UAS will likely operate below 10,000 ft. For the short-range encounter scenarios, 1000 Monte Carlo runs are conducted at each discrete point on the encounter circle. In the long-range encounter scenarios, 100 Monte Carlo simulation runs are conducted for each discrete point on the encounter circle. In both sets of experiments, the Monte Carlo simulations are repeated to include multiple intruder scenarios.

We have implemented the conflict/collision detection method given by Algorithms 2 and 5. In the long-range simulations, the threshold boundary value that defines a conflict is $\tau_{th} = 45$ s with a distance threshold of $d_{th} = 5$ nmi in the horizontal plane and $h_{th} = 1000$ ft in the vertical plane. To define loss of separation, we consider a well clear boundary defined by $\tau = 35$ s with distance thresholds of 4000 ft in the horizontal plane and 700 ft in the vertical plane [34]. In the short-range simulations, the well clear boundary serves as the threshold boundary so that the collision detection algorithm activates the collision avoidance planner once the well clear boundary is penetrated. This ensures that collision avoidance is triggered when the self-separation mode fails to maintain the well clear distance from the intruding aircraft. Our choice of collision volume is a truncated cylinder of radius 500 ft and height of

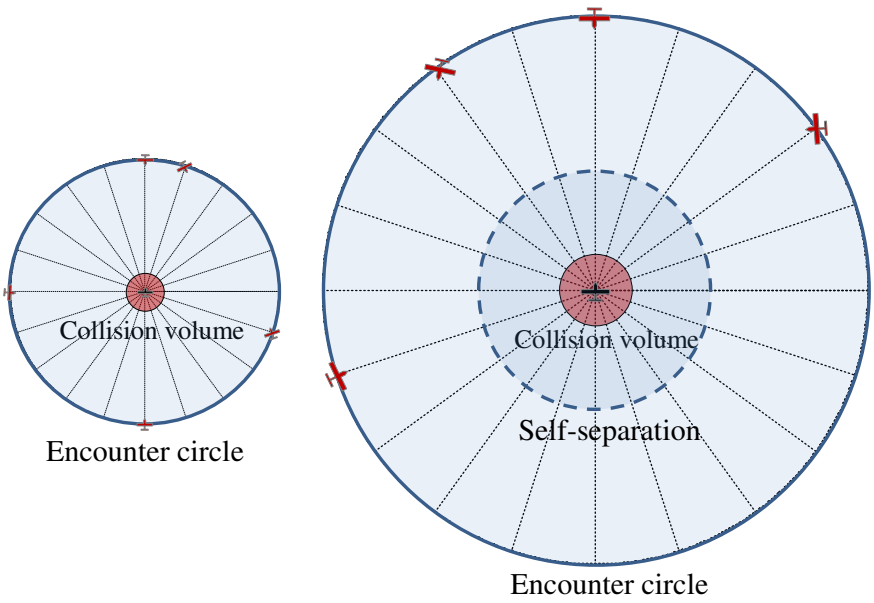
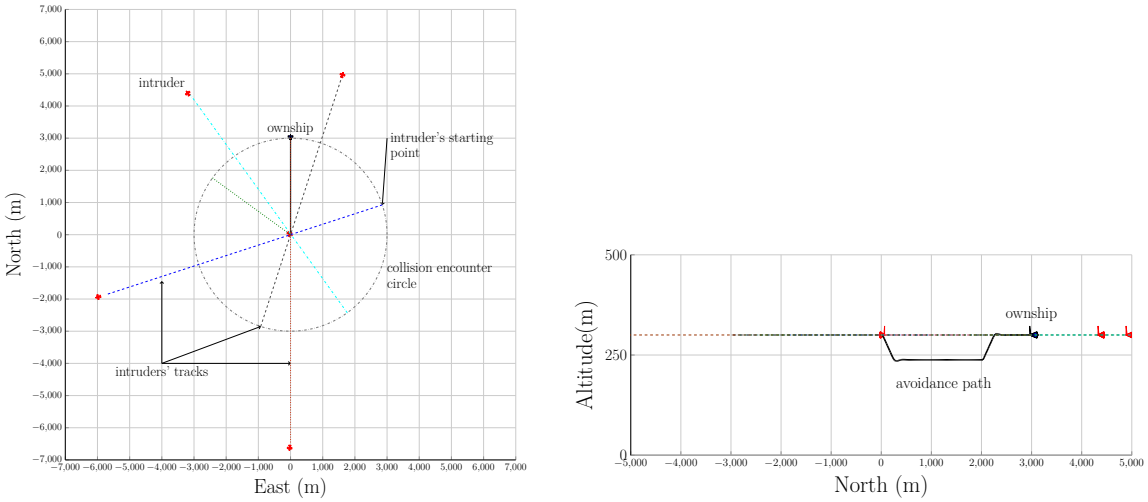


Figure 7.4: Encounter geometries: Short and long range encounter scenarios have radii of 1.62 nmi and 10 nmi, respectively.



(a) Side view of avoidance path.

(b) Overhead view of avoidance path.

Figure 7.5: Example of a collision avoidance encounter scenario.

200 ft. The design parameters used in Algorithm 2 and 5 are

$$m_1 = m_2 = 1,$$

$$\sigma_p = 124 \text{ ft, and } \sigma_v = 8 \text{ kn,}$$

$$\sigma_{p_{rz}} = 75.9 \text{ ft, and } \sigma_{v_{rz}} = 27.96 \text{ ft/min.}$$

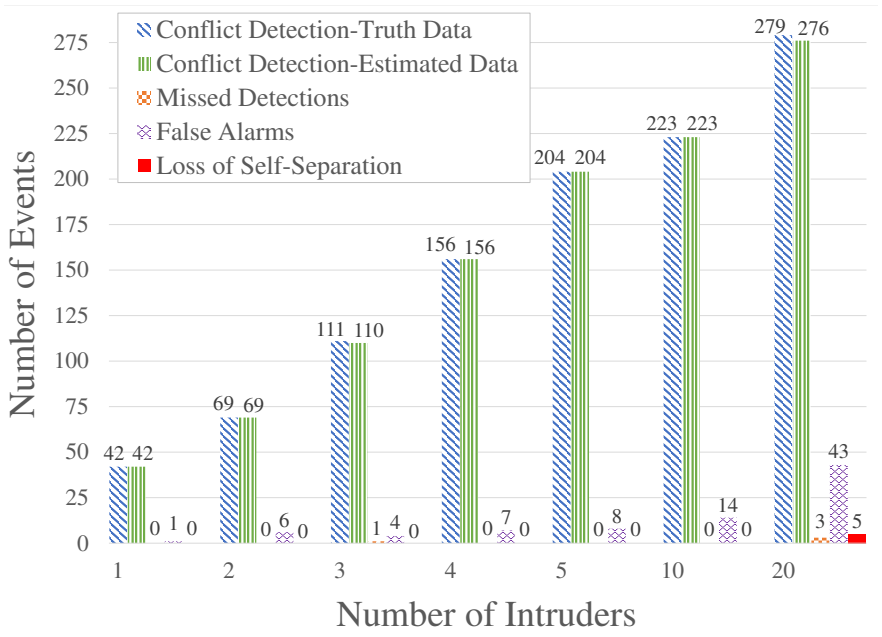


Figure 7.6: Self-separation evaluation metrics of ADS-B-based conflict detection and self-separation algorithms using 2,000 encounter scenarios for each intruder set (a total of 14,000 encounters).

In each Monte Carlo simulation we have also implemented the conflict/collision detection algorithm in Algorithms 1 and 4 to produce the intruder truth tracks needed to calculate the true number of conflict and collision events. The number of true conflict/collision detections is compared to the number of conflict/collision detections that are predicted using Algorithms 2 and 5, which is based on the track estimates of the intruders. We have also recorded the number of collisions and loss of separation incidents for each encounter scenario given that the self-separation/collision avoidance algorithm will be activated once an intruder crosses the self-separation/collision avoidance threshold boundary. We have implemented the local-level frame path planner approach presented previously in the Self-Separation and Collision Avoidance section. The local-level frame map is discretized into three levels with 38 nodes for the collision avoidance algorithm and five levels with 150 nodes for the self-separation algorithm. An example of a collision avoidance scenario with 5 intruders is shown in Figure 7.5. The correct detections, missed detections, and false alarms for both the conflict and collision scenarios are shown in Figures 7.6 and 7.7.

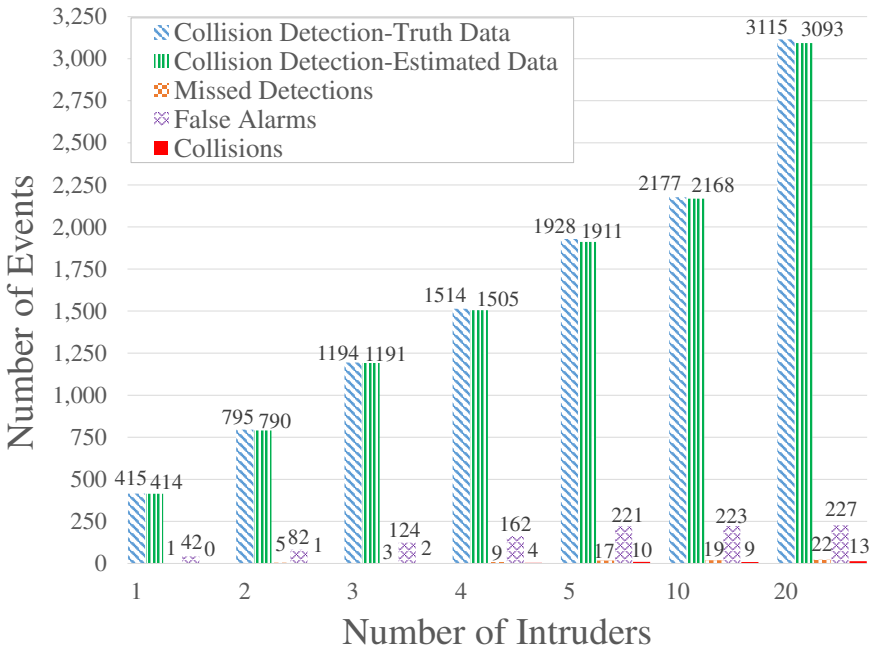
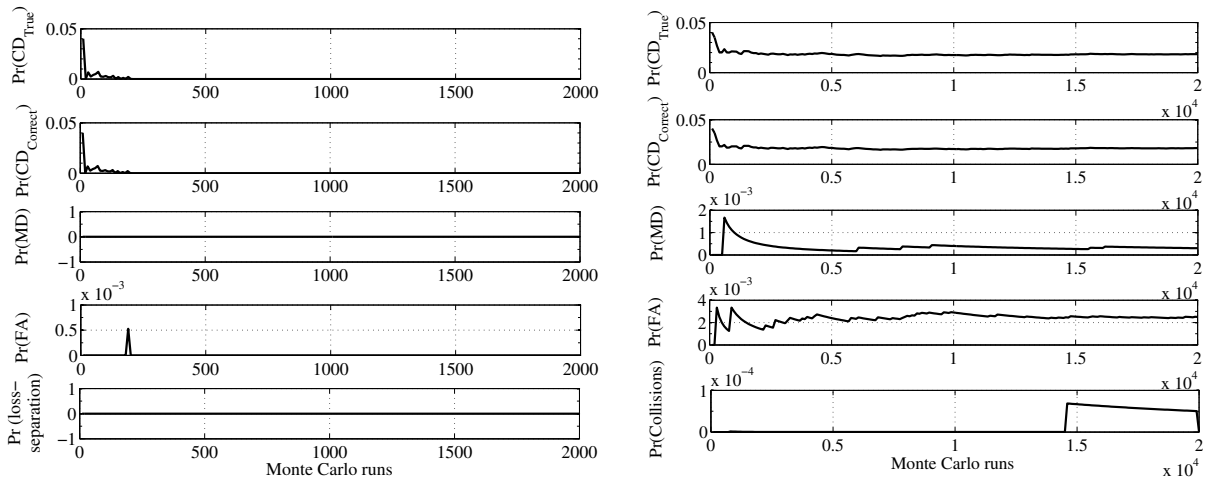


Figure 7.7: Collision avoidance evaluation metrics of ADS-B-based collision detection and avoidance algorithms using 20,000 collision encounter scenarios for each intruder set (a total of 140,000 encounters).

Figure 7.6 shows results from the long-range self-separation assurance simulations, and Figure 7.7 contains results from the short-range tests for collision avoidance. In each figure five metrics are presented for the different numbers of intruders considered. Conflict Detection True refers to the number of collisions detected using truth information for each of the intruders. Missed Detections refers to the number of collisions detected using intruder truth information that were not detected using estimated intruder information. Conflict Detection Correct describes the number of conflicts detected using estimated information that were also detected using truth information. The relationship between Conflict Detection True, Missed Detections, and Conflict Detection Correct can be expressed as $CD_{True} = MD + CD_{Correct}$. False Alarms is the number of collisions detected using estimated information that were not detected using truth information. Finally, Loss of Self-Separation and Collisions are the number of instances that the well clear or collision volumes were penetrated by an intruder.

As shown in Figures 7.6 and 7.7, both conflict and collision detection algorithms are able to correctly detect conflict and collision events with a small number of missed



(a) Self-separation: The 5 intruders scenario. (b) Collision avoidance: The 5 intruders scenario.

Figure 7.8: Statistical Convergence of Monte Carlo Simulation.

detections and few false alarms. The increasing number of conflict and collision incidents with the increasing number of intruders is expected since greater numbers of intruders in the encounter circle result in a higher probability of a conflict or collision with the ownship. Figure 7.8 shows the statistical convergence of the Monte Carlo simulations which indicates a sufficient number of simulation runs.

Table 7.2: Conflict Detection Algorithm: Probability of correct detection, probability of false alarm, and safety ratio.

Scenario	P_{cd}	P_{fa}	Safety Ratio
1 intruder	1.000	0.00051	0.000
2 intruders	1.000	0.00153	0.000
3 intruders	0.990	0.00068	0.010
4 intruders	1.000	0.00089	0.000
5 intruders	1.000	0.00082	0.000
10 intruders	1.000	0.00041	0.000
20 intruders	0.989	0.00206	0.011

The performance of the proposed conflict/collision detection approach is quantified using the probability of correct detection P_{cd} and the probability of false alarm P_{fa} [69, 118].

Table 7.3: Collision Detection Algorithm: Probability of correct detection, probability of false alarm, and safety ratio.

Scenario	P_{cd}	P_{fa}	Safety Ratio
1 intruder	0.998	0.00214	0.002
2 intruders	0.994	0.00209	0.006
3 intruders	0.997	0.00211	0.003
4 intruders	0.994	0.00206	0.006
5 intruders	0.991	0.00225	0.009
10 intruders	0.995	0.00181	0.005
20 intruders	0.993	0.00115	0.007

We further quantify the system performance by computing the safety ratio [69]

$$\text{Safety ratio} = \frac{1 - P_{cd}}{1 - P_{fa}}.$$

The P_{cd} and P_{fa} results for both collision and conflict detection are shown in Tables 7.2 and 7.3. An ideal conflict/collision detection algorithm would result in $P_{fa} = 0$ and $P_{cd} = 1$, with all conflicts/collisions threats correctly detected and no false alarms. This corresponds to the theoretical ideal point (P_{fa}, P_{cd}) =(0,1) in signal detection theory. Unfortunately, due to uncertainty in the intruder state estimates, this ideal is not achievable. The closer the value of (P_{fa}, P_{cd}) to the point (0,1), however, the better the detection performance. In addition, a safety ratio of 0 indicates that the detection system provides perfect protection from loss of separation/collision incidents. A safety ratio of 1, however, indicates that the detection system provides no additional protection from loss of separation/collision incidents. The results shown in Tables 7.2 and 7.3 demonstrate that our approach produces results near the ideal operation point (P_{fa}, P_{cd}) =(0,1) with a safety ratio that is near zero.

The conflict/collision detection algorithm can be also evaluated by measuring the time delay between a conflict/collision event that has been detected using the intruder truth information, and the time instant at which the proposed conflict/collision detection algorithm is able to detect the same event using estimated intruder information. Ideally, the time delay should be zero. Due to the errors in the state estimates of the intruders, however, this cannot be always achieved. The average of the maximum, the 95th percentile, and the maximum of

Table 7.4: Conflict Detection Algorithm: Event detection time delay.

Scenario	Average max. time delay (s)	95th percentile	Max. time delay (s)
1 intruder	1.389	11.280	12.9
2 intruders	1.308	8.900	13.7
3 intruders	1.370	8.160	24.9
4 intruders	1.444	6.255	36.9
5 intruders	1.148	5.200	18.9
10 intruders	1.848	4.770	16.9
20 intruders	1.387	7.200	17.4

Table 7.5: Collision Detection Algorithm: Event detection time delay.

Scenario	Average max. time delay (s)	95th percentile	Max. time delay (s)
1 intruder	2.010	9.990	18.2
2 intruders	2.348	9.900	31.9
3 intruders	2.219	9.900	35.9
4 intruders	2.354	10.245	31.9
5 intruders	2.190	9.900	28.5
10 intruders	2.141	10.10	31.9
20 intruders	1.956	9.900	63.9

simulation time delays is shown in Tables 7.4 and 7.5. If we assume, an ownship UAS similar flying with maximum airspeed of 80 kn encounters an intruder with a maximum airspeed of 240 kn, then the time to loss of separation is 105.5 s given 10 nmi detection range [117]. Then from Table 7.5, the average of the maximum time delay to detect a conflict for all of the seven different scenarios is 1.128 s which leaves the ownship with about 104.372 s to plan and fly an avoidance maneuver. Even with the maximum time delay recorded (36.9 s), the ownship still has about 68.6 s to plan and execute an avoidance maneuver.

Another important aspect of evaluating the performance of the proposed approach is its ability to run in real time. The SAA system must be able to process the measurement information provided by the sensor, provide estimates of the current states of the intruders, assess the encounter risk, and plan an avoidance path, while leaving ample time to avoid

Table 7.6: Conflict Detection: Execution time for one cycle of estimator, and conflict detection, and self-separation algorithms.

Scenario	Average max. computation time (s)	95th percentile	Max. computation time (s)
1 intruder	0.218	0.292	0.363
2 intruders	0.387	0.523	0.649
3 intruders	0.519	0.627	0.645
4 intruders	0.521	1.022	2.131
5 intruders	0.578	0.930	2.403
10 intruders	0.613	0.900	0.929
20 intruders	1.116	1.411	20.572

Table 7.7: Collision Detection: Execution time for one cycle of estimator, and collision detection, and avoidance algorithms.

Scenario	Average max. computation time (s)	95th percentile	Max. computation time (s)
1 intruder	0.108	0.198	0.386
2 intruders	0.108	0.329	0.392
3 intruders	0.072	0.079	0.140
4 intruders	0.014	0.026	0.734
5 intruders	0.019	0.042	0.878
10 intruders	0.071	0.093	0.289
20 intruders	0.126	0.233	0.979

the intruder and ensure that the constraint on the minimum required safe distance from the intruder is not violated. To demonstrate this capability, we have recorded the time required to execute these processes from the point the measurements are provided to the estimator until a new waypoint command is generated by the path planner and is ready to be delivered to the ownship autopilot. The average, 95th percentile, and the maximum time required to execute one cycle of the estimation, the conflict/collision detection, and the self-separation/collision avoidance algorithm are shown in Tables 7.6 and 7.7. The algorithms were run using Matlab on an Intel i7 processor. The ADS-B estimator provides estimates at a sample rate of 0.1 s. To process all of the estimator data, the other algorithms should operate at the same sample rate. Table 7.6 shows that the average of the maximum run time

to execute the estimator, conflict detection, and the self-separation algorithms for all of the seven different scenarios is 0.565 s, and the maximum recorded run time is 20.572 s. Similarly, in Table 7.7 the average of the maximum run time of the estimator, collision detection, and avoidance algorithms for all of the seven different scenarios is 0.074 s, while the maximum recorded run time is 0.979 s. Although the recorded run times are larger than the 0.1 s, these algorithms are running near real time in Matlab. We expect that implementing these algorithms in a compiled language, such as C or C++, will show that real-time execution is feasible using low-cost computational hardware.

In these experiments, the collision volume is large (500 ft radius, 200 ft height) compared to the size of the ownship aircraft. If the collision volume was reduced to more closely reflect the aircraft size, the number of collisions detected drops even further. For example, Figure 7.7 shows a total number of 10 collisions out of the 1928 potential collisions detected using intruder truth information for the five intruder scenario case. If the collision volume is reduced to a third of the original size, then the number of collisions decreases to only one. If the collision volume is further reduced to a cylinder of radius 10 ft, approximating the wingspan of a small UAS, and height of 3 ft, we record zero collisions. From this we can conclude that none of these collision incidents were actual collisions, but only a violation of the surrounding collision volume.

7.5 Conclusions

We have introduced in this chapter an ADS-B-based collision detection and avoidance approach that is technically and computationally feasible for small UAS. Taking advantage of the long-range detection offered by the ADS-B system, the proposed approach is used to detect conflict events over the range of 5 to 10 nmi, and consequently provide self-separation of the host ownship.

The results from Monte Carlo simulations show that the proposed system supports a high level of safety. In the 14,000 executed self-separation simulations, separation was fully maintained up to 10 intruders. For the 140,000 collision scenarios, the number of collision-volume violations was 39, with no actual physical collisions. The run-time results also indicate that the estimation scheme, detection method, and avoidance logic are capable

of running in near real time in Matlab. Considering the increase in computational efficiency that would result from a compiled-code implementation of the algorithm, real-time execution can be easily achieved.

A key result of this work is that it demonstrates that mandating ADS-B Out equipage for all aircraft could play a significant role in the safe integration of UAS into the NAS. While economic, policy, and operational aspects still must be considered, the theoretical basis and positive simulation results presented here demonstrate the technical feasibility of the proposed system. While the FAA 2020 mandate for ADS-B includes only some aircraft, requiring all aircraft to have ADS-B Out could create a suitable environment for implementation of ADS-B-based SAA. Ultimately, the combination of complete ADS-B equipage and SAA systems like the one presented here would be a major step toward allowing UAS to safely operate in the NAS with manned aircraft.

Ultimately, any proposed SAA system needs to coordinate with existing cooperative systems such as TCAS if required to fly alongside TCAS-equipped aircraft. Fortunately, ACAS X is the likely long-term replacement for TCAS, and its surveillance architecture is designed to support surveillance based on GPS data that significantly simplifies development and interoperability with ADS-B equipped UAS.

Chapter 8. Ground-Based Radar SAA for Small UAS

To most people, the sky is the limit. To those who love aviation, the sky is home.

- anon

In this chapter, we present a collision detection and avoidance approach for ground-based SAA systems. We first evaluate the performance of the proposed approach assuming an generic sensor that provides complete and perfect knowledge of the environment. Second, we propose a portable and low-cost ground-based radar system and we use the recursive random sample consensus (R-RANSAC) tracking algorithm to test the performance of the collision detection and avoidance algorithms both in simulation and flight tests. Figure 8.1 depicts a broad application scenario which is used to motivate the work of this chapter. A typical mission requires the ownship to fly from point A to point D and return back to point A. In this work, however, we only consider planing a collision-free path between points B and C.

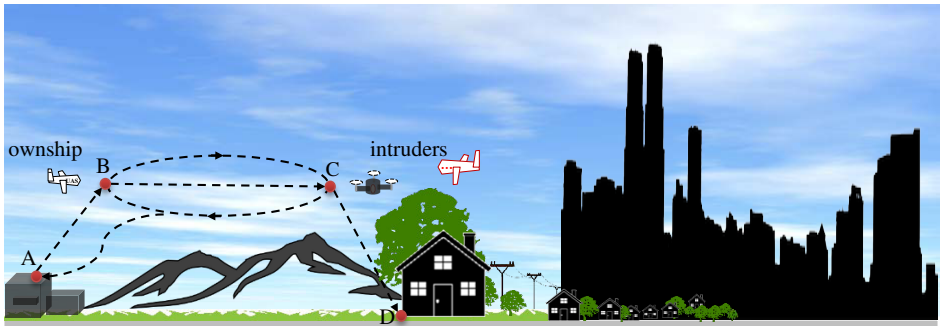


Figure 8.1: An example of a typical mission for small UAS.

8.1 Introduction

Generally, the most suitable SAA configuration depends on the dynamic characteristics, SWaP, payload of the UAS airframe, and the nature of the mission. In this chapter, we present a ground-based sense-and-avoid system as an alternative option to the airborne SAA system. As discussed in Chapter 1, the hardware and algorithms necessary for small UAS sense-and-avoid system make up a large portion of the available size, weight, and power resources. Scaling sensors down to small UAS sizes often requires compromises in range, field of view, measurement accuracy, or processing speed. Such compromises reduce the overall capability of the SAA system, and consequently, decrease the assurance of safety. In addition, carrying sensors on board reduces the UAS payload capability. The ground-based SAA system provide an alternative means of complying with the FAA sense-and-avoid regulations. A ground-based SAA system consist of a ground control station that includes all sensors, communication, processing and logic.

