COLLISION AVOIDANCE ALGORITHM DESIGN FOR UAV BASE ON PARAMETRIC THEOREM AND CIRCLE OVERLAPPING METHOD

NUR FADZILAH BINTI MOHAMAD RADZI

INSTITUTE OF GRADUATE STUDIES

UNIVERSITY OF MALAYA

KUALA LUMPUR

2013

COLLISION AVOIDANCE ALGORITHM DESIGN FOR UAV BASE ON PARAMETRIC THEOREM AND CIRCLE OVERLAPPING METHOD

NUR FADZILAH BINTI MOHAMAD RADZI

DISERTATION SUBMITTED IN FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF ENGINEERING SCIENCE

FACULTY OF ENGINEERING

UNIVERSITY OF MALAYA

KUALA LUMPUR

2013

ABSTRACT

This research presents a collision avoidance algorithm for resolving the conflict between two cooperative UAVs (Unmanned Aerial Vehicle) that fly in a straight trajectory with fixed altitude (2-horizontal dimension). However, it is not applicable to be used for obstacle conflicts. This proposed algorithm will modify the flight plan of hosting UAV (called as UAV in this thesis) to avoid the other UAV (called as target in this thesis) once the UAV's collision avoidance system detects the collision in near future.

Firstly, a mission waypoint path of UAV is given and assumes that all target information is collected from sensors (such as position sensors, angle heading sensor, and velocity sensors). Then, a collision detection algorithm is developed to calculate the potential of collision in future. The algorithm is based on PTCOT (Parametric Theorem and Circle Overlapping Test) between two UAVs. The collision potential will determine whether the avoidance mode needs to be activated or not and collision data (such as collision point, overlap time range, and collision time) will be used in conflict resolution.

Secondly, a collision avoidance algorithm is designed to propose a new flight path in order to perform an avoidance maneuver for hosting UAV. In this research, two collision avoidance commands are proposed. First command is change position command, where the UAV will turn left or right based on relative collision angle between the conflicting agents. The second command is change speed command, where the speed of UAV will increase or decrease depends on the conflict situation.

Finally, the collision avoidance algorithm is verified through MATLAB software. Various cases are tested to demonstrate the robustness of both collision detection and avoidance algorithms.

ABSTRAK

Kajian ini membentangkan sebuah algoritma pengelakkan perlanggaran untuk menyelesaikan konflik antara dua buah koperatif UAV (kenderaan udara tanpa pemandu) yang terbang dalam trajektori lurus dengan ketinggian tetap (2-dimensi mendatar). Walau bagaimanapun, ia tidak sesuai untuk digunakan bagi konflik sesama objek. Algoritma yang dicadangkan ini akan mengubah pelan penerbangan UAV utama (dipanggil sebagai UAV dalam tesis ini) untuk mengelak UAV lain (dipanggil sebagai sasaran dalam tesis ini) apabila sistem pengelakkan perlanggaran UAV mengesan perlanggaran dihadapan.

Pertama, laluan misi UAV diberikan dan menganggap bahawa semua maklumat sasaran dikumpul daripada pengesan (seperti pengesan kedudukan, pengesan sudut tajuk, dan pengesan halaju). Kemudian, algoritma pengesan perlanggaran diaktifkan untuk mengira potensi perlanggaran pada masa akan datang. Algoritma adalah berdasarkan PTCOT (teorem parametrik dan ujian pertindihan bulatan) antara dua UAV. Potensi perlanggaran akan menentukan sama ada mod pengelakkan perlu diaktifkan atau tidak dan data perlanggaran (seperti titik perlanggaran, bertindih julat masa, dan masa perlanggaran) akan digunakan dalam penyelesaian konflik.

Kedua, sebuah algoritma pengelakkan perlanggaran direka untuk mencadangkan laluan penerbangan baru untuk UAV utama. Dalam kajian ini, dua arahan pengelakkan perlanggaran dicadangkan. Arahan pertama adalah arahan perubahan kedudukan, di mana UAV akan bertukar arah ke kiri atau kanan berdasarkan sudut perlanggaran relatif antara agen yang bercanggah. Arahan yang kedua ialah perubahan arahan kelajuan, di mana kelajuan UAV akan meningkat atau menurun bergantung kepada situasi konflik.

Akhirnya, mengelakkan perlanggaran algoritma disahkan melalui perisian MATLAB. Pelbagai kes diuji untuk menunjukkan keteguhan kedua-dua pengesanan perlanggaran dan algoritma mengelakkan.

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to several people who made this research possible. Firstly, my supervisors, Dr. Marizan Mubin and Prof. Dr. Nasrudin bin Abd. Rahim for their encouragement and support throughout the course of this thesis effort. I would like to thank to Mr. Bara' and Dr. Simon for their assistance and simulation test support during research period. The software experiments could not be performed without their help. Secondly, I would also like to thank University of Malaya Research Grant (UMRG) and Postgraduate Research Grant (PPP) for providing me funding in order to make this project success.

Additionally, I would like to express my gratitude to all members who always helping and supporting me especially Siti Salbiah and Roziana Ramli. Finally, I would like to give my appreciation to my parents Mohamad Radzi b. Mursal and Fatimah bt. Abas for their endless support and fully understanding throughout my study.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AAP	Avoidance Action Period
ADS-B	Automatic Dependent Surveillance Broadcast
CA	Collision Avoidance
CAS	Collision Avoidance System
CD	Collision Detection
CDR	Collision Detection and Resolution
COT	Circle Overlapping Test
FAA	Federal Aviation Administration
IPC	Intermittent Positive Control
PSDZ	Protective Separation Distance Zone
PTCOT	Parametric Theorem and Circle Overlapping Test
RCS	Radar Cross Section
SBC	Safety Boundary Circle
SNR	Signal to Noise Ratio
TCAS	Traffic Alert and Collision Avoidance System
UAV	Unmanned Aerial Vehicle

CHAPTER 1

Introduction

In this chapter, a background study of Collision Avoidance System (CAS) for Unmanned Aerial Vehicle (UAV) is described in introduction section. This is followed by the motivation and the objectives of this research. Finally, outline of the thesis is presented in the last section.

1.1 Introduction

UAV in this research is referring to an unmanned aircraft that operate in air without human crew on-board. It is viable solutions for the future civil and military applications. It is because of the characteristics of UAV such as (B. M. Albaker, Rahim, & Mubin, 2009; Radzi, Mubin, & Rahim, 2011; Stephen, Kenneth, Peter, & Linton, 2005) :

1. It is an aircraft that functions autonomously without the pilot and is a self-directing entity.
2. Low cost solution where manned aircraft are currently used.
3. Able to carry out dangerous missions without involving human risks such as reconnaissance and surveillance.
4. UAV are better suited for dull, dirty and dangerous missions than man aircraft.

Since UAV carry out dangerous task, it must able to sense and avoid potential collisions (B. M. Albaker & Rahim, 2009b). Therefore, CAS system is very important and necessary for UAV to allow it operates safely.

CAS is a system of multiple sensors that is placed within the vehicle to take the action of avoiding dangerous collision situations that may lie ahead on the path (Radzi et al., 2011). The information of other UAV such as position, velocity, and heading angle can be measured by radar, lidar, and/or cameras in real time (Watanabe, August 2007; Yizhen, Antonsson, & Grote, 2006). CAS must keep track of the most current vehicle-to-vehicle relative position by process all the information in real time (Yizhen, et al., 2006). Besides that, it should be secured to the equivalent level of safety comparable with manned aircraft in order to fly in civil and military airspace. (Asmat et al., 2006).

The three main parts in CAS architecture system were discussed in many papers. There are sense, detect and avoid parts (Asmat, et al., 2006). Collision detection discusses a method to detect the potential of collision in future after all information is collected from multiple on-board sensors. If the collision potential is high, collision avoidance mode will be activated. A collision avoidance maneuver is proposed based on traffic conflict, essentially to avoid the collision between two UAVs.

Many collision avoidance algorithms were designed recently. Han proposed a collision avoidance algorithm using the conventional proportional navigation guidance law approach (Han, Bang, & Yoo, 2009). Meanwhile, Sigurd and How applied the total field approach to construct an avoidance algorithm (Sigurd & How, 2003). In Saunders' paper presents a method of obstacle avoidance using a series of circular oscillating paths and a single point laser ranger (Saunders, Beard, & McLain, 2007).

In this thesis, the new path or avoidance maneuver is proposed for UAV to follow as its guidance path in collision in near future. The work concentrates on the designing a collision detection algorithm and followed by collision avoidance algorithm. For collision detection, two different methods are investigated which are parametric theorem approach and circle overlapping method. Circle overlapping method is chosen for designing the collision avoidance algorithm. However, UAV motion equations and UAV modelling and control were not included in this thesis. The flight dynamic which controlled by the three dimensional axes which are pitch, roll and yaw is also not covered.

1.2 Motivation

UAV is being widely used especially in military and civilian applications, including mapping, surveillance and reconnaissance, and patrolling. Since it becomes so useful and safe to do dangerous task nowadays, it makes the number of UAV in air operation increase. Consequently, aircraft collisions also increase (B. M. Albaker & Rahim, 2011b; Han, et al., 2009).