Ground sensors, primarily radars, are an alternative to supporting small UAS operations, which do not require modifications to the UAS airframe, and are not restricted by SWaP constraints. A ground-based SAA system, however, may consists of several sensors like the mobile aircraft tracking system (MATS) [133]. The MATS consists of a 2D primary radar, which provides range and azimuth information about targets, an ADS-B receiver and a transponding interrogator. The primary radar of the MATS has a peak output power of 25 kW and provides two modes of instrumented range: 54 nmi with resolution of 180 m or 27 nmi with resolution of 45 m. The performance of the MATS was examined as part of the Smart Skies project using a specially equipped Cessna 172R. The main function of MATS is to detect and track intruding aircraft and provide this information to the UAS pilot located at the ground control station. In other words, the MATS system provides the sense element while the pilot evaluates the risk and performs the avoidance function if needed. Another example is the modern Thales Star 2000 air traffic control (ATC) radar that has a peak output power of 28 kW and provides a coverage of 100 nautical miles with resolution of 230 m. Similar to MATS it provides only range and azimuth information about targets. Not providing the elevation information means that there is no altitude information, which is an essential part of navigation. The MATS and Star 200 system assume low density traffic

where 2D information is sufficient and another cooperative sensor provides the altitude information to supplement the radar data [11]. The sense-and-avoid display system (SAVDS) which uses the Sentinel AN/MPQ-64 air defense radar is a ground-based SAA system that is capable of providing 3D information about targets. It has a peak output power of 23 kW and provides a coverage of 40 nmi with resolution of 150 m [11]. In general, traffic control radar and high-tech military radars tend to be very expensive. For example, the cost guide of the terminal maneuvering area radar is 8 million US dollars [2]. A drawback to using the ground-based radar as an SAA system is that it may be difficult to provide a static coverage volume that is larger than the operating range of the UAS. Also, using ground-based radar introduces the issue of maintaining a reliable, and efficient data link between the ground control station and the ownship. In addition, local terrain may also reduce the surveillance volume, and introduce noise in the measured information.

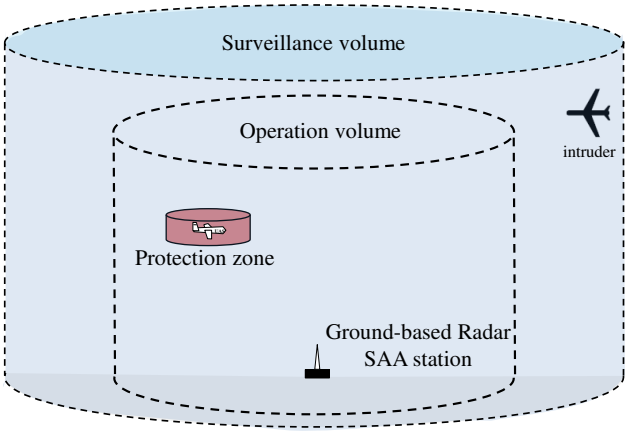


Figure 8.2: Surveillance and operating volumes associated with the ground-based radar SAA system.

Figure 8.2 shows the typical operating volumes associated with a ground-based SAA system. In this configuration, the ground radar detects air traffic in a fixed volume of airspace called the surveillance volume. The ownship flies in a volume of airspace referred as the operation volume. The size and geometry of the operation volume is dependent on the surveillance volume, minimum detection range, and other dynamics characteristics of the UAS, like the minimum turning radius. The size of the operation volume should depend on (1) the minimum required detection range to be able to detect and track the intruder,

(2) the time required to evaluate the encounter scenario, (3) the time required to plan an avoidance maneuver if required, and (4) the time required to take an evasive action.

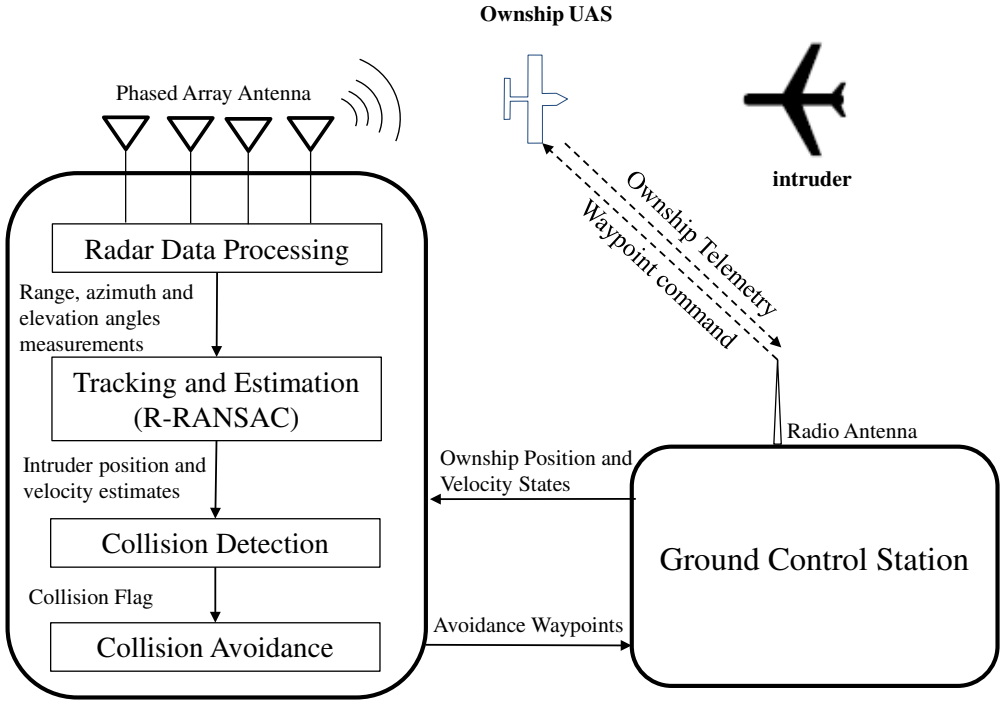


Figure 8.3: Ground-based radar sense-and-avoid system structure diagram.

As shown in Figure 8.3, radar returns from all of the targets are received by a phased array antenna. The radar data is processed to produce ranges and relative bearing angles to all targets. In the tracking step, the target's measurements are processed using the R-RANSAC algorithm [100] to estimate the state estimates of potential intruders, and to distinguish the ownship. After the R-RANSAC filtering, the intruder position and velocity estimates are projected forward in time to identify possible collisions. If a collision threat is detected, the intruder position and velocity estimates and an activation flag are passed into the collision avoidance algorithm. Once the collision level of the avoidance logic has been activated, a new collision-free path is generated. To evaluate encounter situations we use a deterministic approach that computes distance at the closest point of approach. For collision avoidance, we propose a two step collision avoidance algorithm. In the first step, an initial suboptimal path is generated using A* search. In the second step, a simulated

chain of unit masses connected by springs and dampers evolves in a simulated force field. The chain is described by a set of ordinary differential equations that is driven by virtual forces to find the steady-state equilibrium. The ultimate output of the SAA system is a revised set of ownship waypoints that are transmitted to the ownship. The system shown in Figure 8.3 is a complete SAA system for small UAS. It is viable for both fixed wing and multirotor aircraft, and could reasonably be extended for larger UAS outside of the small UAS definition.

8.2 Collision Detection

In this section we propose a simple collision detection logic that evaluates intruders at close proximity based on the geometric relationship between aircraft. In the geometric approach, aircraft trajectory predictions are based on linear projections of current aircraft states forward in time to identify possible future collisions. Linear projections can be computationally efficient and prediction errors are negligible over short time horizons. We assume that there exists a sensor and tracking system that provides the intruders states. We also assume that detected intruders fly at a constant velocity, constant altitude, and follow a straight line path during the entire encounter scenario. Our approach is based on evaluating near-term threats that need immediate action. The proposed collision detection approach constructs a virtual volume surrounding the ownship that, when penetrated by an intruder, declares a collision. The general choice for this volume is a truncated cylinder of radius d_s and height h_s centered at the current location of the UAS. The current manned regulations have no explicit values for the collision volume, however d_s= 500 ft and h_s= 200 ft are generally used in the literature [11].

The encounter geometry is evaluated in the relative coordinate frame where the relative motion of the aircraft is analyzed by investigating the dynamics of the intruder with respect to the ownship. In this relative reference frame, the ownship is stationary and the intruder's location and velocity is determined by the relative position and velocity states. To be more specific, let the jth intruder's position and velocity in the inertial NED reference frame at time t be $\mathbf{p}_{int_j}(t) = (n_{int_j}, e_{int_j}, d_{int_j})^\top$ and $\mathbf{v}_{int_j}(t) = (vn_{int_j}, ve_{int_j}, vd_{int_j})^\top$, respectively. Assume that the ownship's position and velocity are known as $\mathbf{p}_{own}(t) = (n_{own}, e_{own}, d_{own})^\top$

and $\mathbf{v}_{own}(t) = (vn_{own}, ve_{own}, vd_{own})^\top$. Let $\mathbf{p}^*_{int_j}(t) = (n_{int_j}, e_{int_j})^\top$ and $\mathbf{v}^*_{int_j}(t) = (vn_{int_j}, ve_{int_j})^\top$ be the projection of the jth intruder's position and velocity onto the NE-plane. Similarly, we let $\mathbf{p}_{own}(t) = (n_{own}, e_{own})^\top$ and $\mathbf{v}_{own}(t) = (vn_{own}, ve_{own})^\top$ be the projection of the ownship's position and velocity onto the NE-plane. The relative horizontal position and velocity between the jth intruder and the ownship is defined as

$$\mathbf{p}_{r_j} = \mathbf{p}^*_{int_j}(t) - \mathbf{p}_{own}(t), \tag{8.1}$$

$$\mathbf{v}_{r_j} = \mathbf{v}^*_{int_j}(t) - \mathbf{v}_{own}(t). \tag{8.2}$$

We further calculate the relative altitude between the jth intruder and the ownship as

$$h_{r_j} = d_{int_j} - d_{own}. \tag{8.3}$$

Assume that the intruder moves with a constant velocity. Using $\mathbf{p}_r$ and $\mathbf{v}_r$, we predict the time remaining to the closest point of approach (CPA) between the jth intruder and the ownship as

$$t_{cpa_j} \triangleq \begin{cases} \frac{-\mathbf{p}_{r_j}^\top \mathbf{v}_{r_j}}{\left\|\mathbf{v}_{r_j}\right\|^2} & \text{if } \left\|\mathbf{v}_{r_j}\right\| \neq 0, \\ 0 & \text{Otherwise.} \end{cases} \tag{8.4}$$

The horizontal distance, or the slant range, at the CPA between the jth intruder and the ownship is given by

$$d_{cpa_j} = \sqrt{\left\|\mathbf{p}_{r_j}\right\|^2 + t_{cpa_j} \mathbf{p}_{r_j}^\top \mathbf{v}_{r_j}}. \tag{8.5}$$

To predict whether or not the collision will occur for each intruder within the sensor coverage range, we use the simple collision detection logic listed in Algorithm 6. In Algorithm 6, the horizontal distance threshold d_{th} and the time threshold τ_{th} are design parameters that define the collision avoidance threshold. The collision avoidance threshold is a variable boundary around the ownship, larger than the collision volume and depends not only on distance, but time and other factors [30]. It is designed to give a practical separation

Algorithm 6: Collision detection algorithm

```
1: for each intruder within sensor coverage range do
2:    if ‖p_r‖ < d_th or (p_r^⊤ v_r < 0 and t_cpa ≤ τ_th) then
3:       if d_cpa < d_s and |h_r| < h_s/2 then
4:          Collision detected.
5:       else
6:          No collision detected.
7:       end if
8:    end if
9: end for
```

such that if a collision is predicted then the ownship has adequate time to plan an avoidance maneuver and to actually fly the maneuver. The design of the collision avoidance threshold is beyond the scope of this dissertation. We refer the interested reader to [30, 34]. Line 2 in Algorithm 6 checks whether the horizontal range $\|\mathbf{p}_r\|$ to the intruder is less than a horizontal threshold distance d_{th}, or the intruder is converging and the time to CPA is below a time threshold τ_{th}. The product $\mathbf{p}_r^\top \mathbf{v}_r$ characterize whether or not the intruder is horizontally converging, i.e., $\mathbf{p}_r^\top \mathbf{v}_r < 0$, or horizontally diverging i.e. $\mathbf{p}_r^\top \mathbf{v}_r > 0$. Line 3 checks to see if d_{cpa} and the relative altitude h_r are below horizontal and vertical minimum safe distances d_s and $h_s/2$, respectively. If so, then line 4 declares a collision.

8.3 Two-Step Path Planning

The subject of path planning is very broad and has been the focus of a significant body of research especially in the field of autonomous planning and robotics. Search based methods are very popular path planning algorithms in robotics. This can be attributed to the relative ease of implementation and to the early establishment of dynamic graph-based search methods. The A* algorithm is a widely used graph-based search. It uses a best-first search to find the minimum-cost path between the initial node and one or more possible goal nodes. The A* search uses an admissible heuristic estimate of the cost to get from the initial node to the final node to narrow the search. The A* algorithm is proven to be complete subject to the resolution of the grid employed and will always find a solution if one exists [75]. A drawback of A* search is that it is a static algorithm, which means

that when the workspace changes the previous path is no longer valid and the A* algorithm must replan from scratch. The potential fields algorithm is another popular approach for collision avoidance in robotics. In Chapter 4 we proposes a 2D collision avoidance method for a SAA system using a physical analogy of a simulated chain placed in a simulated force field. The collision avoidance algorithm tries to find the proper evasive maneuver observing the specified right-of-way rules. Bortoff presents a method for modeling a UAS path using a series of point masses connected by springs and dampers. This algorithm generates a stealthy path through a set of enemy radar sites of known locations [19]. McLain and Beard present a trajectory planning strategy suitable for coordinated timing for multiple UAS. The paths to the target are modeled using a physical analogy of a chain [88].

Once a collision threat has been detected, the collision avoidance system must select the proper evasive maneuver. The essential requirement for the collision avoidance algorithm is to perform the avoidance maneuver in such a way that the distance at the closest point of approach to the intruder is equal to, or greater than a minimum required safe distance. We propose a reactive path planning technique over look-ahead horizons within the sensor's coverage range. The basic idea is to model the future motion of the ownship using a waypoint path where the waypoints can dynamically move as if they were particles in a force field. The path planning approach taken in this work is carried out in two steps. In the first step, an initial suboptimal path is generated using A* search. In the second step, a chain of unit masses connected by springs and dampers evolves in a simulated force field, using the A* solution as an initial condition. The chain is described by a set of ordinary differential equations that is driven by virtual forces to find the steady-state equilibrium. Assuming a constant velocity model, it is relatively simple to determine roughly where the ownship will be at any given time. This timing information can be used to prevent collisions when creating plans to avoid intruders. In the following sections, we address the key components of the proposed approach.

8.3.1 Graph Search

In this section[1] we will give a brief description of how to use the A* algorithm to generate an initial path. Full implementation details of A* search is beyond the scope of this dissertation. We refer the interested reader to [75]. The region between points B and C as shown in Figure 8.1 is represented by a grid of regularly sized cells, where the locations of the start and the final points, the ownship, and the intruders are known. Let the indices of the grid cells be labeled by $\mathbf{i} = (i, j, k)$ where i is the North index, j is the East index, and k is the altitude index. Note that these indices are positive integers. The index of the start point is $\mathbf{i}_i$ and the index of the goal point is $\mathbf{i}_f$. Unlike many path planning methods that use A* search, the nodes of the graph must take into account timing information. A node on the graph is uniquely defined by its position index, $\mathbf{i}$, and the time of arrival of the ownship at that position, t, and will be denoted as $\mathbf{w} = (\mathbf{i}, t)$.

Let $\mathbf{w}_a$ be the current node that A* is evaluating, with the position indices i_a, and j_a, altitude index k_a, and arrival time t_a. The algorithm determines which possible positions, i_b, j_b, and k_b that the vehicle can travel to and determines the time it will arrive at these positions $t_b = t_a + \Delta t_{\mathbf{i}_a,\mathbf{i}_b}$, where $\Delta t_{\mathbf{i}_a,\mathbf{i}_b}$ is the travel time from position $\mathbf{i}_a$ to $\mathbf{i}_b$. In our current implementation, the possible positions are the ones towards the goal position while the altitude is constant or moving straight and changing altitude. For example, assume that the goal is north of the current node and that the South West corner of the map is $i = 0$, $j = 0$. In this case the possible positions are

$$\mathbf{i}_b \in \left\{ \begin{bmatrix} i_a + 1 \\ j_a - 1 \\ k_a \end{bmatrix}, \begin{bmatrix} i_a + 1 \\ j_a \\ k_a \end{bmatrix}, \begin{bmatrix} i_a + 1 \\ j_a + 1 \\ k_a \end{bmatrix}, \begin{bmatrix} i_a + 1 \\ j_a \\ k_a - 1 \end{bmatrix}, \begin{bmatrix} i_a + 1 \\ j_a \\ k_a + 1 \end{bmatrix} \right\}.$$

Note that any possible position index that exceeds the bounds of the workspace is thrown out. For each possible node $\mathbf{w}_b$, it then calculates the true cost, $g(\mathbf{w}_b)$, to travel to

[1] This section is jointly written with Matthew Argyle and Randal W. Beard as part of conference manuscript for possible publication in the 2016 American Control Conference (ACC) [115].

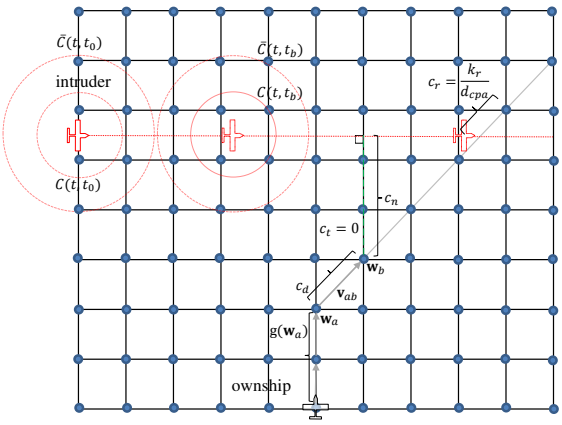


Figure 8.4: An example of costs associated with A^* search in a 2D grid.

$\mathbf{w}_b$ by traveling through $\mathbf{w}_a$ where

$$g\left(\mathbf{w}_b\right) = g\left(\mathbf{w}_a\right) + c_d\left(\mathbf{i}_a, \mathbf{i}_b\right) + c_t(\mathbf{w}_b, \mathbf{p}_{int_j}) + c_n + c_r, \tag{8.6}$$

where c_d is the cost for traveling from position $\mathbf{i}_a$ to $\mathbf{i}_b$ given by

$$c_d\left(\mathbf{i}_a, \mathbf{i}_b\right) = k_{\text{dist}}\sqrt{\left(i_b - i_a\right)^2 + \left(j_b - j_a\right)^2} + k_{\text{alt}}\left|k_b - k_a\right|,$$

and k_{dist} and k_{alt} are positive scaling parameters. To compute the cost of the intruder's threat c_t, we let $\mathbf{p}_{\mathbf{i}} = (n_i, e_j, d_k)^\top \in \mathbb{R}^3$ be the North-East-Down mapping of the grid cell $\mathbf{i}$ in the inertial NED reference frame. Assuming a constant velocity model, we let $\mathbf{p}_{int_j}(t, \ell)$ denote the jth intruder's predicted location at future time step ℓ defined as

$$\mathbf{p}_{int_j}(t, \ell) = \mathbf{p}_{int_j}(t, \ell - 1) + t_b \mathbf{v}_{int_j},$$

where for example, $\mathbf{p}_{int_j}(t, 0)$ is the current location of the jth intruder. Then c_i is given by

$$c_t\left(\mathbf{w}_b, \mathbf{p}_{int_j}\right) = \begin{cases} k_{max}, & \text{if } \mathbf{p}_{\mathbf{i}_b} \in \mathcal{C}(t, t_b), \\ \epsilon k_{max}, & \text{if } \mathbf{p}_{\mathbf{i}_b} \in \bar{\mathcal{C}}(t, t_b), \\ 0, & \text{Otherwise}, \end{cases}$$

where k_{max} is the maximum allowable penalty cost, $\epsilon < 1$ is a positive constant, $\mathcal{C}(t, t_b) = \{\mathbf{D} \in \mathbb{R}^2, H \in \mathbb{R} : \left\|\mathbf{p}^*_{int_j}(t, t_b) - \mathbf{D}\right\| \leq d_s$, and $\left|d_{int_j} - H\right| \leq h_s/2\}$ represent the collision volume centered on the intruder projected at time t_b, and $\bar{\mathcal{C}}(t, t_b) = \{\mathbf{D} \in \mathbb{R}^2, H \in \mathbb{R} : \left\|\mathbf{p}^*_{int_j}(t, t_b) - \mathbf{D}\right\| \leq \varrho d_s$, and $\left|d_{int_j} - H\right| \leq \varrho h_s/2\}$ represent a larger collision volume to penalize nodes that are close to the intruders. The design parameter $\varrho > 1$ determines the volume of $\bar{\mathcal{C}}$, and c_n is the cost of deviation from the nominal path. Assuming that the path is defined as an ordered sequence of waypoints $\mathcal{W} = \mathbf{W}_1, \mathbf{W}_2, \cdots .\mathbf{W}_f$, and at the time of detecting a collision the ownship was following the path segment $(\mathbf{W}_a, \mathbf{W}_b)$, then using Eq. 4.47, c_n is given by

$$c_n = D(\mathbf{W}_a, \mathbf{W}_b, \mathbf{w}_b).$$

The last term in Eq. 8.6 accounts for collisions that may happen when the ownship moves from the current node to the possible node $\mathbf{w}_b$. In other words, if we assume that $\mathbf{w}_b$ is the current node that A* is evaluating, then c_r is the cost that measures the risk of collision if the ownship will be traveling along the edge ($\mathbf{w}_a$,$\mathbf{w}_b$). Let the velocity vector in the direction of node $\mathbf{w}_b$ from node $\mathbf{w}_a$ be $\mathbf{v}_{ab} = v_{own}\frac{\mathbf{p}_{\mathbf{i}_b} - \mathbf{p}_{\mathbf{i}_a}}{\left\|\mathbf{p}_{\mathbf{i}_b} - \mathbf{p}_{\mathbf{i}_a}\right\|}$, where v_{own} is the speed of the ownship, then c_r is given by

$$c_r = \begin{cases} \frac{k_r}{d_{cpa}}, & \text{if Algorithm 6 detects a collision,} \\ 0, & \text{Otherwise,} \end{cases}$$

where k_r is a positive real number and design parameter, and $\mathbf{p}_r$, $\mathbf{v}_r$, and h_r are given by

$$\begin{aligned} \mathbf{p}_r &= \mathbf{p}^*_{int_j}(t_a) - p^*_{\mathbf{i}_b}, \\ \mathbf{v}_r &= \mathbf{v}^*_{int_j}(t_a) - \mathbf{v}^*_{ab}, \\ h_r &= d_{int_j} - d_{k_b}, \end{aligned}$$

where $\mathbf{p}^*_{\mathbf{i}_b}$ and $\mathbf{v}^*_{ab}$ are the projection of $\mathbf{p}_{\mathbf{i}_b}$ and $\mathbf{v}_{ab}$ onto the NE-plane, and d_{k_b} is the height of node $\mathbf{w}_b$ in the inertial NED reference frame.

Finally, the estimated cost from node $\mathbf{w}_b$ to position $\mathbf{i}_f$ is

$$e(\mathbf{w}_b) = c_d\left(\mathbf{i}_b, \mathbf{i}_f\right).$$

The total search cost is then $f(\mathbf{w}_b) = g(\mathbf{w}_b) + e(\mathbf{w}_b)$. The nodes $\mathbf{w}_b$ are then added to the OPEN list and the next node is evaluated. The A* algorithm ends after evaluating all possible nodes without finding a solution, or when it evaluates a node where $\mathbf{i}_a = \mathbf{i}_f$.

8.3.2 Chain Based Potential Field

Once an initial path has been generated by the A* algorithm as shown in Figures 8.5(a), and 8.5(c), it is superimposed by a chain of unit masses that are connected to one another by springs and dampers as shown in Figure 8.5. One end of the chain is connected to the ownship's current location while the other is attached to the goal. The chain is employed for two reasons. The first is to smooth the initial path generated by the A* algorithm, and the second is to dynamically react to changes in the environment. Let $\mathbf{n}_i = (n_i, e_i, d_i)^\top \in \mathbb{R}^3$, $i = 1, 2, 3, ..., l$ be the North-East-Down location of the ith node in the inertial NED reference frame. An $(l-1)$-link chain of waypoints is represented by $\mathbf{c} = \left[\mathbf{n}_1^\top, \mathbf{n}_2^\top, \mathbf{n}_3^\top, \ldots, \mathbf{n}_l^\top\right]^\top$, where the node $\mathbf{n}_i$ is the future planned location of the ownship. If v_o is the speed of the ownship, then the time required by the ownship to fly between nodes is approximately $t_i \approx \frac{\|\mathbf{n}_i - \mathbf{n}_{i-1}\|}{v_o}$. The approximation is due to the fact that the ownship may also follow a circular arc or a fillet maneuver to switch between waypoints. The steady-state equilibrium of the chain system is driven by virtual forces. The springs exert a contractive force, and act to reduce the length of the chain, and the dampers tend to reduce the system oscillation, and remove the kinetic energy stored by the unit masses. Second, the future predicted locations of the intruders at time step ℓ exert a repulsive force applied to the nodes of the ownship's chain associated with time step ℓ. To be consistent with the kinematics of a fixed-wing aircraft, we introduce a horizontal and vertical straightening force to prevent the ownship when following the chain's nodes. The dynamic equation describing the entire chain motion is given by

$$\ddot{\mathbf{c}} = \mathbf{F}, \tag{8.7}$$

where $\mathbf{F}$ is the sum of the applied forces. $\mathbf{F}$ is given by

$$\mathbf{F} = \sum_{i=2}^{l-1}\Big(\mathbf{F}_{s_i} + \mathbf{F}_{d_i} + k_{sh}\mathbf{F}_{sth_i} + k_{sv}\mathbf{F}_{stv_i} + k\sum_{j=1}^{N}\boldsymbol{\mu}_{i,j}\mathbf{F}_{rp}(\mathbf{n}_i, \mathbf{p}_{int_j}(t,i))\Big), \tag{8.8}$$

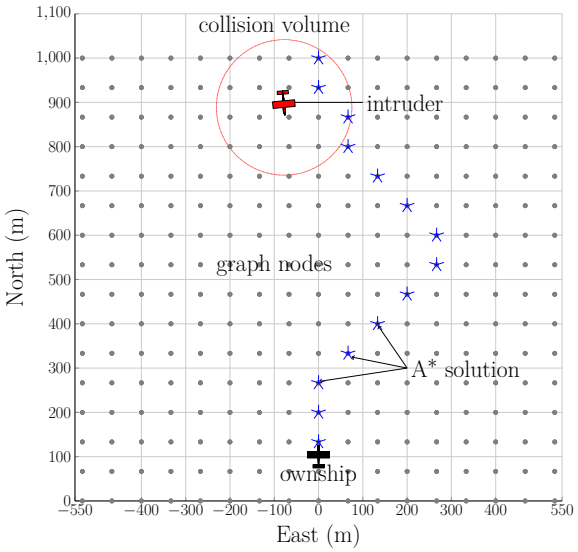
where k is a positive design parameter that represents a tradeoff between repelling from intruders and path length, k_{sh} and k_{sv} are also positive design parameter that represents a tradeoff between repelling from intruders and maintaining aircraft kinematics constraints. The idea is to appropriately weight the forces acting on the system to converge to the optimal path by minimizing the system potential energy. Let $\mathbf{F}_{s_i}$, $\mathbf{F}_{d_i}$ $\mathbf{F}_{sth_i}$, and $\mathbf{F}_{stv_i}$ be the spring restoring force, the damping force, and the horizontal and the vertical straightening force acting on the ith node, respectively. Define N as the number of intruders detected by the sensor, and $\mathbf{F}_{rp}(\mathbf{n}_i, \mathbf{p}_{int_j}(t,i))$ as the repelling force acting on the ith node from the jth intruder at future time step ℓ, and its direction is determined by the unit vector $\boldsymbol{\mu}_{i,j} = (\mathbf{n}_i - \mathbf{p}_{int_j}(t,i))/\left\|\mathbf{n}_i - \mathbf{p}_{int_j}(t,i)\right\|$. As expressed by Eq. (8.8), the state and end nodes of the chain do not change, while the remainder of the chain continuously adapts to changing environmental conditions. The spring restoring force and the damping force in Eq. (8.8) are defined as

$$\begin{aligned}\mathbf{F}_{s_i} &= \kappa(\mathbf{n}_{i+1} - \mathbf{n}_i) - \kappa(\mathbf{n}_i - \mathbf{n}_{i-1}),\\ \mathbf{F}_{d_i} &= b(\dot{\mathbf{n}}_{i+1} - \dot{\mathbf{n}}_i) - b(\dot{\mathbf{n}}_i - \dot{\mathbf{n}}_{i-1}),\end{aligned}$$

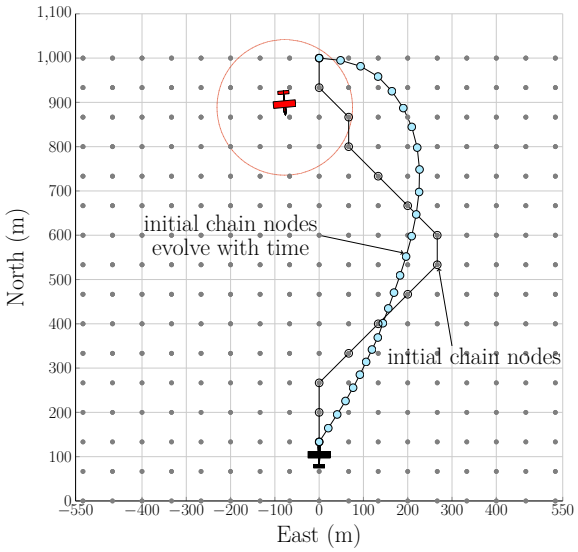
where κ is the spring constant, and b is the damping constant.

To guarantee that the turn radius constraint is not violated, the horizontal straightening force F_{sth_i} is designed to ensure that $|\theta_i| \leq \theta_{max}$, where θ_{max} is the maximum allowable angle that can be formed between adjacent links as shown in Figure 8.6. As depicted in Figure 8.7, the unit vectors $\mathbf{u}_i$, and $\mathbf{u}_{i-1}$ are defined as

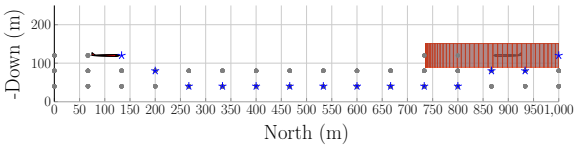
$$\begin{aligned}\mathbf{u}_i &= (\mathbf{n}_i - \mathbf{n}_{i-1})/(\|\mathbf{n}_i - \mathbf{n}_{i-1}\|),\\ \mathbf{u}_{i-1} &= (\mathbf{n}_{i-1} - \mathbf{n}_{i-2})/(\|\mathbf{n}_{i-1} - \mathbf{n}_{i-2}\|\,.\end{aligned}$$



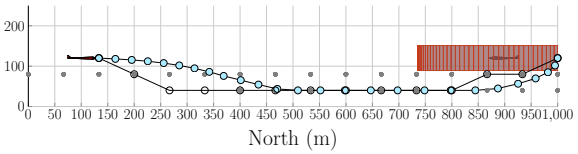
(a) Overhead view of A* initial's path.



(b) Overhead view of chain path.



(c) Side view of A* initial's path.



(d) Side view of chain path

Figure 8.5: The chain-based collision avoidance initialized by the A* algorithm.

Letting $\mathbf{u}_i^* \triangleq P\mathbf{u}_i$ be the projection of $\mathbf{u}_i$ onto the NE-plane, where $P = [1\ 0\ 0 \vdots 0\ 1\ 0 \vdots 0\ 0\ 0]$ is the projection matrix, then $\theta_i = \arccos((\mathbf{u}_i^*)^\top \mathbf{u}_{i-1}^*)$ is the angle between $\mathbf{u}_i^*$ and $\mathbf{u}_{i-1}^*$. As depicted in Figure 8.7, F_{sth_i} acting on node $\mathbf{n}_i$, is expressed as

$$F_{sth_i} = \frac{(\mathbf{u}_i^*)^\perp}{1 + \exp(\lambda(\theta_{max} - \theta_i))},$$

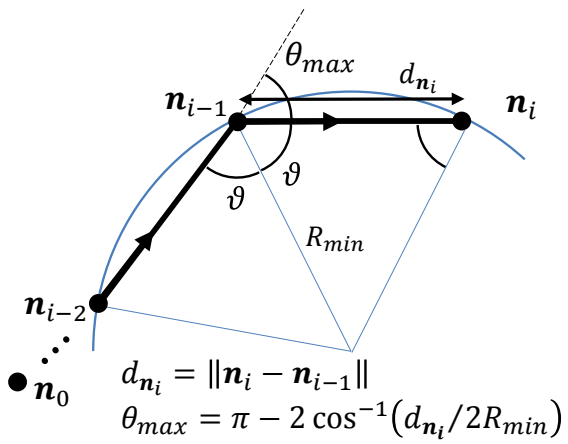


Figure 8.6: Definition of θ_{max}, the maximum allowable turn angle. R_{min} is the ownship's minimum turning radius.

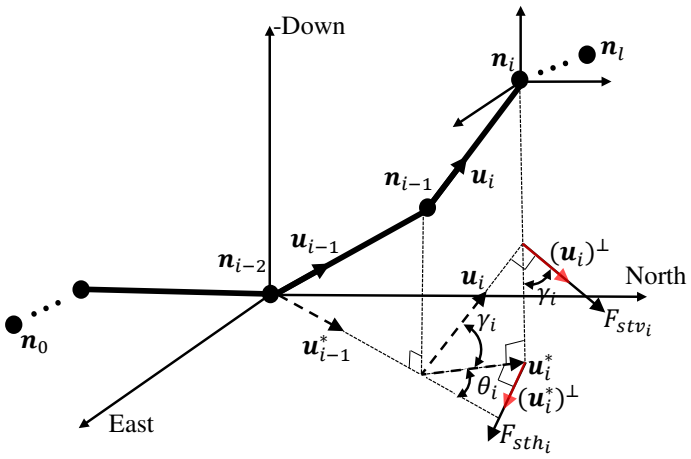


Figure 8.7: The vertical straightening force applied to node $\mathbf{n}_i$ is designed to ensure that $\gamma_i \leq \gamma_{max}$.

where λ is a positive constant that determines how closely the logistic function approximates a step function and $(\mathbf{u}^*_i)^\perp$ is the orthogonal complement of $\mathbf{u}^*_i$ defined as

$$(\mathbf{u}^*_i)^\perp = \begin{cases} \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \mathbf{u}^*_i, & \text{if } \theta^* \leq \pi/2, \\ \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \mathbf{u}^*_i, & \text{if } \theta^* > \pi/2, \end{cases}$$

where $\theta^* \triangleq \arccos(\frac{(\mathbf{u}_i^* - \mathbf{u}_{i-1}^*)^\top (\mathbf{u}_i^*)^\perp}{\left\|\mathbf{u}_i^* - \mathbf{u}_{i-1}^*\right\| \left\|(\mathbf{u}_i^*)^\perp\right\|})$.

In addition, to preventing abrupt altitude changes between chain nodes, the vertical straightening force F_{stv_i} is designed to ensure that $|\gamma_i| \leq \gamma_{max}$, where γ_i is the angle between $\mathbf{u}_i$ and its projection $\mathbf{u}_i^*$ on the NE-plane, $\gamma_{max} = \arcsin(\frac{\dot{h}_{max}}{v_g})$ is the maximum allowable flight path angle, $\dot{h}$ is the climbing rate, and v_g is the ownship ground speed [16]. Then, F_{stv_i} acting on node $\mathbf{n}_i$ is given by

$$F_{stv_i} = \frac{(\mathbf{u}_i)^\perp}{1 + \exp(\lambda(\gamma_{max} - \gamma_i))}.$$

As shown in Figure 8.7, let $(\mathbf{u}_i)^\perp = (u_{n_i}^\perp, u_{e_i}^\perp, u_{d_i}^\perp)^\top \in \mathbb{R}^3$ be the unique orthogonal complement of $\mathbf{u}_i$ such that $u_{z_i}^\perp = \cos(\gamma)$, $\mathbf{u}_i^\top (\mathbf{u}_i)^\perp = 0$, and if $\mathbf{u}_i \times (\mathbf{u}_i)^\perp = a\hat{i} + b\hat{j} + c\hat{k}$, then $c = 0$ i.e the resultant vector of the cross product is parallel to the NE-plane. Letting $\mathbf{u}_i = (u_{n_i}, u_{e_i}, u_{d_i})^\top \in \mathbb{R}^3$, then the required orthogonal complement $(\mathbf{u}_i)^\perp$ is given by

$$\begin{aligned}
u_{n_i}^\perp &= \frac{u_{n_i}}{u_{e_i}} u_{e_i}^\perp, \\
u_{e_i}^\perp &= \frac{u_{e_i} u_{d_i}}{u_{n_i}^2 + u_{e_i}^2} u_{d_i}^\perp, \\
u_{d_i}^\perp &= -\frac{\|\mathbf{u}_i\|^*}{\|\mathbf{u}_i\|} \operatorname{sgn} u_{d_i}.
\end{aligned}$$

Finally, the repelling force acting on the ith link is defined as

$$F_{rp}(\mathbf{n}_i, \mathbf{p}_{int_j}(t,i)) =$$
$$\begin{cases}
f_{max}, & \text{if } d(t,\ell) \leq d_s, \text{ and } h(t,\ell) \leq \frac{h_s}{2} \\
\max\left(\rho_h f_{max}, \rho_v f_{max}\right), & \text{if } d_s < d(t_i) \leq \delta_1 d_s, \text{ and } \frac{h_s}{2} < h(t,\ell) \leq \delta_2 \frac{h_s}{2} \\
\rho_h f_{max}, & \text{if } d_s < d(t_i) \leq \delta_1 d_s, \text{ and } h(t,\ell) \leq \frac{h_s}{2} \\
0, & \text{Otherwise,}
\end{cases}$$

where f_{max} is the maximum allowable repelling force, $\delta_1 > 1$, and $\delta_2 > 1$ are positive constants, and $\rho_h \triangleq \frac{-(d(t,\ell) - d_s)}{d(t,\ell) - \delta_1 d_s} + 1$, and $\rho_v \triangleq \frac{-(h(t,\ell) - h_s/2)}{h(t,\ell) - \delta_2 h_s/2} + 1$. Since the collision volume is defined as a truncated cylinder of radius d_s and height h_s, the distance from the chain nodes to the

intruders are decomposed into a horizonal distance $d(t,\ell)$, and an altitude distance $h(t,\ell)$. Let $\mathbf{n}_i^* = (n_i, e_i)^\top$ be the projection of the ith node location onto the NE-plane, then the horizonal distance between the ith node and the jth intruder is given by

$$d(t,\ell) = \left\| \mathbf{n}_i^* - \mathbf{p}_{int_j}^*(t,\ell) \right\|,$$

where $\mathbf{p}_{int_j}^*(t,\ell) = \mathbf{p}_{int_j}^*(t,\ell-1) + t_i \mathbf{v}_{int_j}^*$. The altitude difference between the ith node and the jth intruder is given by

$$h(t,\ell) = \left| d_i - d_{int_j}(t,\ell) \right|.$$

Once the system has converged, the path is defined as an ordered sequence of the waypoints $\mathcal{W} = \{\mathbf{n}_0, \mathbf{n}_1, \mathbf{n}_2, \cdots .\mathbf{n}_l\}$ given by the steady-state mass locations connected by straight line segments. Next, a path manager is required to follow the waypoint path. One possible approach is to transition waypoints when the ownship enters a ball around the waypoint $\mathbf{n}_i$. A better strategy is to use the half-plane switching criteria that is not sensitive to tracking error [16]. In addition, a smooth transition between the waypoints is required to make the generated path flyable by the ownship. This can be achieved by implementing the fillet maneuver or using the so-called Dubins paths. For further analysis on the path manager, switching between waypoints, and smoothing paths we refer the interested reader to [16].

8.4 Simulation Results with Perfect Sensing

To demonstrate the performance of the proposed collision detection and avoidance algorithm, we developed a simulation environment with a five-degree-of-freedom aircraft model for both the ownship and the intruders. We assume a ground-based sensor configuration that provides a complete and perfect knowledge of the intruders' number and state. The encounter geometry is constructed using typical collision encounters that include multiple intruders flying at different altitudes, approaching head-on, converging, and overtaking scenarios. In all encounter scenarios the ownship initially starts at point B located at $(0, 0, 120)^\top$ in NED coordinates system, with an initial heading of 0 degrees measured from North and follows a

straight line path at constant speed of 20 m/s to reach point C located at $(0, 1000, 120)^\top$ as shown in Figures 8.8, 8.11, and 8.14. In the following simulations, our choice of the collision volume is a cylinder of radius, d_s=153 m (500 ft) and height, h_s =62 m (200 ft) centered on each of the intruders.

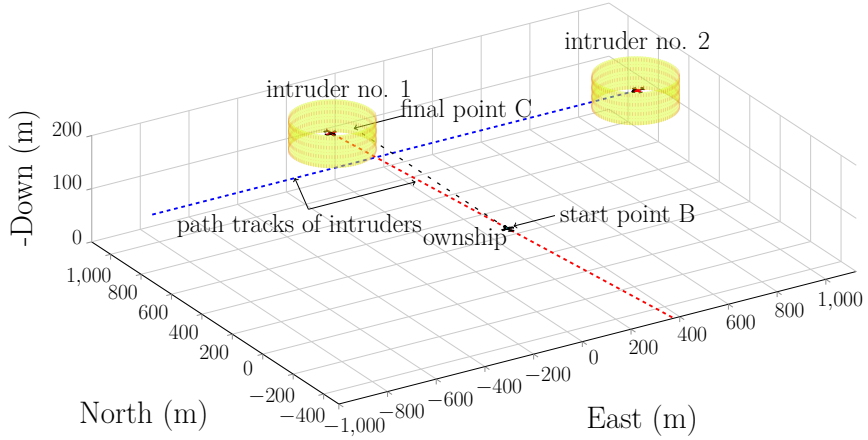


Figure 8.8: Encounter scenario number 1.

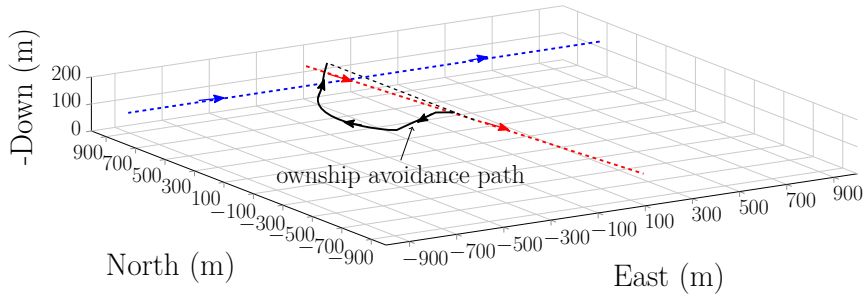


Figure 8.9: Avoidance path followed by the ownship in encounter scenario number 1.

In each encounter scenario the intruders are following a straight line path at a constant velocity and altitude. Figure 8.8 shows the side view of the first encounter scenario. It consists of two intruders: one is approaching head-on and the other is converging from the right. The speed and the altitude of both intruders is 20 m/s and 120 m, respectively. Figure 8.9 shows the avoidance path generated by the proposed algorithm, and Figure 8.10 shows the results of the horizontal range and relative altitude to both intruders. The results shows that no collision incidents occurs. The second encounter geometry consists of three

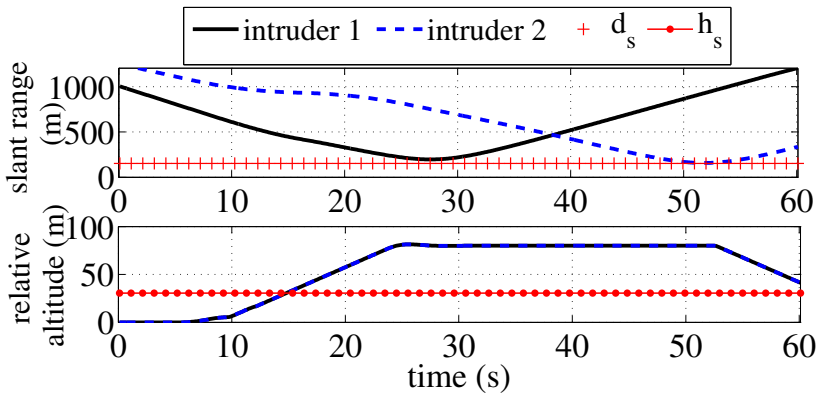


Figure 8.10: Horizontal range and relative altitude to intruders.

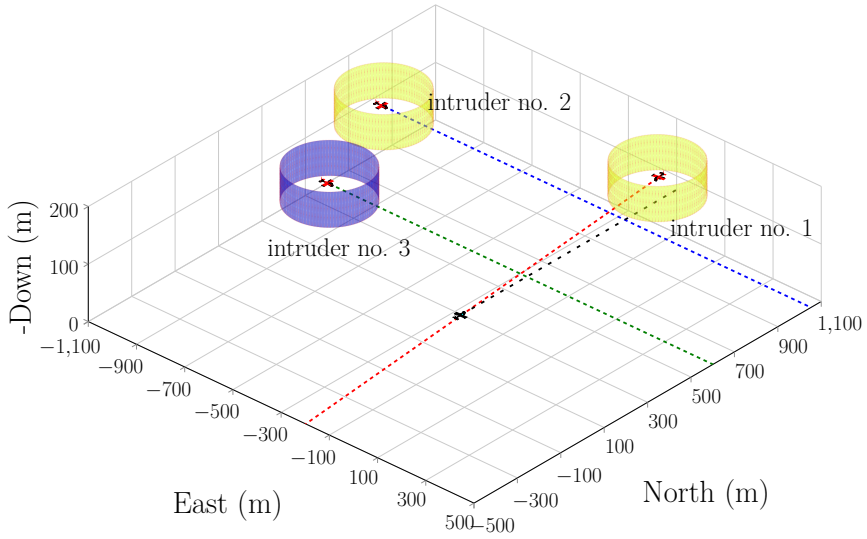


Figure 8.11: Encounter scenario number 2.

intruders: two of the intruders are flying at a fixed altitude of 120 m while the third intruder is at an altitude of 40 m as shown in Figure 8.11. In the second scenario, the speed of all the intruders is 20 m/s. The encounter geometry of the third scenario consists of four intruders: the first intruder is approaching head-on, the second intruder is overtaking the ownship while the third and the fourth are converging from the right and left, respectively as shown in Figure 8.14. In the third scenario, the first and the second intruders are flying at a fixed altitude of 120 m while the third and the fourth intruders are flying at 80 m and 40 m, respectively. The speed of the overtaking intruder is 40 m/s while the speed of the others is 20 m/s.

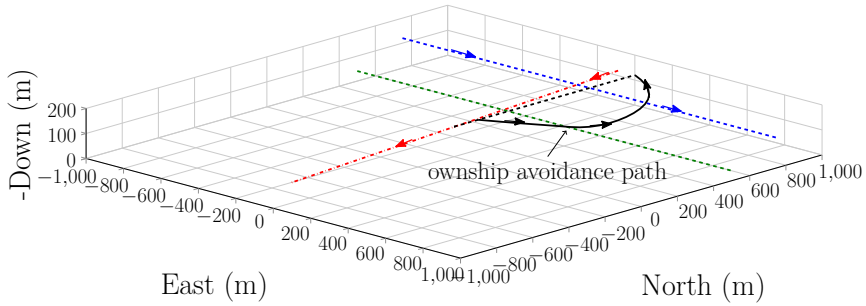


Figure 8.12: Avoidance path followed by the ownship in encounter scenario number 2.

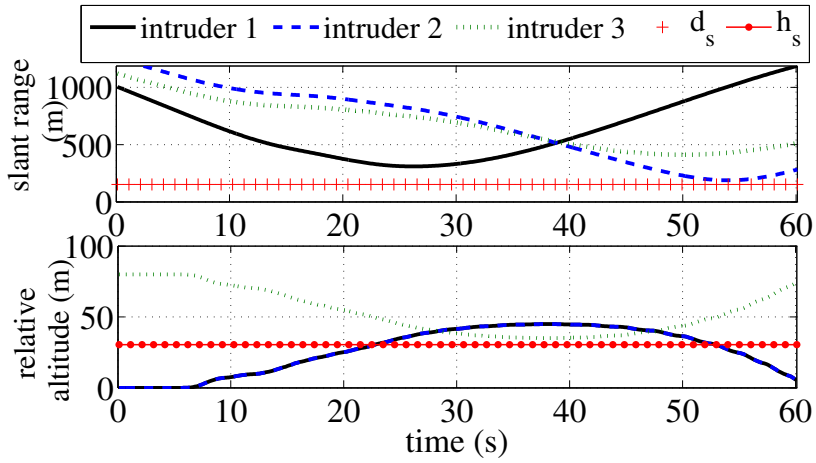


Figure 8.13: Horizontal range and relative altitude to intruders.