Collision avoidance for UAVs has received intensive focus recently. More tasks are being carried out to protect every flying object in shared airspaces. A powerful CAS must be implemented in each autonomous UAV. Thus, self-contained on-board sensors, and effective collision avoidance algorithm are required. It is used to detect other UAV nearby, identify the potential of collision in future, and perform self-avoidance maneuver.

Many research works on collision avoidance were proposed for UAVs (B. M. Albaker & Rahim, 2011a; Han, et al., 2009; Kurnaz, 2009; Sigurd & How, 2003) . The collision avoidance algorithm should be generated in real-time and simple to implement. Therefore, the algorithm must proposed avoidance trajectory when the detected potential of collision in future is high.

1.3 Objectives and Scope of Work

The main objectives of this research presented in this thesis are stated as follows:

1. To develop collision detection algorithms for two cooperative UAVs based on parametric theorem and circle overlapping method.
2. To develop collision avoidance algorithms (position change and speed change commands) for two cooperative UAVs based on circle overlapping test method.

To achieve the above objectives, the scope of work has been listed as follows:

1. The development of collision avoidance system for UAV will be addressed. The advantages and disadvantages on different avoidance methods are described.
2. Robustness between two methods (PT and PTCOT methods) for collision detection has been investigated. PTCOT method is chosen for designing the collision detection algorithm.
3. The mathematical process on designing the collision avoidance algorithm is discussed. Two avoidance commands have been proposed by using the PTCOT method. The processes for both commands are explained in flow chart form.
4. Finally, two collision avoidance commands (position change and speed change) are tested on various collision scenarios or conditions. These results are compared with the results without collision avoidance algorithm.

1.4 Thesis Outline

In Chapter 2, literature review on previous studies about CAS is revised. It can be collected through previous thesis, journals, books and encyclopaedia. This is followed by a case study on CAS architecture for UAV. Then, several methods that have been proposed for collision detection and collision avoidance are investigated.

In Chapter 3, a collision detection algorithm is designed. The algorithm is based on parametric theorem and circle overlapping method. The objective of this algorithm is to calculate or examine the potential of collision in future. Performances of both methods are described.

In Chapter 4, a collision avoidance algorithm design is presented. Using correlation trajectory based on circle overlapping test, two avoidance commands are proposed. One is change position command and the other is change speed command.

In Chapter 5, verification of collision avoidance algorithm using MATLAB is discussed. Different simulation cases are presented and tested to see the robustness of the algorithm. Finally, all the simulation results are discussed and summary of the entire work has been concluded in Chapter 6.

CHAPTER 2

Literature Review

In this chapter, a survey of the literature review related to CAS that applies on UAV system is presented. Firstly, the definition of the protection zone around each UAV is explained. Then, the overview of development and operational deployment of CAS system that implemented in a UAV is discussed. A detail of CAS functional and physical architecture is described. Finally, the focus on collision detection and collision avoidance method from previous research is reviewed.

2.1 Introduction of UAV

UAV operation is remotely controlled semi-autonomous or autonomous aircraft which flies without a human crew on board the aircraft (Gertler, January 3,2012). UAVs are envisioned as an integral part of future civilian and military operations, including intelligence, surveillance and reconnaissance missions, search and rescue missions and surveillance for homeland security (Byrne, Cosgrove, & Mehra, 2006). These missions often require a UAV to fly nap of the earth, risking collision with low level obstacles such as trees and buildings whose position cannot be guaranteed as known before flight. Generally, the UAVs usually being use in situations when it is too dangerous to use manned aircraft (Kwag, Choi, Jung, & Hwang, 2006). Many of these applications will require the UAV to operate in the same condition as manned aircraft. One critical problem that has to be dealt with is how to avoid a collision between the UAV and another aircraft.

Humans are an essential element in this process due to their ability to integrate information and make judgements. However, because failures and operational errors can occur, automated systems have begun to appear both in the cockpit and on the ground to provide decision support and to serve as traffic conflict alerting or resolution systems. These systems use sensor data to predict conflicts between aircrafts and may provide commands or guidance to resolve the conflict.

The smart UAV requires the collision avoidance capability to automatically sense and avoid the stationary obstacle and/or non-stationary moving obstacles along the flight path in the relatively low-attitude flying and rapid maneuvering environment (Kwag, et al., 2006; Kwag & Chung, 2007). The data sensor is used to predict conflicts between aircraft by calculate the level of potential collision in future (Kuchar & Yang, 2000). If the potential collision is detected high, UAV must have the ability to avoid the collision in future by solving the conflict problem (Carbone, Ciniglio, Corraro, & Luongo, 2006). Practical UAVs must include situational awareness based on sensing and perception of the immediate environment to locate collision dangers and plan an appropriate avoidance path (Byrne, et al., 2006; Sinopoli, Micheli, Donato, & Koo, 2001).

2.2 CAS System

2.2.1 CAS Overview

CAS is a system consists of sensors, detection, extraction, decision making, reaction, interception to detect and avoid autonomously from any dangers that may lie ahead on the path of vehicle (Radzi, Mubin, Rahim, et al., 2011). Some of the dangers that these sensors can pick up on include how close the vehicle is to other intruder surrounding it and what action that vehicle needs to take to avoid those dangers so that it can prevent that vehicle from potentially getting into a serious accident.

CAS is an example of a system that is designed to prevent accidents to minimize the force and damage of collision in case of accidents. The goal of the CAS is to help avoiding vehicles from collide with each other. The implementation of CAS in automotive is expected to improve traffic safety significantly. These electronic systems scan and monitor the dynamic environment of the vehicle for potential dangers. When the potential danger of crashing is determined and, if so, a collision avoidance action is undertaken by controlling the speed or direction of the vehicle or warning drivers with visual or sound alert.

Thus, the CAS should be considered as a part of the guidance system which generates the necessary maneuver commands to guide the vehicle towards its goal while avoiding other aircraft on its path. The goal of the CAS is to help avoiding vehicle crashes (Ajith Kumar & Ghose, 2001). There are three subject areas of collision occur that shown in Figure 2.1:

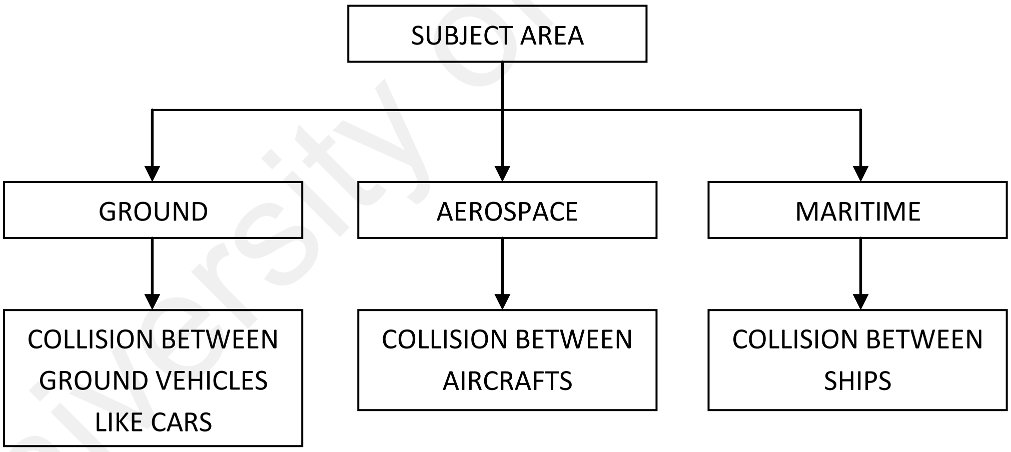

SUBJECT AREA
GROUND
AEROSPACE
MARITIME
COLLISION BETWEEN GROUND VEHICLES LIKE CARS
COLLISION BETWEEN AIRCRAFTS
COLLISION BETWEEN SHIPS

Figure 2.1: Subject areas of collision

A CAS should have the following functions for manual and full autonomous avoidance cases (Dear & Sherif, 1991):

i. *Manual avoidance case:*

1. To detect and evaluate potential collision hazards to the aircraft involved.
2. To indicate the threat to the pilots involved with sufficient advanced warning.

3. To provide the pilot with appropriate command signals indicating a satisfactory avoidance maneuver.

ii. *Full autonomous avoidance case:*

1. To detect and evaluate potential collision hazards to the aircraft involved.
2. Able to estimate the future collision data.
3. Able to avoid the collision automatically.

Many CAS technologies had been proposed and investigated by the Federal Aviation Administration (FAA). For example of these systems include the following (Dear & Sherif, 1991; Hawkes, 1998; Kuchar & Drumm, 2007):

1. Traffic Alert and Collision Avoidance System (TCAS I , TCAS II, TCAS III & TCAS IV)
2. Airborne Collision Avoidance System
3. Ground-based Collision Avoidance – Intermittent Positive Control (IPC) system.
4. Automatic Dependent Surveillance Broadcast (ADS-B)

2.2.2 CAS Architecture

The complete CAS system for UAV is shown in Figure 2.2. In collision avoidance system, there are three critical functions of operation required to decide the status of the collision risk: sense (to sense any intruders that enter possible collision ranges), detection (to detect any possibilities of collision to be occurred by collecting all the information needed such as velocity, bearing, etc., and evaluate the threat of a near mid-air collision) and avoidance (the system executes an appropriate avoidance maneuver by turning the direction of flight path or changing the velocity of own UAV in order to avoid the dangerous intruders) (Kwag, et al., 2006; Kwag & Chung, 2007).