Figures 8.12 and 8.15 shows the planned avoidance path in encounter scenarios 2 and 3. The horizontal range and relative altitude to intruders is shown in Figures 8.13 and 8.16. The results shows that the avoidance path maneuvers the ownship safely without colliding. Another important aspect to evaluate the performance of the proposed algorithm is its ability to reduce the length of the avoidance path while avoiding the intruders. This is important because it reduces the amount of deviation from the original path, and ultimately the flight

Table 8.1: Length of the avoidance path.

Scenario number	Initial path length (m)	Avoidance path length (m)
1	1000	1222.2
2	1000	1258.5
3	1000	1044.6

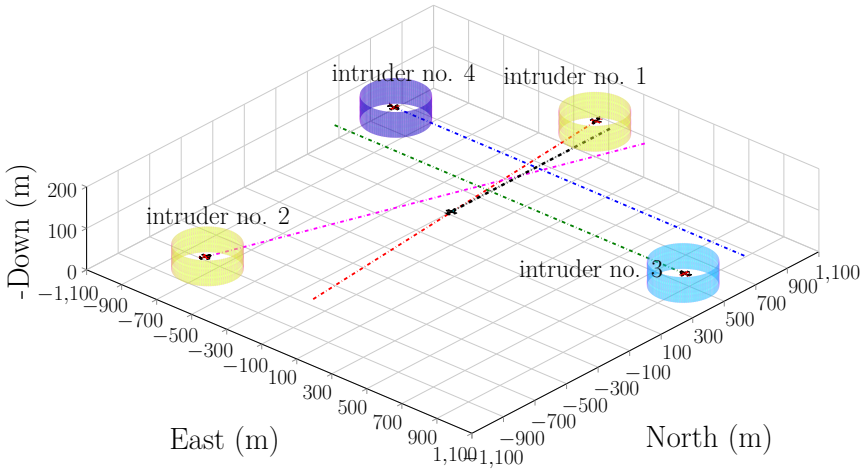


Figure 8.14: Encounter scenario number 3.

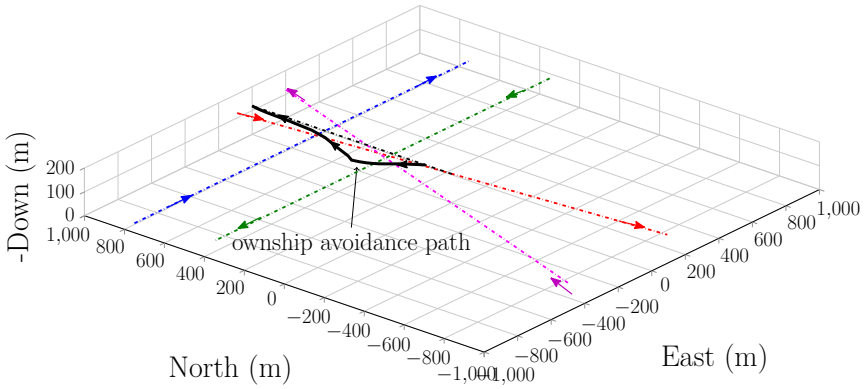


Figure 8.15: Avoidance path followed by the ownship in encounter scenario number 3.

time which is of critical importance for small UAS with limited power resources. Table 8.1 shows that the length of the avoidance paths is fairly acceptable compared to the initial path length. We have also recorded the time required to execute the collision avoidance algorithm. The proposed collision avoidance algorithm was run using Matlab on an Intel i7 processor. It

Table 8.2: Collision avoidance algorithm run time.

Scenario number	A* run time (one cycle)(s)	chain max. run time (s)	chain average run time (s)
1	1.147	0.328	0.328
2	4.050	1.157	1.157
3	6.379	1.007	0.799

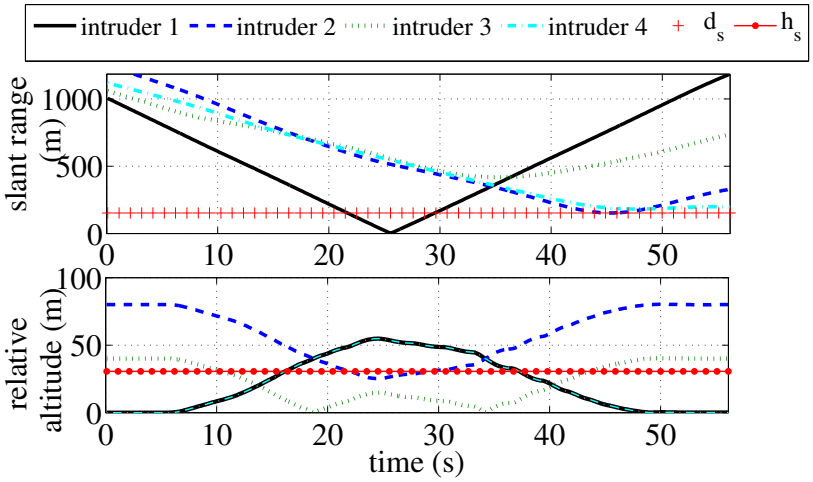


Figure 8.16: Horizontal range and relative altitude to intruders.

executes the A* algorithm once to initialize the chain, while the chain based algorithm may be executed several times. This is because following the avoidance path may result in another pop-up collision threat. This situation requires the collision avoidance method to execute the chain algorithm to respond to the pop-up collision threat. Table 8.2 shows that the time required to execute one cycle of the A* algorithm, and the maximum time and average time required to run the chain algorithm. The results show that A* algorithm requires a larger amount of time compared to the chain algorithm, however, the current implementation of the A* and the chain algorithm, are coded in Matlab and were not optimized for run times. Significant speed increases could be achieved by optimizing the code and by porting it to C/C++.

8.5 Simulation and Flight Test Results with Radar

This[1] section describes the development of a low SWaP phased array radar designed specifically for small UAS sense-and-avoid system. In this section, we present a complete sense-and-avoid solution for small UAS including reliable ground-based radar intruder sensing, tracking and estimation, collision detection, and collision avoidance.

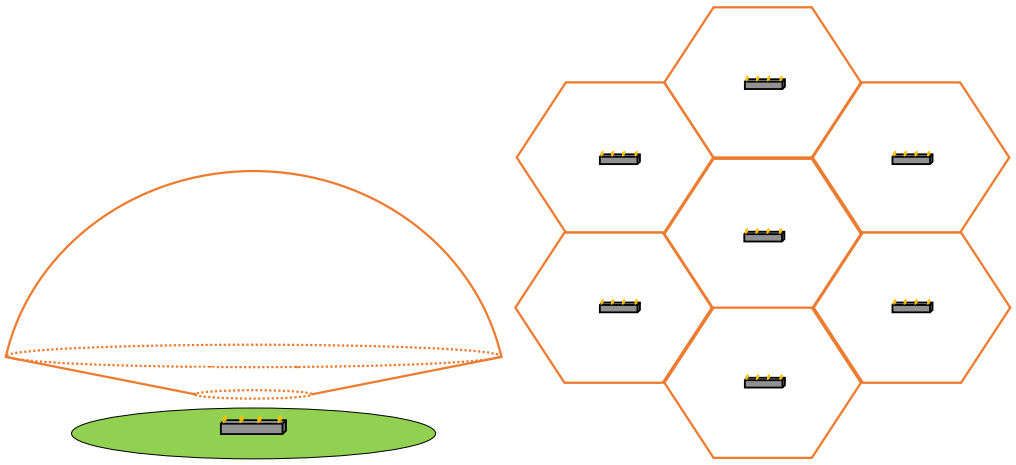
[1]This section is jointly written with Jared Wikle, Jonathan Spencer, Michael Boren, Kaleo Roberts, Randal Beard, Timothy McLain and Karl Warnick as part of journal manuscript for possible publication.

8.5.1 Ground-Based Radar Model

Air traffic control radars use large ground based systems that can detect targets approaching from up to 400 kilometers away [128]. These radar systems achieve a long detection range by using large, high gain antennas to focus the radar energy into beamwidths ranging from 2 to 5 deg. The antennas are placed on large mechanical gimbals that rotate the antenna to scan the narrow beam. Since most aircraft fly at predictable altitudes, these systems scan over a relatively narrow set of angles vertically, while providing 360 deg of horizontal coverage. Although the large antennas and high transmitted power permit long-range detection, the system has a doughnut shaped detection pattern that is blind to targets directly overhead. These systems also avoid scanning too close to the horizon in order to avoid ground clutter reflections from man-made structures.

In recent years, miniaturized radar systems have shown promise as on-board SAA sensors for UAS. There currently exist solutions that meet the SWaP constraints of small UAS and achieve the field of view requirements for fixed wing flight [97]. As the field of view of these radar systems expand to meet the needs of more general multirotor systems, they will provide an attractive alternative to ATC radar for localized collision avoidance. An improved ground-based sensor could also resolve the current blind spots of ATC radar. Although miniature scale versions of current ATC radar exist, they are not viable for small UAS because they provide only 2D sensing where UAS sensing requires three-dimensional spacial localization of small targets. Gimballed radar systems not only have blind spot issues overhead, but their update rate is very slow: on the order of a few seconds [11]. For long range radar systems, this is not an issue since their detection range is long enough to compensate for a slow update rate, but for short range radar systems, aircraft could travel most of the way through the field of view in a single update interval. A solution to resolve the issues of a mechanically steered antenna would be to replace it with an electronically steered array (ESA) of antennas. An ESA or phased array antenna has the same narrow beam and high resolution as a mechanically steered antenna, but it can be steered with a wider field of view and a much higher update rate.

A potential solution for ground based SAA for UAS would be to create a planar phased array radar with a wide field of view angled directly at the sky as shown in Fig. 8.17(a). This



(a) Single 2D Phased Array (b) Radar Network Cells

Figure 8.17: UAS surveillance using planar phased arrays

would create a 3D hemispherical bubble where the radar can detect targets and guarantee safe flight within the bubble. The update rate for a phased array radar that implements digital beamforming is the pulse rate of the system, on the order of milliseconds. As a result, such a system can provide high resolution in range, elevation angle, azimuth angle, and time. These 3D phased array systems have a wide field of view that covers the full extent of UAS flight paths, and can implement filtering methods to remove background reflections from the environment.

In order to expand the coverage of the radar along the ground, multiple radar systems could be arranged into a grid similar to the current cell phone antenna network. This configuration, shown in Fig. 8.17(b), is designed to establish coverage over a wide area while maintaining low radiated power and ensuring that a single antenna is not overwhelmed with traffic. Since UAS only need to know about nearby air traffic, this cellular radar sensor network also provides an efficient solution to manage the volume of data produced by air traffic monitoring. A maneuvering aircraft will communicate only with nearby radar sensors to obtain local air traffic information rather than being required to communicate with a central source that transmits global air traffic. This type of a system is scalable depending on the amount of air traffic expected. In rural areas, the cells could be very large or the system could just use on-board sensors for collision avoidance since low volume of intruding

traffic is expected. In densely populated urban environments where a lot of traffic is expected cells could be made smaller to ensure adequate coverage and load balancing.

A step in developing a 3D sensor is the 2D line array radar system that is introduced in Section 2.2.2, which could be used interchangeably as the primary sensor on board the aircraft or as a ground based sensor. The key design parameters are listed in Table 8.3. The design parameters were driven primarily for its use as an onboard sensor for small UAS. The main design constraints were to minimize SWaP and to provide a reasonably large field of view of approximately 120 deg horizontally and 30 deg vertically.

Table 8.3: Radar sensor parameters

Parameter	**Value**	**Parameter**	**Value**
Weight	120 g (0.26 lbs)	Size	2.25in x 4in x 1in
Consumed Power	15 watts	Transmitted Power (P_{trans})	2 watts
Center Frequency	10.25 GHz	Radio Frequency Bandwidth	500 MHz
Sweep Duration (T_c)	2 ms	Intermediate Frequency Bandwidth	1 MHz
System Noise Figure (F)	8 dB	Antenna Elevation Beamwidth	18 deg
Antenna Gain	12 dB	Antenna Azimuth Beamwidth	110 deg
Array Steerable Range	110 deg	Number of Receive Elements	4
Peak Channel Coupling	approx. -20dB	Synthesized Azimuth Beamwidth	25 deg

8.5.2 States Estimation and Tracking

The requirement of tracking multiple intruders using ground-based radar motivates the use of the the R-RANSAC algorithm [100]. One of the advantages of the R-RANSAC algorithm is its modularity and applicability to a wide range of tracking scenarios. It has the ability to track multiple targets in clutter without requiring a prior knowledge of the number of existing targets. While not a cure-all, the R-RANSAC algorithm is robust to clutter and intermittent measurements, simple to implement, and computationally efficient, and fully autonomous while maintaining accuracy.

In this section, we briefly introduce the R-RANSAC algorithm. The full mathematical framework and implementation of R-RANSAC algorithm is beyond the scope of this dissertation, we refer the interested reader to [100]. Let the state vector $\mathbf{x}[k]$ represent the

Cartesian coordinates of the detected target in the inertial NED reference frame at time step k

$$\mathbf{x}[k] = \begin{pmatrix} x_1[k] \\ x_2[k] \\ x_3[k] \\ x_4[k] \\ x_5[k] \\ x_6[k] \end{pmatrix} \triangleq \begin{pmatrix} n[k] \\ e[k] \\ d[k] \\ vn[k] \\ ve[k] \\ vd[k] \end{pmatrix} \tag{8.9}$$

where $n[k]$, $e[k]$, and $d[k]$ and $vn[k]$, $ve[k]$ and $vd[k]$ are the North-East-Down position and velocity of the target. Let r, α and ε be the range, azimuth and elevation measurement provided by the radar at time step k, respectively. These measurements include all aircraft in the field of view of the radar, i.e. the intruders and the ownship. Let the measurement vector $\mathbf{h}[k]$ given by

$$\mathbf{h}[k] = \begin{pmatrix} r[k] \\ \alpha[k] \\ \varepsilon[k] \end{pmatrix}, \tag{8.10}$$

where

$$\begin{aligned} r[k] &= \sqrt{(x_1 - n_{_{rd}})^2 + (x_2 - e_{_{rd}})^2 + (x_3 - d_{_{rd}})^2}, \\ \alpha[k] &= \tan^{-1}\left(\frac{x_2 - e_{_{rd}}}{x_1 - n_{_{rd}}}\right), \\ \varepsilon[k] &= \sin^{-1}\left(\frac{x_3 - d_{_{rd}}}{\sqrt{(x_1 - n_{_{rd}})^2 + (x_2 - e_{_{rd}})^2 + (x_3 - d_{_{rd}})^2}}\right), \end{aligned} \tag{8.11}$$

where $(n_{_{rd}}, e_{_{rd}}, d_{_{rd}})^\top$ is the North-East-Down location of the ground-based radar station.

The R-RANSAC algorithm implements a linear Kalman filter that assumes the process and measurement equations can be modeled according to

$$\begin{aligned} \mathbf{x}[k] &= A\mathbf{x}[k-1] + \xi[k], \\ \mathbf{y}[k] &= C\mathbf{x}[k] + \eta[k]. \end{aligned} \tag{8.12}$$

As shown in Eq. (8.11), the range, azimuth, and elevation measurements are nonlinear in the states. To simplify the processing we perform a nonlinear transformation of the range, azimuth, and elevation into orth-East-Down position, as

$$\begin{aligned} n &= r\cos(\alpha)\cos(\varepsilon) + n_{_{rd}}, \\ e &= r\sin(\alpha)\cos(\varepsilon) + e_{_{rd}}, \\ d &= r\sin(\varepsilon) + d_{radar}. \end{aligned} \tag{8.13}$$

Since tracking, estimation and path planning is conducted on ground, the R-RANSAC algorithm is modified to distinguish ownship from the intruders. This modification is required because the radar sensor provides measurements of all detected aircraft with its surveillance range. That implies a reliable communication link between the ownship and the ground control station.

8.5.3 Simulation Results

To demonstrate the performance of the proposed ground-based radar sensor model, the R-RANSAC estimation scheme, and the collision detection and avoidance algorithm, we developed a simulation environment with a six-degree-of-freedom aircraft model for both the ownship and the intruders. The encounter geometry is constructed using typical collision encounters that includes multiple intruders flying at different altitudes, approaching head-on, converging, and overtaking scenarios.

In the first encounter scenario the ownship starts at $(-400, 0, -200)^\top$ in NED coordinates system, with an initial heading of 0 deg measured from North and follows a straight line path at constant speed of 13 m/s to reach waypoint located at $(500, 0, -200)^\top$ as shown

in Figure 8.18. The radar system is located at $(0, 0, 0)^\top$ and uses a simulated transmit power of 5 kW. In the following simulations, our choice of the collision volume is a cylinder of radius, d_s= 153 m (500 ft) and height, h_s = 61 m (200 ft) centered on each of the intruders. All aircraft are assumed to have a simulated radar cross section of 0.1 m^2.

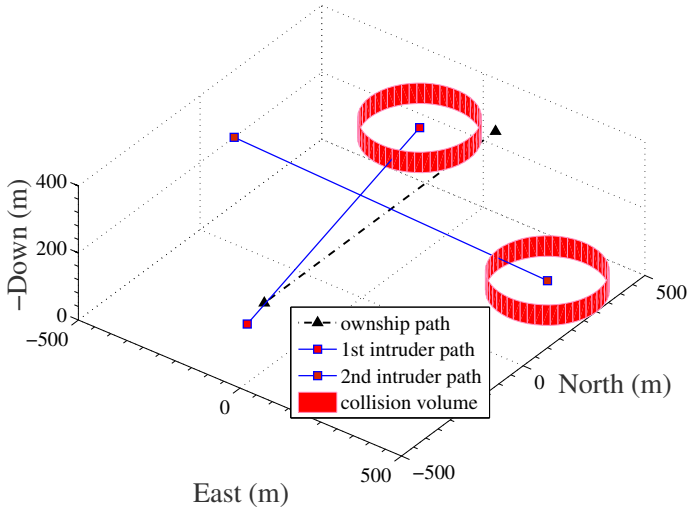


Figure 8.18: Encounter geometry of scenario number 1.

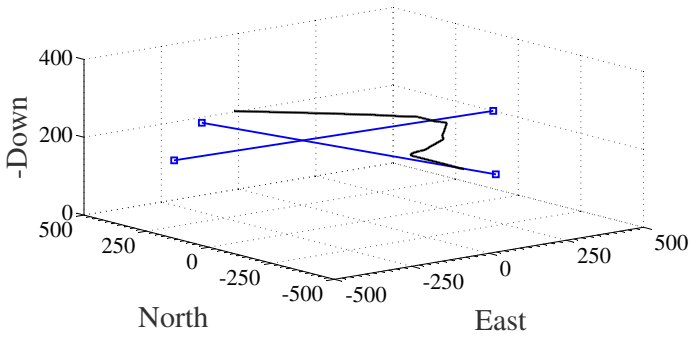
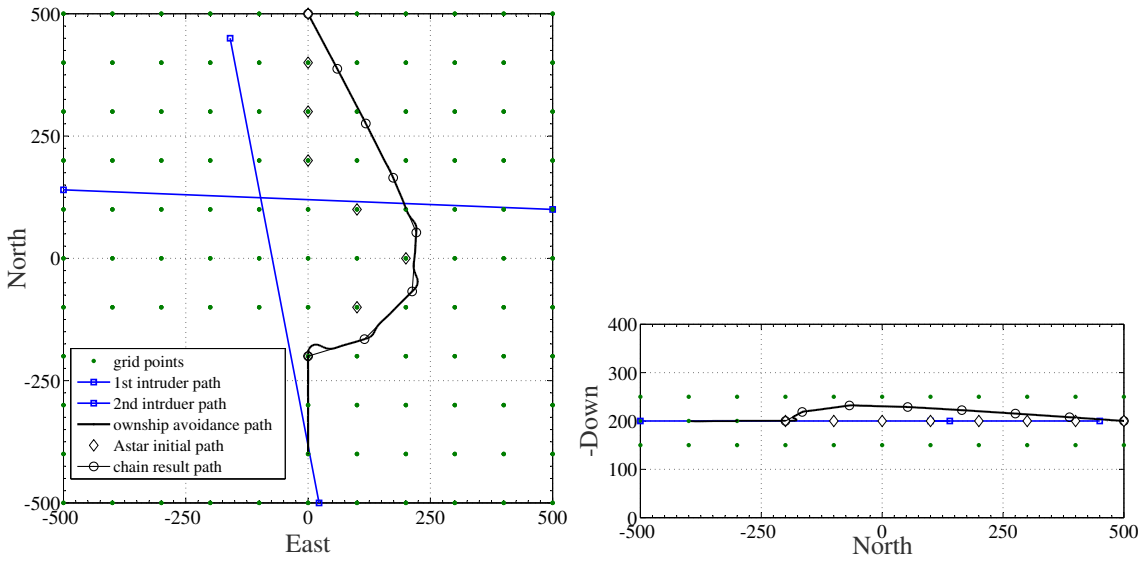


Figure 8.19: The avoidance path of the ownship.

In each encounter scenario the intruders are following a straight line path at a constant velocity and altitude. Figure 8.8 shows the first encounter scenario. It consists of two intruders: one is approaching head-on and the other is converging from the right. The speed of the intruders is 17 m/s and 15 m/s, respectively. The altitude of both intruders is 200 m. If no collision avoidance is planned, the d_{cpa} with respect to the first and second intruders



(a) Top view of the ownship's avoidance path. (b) side view of the ownship's avoidance path.

Figure 8.20: Avoidance path followed by the ownship in encounter scenario number 1.

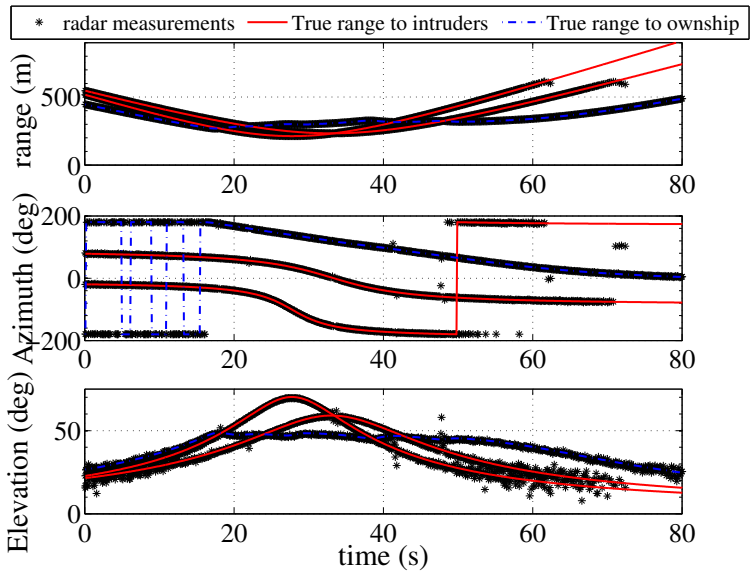
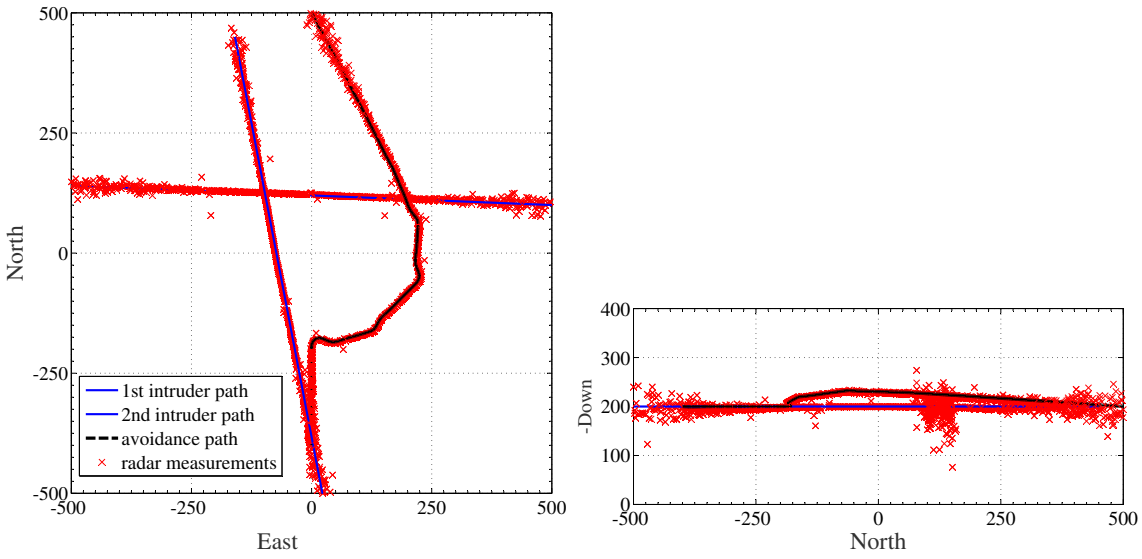


Figure 8.21: Radar measurements: range, azimuth and elevation.

is approx. 78.2 m and 10.5 m, respectively. Since, all aircraft are flying at same altitude and the d_{cpa} is less than the horizontal safety distance d_s, then these encounters will lead to a collision. Figures 8.19, 8.20(a) and 8.20(b) shows the intruder paths and the avoidance path planned by the ownship. Figure 8.21 shows the range, azimuth and elevation to all aircraft measured by the radar system. These figures predict that the principal signal decay



(a) Top view of the radar measurements in the NED inertial frame.