The outputs for sense and detect parts are interchangeable as they are depending on real-time cooperative UAV location. Avoidance function has to play an important role to make a right decision to ensure that the mission given becomes a success without any crash. The autonomous UAV should perform the collision avoidance maneuver on its flight path. Meanwhile, communication function refers to the ability to notify external systems of appropriate events regarding the flight of UAV equipped with CAS and to communicate with other cooperative UAV, in the vicinity.

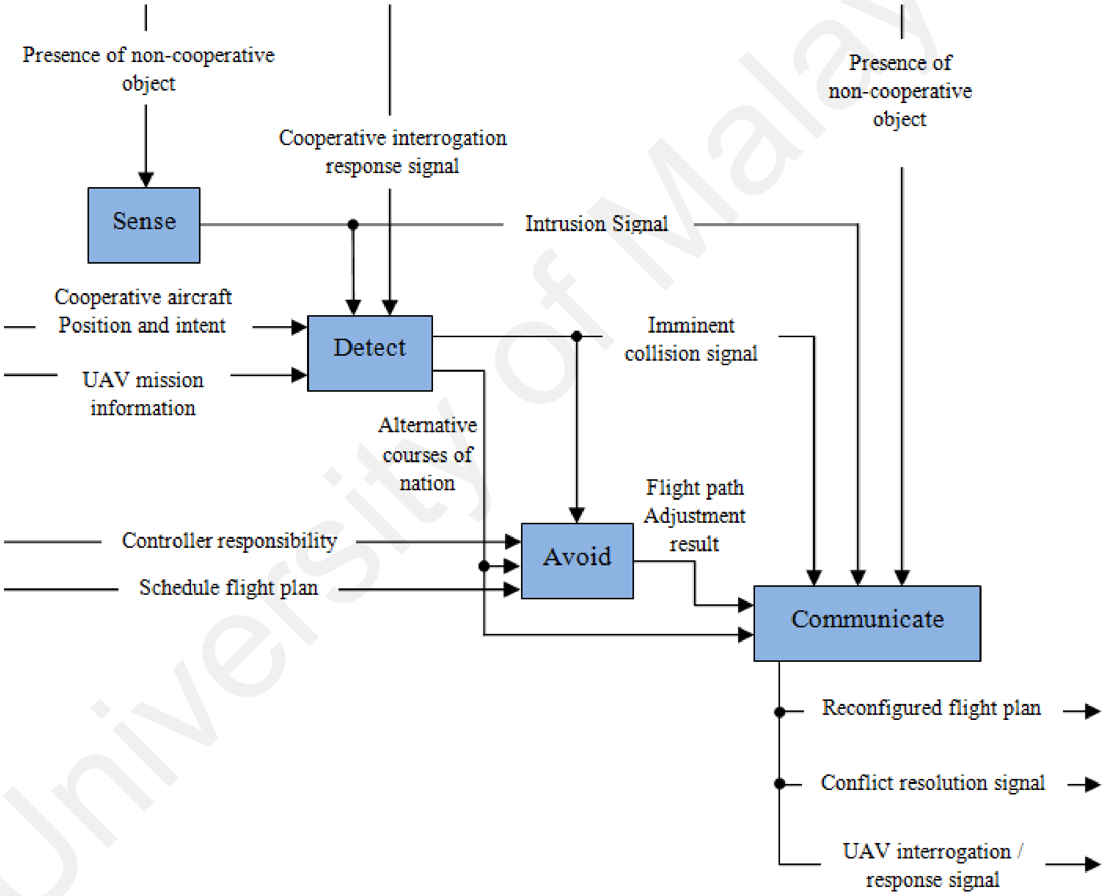

Presence of non-cooperative object
Cooperative interrogation response signal
Presence of non-cooperative object
Sense
Intrusion Signal
Cooperative aircraft Position and intent
UAV mission information
Detect
Imminent collision signal
Alternative courses of nation
Flight path Adjustment result
Controller responsibility
Schedule flight plan
Avoid
Communicate
Reconfigured flight plan
Conflict resolution signal
UAV interrogation / response signal

Figure 2.2: Functional Architecture Diagram for UAV (B. M. Albaker & Rahim, 2009a)

The overview of generic physical architecture of CAS for UAV is proposed in Figure 2.3. It defines the hierarchy of physical components that cover the system. Since the UAV is an aircraft that flies without a human crew on board the aircraft, it needs a robust CAS that can sense, detect and avoid the obstacles for hosting UAV.

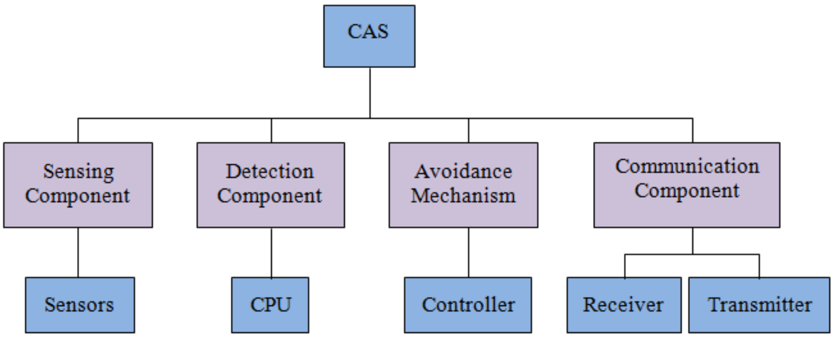


Figure 2.3: CAS Generic Physical Architecture for UAV (Asmat, et al., 2006)

2.3 Collision Detection (CD)

Recently, interest has grown toward developing more advanced automation tools to detect traffic conflicts and assist in their resolution. Relatively simple conflict predictors have been a part of air traffic control automation for several years, and the Traffic Alert and Collision Avoidance System (TCAS) has been in place of the on-board domestic transport aircraft since the early 1990s (Britt, 1994; Kuchar & Yang, 2000; Livadas, Lygeros, & Lynch, 2000; Yuan & Song, 2009).

Generally, each cooperative aircrafts should have three fundamental zones around them in order to detect the conflict arises and in that way to resolve it (B. M. Albaker, et al., 2009). Figure 2.4 represent UAV zones which are communication zone, detection zone and protective zone.

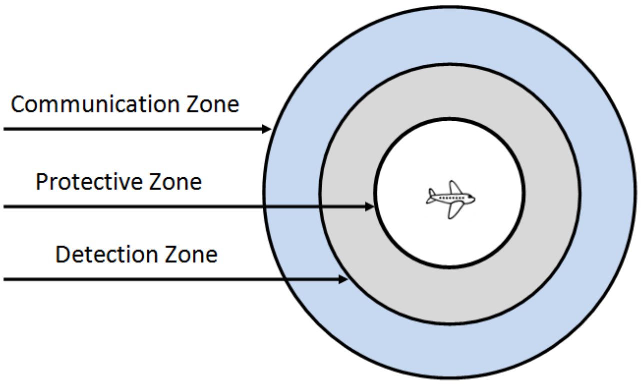


Figure 2.4: UAV Zones (B. M. Albaker & Rahim, 2009a)

Conflict can be defined as a violation of separation assurance standard or in other words two or more aircrafts experience a loss of minimum separation (Kuchar & Yang, 2000; Park, Oh, & Tahk, 2009; Wallace, Collins, Rapids, & Iowa, 1999). Collision detection algorithm must be able to detect whether the protected zone of UAV is violated or not. Generally, a collision occurs when the aircraft separation distance between two or more aircrafts equals to minimum separation required in all dimensions as shown in Figure 2.5 and Figure 2.6. Whereas when the protection zones of two aircrafts are overlap, it is called as conflict. If it does, then the collision avoidance algorithm of UAV should solve the violation using proper way to avoid the conflict. The methods of maintaining separation between aircraft in the current airspace system have been built from a foundation of structured routes and evolved procedures (Kuchar & Yang, 2000).

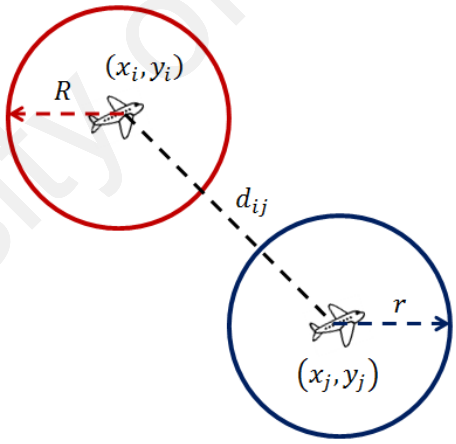


Figure 2.5: Collision Detection (Dear & Sherif, 1991)

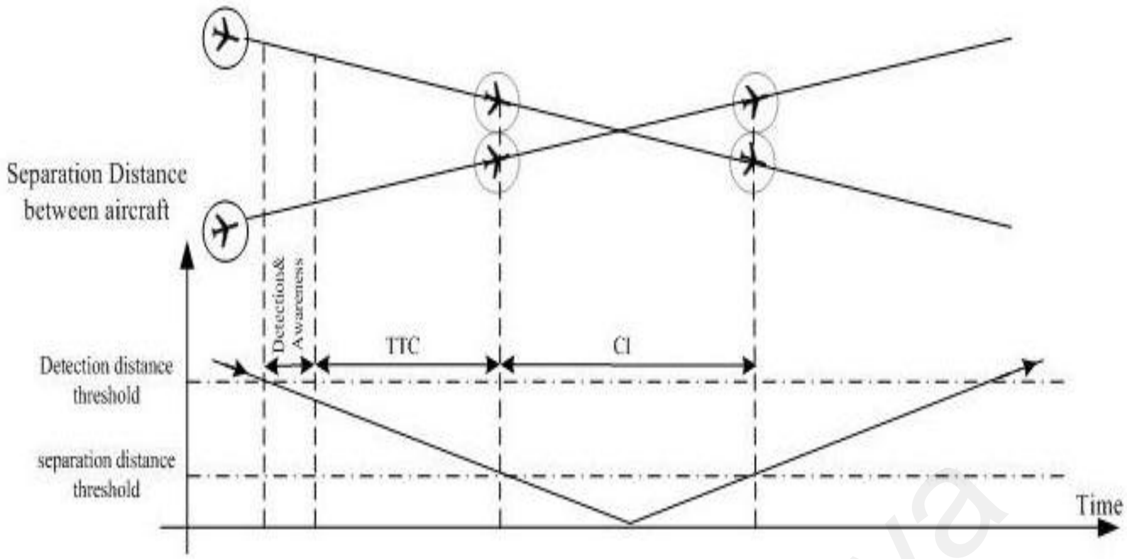

Separation Distance between aircraft
Detection& Awareness
TTC
CI
Detection distance threshold
separation distance threshold
Time

Figure 2.6: Demonstrates the collision parameters and separation distance between computing aircraft and encounter in a future course collision scenario (B. M. Albaker & Rahim, 2009a)

In total, over 60 different methods have been proposed by various researchers to address collision detection and resolution (CDR). These methods have been developed not only for aerospace, but also for ground vehicle, robotics, and maritime applications because the fundamental conflict avoidance issues are similar across transportation modes. For example, Lindsten propose angle-only based collision risk assessment method to continuously evaluate the risk for collision (Lindsten, 2008).

The design of the CAS system is based on eight collision detection and resolution factors (Kuchar & Yang, 2000). These factors are:

i. State propagation:

It explains how current states are projected into future. Four fundamental extrapolation methods have been identified: (a) nominal, (b) worst case, (c) probabilistic, and (d) flight plan sharing. The four methods are shown schematically in Figure 2.7.

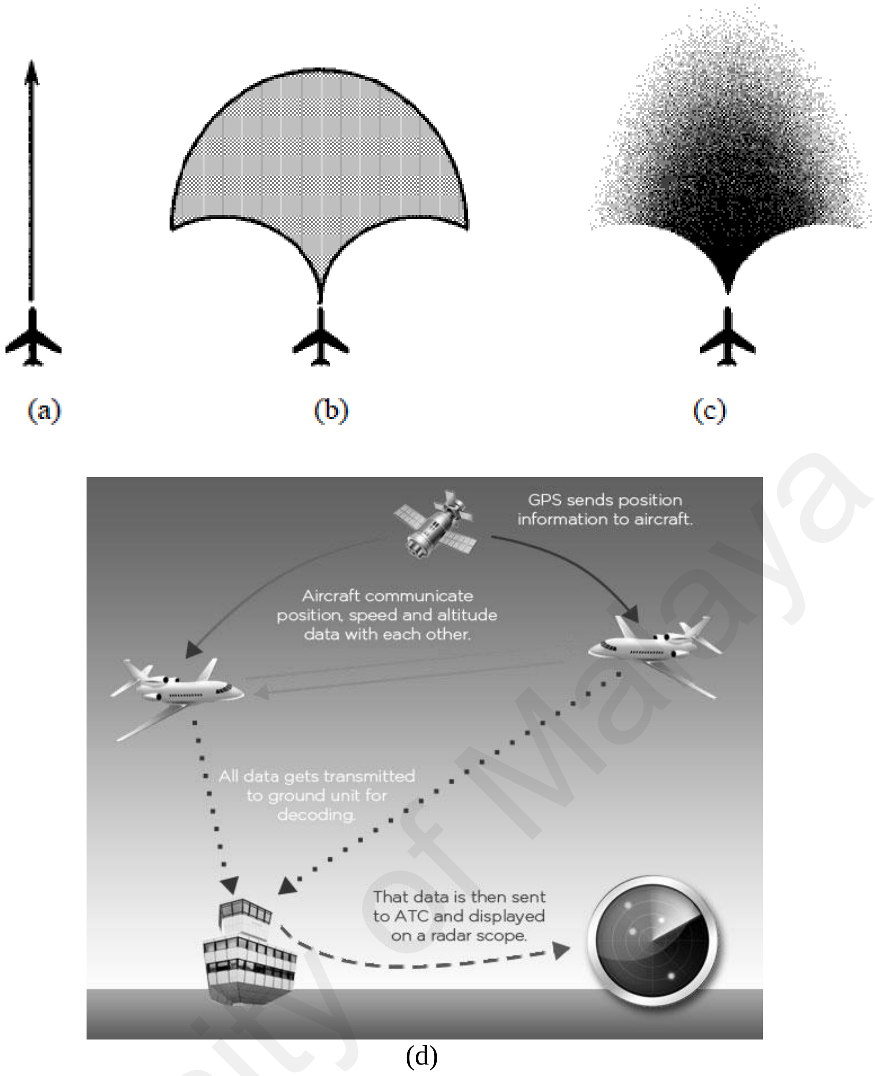

(a)
(b)
(c)
GPS sends position information to aircraft.
Aircraft communicate position, speed and altitude data with each other.
All data gets transmitted to ground unit for decoding.
That data is then sent to ATC and displayed on a radar scope.
(d)

Figure 2.7: State propagation methods (Kuchar & Yang, 2000; O'Brien, 2009)

ii. State dimension:

It represents operating dimension considered by the system. Types: Horizontal plane, vertical plane, and both (horizontal and vertical plane).

iii. Conflict detection:

It represents how system detects threats. The variables required in this process include the estimation of:

1) Threat position: range, azimuth and elevation.

2) Probability of false targets: probability of detection/ missed-detection.

3) Detection of the robustness to the environments such as wind, rain, etc.

iv. Conflict awareness:

It predicts whether a potential conflict will occur in the near future or not.

v. Conflict resolution:

It explains the solution to a conflict. Categories: Prescribed, optimal, force field.

vi. Resolution maneuver:

It indicates dimension of resolution. Divisions: Turn, vertical maneuver, and speed changes.

vii. Multiple conflicts:

In a realistic traffic environment, it will be necessary that a CDR system should be able to manage encounters involving more than two aircraft. In a pair wise approach, if one conflict solution induces a new conflict, the original solution may be modified until a conflict-free solution is found. A global solution that considers more than one other aircraft at a time requires more complex and may be more robust. For example consider the situation shown in Figure 2.8.

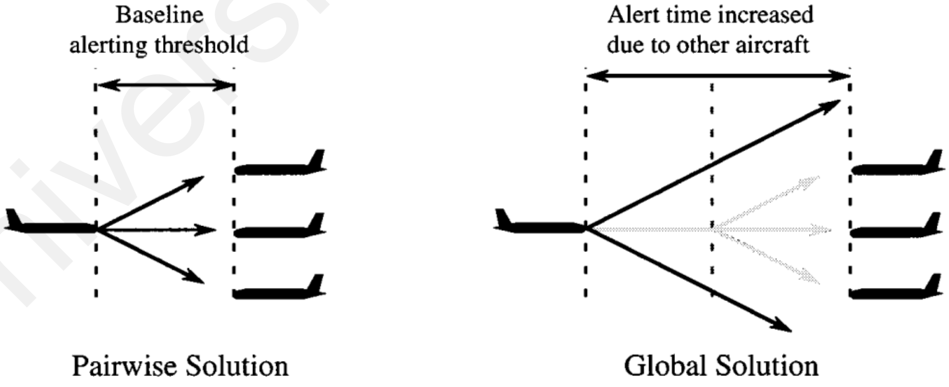


Figure 2.8: Multiple aircraft conflict detection and resolution (Kuchar & Yang, 2000)

viii. Coordinated maneuvers:

Coordinating conflict resolution between aircraft has two primary benefits. First, the required magnitude of maneuvering for a given aircraft may be reduced when two aircrafts maneuver cooperatively when compared against a case in which only one aircraft maneuvers. Second, coordination helps to ensure that aircraft do not maneuver in a direction that could prolong or intensify the conflict. However, coordination may increase controller or pilot workload due to the need to monitor several changes in the air traffic situation at one time.

2.3.1 Collision Detection Methods

Park et al. (Park, et al., 2009) proposed a method of conflict detection and resolution using geometric approach. Using geometric equation for point of closest approach, the algorithm conflict detection and resolution is induced by intuition simply. This paper deals with simple dynamics for the UAVs which considered as point masses with constant velocity. However, both en route UAVs must be linked by real time bases like ADS-B so that all UAVs share the information each other.