(b) side view of the radar measurements in the NED inertial frame.

Figure 8.22: Aircraft's paths with radar measurements in encounter scenario number 1.

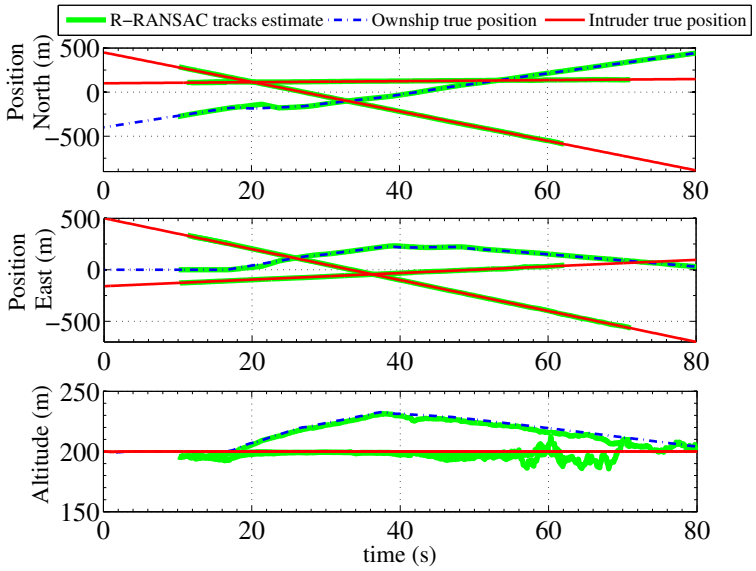


Figure 8.23: R-RANSAC tracks: position estimates of aircraft.

happens at low elevation angles as the aircraft enter and exit the antenna beam. Despite the noise in the elevation angle data, range and azimuth measurements maintain a high degree of accuracy until the aircraft exits the radar sensing volume at about 600 m. The aircraft with radar measurements in the NED inertial frame is shown in Figure 8.22. Figure 8.22(a) shows that the North and East coordinates constructed from radar measurements have a

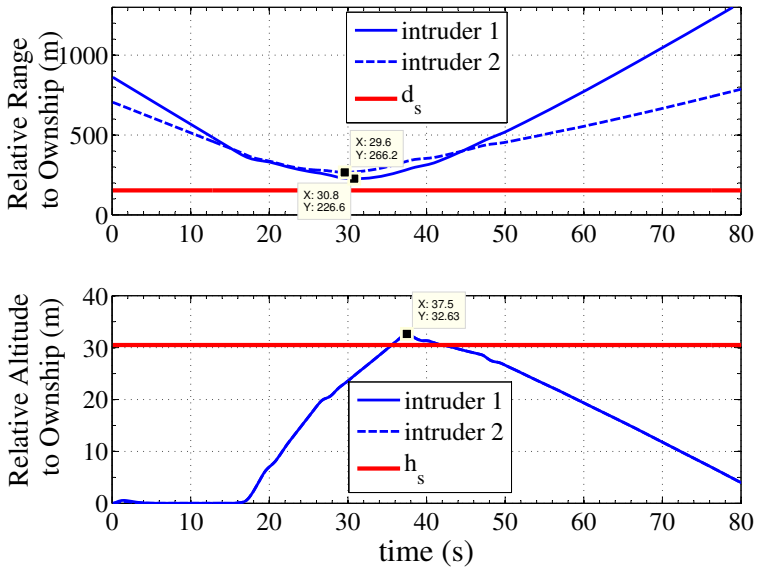


Figure 8.24: Relative range and altitude to intruders.

high degree of accuracy, whereas Figure 8.22(b) shows that the altitude has a lower degree of accuracy because of noisy elevation measurements. The state estimates of position are shown in Figure 8.23. Figure 8.23 shows that R-RANSAC is able to track all aircraft and distinguish the ownship from the intruders. From the figures we see that R-RANSAC takes about 10 seconds for good models to appear. This is due to the initial noisy radar elevation measurements, the sample rate at which R-RANSAC is running, and the underlying R-RANSAC design parameters. Both intruder tracks die slightly after we stop receiving radar measurements, due to the aircraft being outside the visible field of view of the radar; however, the ownship's track never dies. We also see that the estimates become more noisy at about 55 seconds due to the noisy elevation measurements that start occurring as the aircraft leave the field of view of the radar. Figure 8.24 shows the results of the relative range and altitude to both intruders. The results shows that no collision incidents occurs. The collision avoidance path increases the d_{cpa} to 226.6 m and 266.2 m with respect to the first and second intruders.

In the second encounter geometry, the ownship initially starts at $(-300, 0, -200)^\top$ in NED coordinates system, with an initial heading of 0 deg measured from North and follows a straight line path at constant speed of 13 m/s to reach a waypoint located at $(500, 0, -200)^\top$ as shown in Figures 8.25. This encounter scenario consists of four intruders flying at speed

of 16 m/s, 14 m/s, 13 m/s and 17 m/s, respectively. The intruder are flying at constant altitude of 250 m, 150 m, 220 m, and 200 m as shown in Figures 8.25.

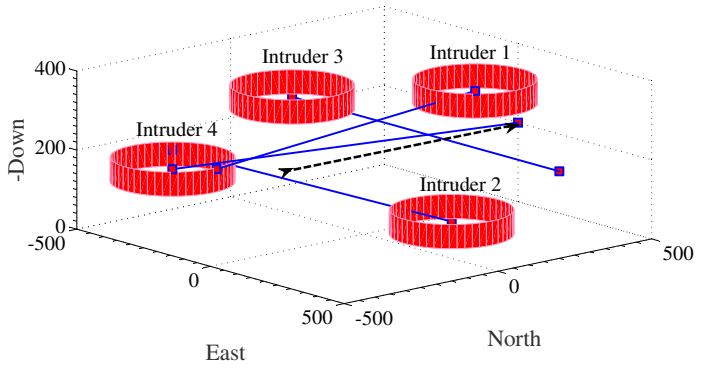


Figure 8.25: Encounter scenario number 2.

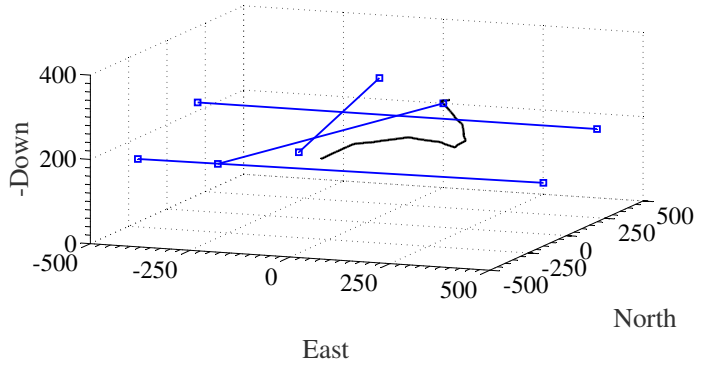
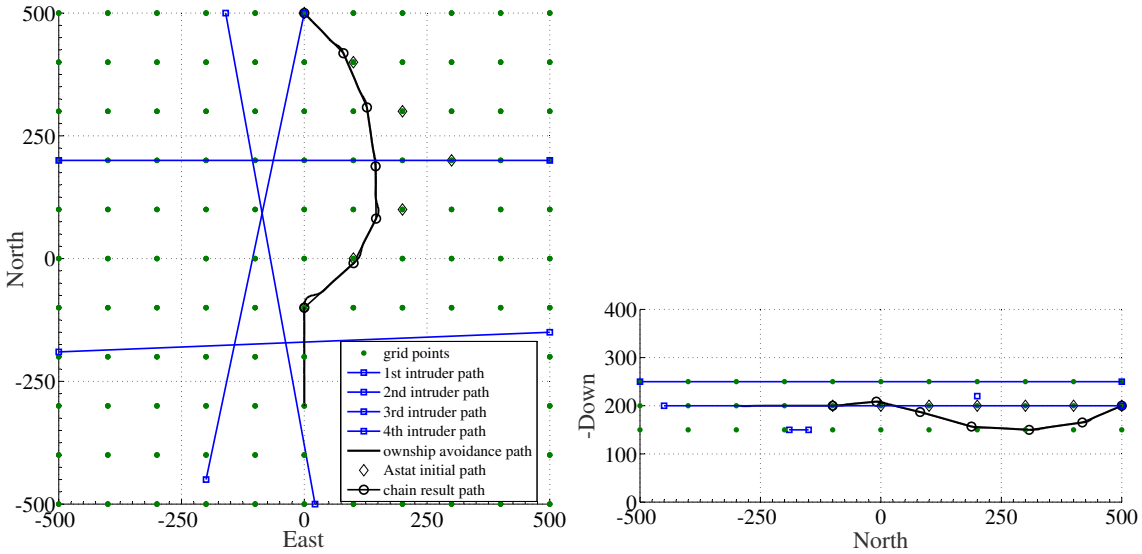


Figure 8.26: The avoidance path of the ownship.

Figures 8.26, 8.27(a) and 8.27(b) shows the planned avoidance path in the second encounter scenario. The d_{cpa} with respect to the four intruders is approx. 79.9 m, 240.3 m, 0 m and 40 m. Since all aircraft are flying at different altitudes, there will be a collision with the third and fourth intruder. The ownship, however, should plan an avoidance maneuver that does not lead into a collision with the remaining aircraft. Figures 8.26, 8.27(a) and 8.27(b) shows the intruder paths and the avoidance path planned by the ownship. Figures 8.28 and 8.29 demonstrate similar results as Figures 8.21 and 8.22. The addition of more intruders does not significantly degrade the radar's ability to detect multiple targets. Figure 8.29(a) demonstrate that the aircraft remain well resolved , even when in close proximity. Figure 8.30



(a) Top view of the ownship's avoidance path. (b) side view of the ownship's avoidance path.

Figure 8.27: Avoidance path followed by the ownship in encounter scenario number 1.

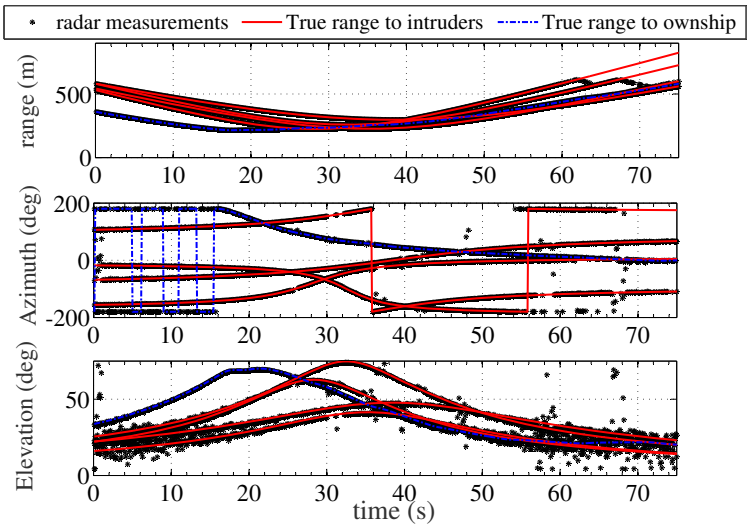
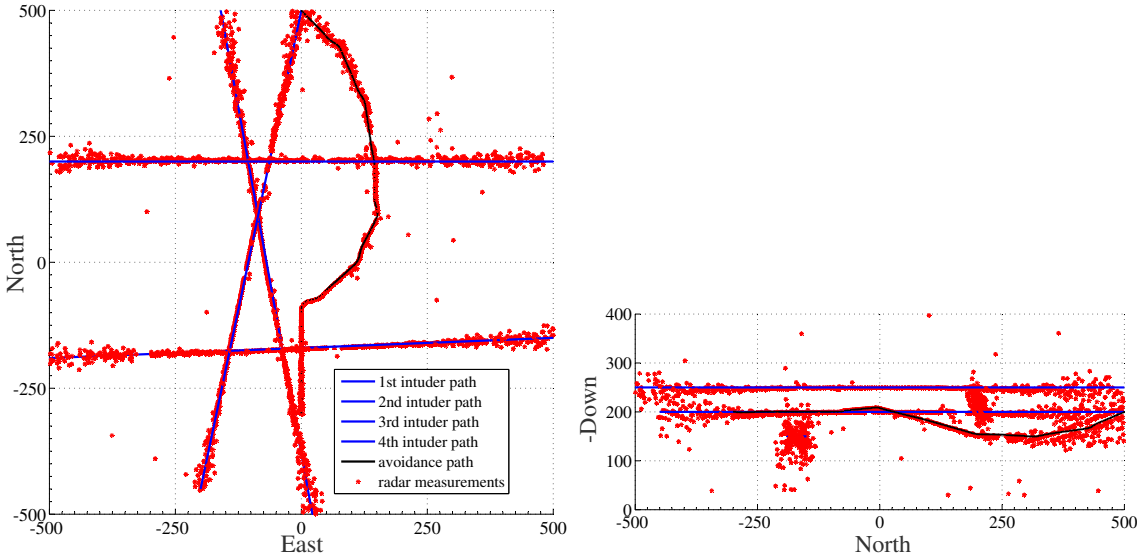


Figure 8.28: Radar measurements: range, azimuth and elevation.

shows similar results to those in the first scenario, in which we notice that the R-RANSAC algorithm is able to track multiple intruders. Figure 8.31 shows the results of the relative range and altitude to all intruders. The results shows that the avoidance path maneuvers the ownship safely without colliding. In Figure 8.31, the avoidance planner makes sure that when the relative horizontal range is less than d_s, the relative altitude is greater than h_s. For example, the relative range to the third intruder over time interval [39, 54.9] s is between



(a) Top view of the radar measurements of the ownship's avoidance path.

(b) side view of the radar measurements of the of ownship's avoidance path.

Figure 8.29: Aircraft's paths constructed using radar measurements in encounter scenario number 1.

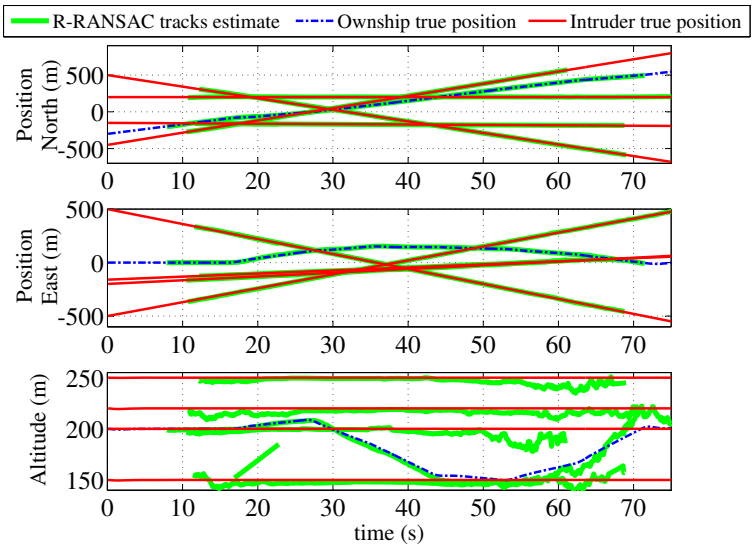


Figure 8.30: R-RANSAC tracks: position estimates of aircraft.

151.7 and 152.9 meters, however, over the same time interval the relative altitude is between 48.65 and 68.54 meters which is above the vertical safe distance h_s.

Another important aspect to evaluate the performance of the proposed algorithm is its ability to reduce the length of the avoidance path while avoiding the intruders. This is important because it reduces the amount of deviation from the original path, and ultimately

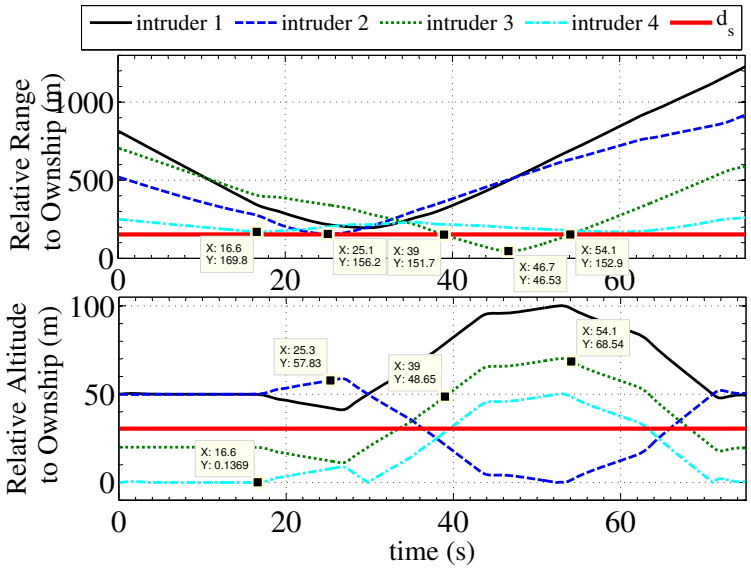


Figure 8.31: Relative range and altitude to intruders.

Table 8.4: Length of the avoidance path.

Scenario number	Initial path length (m)	Avoidance path length (m)
1	900	1564.2
2	800	1334.6

the flight time which is of critical importance for the small UAS with limited power resources. Table 8.4 shows that the length of the avoidance paths is fairly acceptable compared to the initial path length. We have also recorded the time required to execute the radar measurements processing logic T1, the R-RANSAC estimator T2, collision detection T3, and

Table 8.5: Ground-based radar SAA algorithms run time.

Scenario	1		2	
Time (s)	average	max.	average	max.
T1	0.2123	0.2926	0.3325	0.3899
T2	0.0075	0.0948	0.0102	0.0856
T3	0.00019	0.0012	0.00022	0.00071
T4	0.0527	0.2870	0.0484	0.2985

avoidance algorithm T4. Table 8.5 shows that the average and maximum time required to execute one cycle of these processes. These algorithms are running near real time in Matlab on an Intel i7 processor. The results show that the SAA algorithms requires at maximum a time of less than 1 second, however, the current implementation is coded in Matlab and was not optimized for run time. Significant speed increases could be achieved by optimizing the code and by porting it to C++.

8.5.4 Flight Test Results

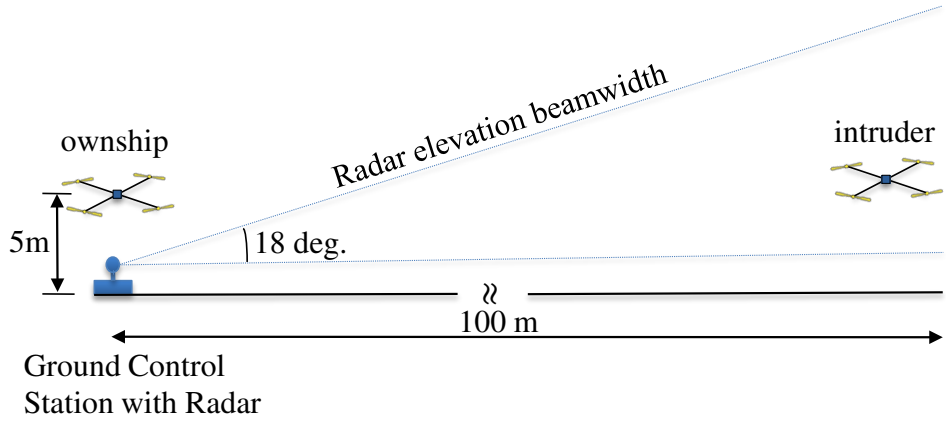


Figure 8.32: Sketch of the encounter geometry of flight test (not to scale).

To further examine the performance of the collision detection and avoidance algorithms, we conducted flight tests using 3D Robotics X8 multicopter for both the ownship and the intruder. An explanatory sketch of the flight experiments is depicted in Figure 8.32. In these experiments, we used simulated radar return measurements and simulated intruder telemetry states. The radar and the intruder dynamic model along with the R-RANSAC algorithm, and the collision detection and avoidance algorithms are running in Matlab on Lenovo Intel core i5 processor. The experimental setup is depicted in Figure 8.33. The telemetry of the ownship including position and velocity states and the avoidance waypoints are communicated between the ground-station laptop and the X8 multicopter ownship using the 915 MHz 3DR telemetry radio set. The flight tests are conducted over a region of 100×100 square meters, hence our choice of the collision volume is a scaled-down circular

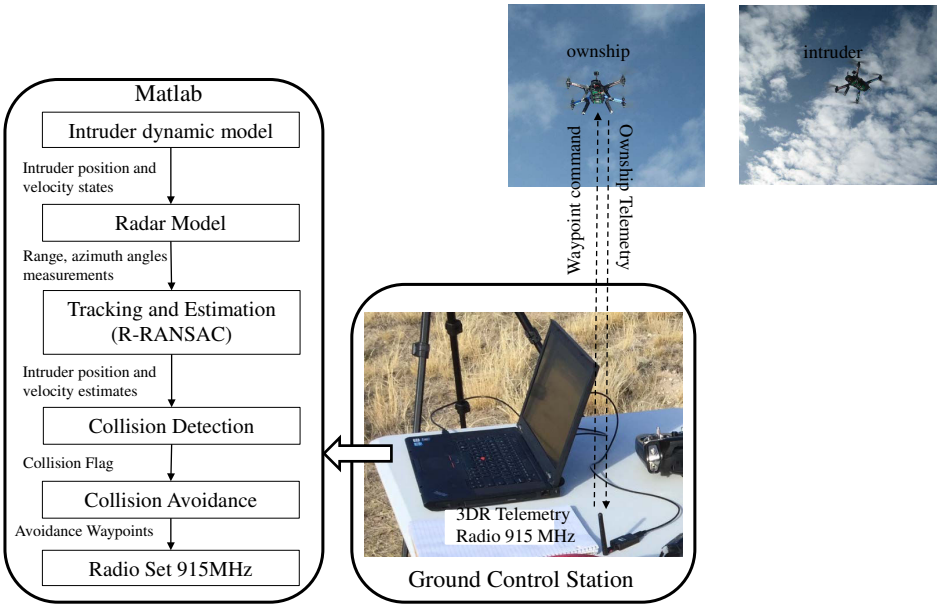


Figure 8.33: The ground-based radar SAA experiment structure diagram.

disk of radius d_s= 10 m. The collision detection and avoidance is limited to 2D path planning. Both the ownship and the intruder maintain a constant altitude of approximately 5 m above the ground.

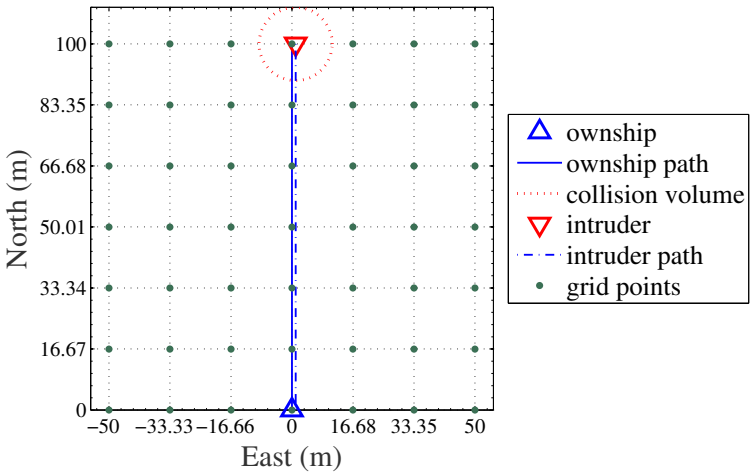


Figure 8.34: Encounter geometry of ground-based SAA flight test number 1.

In the first flight test we use a recursive implementation of depth-first search (DFS) algorithm [60]. Initially our focus was on testing the communication between different software and hardware pieces, and making sure that we are able to communicate with the X8

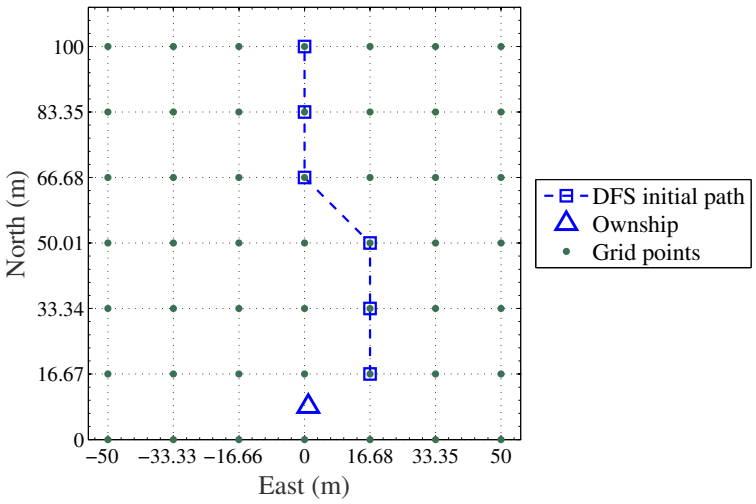


Figure 8.35: The depth-first search avoidance path of flight test number 1.