Cooperative, peer to peer negotiation technique is one of the collision detection methods for UAV which is proposed by (B. M. Albaker, & Rahim, N. A., 2009). In order to detect the conflict, three collision angle sectors in the detection zone are proposed to define the area in which the aircraft detect other aircraft. The advantage of using the straight projection is because of its simplicity and time requirement in calculation. Even so, the angle sectors only consider 180 degree of front view of aircraft.

Based on various sensors, radar sensor is selected for the collision detection due to the real-time range and range-rate acquisition capability of the stationary and moving aircraft even under all-weather environment. The conceptual radar sensor with design parameters to detect the probability of collision detection and avoidance is presented by Kwag and Chung (Kwag & Chung, 2007). Radar detection probability varies depending on Signal to Noise Ratio (SNR) and/or false alarm rate. SNR varies depending on Radar Cross Section (RCS) which may be fluctuated in amplitude and/or in phase. Since the atmospheric attenuation is the main reason for decreasing SNR in milimeter wave bands, the attenuation according to range should be considered in radar equation.

2.4 Collision Avoidance (CA)

Aircraft collision is a serious concern as the number of aircraft in operation increases. Ground air traffic control load has been also increasing due to heavy workload (Tomlin, Pappas, & Sastry, 1998). In particular, UAVs may be additional flight objects for consideration of collision avoidance in the future. Autonomous UAVs will require sophisticated avoidance systems with conventional aircraft flying together. So far, most of air traffic control is performed by ground station command (Williams, 2004).

However, with rapid increase of air traffic, the ground station-based air traffic control may not be sufficient to cover every flying object. Thus self-contained on-board air traffic control or collision avoidance systems have been considered. Autonomy of collision avoidance system requires avoidance laws among multiple aircraft in operation. The UAV or aircraft should carry sensors which can detect other aircraft nearby. Information of other aircraft can be used to design the avoidance command. The developed collision avoidance algorithm can help avoid the collision from any static obstacles and any dynamic objects such as an intruder UAV (Bicchi & Pallottino, 2000; Han, et al., 2009; Pallottino, Feron, & Bicchi, 2002).

A smart development of collision avoidance algorithm allows each aircraft to negotiate with each other in a peer to peer manner to determine a safe and acceptable resolution when a potential conflict is detected. Thus conflict resolution will be obtained quickly allowing it to be implemented in time of the critical situations. The solution taken for avoidance needs to be clarified before it can predefine maneuver. The predefined maneuver will depend directly onto the identification of relative collision angle between colliding aircraft. The demonstration of maneuver in different collision scenarios are shown in Figure 2.9.

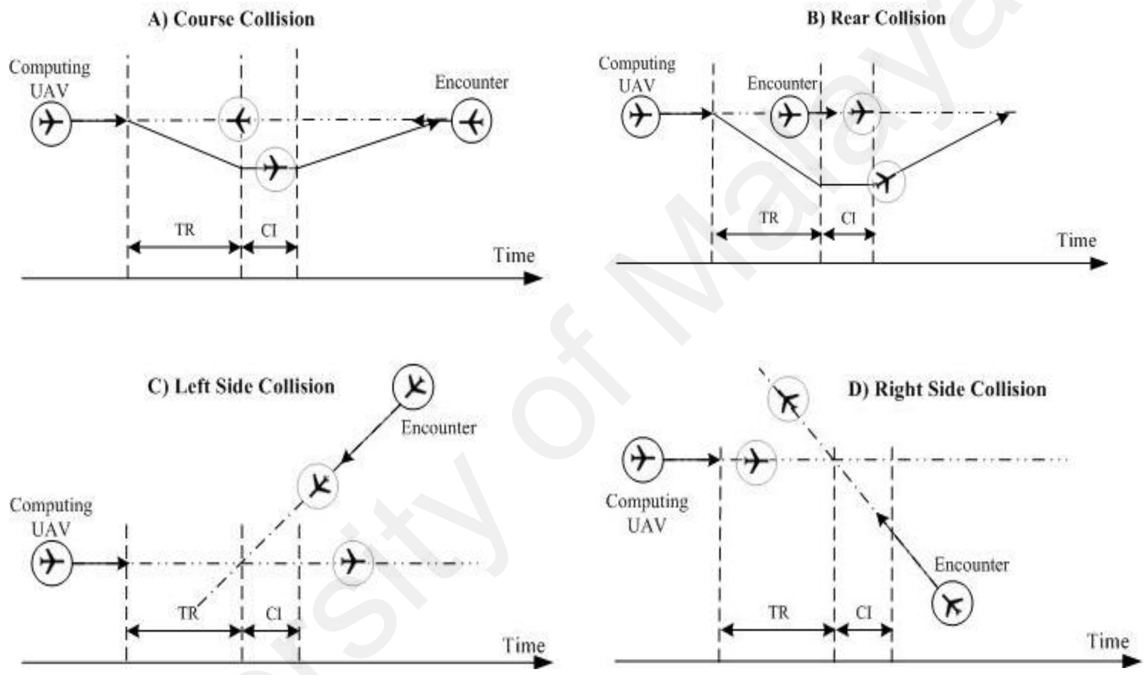


Figure 2.9: Demonstrates maneuver realization processes in different collision scenarios (B. M. Albaker & Rahim, 2010)

2.4.1 Collision Avoidance Methods

Sinopoli et al. (Sinopoli, et al., 2001) develop a system for autonomous navigation of UAVs based on computer vision. A global offline computation and a local online computation are distinguished based on a coarse known model of the environment and the information coming from the vision system. Their work address the problem of vision based autonomous navigation in partially known environment. Multi resolution via wavelet transform allows to localize the level of accuracy required to minimize collision probability, making the approach scalable with respect to the size of the map.

Protocol-based is one of decentralized collision avoidance methods. It offers a very elegant solution to conflict-free navigation with an inter-agent communication. (Hill, 2005; D. Sislak, Pechoucek, M., Volf, P., Pavlicek, D., Samek, J., Marik, V., & Losiewicz, P., 2007; D. Sislak, Volf, P., Komenda, A., Samek, J., & Pechoucek, M., 2007; Wangermann, 1999) address the problem of distributed collision avoidance among autonomous unmanned aircraft using multi-agent technology. They approaches were based on computing utility functions for each maneuver and aircraft involved in conflict resolution process, then finding the best solution for the avoidance. This will result in an efficient and more optimal solution but on the expenses of time.

Therefore, Albaker et al. (B. M. Albaker & Rahim, 2010) proposed another method using cooperative agent-based negotiation approach to be implemented in time critical situations. The proposed approach relies on predefined maneuvers and simple negotiation protocol to solve the conflict. It considers the problem of a finite number of autonomous aircraft sharing the same airspace. This will results in more efficient conflict resolution approach with less flight path maneuvering cost. However, it uses simple negotiation peer to peer protocol to solve the conflicts between two aircrafts. This peer to peer approach considers multiple collisions by executing it iteratively.

Saunders et al. (Saunders, et al., 2007) developed a method of obstacle avoidance for fixed-wing miniature air vehicles using a series of circular oscillating paths and a single point laser ranger. The oscillating paths allow the laser ranger to scan for obstacle and possible escape paths in case of obstacle detected. The circular paths are generated along waypoints and transition between waypoint paths without loss of scanning capabilities. Nevertheless, the use of laser ranger is of limited capabilities to detect the environment. Moreover, this type of sensors is considered costly and suitable for short-detection only.

The Proportional-Navigation (PN) guidance law is one of the most strategies for missile engagement scenario. However, it can be applied to collision avoidance problems by simple modification of PN guidance law as introduced by Han et al. (Han, et al., 2009). A collision avoidance vector is first defined, and then the vector defining heading angle of the aircraft is guided to the pre-defined collision avoidance vector. It also can be applied to 3-dimensional maneuver problems. Unfortunately, it is not suitable to be used for multiple unmanned aircraft problem.

Hachour (Hachour, 2009) introduced a linear parametric smoothed curves of navigation for path planning which connections are made to determine the form of the path. The proposed algorithm generates a collision-free trajectory between the starting configuration and the goal configuration in a static or dynamic environment. Therefore, the capability to build a map of the environment and updating the map must be fast and in real-time process.

In this thesis, parametric theorem and collision overlapping test (PTCOT) method is introduced. This method proposed a new collision avoidance algorithm which a new flight path is produced once the collision detected high. The advantages of this propose method is that it can be used for the CD and CA. Besides, this method can generates two types of avoidance commands for CA. These are position change command (controlling the changes on heading angle) and speed change command (controlling the speed by accelerating or decelerating). However, this method can be used for two cooperative UAVs conflict which are flying in a straight path.

CHAPTER 3

Collision Detection Algorithm

Collision detection is a process of detecting the potential of collision in near future. Since this thesis did not focus on sensing part, all the information such as position, velocity, and heading angle of UAV and target are assumed. In this chapter, collision detection algorithm is proposed based on parametric theorem, (PT) method and then it is improved by using the parametric theorem and circle overlapping test, (PTCOT) method. The robustness of algorithm is examined through the future analysis from simulation results.

3.1 PT Method

The main idea for collision detection algorithm is to detect the conflict by defining the intercept point. The conflict situation between two cooperative aircrafts (which one of aircraft is host UAV and another is target) which is flying in a straight flight path is illustrated in Figure 3.1.

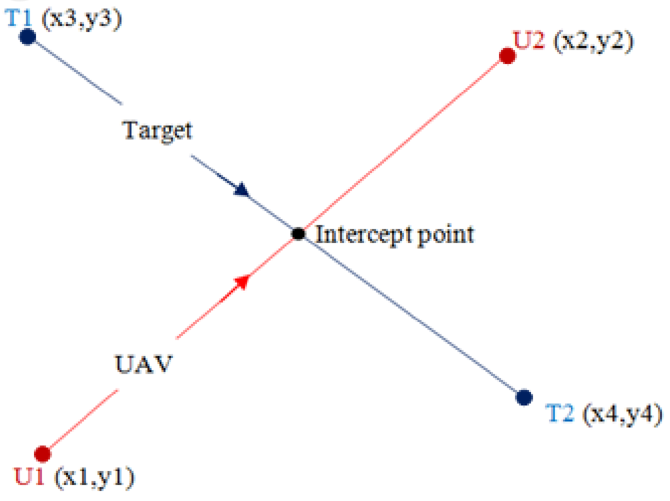


Figure 3.1: Conflict of two straight flight path lines

3.1.1 Equation of a Line

In this section, equation of a line is presented. It is used to detect the collision in the future and interception point as shown below.

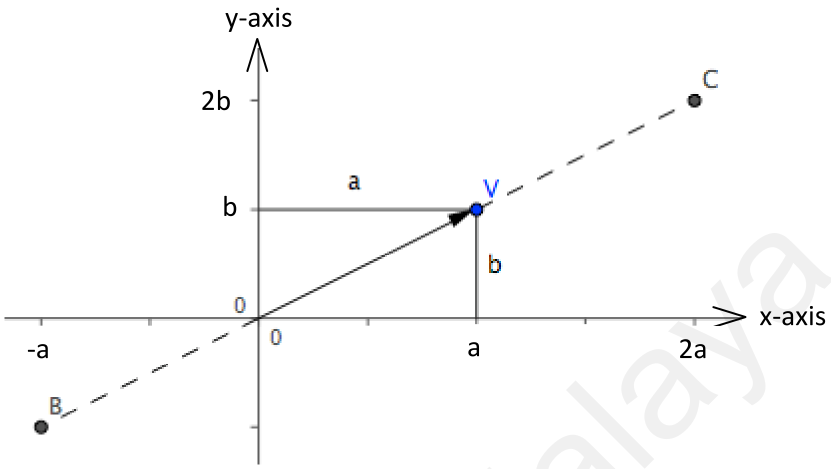


Figure 3.2: When $\mathrm{k} = 2$, point $\vec{\mathbf{V}}$ becomes point $\vec{\boldsymbol{C}}$ and when $\mathrm{k} = -1$, point $\vec{\mathbf{V}}$ become point $\vec{\mathbf{B}}$

Consider the point $\vec{\boldsymbol{V}} = \{a, b\}$ as shown in Figure 3.2 is multiply by k, it gives a scaling point. The relationship between value of k and point $\{a, b\}$ are described as below (Lee, 2006):

1. If $k > 1$, then point $\vec{\mathbf{V}}$ gets farther from origin.
2. If $k = 1$, then point $\vec{\mathbf{V}}$ remains the same.
3. If $0 < k < 1$, then point $\vec{\mathbf{V}}$ is shrinking towards the $\{0,0\}$.
4. If $k < 0$, then $k\{\mathrm{a}, \mathrm{b}\}$ is reflected thru the origin.

Let k vary, $k\{a, b\}$ is a line passing the origin which it is a straight line. If x denote the x-coordinate component, and y denote the y-coordinate, then the parametric formula for a line is:

$$\{x, y\} = k\{a, b\}, \tag{3.1}$$

where; $x = a \times k$ and $y = b \times k$.

If the line passing through a point $\{c, d\}$, the parametric formula is prescribed as

$$\{x, y\} = k\{a, b\} + \{c, d\}, \tag{3.2}$$

The algebraic equation can be derived from the parametric formula of a line passing two points $\{a, b\}$ and $\{c, d\}$. Given that the vector from $\{a, b\}$ to $\{c, d\}$ is $\{c, d\} - \{a, b\}$. So, a line with that slope is $k(\{c, d\} - \{a, b\})$. Finally, the algebraic equation of a line passing two points $\{a, b\}$ and $\{c, d\}$ passing through a point $\{c, d\}$ is

$$\{x, y\} = \{c, d\} + k(\{c, d\} - \{a, b\}), \tag{3.3}$$

where; $x = c + k(c - a)$ and $y = d + k(d - b)$.

3.1.2 Interception Point between UAV and Target

From (3.3), the equation of a line for UAV and target as shown in Figure 3.1 are expressed as (Bourke, April 1989) :

$$\boldsymbol{P}_U = \boldsymbol{U}1 + u_a(\boldsymbol{U}2 - \boldsymbol{U}1), \tag{3.4}$$

$$\boldsymbol{P}_T = \boldsymbol{T}1 + u_b(\boldsymbol{T}2 - \boldsymbol{T}1), \tag{3.5}$$

where; $\boldsymbol{P}_U = \{x_U, y_U\}$ that is a point located on UAV line, $\boldsymbol{P}_T = \{x_T, y_T\}$ that is a point located on target line, u_a and u_b are the scale of point $\boldsymbol{P}_U$ and $\boldsymbol{P}_T$.

To find the intercept point in Figure 3.1, the value of point $\boldsymbol{P}_U$ and $\boldsymbol{P}_T$ must be same. Since $\boldsymbol{P}_U = \boldsymbol{P}_T$, the following two equations below can be obtained.

$$x1 + u_a(x2 - x1) = xt1 + u_b(x4 - x3), \tag{3.6}$$

$$y1 + u_a(y2 - y1) = yt1 + u_b(y4 - y3), \tag{3.7}$$

The expression of u_a and u_b can be determined by solving (3.6) and (3.7) (Bourke, April 1989):

$$u_a = \frac{(x4-x3)(y1-y3)-(y4-y3)(x1-x3)}{(y4-y3)(x2-x1)-(x4-x3)(y2-y1)}, \tag{3.8}$$

$$u_b = \frac{(x2-x1)(y1-y3)-(y2-y1)(x1-x3)}{(y4-y3)(x2-x1)-(x4-x3)(y2-y1)}, \tag{3.9}$$

To determine whether the existence of intercept point between UAV’s path and target’s path, the value of u_a and u_b must lie between 0 and 1. Therefore, intercept point within both flight path line segments only exist if and only if

$$(0 < u_a < 1) \cap (0 < u_b < 1), \tag{3.10}$$

However, CD based on parametric theorem by investigating the intercept point only is not very robust. The collision in near future cannot be detected if only the intercept point is existed. It is because both aircrafts may not be lie on the intercept point at the same time because of the velocity difference. Besides, collision is defined as an incident in which two protective zones of aircrafts experience a loss of minimum separation where overlapping between two circles zone is happened as explain in Figure 3.4c. Therefore, some modification on CD is investigated by using PTCOT method.

3.2 PTCOT Method

3.2.1 Equation of Motion

Equation of motion for a straight path scenario based on parametric theorem describes the behavior of two moving UAVs as shown in Figure 3.3.

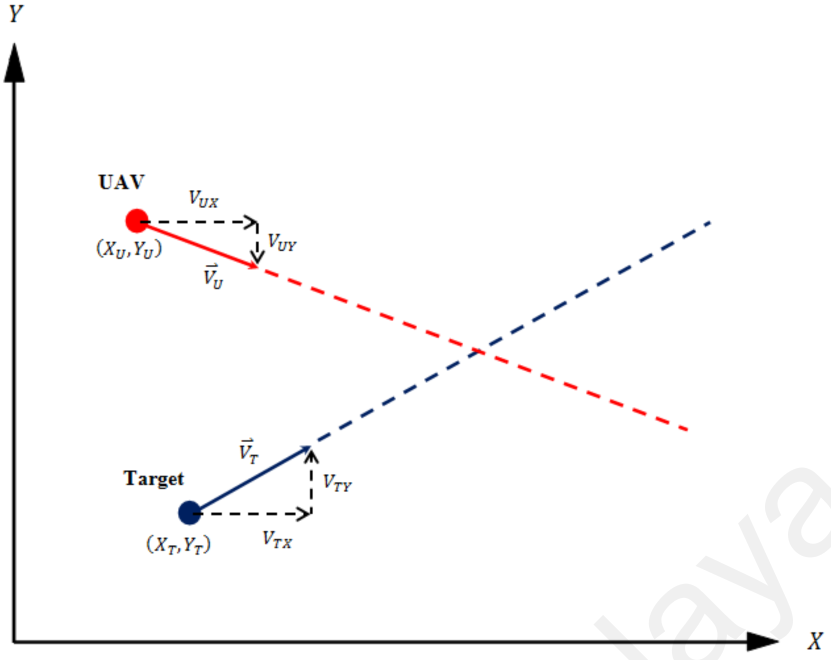


Figure 3.3: Relative motion of two moving aircrafts

These equations can be utilized for motion that can be described as being either a constant velocity motion or a constant acceleration motion. It can never be applied over any time period during which the acceleration is changing. Consider the kinematic equations for constant velocity and constant acceleration are shown in (3.11) and (3.12), respectively, which describes an aircraft's motion for straight path.

$$\vec{P}_{t_f} = \vec{P}_{t_i} + \vec{V}\Delta t \qquad (3.11)$$

$$\vec{P}_{t_f} = \vec{P}_{t_i} + \vec{V}\Delta t + \tfrac{1}{2}\vec{a}\Delta t^2 \qquad (3.12)$$

Let $\vec{P}_{t_f}$ be the position vector of the aircraft at time t, $\vec{P}_{t_i}$ be the position vector at the beginning of the frame, $\vec{V}$ and $\vec{a}$ is the velocity and acceleration vectors of the aircraft, respectively. Thus, time period of frame is $\Delta t = t_f - t_i$.