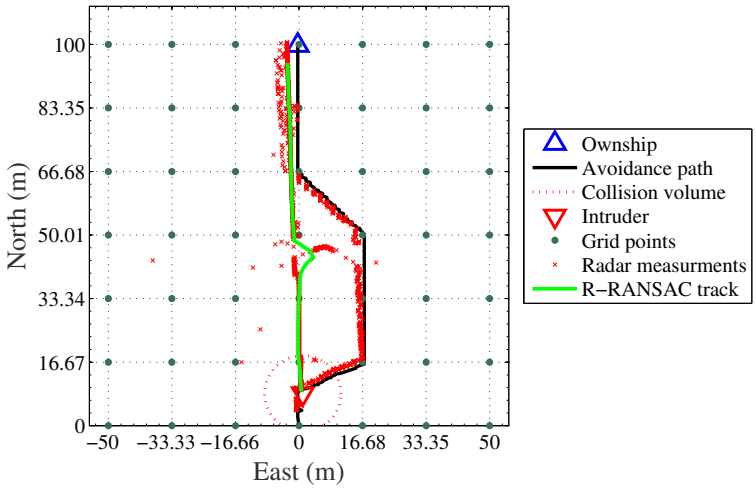


Figure 8.36: The avoidance path of the ownship and the radar measurements in the NED inertial frame of the aircraft of flight test number 1.

multicopter ownship and overwrite the initial mission waypoints with new avoidance waypoints. Therefore, as a first step we have used the DFS algorithm because it is simple and relatively easy to implement. The region is divided into a 7×7 grid. The cost for traveling between nodes on the grid is computed using Eq. (8.6). The geometry of the encounter scenario is shown in Figure 8.34. The ownship starts at $(0, 0, -5)^\top$ in NED coordinates system, with an initial heading of 0 deg measured from North and follows a straight line path at constant speed of 1 m/s to reach waypoint located at $(100, 0, -5)^\top$. The intruder start location is $(100, 1, -5)^\top$, with an initial heading of 180 deg measured from North and follows

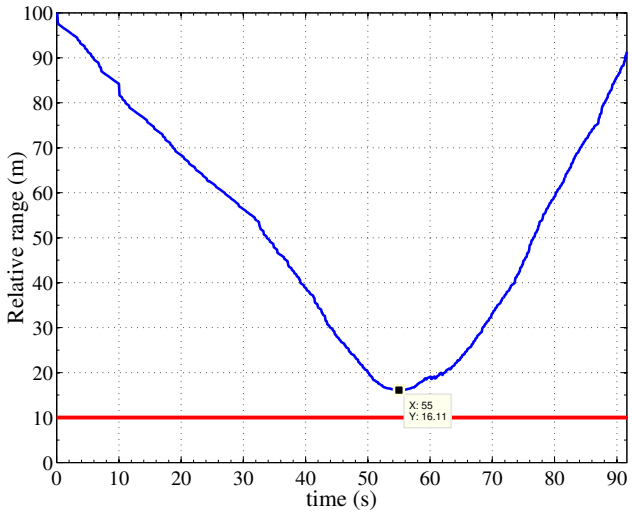
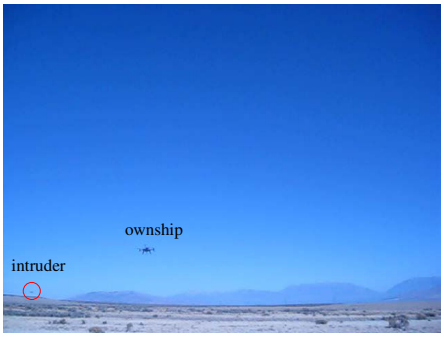


Figure 8.37: Relative range to the intruder of flight test number 1.

(a) Snap shot at t=0 s.

(b) Snap shot at t=0.22 s.



(c) Snap shot at t=0.28 s.

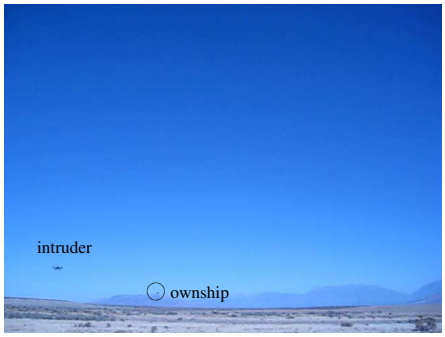
(d) Snap shot at t=0.45 s.

Figure 8.38: First set of snap shots of the ownship performing collision avoidance in encounter scenario number 1.

a straight line path at constant speed of 1 m/s as shown in Figure 8.34. This encounter is a collision scenario with t_{cpa} = 50 s and d_{cpa} = 1 m. The result of the depth-first search

algorithm is shown in Figure 8.35. The avoidance path followed by the ownship is shown in Figure 8.36. Figure 8.36 shows the the R-RANSAC tracks and the radar measurement transformed into the NED reference frame. Figure 8.37 shows the relative range to the intruder aircraft. The results shows that no collision incidents occurs. The collision avoidance path increases the d_{cpa} to 16.11 m which is considered safe with respect to the defined minimum safe distance, $d_s = 10$ m. A snap shots at several time instants of the ownship performing collision avoidance in encounter scenario number 1 are shown in Figures 8.38 and 8.39.

(a) Snap shot at t=1.10 s.



(b) Snap shot at t=1.35 s.

(c) Snap shot at t=1.50 s.

(d) Snap shot at t=1.58 s.

Figure 8.39: Second set of snap shots of the ownship performing collision avoidance in encounter scenario number 1.

In the second flight test we use the same setup geometry and structure depicted in Figures 8.32 and 8.33. We implement the A* and the chain based collision avoidance approach. Similar to the first experiment, we use simulated radar measurements and simulated intruder telemetry states. The region is divided into an 11×11 grid. The cost for traveling between nodes on the grid is computed using Eq. (8.6) and (8.8). The geometry of the encounter scenario is shown in Figure 8.40. The ownship starts at $(0, 0, -5)^\top$ in NED coor-

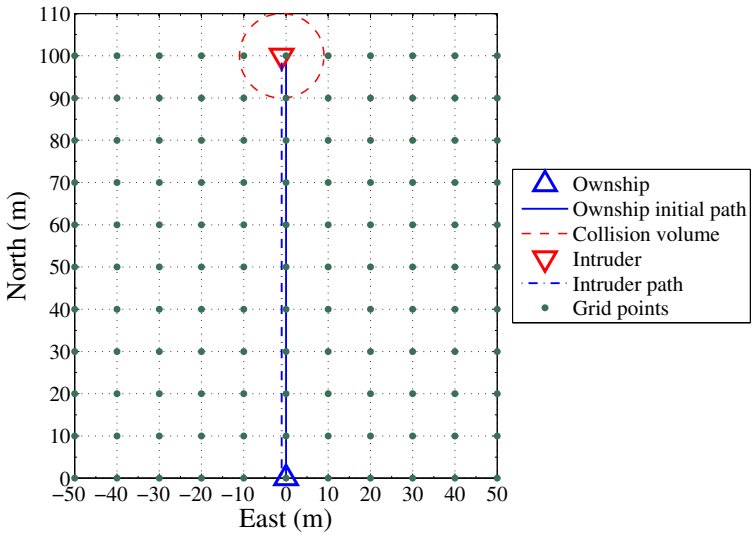


Figure 8.40: Encounter geometry of ground-based SAA flight test number 2.

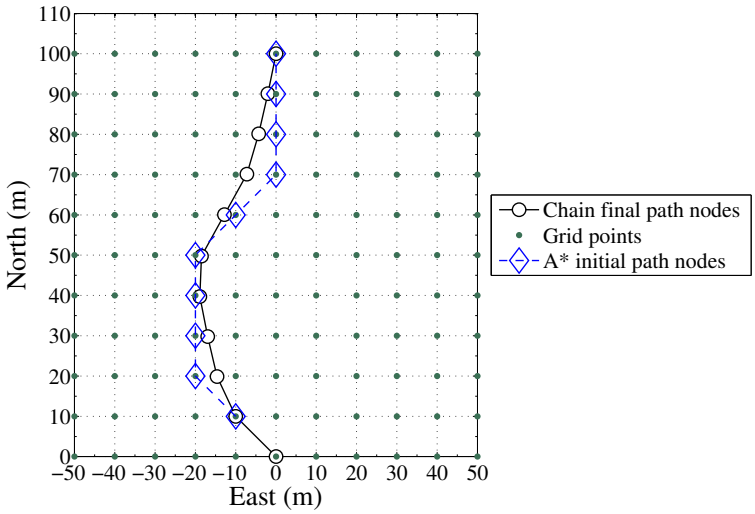


Figure 8.41: The A* initial search and the chain-based avoidance path of flight test number 2.

dinates system, with an initial heading of 0 deg measured from North and follows a straight line path at constant speed of 1 m/s to reach waypoint located at $(100, 0, -5)^\top$. The intruder start location is $(100, 1, -5)^\top$, with an initial heading of 180 deg measured from North and follows a straight line path at constant speed of 1 m/s as shown in Figure 8.40. This encounter is a collision scenario with $t_{cpa} = 50$ s and $d_{cpa} = 1$ m. The result of the A* initial search and the chain-based avoidance path are shown in Figure 8.41. The chain-based path smooth out the initial path provided by the A* search. The avoidance path followed by

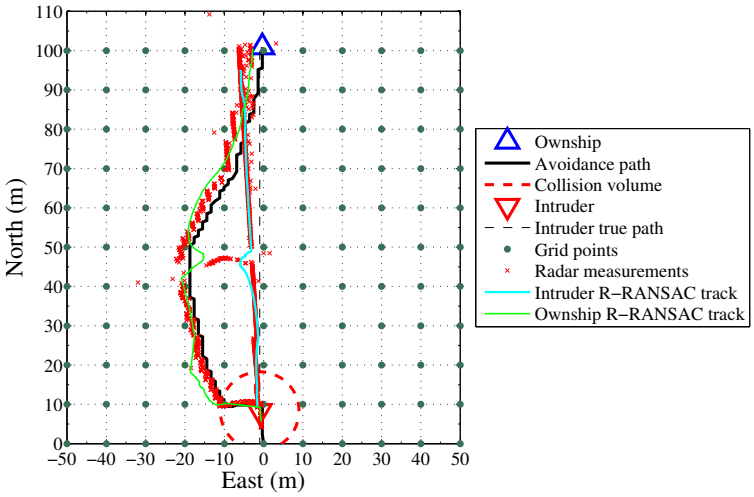


Figure 8.42: The avoidance path of the ownship and the radar measurements in the NED inertial frame of the aircraft of flight test number 2.

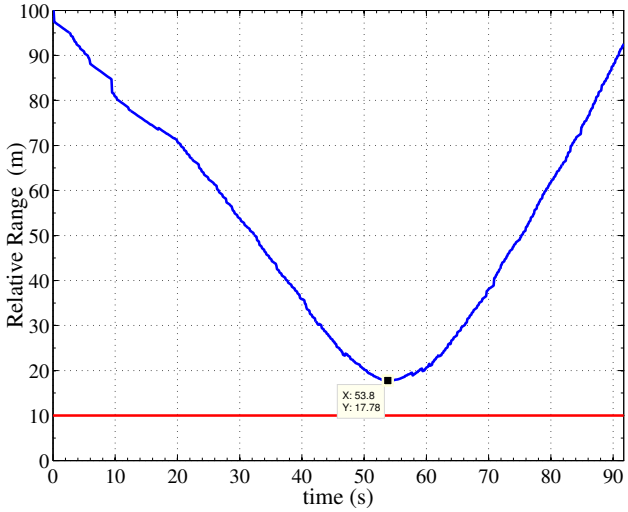


Figure 8.43: Relative range to the intruder of flight test number 2.

the ownship is shown in Figure 8.42. Figure 8.42 shows the the R-RANSAC tracks and the radar measurement transformed into the NED reference frame. Figure 8.43 shows the relative range to the intruder aircraft. The results shows that no collision incidents occurs. The collision avoidance path increases the d_{cpa} to 17.78 m which is considered safe with respect to $d_s = 10$ m.

8.6 Conclusions

The proposed approach can trade-off a collision-free path versus path length. One of the advantages of our approach is that timing information is specifically embedded in the future path representation of the vehicle, thus allowing the ownship to cross the physical path of the intruder as long as the crossings are well separated in time. It is also flexible, in that it can easily accommodate dynamic environment. A unique strength of this approach is the ability to use the dynamic chain analogy to smooth the path into a flyable shape while maintaining a safe distance from intruders. We evaluate the performance of the proposed collision detection and avoidance approach in typical encounter scenarios that include multiple intruders flying at different altitudes using simulated radar data and the R-RANSAC tracking algorithm. We demonstrate with flight tests using an X8 multicopter the capability of the proposed collision detection and avoidance approach to detect imminent collisions and plan avoidance paths in near real time. The proposed algorithms are running in near real time in Matlab. We expect that implementing these algorithms in a compiled language, such as C or C++, will show that real-time execution is feasible using hardware. The proposed architecture of the ground-based SAA system shown in Figure 8.3 assumes that all the sense-and-avoid tasks are computed on the ground, and that the telemetry and commands are communicated to the ownship. The proposed tracking, collision detection, and avoidance algorithms, however, are also viable for airborne solutions. In other words, the sensor measurements are processed on the ground and communicated to the ownship.

Chapter 9. Conclusion and Future Directions

I think and think for months and years. Ninety-nine times, the conclusion is false. The hundredth time I am right.

- Albert Einstein

In this dissertation, we have presented collision detection and avoidance techniques that further the development of an airborne sense-and-avoid systems for small unmanned aircraft systems. A key result in this work is developing collision detection and avoidance algorithms that are sensor independent. Another, important aspect of this work is the run-time of the proposed approaches. These approaches have been evaluated using different sensors modalities and applied for both airborne and ground-based SAA system. The run-time results indicates that the proposed collision detection and avoidance algorithms are capable of running in real time.

We have derived a close form analytical solution to compute the minimum sensing range required for the SAA system. We have developed two collision detection and risk assessment approaches based on deterministic and probabilistic framework. In the deterministic-based approach, we design a collision detection algorithm that evaluate an encounter based on distance at the CPA and time to the CPA. We account for uncertainties in the state estimates of the intruder by deriving an analytic expressions to propagate the error variance using a first-order Taylor series approximation. To better understand the assumptions and limitations of the error variance equation we derived it. The full derivation is presented in Appendix B. To address unanticipated intruder's maneuvers, we have developed a probabilistic-based collision detection framework using the concept of the reachability set and the transition probabilities contained in the UEM developed by MIT Lincoln Laboratory. For the collision avoidance part, we have proposed a collision avoidance algorithm

based on a simulated chain of waypoints that responds to a virtual force field produced by the encountering intruders. We have also developed a collision avoidance logic using an ownship centered coordinate system. The technique builds graph in the local-level frame and use DijkstraŠs algorithm to find the path with the least cost. Both algorithms account for multiple intruders in various encounter scenarios, and for uncertainties in the state estimates of the detected intruders.

We have implemented the proposed approaches using several sensors modalities including camera, radar and ADS-B with suitable tracking and estimation schemes. We have evaluated the performance of the proposed approaches with UEM LL MIT, and typical designed encounter scenarios using Monte Carlo Simulations.

To address ground-based radar SAA system for small UAS, we have developed two-step path planner approach based on the A* search algorithm and a chain of unit masses connected by springs and dampers. We have evaluated the performance of the proposed approach in typical encounter scenarios using simulated radar data and the R-RANSAC tracking algorithm. In addition, the A* search and the chain-based collision avoidance algorithm have been demonstrated in flight tests using an X8 multicopter platform. Finally, we conclude this dissertation with a summary of several avenues for potential future research building on what has been presented here.

9.1 Future Works

There are number of prospective research topics that may extend from this work. In Chapter 2 we present a close form solution to calculate the minimum detection range required for sense-and-avoid system. The approach is based on a geometric approximation of the point at which the velocity vector of the intruder is tangent to the collision volume. The geometry in Figure 2.2 is used to approximate the angle θ. The tangent point can be better determined if using the closest point of approach, i.e., the point at which the rate of change of the relative range between aircraft is zero.

In Chapter 3, the current implementation of probabilistic-based collision detection algorithm generates the reachable set using the current estimate of the intruder's position and velocity. As shown in Figure 9.1, the start point of the reachable set is the center of the

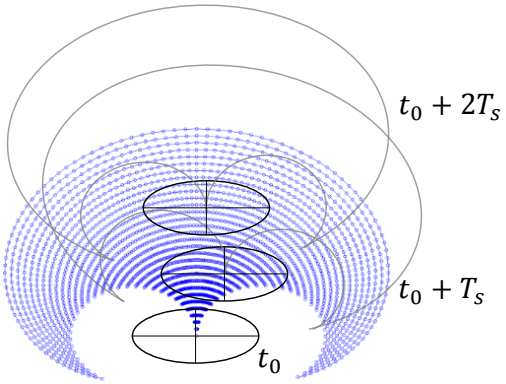


Figure 9.1: Top view of the 2D probability distribution function of uncorrelated position estimates.

ellipse that represent the 2D probability distribution of the position estimates. To improve the computation of the reachable set, the estimate of the intruder's turn rate can be used to reduce the size of the reachability set, and hence speedup the computation and decrease false alarms. For example, assume that the SAA system detects an intruder that is following a straight-line path. Initially, the collision detection algorithm builds the reachability set using the turn rates candidates $\dot{\psi} \in [-8, 8]$ deg/s. Observing turn rate estimates, in addition to the position and velocity estimates, can used to reduce the size of the turn rate candidates interval to $\dot{\psi} \in [-2, 2]$ deg/s or $\dot{\psi} \in [-1, 1]$ deg/s, and hence the size of the reachability region. In addition, extending the work to 3D and non level flight case is an important step forward for real-time implementation where altitude rate variations exists.

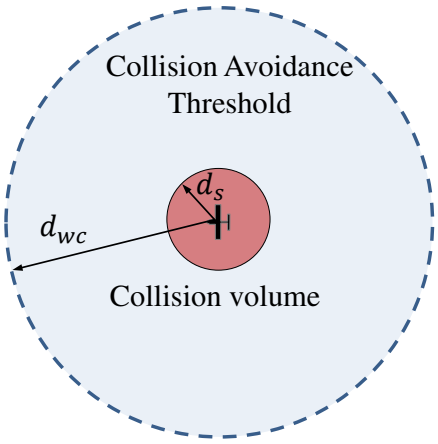


Figure 9.2: Well clear boundary and collision volume centered on current location of the ownship.

Through out this dissertation we used a cylindrical collision volume with radius of 500 ft and 200 ft. In Chapter 7, we adopted the well clear candidate selected by the Sense and Avoid Research Panel (SARP) [34]. The well clear boundary or the collision avoidance threshold is defined by a truncated cylinder that consists of a modified τ value of 35 s with distance of 4000 ft in the horizontal plane, and a vertical distance from the ownship of 700 ft. The question becomes: What is the optimal ratio between the size of collision volume and the size of the well clear boundary? The optimality can be defined based on minimizing the probability of near-mid air collisions (NMAC) given the size ratio between the collision volume and well clear boundary. In other words, if there exists a probability of collision ϱ, that need not to be exceeded, then what is the optimal size of the well clear volume (WCV) with respect to the size of the collision volume such that the SAA system should satisfy

$$Pr(\text{NMAC}|\text{WCV}) \leq \varrho.$$

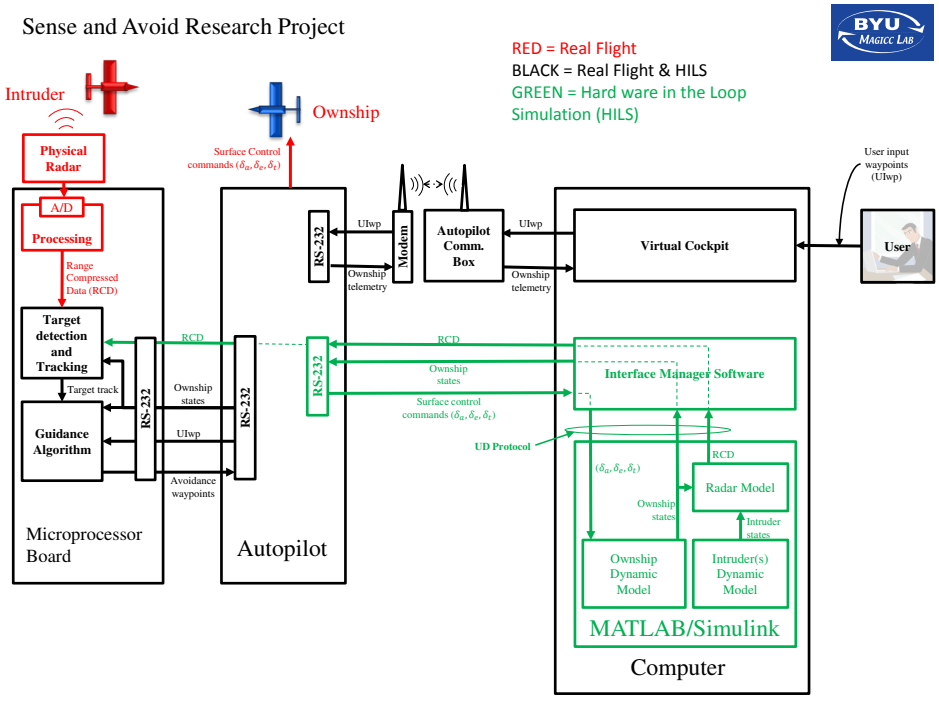


Figure 9.3: Hardware-in-the-loop simulation functional block diagram of sense and avoid system.

An important step forward to move towards a deployable airborne SAA system is to test and evaluate the performance of the close-loop of sensor, tracker, collision detection, and collision avoidance using hardware-in-the-loop simulation (HILS). Figure 9.3, shows an example of a functional block diagram of HILS for radar-based sense-and-avoid system. The same concept can be applied to different sensors. This step allows a better understanding of the behavior of the sense-and-avoid system when there exists an actual pieces of hardware. This step is fairly required to fine tune the estimator, and collision detection and avoidance algorithms. Also, it helps to better measure time-delays and account for them.

In Chapters 5, 6, and 7, we have used Monte Carlo simulations to evaluate the performance of one or more functions of the sense-and-avoid system. In these evaluation routines, we have either designed the encounter scenarios geometry or used the UEM developed by Lincoln Laboratory. Practically, the deployment of any SAA system requires a lengthy and comprehensive development process followed by a rigorous certification process. This work along with others, however, lay the initial technical foundation which require further analysis including using higher fidelity models of encounter airspace, representative number of simulations, and hardware-in-the-loop simulation. Unlike existing collision avoidance systems, an encounter model cannot be constructed solely from observed data, as UAS are not yet integrated in the NAS and good data does not exists. That will require, however, building encounter models similar to those developed to support the evaluation and certification of TCAS. Flight tests can play a role as part of the evaluation, however, this is not sufficient simply because system reliability and safety assessment cannot be determined by a few flight tests.

Appendix A. Manned Aviation Regulations

Aviation regulations, collectively known as Federal Aviation Regulation (FAR), are codified in the Code of Federal Regulations (CFR), Title 14, Chapter I. The FAR consists of several sections related to airworthiness certification (21-39), maintenance (43), aircraft registration and marking (45-49), pilot certification (61-67), airspace classes (71-77), operating rules (91-99) and special classes of vehicles (101-105). The CFR along with supplementary material like handbooks, orders, advisory circulares and technical standards orders issued by FAA, define appropriate standards, procedures, and practices to ensure that manufactures and operators are able to establish a minimum level of safety and reliability required for civil operations [36].

A.1 Airspace Categories and Classes

The airspace is defined as the portion of the atmosphere controlled by a country above its territory, including its territorial waters or, more generally, any specific three-dimensional portion of the atmosphere [131]. According to CFR, Title 14, part 71-73 and the Aeronautical Information Manual the national airspace is divided into two categories and four types based on the complexity or density of aircraft movements, proximity to airports, altitude, the nature of the operations conducted within the airspace, the level of safety required, and the national and public interest:

1. Regulatory that includes airspace class A, B, C, D and E, restricted and prohibited areas.

2. Non regulatory which include airspace class G, military operations areas (MOAs), warning areas, alert areas, and controlled firing areas.