When acceleration in (3.12) is zero, the equation reverts to a rearranged constant velocity equation as (3.11). Thus, UAV's motion and target's motion for constant acceleration are given in (3.13) and (3.14), respectively.

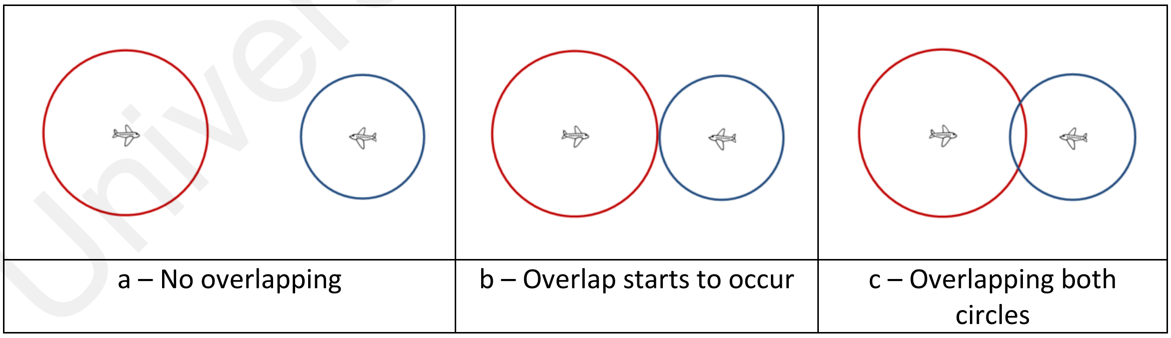
$$\vec{U}_{t_f} = \vec{U}_{t_i} + \vec{V}_U \Delta t + \frac{1}{2}\vec{a}_U \Delta t^2 \tag{3.13}$$

$$\vec{T}_{t_f} = \vec{T}_{t_i} + \vec{V}_T \Delta t + \frac{1}{2}\vec{a}_T \Delta t^2 \tag{3.14}$$

Let $\vec{U}_{t_f}$ and $\vec{T}_{t_f}$ be the new positions vectors of UAV and target at the time t_f. Meanwhile, let time t_i at the beginning of the frame of information data received from sense part in CAS system. Also let $\vec{V}_U$ and $\vec{a}_U$ be the initial velocity and acceleration vectors of the UAV, and let $\vec{V}_T$ and $\vec{a}_T$ be the initial velocity and acceleration vectors of the target.

3.2.2 Safety Boundary Circle (SBC)

Generally, each aircraft has their own SBC. The radius of each SBC is depending on the size of each aircraft. The simplest method to detect a collision is to check if the circles are overlapping as shown in Figure 3.4.



a – No overlapping

b – Overlap starts to occur

c – Overlapping both circles

Figure 3.4: Overlapping situation of two different sizes of aircrafts

3.2.3 Collision Detection Algorithm Design

3.2.3.1 Separation Distance Analysis

The idea of collision detection is to check the separation distance between UAV and target at each time of frame, $t_i < t < t_f$ as shown in Figure 3.5. Collision detection is detected when UAV's circle and target's circle start to overlap. Geometric configuration for collision detection between two aircrafts is shown in Figure 3.6.

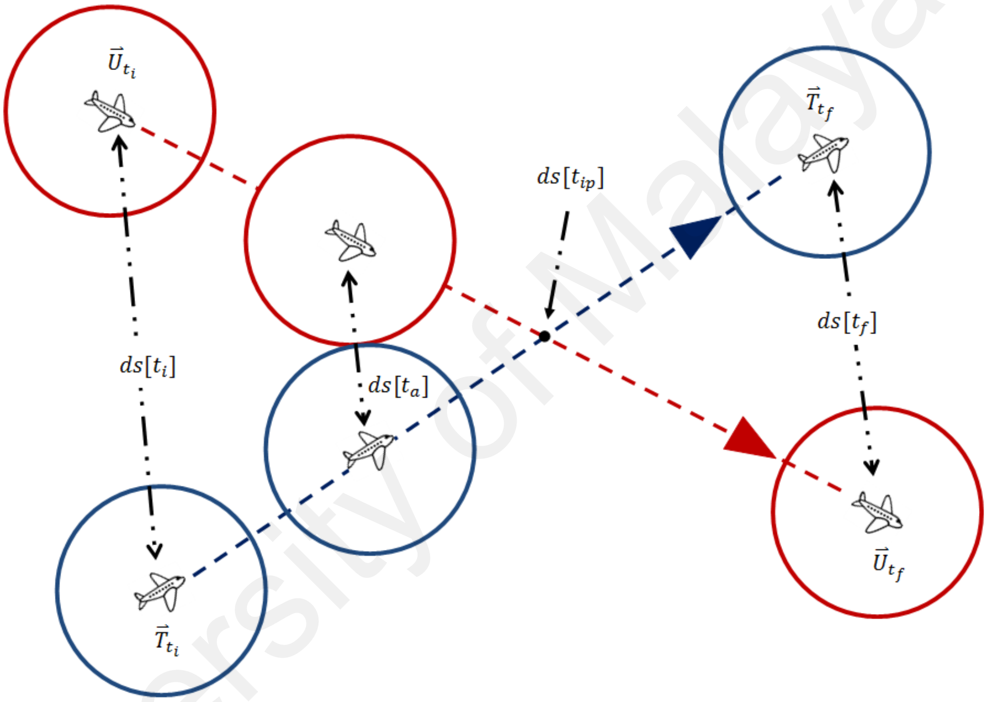


Figure 3.5: Separation distance between moving UAV and target

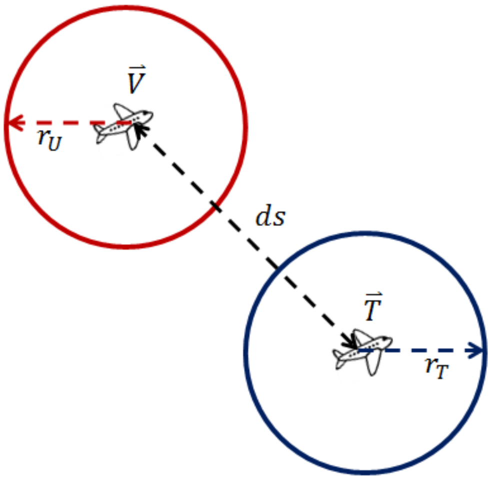


Figure 3.6: Separation distance between aircrafts

From Figures 3.5 and 3.6,

t_i = initial time frame (current data time at UAV's sensor detect the obstacle).

t_a = initial overlapping time.

t_{ip} = intercept point time.

t_b = final overlapping time.

t_f = final time frame.

r_U = radius of SBC for UAV.

r_T = radius of SBC for target.

ds = separation distance or relative distance between UAV and target.

In two dimensional coordinates, the kinematic equation of motion for UAV and target for constant heading angles are presented in (3.15) and (3.16).

$$\begin{bmatrix} X_U \\ Y_U \end{bmatrix}_{t_f} = \begin{bmatrix} X_U \\ Y_U \end{bmatrix}_{t_i} + \begin{bmatrix} V_{UX} \\ V_{UY} \end{bmatrix} \Delta t + \frac{1}{2} \begin{bmatrix} a_{UX} \\ a_{UY} \end{bmatrix} \Delta t^2 \tag{3.15}$$

$$\begin{bmatrix} X_T \\ Y_T \end{bmatrix}_{t_f} = \begin{bmatrix} X_T \\ Y_T \end{bmatrix}_{t_i} + \begin{bmatrix} V_{TX} \\ V_{TY} \end{bmatrix} \Delta t + \frac{1}{2} \begin{bmatrix} a_{TX} \\ a_{TY} \end{bmatrix} \Delta t^2 \tag{3.16}$$

Assumption 3.2.3.1.1. The separation distance between two aircrafts will be just the magnitude of the relative position vector.