While classes are mutually exclusive, categories can be defined within any class as needed. Categories typically used to provide additional information to pilots about hazards activities common to the airspace. There are four types within these two categories:

1. Controlled airspace,
2. Uncontrolled airspace,
3. Special use airspace and
4. Other airspace.

Airspace Classes A through E, ordered from more restrictive to less restrictive, corresponds to controlled airspace, however when overlapping airspace designations apply to the same airspace, the operating rules associated with the more restrictive airspace designation apply [CFR §71.9].

- **Airspace class A:** Generally most aircraft that fly above 18,000 ft above mean sea level (MSL) are capable of Instrument Flight Rules (IFR). Airspace class A was designed from 18,000 ft (FL180) above mean sea level (MSL) up to and including 60,000 ft (FL600). Unless otherwise authorized as per the [CFR §71.33] and the [CFR §91.167] through the [CFR §91.193] all persons must operate their aircraft under IFR and an aircraft needs to receive Air Traffic Control (ATC) clearance before entering Visual Flight Rules (VFR) weather minimums. There are also requirements for communication and transponder equipment. Ultralight vehicles and parachute jumps are prohibited without prior ATC permission.
- **Airspace class B:** Generally, the airspace from the surface to 10,000 ft MSL surrounding the busiest airports in terms of IFR operations or passengers. About 37 airports in the US are considered a Class B airspace. The configuration of each Class B airspace area is individually tailored and consists of a surface area and two or more layers (some Class B airspace areas resemble upside-down wedding cakes), and is designed to contain all published instrument procedures once an aircraft enters the airspace, see the [CFR §91.131]. An ATC clearance is required from the facility controlling that

area for all aircraft to operate in the area, and all aircraft that are so cleared receive separation services within the airspace. Aircraft must be equipped for all operations with an operable two-way radio capable of communicating with ATC on appropriate frequencies for that area and an operable radar beacon transponder with automatic altitude reporting equipment.

- **Airspace class C:** It extends airspace surrounding airports that have an operational control tower, but lack sufficient traffic to be considered Class B. About 123 Airports in the US are considered a Class C airspace. Class C has a mandatory requirement to communicate with the ATC before entering and to maintain the communication while within the class. The airspace usually consists of a 5 nmi radius core surface area that extends from the surface to 4,000 ft above ground level (AGL), and a 10 nmi radius shelf area that extends no lower than 1,200 ft up to 4,000 ft AGL. Aircraft should be equipped with a two-way radio and, unless otherwise authorized by ATC, an operable radar beacon transponder with automatic altitude reporting equipment, see the [AIM 3-2-4 (a)].

- **Airspace class D:** It extends from the surface to 2,500 ft AGL surrounding those airports that have an operational control tower, but lack sufficient traffic to be considered Class C. All pilots operating in Class D are required to communicate with the tower, regardless of weather conditions. When arriving, departing, or passing through Class D airspace, communications must also be established with the tower. When the tower is not operating but weather information is available, the airspace reverts to surface based Class E. If weather information is not available, the airspace reverts to Class G.

- **Airspace class E:** Generally, if the airspace is not class A, class B, class C, or class D, and it is controlled airspace, it is class E airspace [AIM 3-2-6 (a)]. In other words, it corresponds to the rest of controlled airspace. Class E extends from 14,500 ft up to class A boundary as well as the airspace above 60,000 ft [CFR §71.71]. Near airports, Class E may extend down to the surface and in the proximity of federal airways, and it extends from 700 or 1,200 ft AGL. Although no specific equipment is required, the pilot must establish two-way radio communication with the ATC when near airports.

No ATC clearance is required and flight can be VFR or IFR and no separation services are provided to VFR aircraft.

- **Airspace class G:** It corresponds to the uncontrolled airspace. In other words, any airspace that has not been designated as class A, B, C, D, or E airspace is class G. This class extends from the surface to 1200 ft AGL in all areas that are not designated by another classes and can extends above 1200 ft AGL in unpopulated areas. Since airspace class G is defined with respect to AGL, it rises and falls with the contours of the earth. Rules governing VFR flight have been adopted to assist the pilot in meeting the responsibility to see and avoid other aircraft. Minimum flight visibility and distance from clouds required for VFR flight are contained in [14 CFR §91.155].

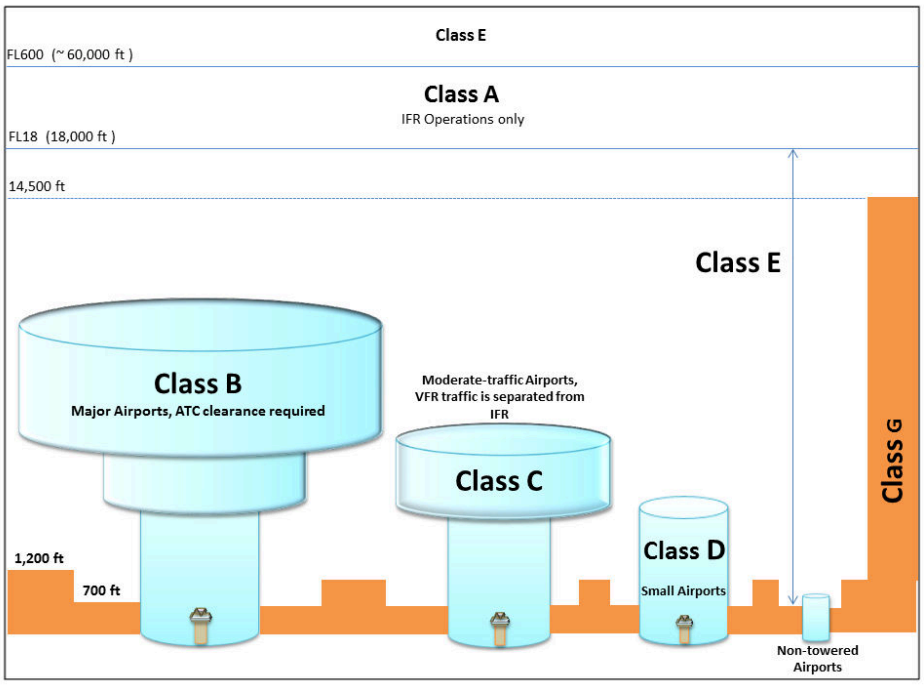


Figure A.1: The US national airspace classes.

A.2 General Operation and Flight Rules

Code of Federal Aviation, Title 14, part 91 has established right-of-way rules, aircraft speed, minimum safe altitudes, equipment, instrument, and certificate requirements, main-

tenance, preventive maintenance, special flight operations i.e. aerobatic flight, flight test areas, etc. Federal regulations do not permit pilots to fly below 10,000 ft or in proximity of airspace Class B, C, and D at speed exceeding 250 knots. It is not allowed for aircraft to operate close to another aircraft in a way that creates a collision hazard. The pilot as well must contribute to collision avoidance by being alert to "*see and avoid*" other aircraft [CFR §91.111]. The following is a summary of few flight rules that may be helpful in the design of sense-and-avoid system for small UAS.

- **Right-of-way rules – Except water operations:** When an aircraft has the right-of-way, the another aircraft shall give way to that aircraft and may not pass over, under, or ahead of it unless well clear [CFR §91.113]. The right-of-way rules are summarized as follows:
  - The aircraft that is in distress has right of way over all other air traffic.
  - Landing aircraft or aircraft on final approach to land have the right of way over other aircraft in flight or operating on the surface. Two or more aircraft are landing then the aircraft at the lower altitude has the right of way.
  - Aircraft being overtaken has the right of way. Overtaking aircraft shall alter course to the right to pass well clear.
  - In the approach head-on scenario, each aircraft shall alter course to the right to pass well clear.
  - The converging scenario has the following classification
    1. When aircraft of the same category then the aircraft to the other's right has the right of way.
    2. If the aircraft are of different categories then the following applies:
       (a) A balloon has the right-of-way over any other category of aircraft.
       (b) A glider has the right-of-way over an airship, powered parachute, weight-shift-control aircraft, airplane, or rotorcraft.
       (c) An airship has the right-of-way over a powered parachute, weight-shift-control aircraft, airplane, or rotorcraft.

(d) Aircraft towing or refueling other aircraft has the right of way over all other engine-driven aircraft.

- **Aircraft speed:** The minimum safe airspeed defined by the [CFR; §91.117] is 250 knots (288 mph) for any particular aircraft operating below 10,000 ft MSL. However, the minimum safe airspeed is 200 knots (230 mph) for any aircraft operating at or below 2,500 ft above the surface within 4 nmi of the primary airport of a Class C or Class D airspace area. Regarding Class B, [CFR §91.117] indicates that no person may operate an aircraft in the airspace underlying a Class B airspace designated for an airport or in a VFR corridor designated through Class B airspace at an indicated airspeed of more than 200 knots (230 mph).

- **Minimum safe altitudes:** According to [CFR §91.199] the minimum safe altitudes are categorized as follows:

  - Over congested areas: Any person operate an aircraft shall maintain an altitude of 1,000 ft above the highest obstacle within a horizontal radius of 2,000 ft of the aircraft.
  - Non-congested areas: Minimum of 500 ft AGL shall be maintained (except over open water or sparsely populated areas).
  - Anywhere else: any person operate an aircraft should maintain an altitude such that if the engine fails, an emergency landing may be executed without undue hazard to persons or property on the surface.

The CFR Title 14, Section 91 defines two types of flight rules, namely visual flight rules and instrument flight rules, which apply to all aircraft except unmanned rockets, unmanned free balloons, moored balloons, and ultralights. The following sections is a brief of both flight rules.

A.2.1 Visual Flight Rules (VFR)

In general, VFR are set of regulations under which a pilot operates an aircraft by controlling the aircraft's attitude and navigating to avoid obstacles and other aircraft. Ta-

Table A.1: Weather Minimums for VFR operations, see the [CFR §91.155], [36].

Airspace Class	Visibility (miles[1])	Distance from Clouds (ft)		
		Above	Below	Horizontal
Class A	N/A	Not Applicable		
Class B	3	Clear of Clouds		
Class C	3	1000	500	2000
Class D	3	1000	500	2000
Class E ($< 10,000$ ft)	3	1000	500	2000
Class E ($>= 10,000$ ft)	5	1000	500	1 mile[1]
Class G (day, $<= 1,200$ ft)	1	1000	500	2000
Class G (day, $< 10,000$ ft)	1	1000	500	2000
Class G (night, $< 10,000$ ft)	3	1000	500	2000
Class G ($>= 10,000$ ft)	5	1000	500	1 mile[1]

1: statute mile or land mile most commonly 5,280 ft as distinguished from the nautical mile, about 6,076.1 ft.

ble A.1 summarized the minimum visibility range and distance from clouds, known as the weather minimums as defined by the [CFR §91.155]. The VFR pilot is responsible for proper navigation separation from all other aircraft and required to *see and avoid* obstacles and other aircraft although they may be provided by advisory ATC service when flying in certain airspace classes. A VFR aircraft may be required to have a transponder in certain airspace classes. Flying under VFR conditions require the operator to make sure that the aircraft has sufficient fuel (considering wind and forecast weather conditions) to reach its destination, with a reserve to allow an additional 30 minutes of flight during the day and 45 minutes at night. Moreover, the [CFR §91.153] requires each person filing a VFR flight plan to include aircraft and pilot identification data, the point and proposed time of departure, the proposed route, cruising altitude, and true airspeed at that altitude.

There are additional flight rules categorized under VFR as special VFR (SVFR) and night VFR. The [CFR §91.157 defines relaxed weather minimums that are allowed under SVFR, making it a special case of operating under visual flight rules. Night VFR operations are not permitted in most countries; night flying is conducted by IFR. However, in the US, night VFR is allowed if and only if the airplane is equipped with functioning instruments

required for IFR operations and with electric landing lights required for night operations as per [CFR §91.205(d)].

A.2.2 Instrument Flight Rules (IFR)

Flying under IFR is also subject to the same restriction requirements in terms of fuel and should receive ATC clearance before flying. The IFR aircraft is required to select an operational and properly maintained radio navigation system to enable the aircraft to determine its position and stay on course by receiving radio signals transmitted by a network of fixed ground radio beacons. According to the [CFR §91.183], the IFR pilot is required to maintain appropriate communication frequency and report to ATC when it reaches pre-determined points, encounters unforecasted weather conditions, or any other problem that may affect flight safety.

A.3 UAS Rules and Regulations

The UAS access into the NAS is restricted by safety issues and these concerns are working against a rapid integration. The FAA has highlighted these concerns which are also shared by other NAS stakeholders including the Aircraft Owners and Pilots Association (AOAP) which insists that all sizes of UAS should be governed by the same rules that apply to manned aircraft. The AOAP is also against the use of restricted airspace to segregate UAS operations, as this will require pilots to circumnavigate the airspace, thus raising their expenses [36, 132].

The first efforts towards the UAS regulation took place as early as 1991, at which the time the FAA issued a notice for proposed rule making and formed an industry support group. In 2001, New Mexico State University published the first version of a document known as "High Altitude Long Endurance (HALE) UAV Certification and Regulatory Road-map" with the goal of finding a common language for discussion between the FAA, industry, and others for establishing regulations for airworthiness and flight standards that will allow safe operation of HALE UAS in the NAS [36]. Among different programs sponsored by NASA, FAA, DoD, and other stakeholders, the FAA seems to have adopted a cautious approach,

mainly because UAS technology is still under development and cannot considered mature. Their belief is that the process of UAS airworthiness certification is challenged by technical problems like sense and avoid systems and communication issues with ground stations, the ATC, and other aircraft [113].

Before October 2003, obtaining a Certificate of Authorization (COA) for UAS operations in the NAS has been limited to military operations based on agreements between the DoD and the FAA. After this date, the FAA has allowed non-military operations, mainly for proof-of-concept demonstrations. In September, 2005, the FAA issued "AFS-400 UAS Policy 05-01" which provided guidance on issuing COAs in which it was clarified that COA applications would not be available for civil UAS. This policy requires:

1. Observers to ensure that the UAS complies with the right-of-way rules and sense and avoid requirements outlined in the [CFR §91].

2. A safety analysis indicating that the risk of a mid-air-collision is extremely improbable.

The policy also requires pilot and observer minimum qualification and as well as operations guidelines. However, the FAA issued an updated guidance document titled "Interim Operational Approval Guidance 08-01", on March 2008 that replaced AFS-400. This document indicates that all UAS operators must at minimum comply with the CFR part 61 and 91. Although a "Sense and Avoid" requirement is still considered important, the presence of observers is no longer mandatory provided that other risk mitigation measures are proposed and validated through appropriate safety studies [36].

In March 2012, the FAA created a new UAS Integration office, headed by a single executive, which brings together specialists from the aviation safety and air traffic organizations. By July, the agency will request proposals to manage test sites with final decision making in December. The FAA has been working for several years to implement the provisions of Section 333 of the FAA Modernization and Reform Act of 2012 and move forward with UAS integration before proposing a small UAS rule. Several companies film production have approached the FAA and are also considering filing exemption requests. While UAS operations have increased as a result of the Section 333 exemptions approved since September of 2014, the overall realized benefit of UAS operations is still a small fraction of the

demand. Additionally Section 333 exemptions are not a long-term solution to supporting UAS in the National Airspace System. In laying the foundation for a long-term solution for UAS in the NAS, the Federal Aviation Administration has mandated that UAS be capable of an equivalent level of safety to the see-and-avoid mandate for manned aircraft [44, 55].

Appendix B. Derivation of Statistical Error Propagation

Definition 3. Let $X \in \mathbb{R}^n$ be an $n \times 1$ vector of jointly-Gaussian random variable (r.v.) with mean $\boldsymbol{\mu}_X$, and covariance matrix $\mathbf{Q} \in \mathbb{R}^{n \times n}$. Let Y be a new r.v. that is not measured directly but determined by $Y = g(X)$, where $g(\cdot) : \mathbb{R}^n \to \mathbb{R}$ be a function of X. If $g(X)$ is differentiable, and up to the first-order Taylor series approximation, $Y \approx g(\boldsymbol{\mu}_X) + \nabla g(X - \boldsymbol{\mu}_X)$, where $\nabla g(X) \in \mathbb{R}^{n \times 1}$ is the gradient vector $\frac{\partial g}{\partial X}$ evaluated at $\boldsymbol{\mu}_X$. Then, the r.v. Y has mean $\mu_{_Y} \approx g(\boldsymbol{\mu}_X)$ and variance given by

$$\sigma_{_Y}^2 \approx (\nabla g)^\top \mathbf{Q} \nabla g. \tag{B.1}$$

If the r.v. in X are uncorrelated, Eq (B.1) simplifies further to

$$\sigma_{_Y}^2 \approx (\nabla g)^\top \mathbf{Q}_d \nabla g, \tag{B.2}$$

where $\mathbf{Q}_d = diag([\sigma_{X_1}^2, \sigma_{X_2}^2, \cdots, \sigma_{X_n}^2])$.

Derivation: The Taylor series expansion of $Y = g(X)$ expanded about the mean $\bar{X}_i \triangleq E[X_i]$ where $i = 1, 2, \cdots, n$, and $E[\,]$ denotes the expectation operator, is given by

$$Y = g(\bar{X}) + \sum_{i=1}^{n} \frac{\partial g(X)}{\partial X_i}(X_i - \bar{X}_i) + \frac{1}{2!} \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\partial^2 g(X)}{\partial X_i \partial X_j}(X_i - \bar{X}_i)(X_j - \bar{X}_j) + H.O.T, \tag{B.3}$$

where $H.O.T$ stands for high order terms. The partial derivatives are evaluated at the mean $X_i = \bar{X}_i$. If the second and the higher order terms are neglected in Eq. (B.3), then it reduces to the first order Taylor series approximation

$$Y \approx g(\bar{X}) + \sum_{i=1}^{n} \frac{\partial g(X)}{\partial X_i}(X_i - \bar{X}_i). \tag{B.4}$$

The mean estimate of the output Y is defined by $\bar{Y} = E[Y]$. The mean estimate of the output $\bar{Y}$ is approximated using the first order Taylor series expansion given in Eq. (B.4) as

$$\begin{aligned} \bar{Y} &= E\Big[g(X)\Big], \\ &\approx E\Big[g(\bar{X})\Big] + E\Big[\sum_{i=1}^{n} \frac{\partial g(X)}{\partial X_i}(X_i - \bar{X}_i)\Big], \\ &\approx g(\bar{X}), \end{aligned} \tag{B.5}$$

where the expectation of the first central moment $E[(X_i - \bar{X}_i)]$ is zero. Subtracting Eq. (B.5) from Eq. (B.4) gives

$$Y - \bar{Y} \approx \sum_{i=1}^{n} \frac{\partial g(X)}{\partial X_i}(X_i - \bar{X}_i). \tag{B.6}$$

Squaring and taking the expectation of Eq. (B.6) gives

$$\begin{aligned} E[(Y - \bar{Y})^2] &\approx E\Big[\Big(\sum_{i=1}^{n} \frac{\partial g(X)}{\partial X_i}(X_i - \bar{X}_i)\Big)^2\Big], \\ \sigma_Y^2 &\approx \sum_{i=1}^{n}\Big(\frac{\partial g(X)}{\partial X_i}\Big)^2 E\Big[(X_i - \bar{X}_i)^2\Big] + \\ &\quad 2\sum_{j=i+1}^{n} \frac{\partial g(X)}{\partial X_i}\frac{\partial g(X)}{\partial X_j} E\Big[(X_i - \bar{X}_i)(X_j - \bar{X}_j)\Big], \\ \sigma_Y^2 &\approx \sum_{i=1}^{n}\Bigg(\Big(\frac{\partial g(X)}{\partial X_i}\Big)^2 \sigma_{X_i}^2 + 2\sum_{j=i+1}^{n} \frac{\partial g(X)}{\partial X_i}\frac{\partial g(X)}{\partial X_j}\sigma_{X_i X_j}^2\Bigg), \end{aligned} \tag{B.7}$$

where σ_Y^2 and $\sigma_{X_i}^2$ are the variances in the output Y and the measurements X_i, respectively, and the $\sigma_{X_i X_j}^2$ term is the covariance between X_i and X_j, $i \neq j$. We can further simplify Eq (B.7) by assuming that X_i are independent random variables, then the covariance term $\sigma_{X_i X_j}^2 = 0$, and Eq (B.7) reduces to

$$\sigma_Y^2 \approx \sum_{i=1}^{n}\Big(\frac{\partial g(X)}{\partial X_i}\Big)^2 \sigma_{X_i}^2. \tag{B.8}$$

Bibliography

[1] "Collision Avoidance Functional Requirements for Step 1," NASA, Edwards, CA, Tech. Rep., February 2006. 8

[2] "Guidance Material on Comparison of Surveillance Technologies (GMST)," International Civil Aviation Organization Asia And Pacific (ICAO), Tech. Rep., September 2007. 130

[3] "Standard Specification for Design and Performance of an Airborne Sense-and-Avoid System," ASTM International, West Conshohocken, PA, Tech. Rep. TR F2411-07, 2007. 6, 7

[4] "FAA Modernization and Reform Act of 2012," H.R. 658, pp. 1–145, Feb. 2012. 2, 3, 11

[5] "Unmanned Aircraft System (UAS) Service Demand 2015-2035," U.S. Department of Transportation, Tech. Rep., 2013. 2

[6] "Six Companies Can Now Fly Small UAS Following FAA-approved Safety Procedures," [online]:www.faa.gov/news/, September 25, 2014. 3

[7] J. W. Adaska, "Computing Risk For Unmanned Aircraft Self Separation With Maneuvering Intruders," in *2012 IEEE Digital Avionics Systems Conference*, 2012. 12

[8] V. J. Aidala and S. E. Hammel, "Utilization of Modified Polar Coordinates for Bearings-Only Tracking," *IEEE Transactions on Automatic Control*, vol. 28, pp. 283–294, 1982. 83, 94, 98, 100

[9] B. M. Albaker and N. A. Rahim, "A Survey of Collision Avoidance Approaches for Unmanned Aerial Vehicles," *International Conference for Technical Postgraduates (TECHPOS)*, pp. 1–7, 2009. 15

[10] ——, "Unmanned Aircraft Collision Detection and Resolution: Concept and Survey," *2010 5th IEEE Conference on Industrial Electronics and Applications*, pp. 248–253, 2010. 11

[11] P. Angelov, *Sense and Avoid in UAS: Research and Applications*. John Wiley & Sons, Ltd., 2012. 6, 7, 11, 12, 15, 130, 132, 150

[12] N. Archives, O. o. t. F. R. O. Records Administration (NARA), and T. G. P. O. (GPO), "Electronic Code of Federal Regulation (e-CFR)," [online] http://ecfr.gpoaccess.gov. 3

[13] M. E. Argyle, C. Chamberlain, and R. W. Beard, "Chain-Based Path Planning for Multiple UAVs," in *50th IEEE Conference on Decision and Control and European Control Conference*, Orlando, FL, USA, Dec. 2011. 15, 58, 60, 64

[14] Y. Bar-Shalom, X. R. Li, and K. Thiagalingam, *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, Ltd, 2001. 97, 99

[15] J. Barreiro, M. Do, and D. E. Smith, "Intelligent UAS Sense-and-Avoid Utilizing Global Constraints," ICAPS Schedualling and Planning Applications Workshop (SPARK), Portsmouth, NH, 2014. 17

[16] R. W. Beard and T. W. McLain, *Small Unmanned Aircraft: Theory and Practice*. Princeton University Press, 2012. 5, 14, 23, 79, 81, 99, 109, 143, 144

[17] K. D. Bilimoria, "A Geometric Optimization Approach to Aircraft Conflict Resolution," in *Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit*, 2010. 15

[18] A. N. Bishop, P. N. Pathirana, and A. V. Savkin, "Target Tracking with Range and Bearing Measurements via Robust Linear Filtering," in *3rd International Conference on Intelligent Sensors, Sensor Networks and Information*, no. 1, 2007, pp. 131–135. 98

[19] S. A. Bortoff, "Path Planning for UAVs," in *Proceedings of the American Control Conference*, Chicago, Illinois, June 2000, pp. 364–368. 15, 135

[20] J. D. Boskovic, J. A. Jackson, and R. K. Mehra, "Sensor and Tracker Requirements Development for Sense and Avoid Systems for Unmanned Aerial Vehicles," in *AIAA Modeling and Simulation Technologies (MST) Conference*, Boston, MA, Sep 2013. 7, 9

[21] A. Chakravarthy and D. Ghose, "Obstical Avoidance in a Dynamic Enviroment: A Collision Cone Approach," *IEEE Transactions on System, Man and Cybernitics, Part A: Systems and Humans*, vol. 28, no. 5, pp. 562–572, 1998. 15

[22] C. Chamberlain, "System Identification, State Estimation, and Control of Unmanned Aerial Robots," Master's thesis, Brigham Young University, Apr. 2011. 58

[23] R. H. Chen, A. Gevorkian, A. Fung, W. Chen, and V. Raska, "Multi-Sensor Data Integration for Autonomous Sense and Avoid," in *Proceedings of the AIAA Infotech at Aerospace*, 2011. 9, 16

[24] W. Chen, J. Kay, and V. M. Raska, "Autonomous Sense and Avoid (SAA) for Unmanned Air Systems (UAS)," in *NATO Research and Technology Organisation (RTO), RTO-SCI-202*, 2009. 12

[25] Y. Chiang, J. T. Klosowski, C. Lee, and J. S. B. Mitchell, "Geometric Algorithms for Conflict Detection/Resolution in Air Traffic Management," in *Proceedings of the 36th IEEE Conference on Decision and Control*, vol. 2, 1997, pp. 1835–1840. 13, 51

[26] J. P. Chryssanthacopoulos and M. J. Kochenderfer, "Hazard Alerting Based on Probabilistic Models," *Guidance, Control, and Dynamics*, vol. 35, no. 2, pp. 442–450, 2012. 14, 51

[27] M. Cirillo, "Air Traffic Bulletin: New Technology - ADS-B, TIS-B, and FIS-B," 2005. 107

[28] R. Clothier, J. Palmer, R. Walker, and N. Fulton, "Definition Of An Airworthiness Certification Framework For Civil Unmanned Aircraft Systems," *Safety Science*, vol. 49, pp. 871–885, 2011. 3

[29] E. J. Cockayne and G. W. C. Hall, "Plane Motion of a Particle Subject to Curvature Constraints," *SIAM*, vol. 13, no. 1, pp. 197–220, 1975. 13

[30] M. Consiglio, V. Carreno, and D. Williams, "Conflict Prevention and Separation Assurance Method in Small Aircraft Transportation System," in *AIAA 5th Aviation, Technology, Integration, and Operations Conference (ATIO)*, Arlington, Virginia, 2005. 39, 40, 114, 133, 134

[31] M. Consiglio, J. Chamberlain, C. Munoz, and K. Hoffler, "Concept Of Integration For UAS Operations In The NAS," in *28th International Congress of the Aeronautical Sciences (ICAS)*, Brisbane, Australia, 2012. 8, 114

[32] V. M. Contarino and M. R. Contarino, "Using ADS-B to Avoid Collisions Between UAS," in *Association for Unmanned Vehicle Systems International*, 2013. 11

[33] V. M. Contarino and L. Scire Consultants, "All Weather Sense and Avoid System for UASs," Report to the Office of Naval Research, Tech. Rep., 2009. 9, 10

[34] S. P. Cook, D. Brooks, R. Cole, D. Hackenberg, and V. Raska, "Defining Well Clear for Unmanned Aircraft Systems," in *AIAA Infotech @ Aerospace*. Kissimmee, Florida: AIAA, January 2015. 40, 114, 115, 118, 134, 174

[35] K. Dalamagkidis, K. P. Valavanis, and L. A. Piegl, "Current Status and Future Perspectives for Unmanned Aircraft System Operations in the US," *Journal of Intelligent and Robotic Systems*, vol. 52, pp. 313–329, 2008. 3

[36] ——, *On Integrating Unmanned Aircraft Systems into the National Airspace System*, 2nd ed. Springer Science+Business Media B.V., 2012, vol. 52. viii, 3, 6, 176, 182, 183, 184

[37] D. Dey, C. Geyer, S. Singh, and M. Digioia, "Passive, Long-range Detection of Aircraft: Towards a Field Deployable Sense and Avoid System," in *Field & Service Robotics*, 2009. 10

[38] E. W. Dijkstra, "A Note On Two Problems In Connection With Graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959. 16

[39] L. E. Dubins, "On Curves of Minimal Length with a Constriant on Average Curvature and with Prescribed Initial and Terminal Positions and Tangents," *American Journal of Mathematics*, vol. 79, no. 3, pp. 497–516, 1957. 46

[40] J. J. Enright, K. Savla, E. Frazzoli, and F. Bullo, "Stochastic and Dynamic Routing Problems for Multiple Uninhabited Aerial Vehicles," *Journal of Guidance, Control, and Dynamics*, vol. 32, no. 4, pp. 1152–1166, 2009. 13

[41] A. Farina, "Target Tracking with Bearings Only Measurements," *Signal Processing*, vol. 78, no. October 1998, pp. 61–78, 1999. 94, 98, 100

[42] Federal Aviation Administration, "Advisory Circular 20-165: Airworthiness Approval of Automatic Dependent Surveillance - Broadcast (ADS-B) Out Systems," pp. 1–67, 2010. 109

[43] ——, "Automatic Dependent Surveillance Broadcast (ADS-B) Out Performance Requirements To Support Air Traffic Control (ATC) Service," pp. 1–37, 2010. 10, 107

[44] ——, "Subchapter F - Air Traffic and General Operating Rules," pp. 579–844, 2015. 2, 109, 185

[45] P. Fiorini and Z. Shiller, "Motion Planning in Dynamic Environments Using Velocity Obstacles," *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998. 15

[46] E. Frew and R. Sengupta, "Obstacle Avoidance With Sensor Uncertainty For Small Unmanned Aircraft," in *43rd IEEE Conference on Decision and Control*, vol. 1, 2004, pp. 614–619. 13

[47] A. Ganguli and S. Avadhanam, "On the limits of collision detection performance of a sense-and-avoid system for non-cooperative air traffic," in *Proceedings of the AIAA Infotech Conference*. Atlanta, Georgia: AIAA, 2010. 85

[48] J. George and D. Ghose, "A Reactive Inverse PN Algorithm for Collision Avoidance among Multiple Unmanned Aerial Vehicles," in *American Control Conference*. St. Louis, MO, June 10-12: IEEE, pp. 3890–3895. 15

[49] S. George, "FAA Workshop on Sense and Avoid (SAA) for Unmanned Aircraft Systems (UAS)," 2009. 3, 8, 114

[50] A. Geser and C. Munoz, "A Geometric Approach to Strategic Conflict Detection And Resolution," in *Proceedings of the 21st IEEE Digital Avionics Systems Conference*, vol. 1, 2002. 13, 51

[51] C. Geyer, S. Singh, and L. Chamberlain, "Avoiding Collisions Between Aircraft: State of the Art and Requirements for UAVs operating in Civilian Airspace," Carnegie Mellon University, Tech. Rep., 2008. 6, 7, 12, 17, 21

[52] R. Gnanadesikan and J. R. Kettenring, "Robust Estimates, Residuals, and Outlier Detection with Multiresponse Data," *Biometrics*, vol. 28, no. 1, pp. 81–124, 1972. 69

[53] Grewal, Mohinder, S. and A. P. Andrews, *Kalman Filtering: Theory and Practice Using MATLAB*. New York: John Wiley & Sons, Ltd, 2008. 99

[54] S. Hammel and V. J. Aidala, "Observability Requirements For Three-Dimensional Tracking Via Angle Measurements," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 21, no. 2, pp. 200–207, 1985. 83

[55] S. B. Hottman, K. R. Hansen, and M. Berry, "Literature Review on Detect, Sense, and Avoid Technology for Unmanned Aircraft Systems," Tech. Rep., 2009. 2, 185

[56] J. Hu, J. Lygeros, M. Prandini, and S. Sastry, "Aircraft Conflict Prediction and Resolution Using Brownian Motion," in *Proceedings of the 38th Conference on Decision & Control*, no. December, Phoenix, 1999. 12

[57] Y. K. Hyunjin, Choi, "Reactive Collision Avoidance of Unmanned Aerial Vehicles Using a Single Vision Sensor," *AIAA Guidance, Control, and Dynamics*, vol. 36, no. 4, pp. 1234–1240, 2013. 15

[58] J. Jackson and J. Boskovic, "Application of Airspace Encounter Model for Prediction of Intruder Dynamics," in *AIAA Modeling and Simulation Technologies Conference*, ser. Guidance, Navigation, and Control and Co-located Conferences. AIAA, 2012. 13, 14, 45, 51

[59] L. E. Kavraki, P. Svestka, J. C. Latombe, and M. H. Overmars, "Probabilistic Roadmaps for Path Planning in High-dimensional Configuration Spaces," *IEEE Transactions on on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996. 15, 16

[60] J. Kleinberg and É. Tardos, *Algorithm Design*. Pearson Education India, 2006. 164

[61] M. J. Kochenderfer, J. K. Kuchar, L. P. Espindle, and J. D. Griffith, "Uncorrelated Encounter Model of the National Airspace System Version 1.0," Lincoln Laboratory, Masschusetts Institute of Technology, MA, Tech. Rep., 2008. 13, 49, 54, 118

[62] M. J. Kochenderfer, J. E. Holland, and J. P. Chryssanthacopoulos, *Lincoln Laboratory Journal*. 16

[63] K. J. K. G. J. D. Kochenderfer Mykel J., Espindle Leo P., "A Comprehensive Aircraft Encounter Model of the National Airspace System," *Lincoln Laboratory Journal*, vol. 17, pp. 41–53, 2008. 49

[64] Y. Koren and J. Borenstein, "Potential Field Methods And Their Inherent Limitations For Mobile Robot Navigation," in *IEEE International Conference On Robotics And Automation*, vol. 2. IEEE, 1991, pp. 1398–1404. 16

[65] M. Kothari and I. Postlethwaite, "A Probabilistically Robust Path Planning Algorithm for UAVs Using Rapidly-Exploring Random Trees," *International Robotic Systems*, vol. 71, pp. 231–253, 2013. 16

[66] J. Krozel, "Strategic Conflict Detection and Resolution for Free Flight," in *36th Conference of Decision & Control*, San Diego, CA, December 1997, pp. 1822–1828. 116

[67] J. Krozel, D. Andrisani, M. A. Ayoubi, T. Hoshizaki, and C. Schwalm, "Aircraft ADS-B Data Integrity Check," in *AIAA Aircraft Technology, Integration, and Operations Conference*, Chicago, IL, 2004, pp. 1–11. 112

[68] J. K. Kuchar and L. C. Yang, "A Review of Conflict Detection and Resolution Modeling Methods," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 4, pp. 179–189, Dec. 2000. 11, 15

[69] J. K. Kuchar, "Methodology for Alerting-System Performance Evaluation," *Guidance, Control, and Dynamics*, vol. 19, no. 2, pp. 438–444, 1996. 122, 123

[70] J. K. Kuchar and A. C. Drumm, "The Traffic Alert and Collision Avoidance System," *Lincoln Laboratory Journal*, vol. 16, no. 2, pp. 277–296, 2007. 17

[71] J. Lai, J. J. Ford, L. Mejias, P. O. Shea, and R. Walker, "See and Avoid Using Onboard Computer Vision," in *Sense and Avoid in UAS: Research and Applications*, 2012, ch. 10, pp. 265–294. 9

[72] T. M. Lam, M. Mulder, M. Van Paassen, J. A. Mulder, and F. C. Van Der Helm, "Force-Stiffness Feedback in Uninhabited Aerial Vehicle Teleoperation with Time Delay," *AIAA Guidance, Control, and Dynamics*, vol. 32, no. 3, pp. 821–835, 2009. 15

[73] T. A. Lauderdale, "Probabilistic Conflict Detection for Robust Detection and Resolution," in *12th AIAA Aviation Technology, Integration and Operation (ATIO) Conference*, Indianapolis, IN., Sep 2012. 14, 51

[74] S. M. LaValle, "Rapidly-Exploring Random Trees: A New Tool for Path Planning," 1998. [Online]. Available: http://msl.cs.uiuc.edu/~lavalle/rrtpubs.html 15, 16

[75] ——, *Planning Algorithms*. Cambridge University Press, 2006. 16, 80, 134, 136

[76] E. S. Lee and R. N. Forthofer, *Analyzing Complex Survey Data*, 2nd ed. Thousand Oaks, CA: Sage Publications, Inc., 2006. 40

[77] S. M. Lee, C. Park, M. A. Johnson, and E. R. Mueller, "Investigating Effects of Well Clear Definitions on UAS Sense-And-Avoid Operations," in *Aviation Technology, Integration, and Operations Conference*. Los Angeles, CA: AIAA, 2013. 8

[78] Y. Lee, J. H. Yang, J. Kuchar, and E. Feron, "A Real-Time Monte Carlo Implementation for Computing Probability of Conflict," in *Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit*, vol. 178, 2004, pp. 1835–1840. 14, 51

[79] D. Lerro and Y. Bar-Shalom, "Tracking with Debiased Consistent Converted Measurements Versus EKF," *IEEE Transactions on Aerospace and Electronic Systems, AES*, vol. 29, no. 3, pp. 1015–1022, 1993. 98

[80] A. Lindgren and K. Gong, "Position And Velocity Estimation Via Bearings Observations," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 14, no. 4, pp. 564–577, JulY 1978. 83

[81] F. Lindsten, P. Nordlund, and F. Gustafsson, "Conflict Detection Metrics for Aircraft Sense and Avoid Systems," in *Proceedings of the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, Spain, 2009. 14, 51

[82] B. Luders, S. Karaman, E. Frazzoli, and J. P. How, "Bounds on Tracking Error using Closed-Loop Rapidly-Exploring Random Trees," in *American Control Conference*. Baltimore, MD: IEEE, 2010. 16

[83] B. Luders, S. Karaman, and J. P. How, "Robust Sampling-based Motion Planning with Asymptotic Optimality Guarantees," in *Guidance, Navigation, and Control (GNC) Conference*. Boston, MA: AIAA, 2013. 16

[84] J. Mackie, J. Spencer, and K. F. Warnick, "Compact FMCW Radar for GPS-Denied Navigation and Sense and Avoid," in *IEEE Antennas and Propagation Society, AP-S International Symposium*, 2014, pp. 989–990. 9

[85] P. C. Mahalanobis, "On The Generalised Distance In Statistics," *Proceedings of the National Institute of Sciences of India 2*, vol. (1), pp. 49–55, 1936. 69

[86] F. Martel, M. Mullins, N. Kaabouch, and W. Semke, "Flight Testing of an ADS-B-based Miniature 4D Sense and Avoid System for Small UAS," in *AIAA Infotech@Aerospace Conference, pages = 1–7, month = March, year = 2011.* 11

[87] F. Martel, R. R. Schultz, W. H. Semke, Z. Wang, and M. Czarnomski, "Unmanned Aircraft Systems Sense and Avoid Avionics Utilizing ADS-B Transceiver," in *AIAA Infotech@Aerospace Conference*, April 2009, pp. 1–8. 11

[88] T. W. McLain and R. W. Beard, "Trajectory Planning For Coordinated Rendezvous Of Unmanned Air Vehicles," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, vol. 4369. AIAA Reston, VA, 2000, pp. 1–8. 15, 58, 63, 135

[89] K. Mehrotra and P. R. Mahapatra, "A Jerk Model for Tracking Highly Maneuvering Targets," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, no. 4, pp. 1094–1105, 1997. 112

[90] R. Melnyk, S. Daniel, V. Vitali, and J. Hernando, "Sense and Avoid Requirements for Unmanned Aircraft Systems Using a Target Level of Safety Approach," *Risk Analysis*, vol. 34, no. 10, pp. 1894–1906, 2014. 7

[91] K. S. Miller and D. M. Leskiw, "Nonlinear Estimation With Radar Observations," *IEEE Transactions on Aerospace and Electronic Systems, AES*, vol. AES-18, no. 2, pp. 192–200, 1982. 98

[92] C. Mirolo and E. Pagello, "A Cell Decomposition Approach to Motion Planning Based on Collision Detection," in *Proceedings of the 1995 International Conference on Advanced Robotics*, 1995, pp. 481–488. 16

[93] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, "A Time-Dependent Hamilton-Jacobi Formulation of Reachable Sets for Continuous Dynamic Games," *IEEE Transactions on Automatic Control*, vol. 50, no. 7, pp. 947–957, 2005. 13

[94] S. C. Mohleji and G. Wang, "Modeling ADS-B Position and Velocity Errors for Airborne Merging and Spacing in Interval Management Application," Center for Advanced Aviation System Development, McLean, VA, Tech. Rep., 2010. 108, 109

[95] C. Moody and R. Strain, "Implementation Consideration for Automatic Dependent Surveillance - Broadcast on Unmanned Aircraft Systems," in *AIAA Infotech@Aerospace Conference*, Reston, Virigina, Apr. 2009, pp. 1–8. 10, 11, 110

[96] W. Morchin, *Radar Engineer's Sourcebook*. Artech House, 1993, pp. 110–118. 101, 102

[97] A. Moses, M. J. Rutherford, and K. P. Valavanis, "Radar-based detection and identification for miniature air vehicles," in *IEEE International Conference on Control Applications (CCA)*. IEEE, 2011, pp. 933–940. 150

[98] C. Munoz, A. Narkawicz, and J. Chamberlain, "A TCAS-II Resolution Advisory Algorithm," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, no. AIAA-2013-4622, Boston, MA, 2013. 13, 37, 51

[99] S. C. Nardone and V. J. Aidala, "Observability Criteria For Bearings-Only Target Motion Analysis," *IEEE Transactions on Aerospace and Electronic Systems, AES*, vol. 17, no. 2, pp. 161–166, march 1981. 83

[100] P. C. Niedfeldt, "A Novel Multiple Target Tracking Algorithm in Clutter:Recursive-RANSAC," Ph.D. dissertation, Department of Electrical and Computer Engineering, Brigham Young University, June 2014. 131, 152

[101] O. Shakernia W. Chen and V. Raska, "Passive Ranging For UAV Sense And Avoid Applications," in *Proceedings of the AIAA Infotech at Aerospace*, 2005. 83

[102] R. A. Paielli and H. Erzberger, "Conflict Probability Estimation for Free Flight," *AIAA Guidance, Control, and Dynamics*, vol. 20, no. 3, pp. 588–596, 1997. 12, 14, 51

[103] A. Papoulis and S. U. Pillai, *Probability, Random Variables, and Stochastic Processes*, 4th ed. McGraw-Hill New York, 2002. 108

[104] A. Payne, "Observability problem for bearings-only tracking," *International Journal of Control*, vol. 49, pp. 761–776, 1989. 83

[105] M. Prandini, J. Hu, J. Lygeros, and S. Sastry, "A Probabilistic Approach to Aircraft Conflict Detection," *IEEE Transactions on Inteliget Transportation Systms*, vol. 1, no. 4, pp. 199–220, 2000. 6

[106] X. Prats, D. Luisn, R. Jorge, R. Pablo, and P. Enric, "Requirements, Issues, and Challenges for Sense and Avoid in Unmanned Aircraft Systems," *Aircraft*, vol. 49, no. 3, pp. 677–687, 2012. 6

[107] Radio Technical Commision for Aeronautics, "Minimum Aviation System Performance Standards for Aircraft Surveillance Applications," 2003. 109

[108] ——, "DO-282B Minimum Operational Performance Standards for Universal Access Transceiver (UAT) Automatic Dependent Surveillance - Broadcast (ADS-B) DRAFT Final Review and Comment," 2009. 109

[109] ——, "DO-260B Minimum Operational Performance Standards for 1090 MHz Extended Squitter Automatic Dependent Surveillance & Broadcast (ADS-B) and Traffic Information Services & Broadcast (TIS-B)," 2011. 109

[110] S. Rajnikant, J. B. Saunders, and R. Beard W., "Reactive Path Planning for Micro Air Vehicles Using Bearing-only Measurements," *International Robotic Systems*, vol. 65, no. 1-4, pp. 409–416, 2012. 15

[111] M. A. Richards, J. Scheer, and W. A. Holm, *Principles of Modern Radar: Basic Principles*. SciTech Publishing, Inc., 2010. 29, 95, 96

[112] S. M. S. and K. Kroschel, "Limits in Tracking with Extended Kalman Filters," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 40, no. 4, pp. 1351–1359, 2004. 98

[113] N. Sabatini, "Assuring the safe integration of UAS," *Unmanned Aircraft Systems: The Global Perspective*. 184

[114] SAE International, "AS8002A Air Data Computer - Minimum Performance Standard," pp. 1–19, 1996. 109

[115] L. R. Sahawneh, M. E. Argyle, and R. W. Beard, "3D Path Planning for Small UAS Operating in Low-Altitude Airspace," in *The 2016 American Control Conference*, Boston, MA, 6-8, July 2016. 20, 136

[116] L. R. Sahawneh and R. W. Beard, "A probabilistic Framework for Unmanned Aircraft Systems Collision Detection and Risk Estimation," in *53rd IEEE Conference on Decision and Control*, Los Angeles, CA, December 15-17. 20

[117] L. R. Sahawneh, M. O. Duffield, R. W. Beard, and T. W. McLain, "Detect and Avoid for Small Unmanned Aircraft Systems using ADS-B," *Air Traffic Control Quartely: An International Journal of Engineering and Operations*, vol. 23(2/3), no. 1-38, 2015. viii, x, 20, 105, 106, 107, 108, 112, 124

[118] L. R. Sahawneh, J. Mackie, J. Spencer, R. W. Beard, and K. F. Warnick, "Airborne Radar-Based Collision Detection and Risk Estimation for Small Unmanned Aircraft Systems," *Aerospace Information Systems*, pp. 1–11, 2015. 9, 20, 94, 122

[119] L. R. Sahawneh, J. Spencer, R. W. Beard, and K. F. Warnick, "Minimum Required Sensing Range for UAS Sense and Avoid Systems," in *AIAA Infotech@Aerospace*. San Diego, California: AIAA, 4-8 January 2016. viii, 20, 28, 30

[120] L. R. Sahawneh, B. R. W., S. Avadhanamz, and H. Bai, in *AIAA Guidance, Navigation, and Control (GNC) Conference*, Boston, MA. 15, 20, 85

[121] J. Saunders and R. W. Beard, "Vision-based Reactive Multiple Obstacle Avoidance for Micro Air Vehicles," in *IEEE American Control Conference ACC'09*, St. Louis, MO, June 10-12, 2009, pp. 5253–5258. 15

[122] R. J. E. Subhash Challa, Mark R. Morelande, Darko Mušicki, *Fundamentals of Object Tracking*. New York: Cambridge University Press, 2011. 97

[123] S. Temizer, M. J. Kochenderfer, L. P. Kaelbling, T. Lozano-Pï£ · rez, and J. K. Kuchar, "Collision Avoidance for Unmanned Aircraft using Markov Decision Processes," in *Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit*, 2010. 12, 13, 17

[124] F. E. Udwadia and R. E. Kalaba, *Analytical Dynamics: A New Approach*. Cambridge University Press, 1996. 63

[125] J. Upchurch, C. Muñoz, A. Narkawicz, J. Chamberlain, and M. Consiglio, "Analysis of well-clear boundary models for the integration of uas in the nas," Langley Research Center, Hampton, Virginia, Tech. Rep. NASA/TMŰ2014Ű218280, 2014. 37

[126] US Department of transportation and Federal Aviation Adminstration, *Aeronautical Information Manual Official Guide to Basic Flight Information and ATC Procedures.* 7

[127] C. E. van Daalen and T. Jones, "Fast Conflict Detection Using Probability Flow," *Automatica*, vol. 45, pp. 1903–1909, 2009. 14, 51

[128] M. E. Weber, "Faa surveillance radar data as a complement to the wsr-88d network," in *Preprints, 9th Conference on Aviation Range and Aerospace Meteorology*, 2000. 150

[129] R. E. Weibel, M. W. M. Edwards, and C. S. Fernandes, "Establishing a Risk-Based Separation Standard for Unmanned Aircraft Self Separation," in *Proceedings of the Ninth USA/Europe Air Traffic Management Research & Development Seminar*, no. June, Berlin, Germany, 2011. 12

[130] B. A. White and H. S. Antonios, "UAV Obstacle Avoidance using Differential Geometry Concepts," in *18th IFAC World Congress*, no. 3, Milano, Italy, 2011, pp. 6325–6330. 15

[131] Wikipedia: The Free Encyclopedia. Airspace. 176

[132] H. J. Williams, "Draft Enviromental Impact Study For The BRAC Beddown And Flight Operations Of Remotely Piloted Aircraft," Letter of the Aircraft Owners and Pilots Association to Mr. Allbright, 2010. 183

[133] M. Wilson, "Ground-based sense and avoid support for unmanned aircraft systems," in *28th Congress of the International Council of the Aeronautical Sciences*, 2012. viii, 31, 129

[134] Z. Z., X. R. Li, and V. P. Jilkov, "Best Linear Unbiased Filtering with Nonlinear Measurements for Target Tracking." *IEEE Transactions on Aerospace and Electronic Systems*, vol. 40, no. 4, pp. 1324–1336, 2004. 98

[135] Zarchan, Paul and H. Musoff, *Fundamentals of Kalman Filtering: A Practical Approach.* New York: AIAA, 2008. 99

[136] A. D. Zeitlin and M. P. McLaughlin, "Safety Of Cooperative Collision Avoidance For Unmanned Aircraft," *IEEE Aerospace and Electronic Systems Magazine*, vol. 22, no. 4, pp. 9–13, 2007. 17