The general equation of separation distance, ds between UAV and target is given by:

$$ds^2 = \Delta X_{t_f}^2 + \Delta Y_{t_f}^2$$

$$ds^2 = (X_U - X_T)_{t_f}^2 + (Y_U - Y_T)_{t_f}^2$$

$$ds^2 = (X_U)_{t_f}^2 + (X_T)_{t_f}^2 - 2(X_U X_T)_{t_f} + (Y_U)_{t_f}^2 + (Y_T)_{t_f}^2 - 2(Y_U Y_T)_{t_f} \tag{3.17}$$

3.2.3.2 Time Overlapping Equation Analysis

Detection of time overlapping for two circle boundaries can be expressed by substituting (3.15) and (3.16) into (3.17). Referring to Figure 3.4b, the magnitude of separation distance is equal to the total of UAV's radius and target's radius. Thus, the overlapping time range between UAV and target can be investigated by substituting (3.18) into (3.19).

$$ds = r_U + r_T \tag{3.18}$$

Furthermore, by substituting (3.15) and (3.16) into (3.17), the relationship between ds and Δt is obtained as

$$ds^2 = A\Delta t^4 + B\Delta t^3 + C\Delta t^2 + D\Delta t + E \tag{3.19}$$

where,

$$A = (a_{UX}^2 + a_{UY}^2 + a_{TX}^2 + a_{TY}^2) - 2(a_{UX}a_{TX} + a_{UY}a_{TY})$$

$$B = (a_{UX}V_{UX} + a_{UY}V_{UY} + a_{TX}V_{TX} + a_{TY}V_{TY}) - (a_{UX}V_{TX} + a_{UY}V_{TY} + a_{TX}V_{UX} + a_{TY}V_{UY})$$

$$C = (V_{UX}^2 + V_{UY}^2 + V_{TX}^2 + V_{TY}^2) + \left(a_{UX}[X_U]_{t_i} + a_{UY}[Y_U]_{t_i} + a_{TX}[X_T]_{t_i} + a_{TY}[Y_T]_{t_i}\right) - \left(a_{UX}[X_T]_{t_i} + a_{UY}[Y_T]_{t_i} + a_{TX}[X_U]_{t_i} + a_{TY}[Y_U]_{t_i} + 2V_{UX}V_{TX} + 2V_{UY}V_{TY}\right)$$

$$D = 2\left(V_{UX}[X_U]_{t_i} + V_{UY}[Y_U]_{t_i} + V_{TX}[X_T]_{t_i} + V_{TY}[Y_T]_{t_i}\right) - 2(V_{UX}[X_T]_{t_i} + V_{UY}[Y_T]_{t_i} + V_{TX}[X_U]_{t_i} + V_{TY}[Y_U]_{t_i})$$

$$E = [X_U]_{t_i}^2 + [Y_U]_{t_i}^2 + [X_T]_{t_i}^2 + [Y_T]_{t_i}^2 - 2[X_U]_{t_i}[X_T]_{t_i} - 2[Y_U]_{t_i}[Y_T]_{t_i}$$

Therefore, PTCOT is the quartic equation that is derived from (3.19) becomes

$$A\Delta t^4 + B\Delta t^3 + C\Delta t^2 + D\Delta t + (E - ds^2) = 0 \tag{3.20}$$

When the PTCOT is solved, it gives four answers of Δt. The potential of collision in near future can be determined by examining all the four answers ($\Delta t_1, \Delta t_2, \Delta t_3, \Delta t_4$). Remember that for constant velocity, the acceleration value is zero. Thus quartic equation becomes quadratic equation and it gives only two answers ($\Delta t_1, \Delta t_2$). Table 3.1 shows the types of answers for different scenario as described below:

1- Case 1 = No interception point, so no collision.

2- Case 2 = Has interception point and collision.

3- Case 3 = Has interception point but no collision.

Table 3.1: Types of answers for different cases

	$\boldsymbol{\Delta t}$	**CASE 1**	**CASE 2**	**CASE 3**
Constant Velocity, $a = 0$	Δt_1	-	Real Number	Imaginary Number
	Δt_2	-		
	Δt_3	-	-	-
	Δt_4	-	-	-
Constant Acceleration, $a \neq 0$	Δt_1	-	Real Number	Imaginary Number
	Δt_2	-		
	Δt_3	-	Imaginary Number	
	Δt_4	-		

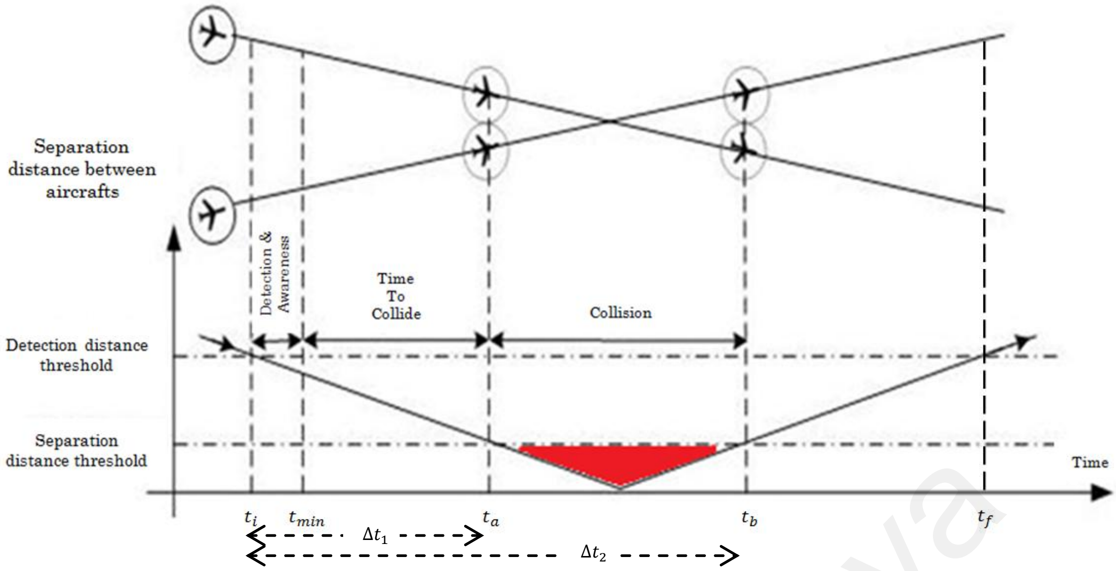


Figure 3.7: Geometric configuration between moving UAV and target

The two real numbers of t_a and t_b as presented in Figure 3.7 are described in (3.21)-(3.22). The collision time range is represent by the red shaded area can be defined as (3.23).

$$t_a = \Delta t_1 + t_i \tag{3.21}$$

$$t_b = \Delta t_2 + t_i \tag{3.22}$$

$$t_a \leq collision\ time \leq t_b \tag{3.23}$$

Furthermore, the intercept point time, t_{ip} can be defined by substituting the value of $ds = 0$ into the (3.20).

Assumption 3.2.3.2.1. The intercept point,t_{ip} exists only if UAV and target reach at the intercept point at the same time.

3.2.4 Simulation

In this section, the performance of collision detection algorithm based on PTCOT is verified using MATLAB simulink. Let all information of position, constant velocity and heading angle for UAV and target are assumed in this thesis. Three different cases are considered with the following assumption as shown in Table 3.2 are investigated.

Table 3.2: Assumption of the information for UAV and target

		Case 1	Case 2	Case 3
UAV	Initial Position (km)	(-10,10)	(-10,10)	(-10,10)
	Final Position (km)	(10,10)	(10,10)	(10,10)
	Velocity (m/s)	100	100	100
	Heading Angle (degree)	0	0	0
Target	Initial Position (km)	(-10,20)	(0,0)	(0,0)
	Final Position (km)	(10,20)	(0,20)	(0,20)
	Velocity (m/s)	100	100	50
	Heading Angle (degree)	0	90	90

Where each case represents:

1. Case 1 – No interception point.
2. Case 2 – Has interception point and collision in near future.
3. Case 3 – Has interception point but no collision in near future.

3.2.4.1 Simulation Results

The prediction of collision in near future for four cases using PTCOT is tested on developed algorithm. The algorithm results for each case are presented in Table 3.3.

Table 3.3: Simulation results of collision detection algorithm using circle overlapping test

	CASE 1	**CASE 2**	**CASE 3**
Δt_1 (sec)	-	85.86	$120 - 35.77\text{i}$
Δt_2 (sec)	-	114.14	$120 + 35.77\text{i}$
Δt_3 (sec)	-	-	-
Δt_4 (sec)	-	-	-
Collision Detected	No	Yes	No

Null answers for case 1 means that no overlapping in near future between moving UAV and target. Therefore, no collision in the near future is detected and collision avoidance mode will never be activated.

Meanwhile, case 2 gives two real numbers of overlapping time, $\Delta t_1 = 85.86\ sec$ and $\Delta t_2 = 114.14\ sec$. It explains the range of overlapping between moving UAV and target. Therefore, collision in the near future is detected. Hence, the collision avoidance mode needs to be activated.

Finally, for case 3, it gives two imaginary numbers, $\Delta t_1 = 120 - 35.77\text{i}\ sec$ and $\Delta t_2 = 120 + 35.77\text{i}\ sec$ which means that no overlapping in near future between moving UAV and target. Therefore, no collision in the near future is detected and collision avoidance mode will never be activated.

3.3 Analysis of Future Scenario

Future analysis of three cases scenarios are shown in Figure 3.8, Figure 3.9, and Figure 3.10. The accuracy of prediction of collision in near future data that is reported in Table 3.3 is discussed based on PTCOT method.

3.3.1 Discussion of PTCOT Method

Future analysis of three cases scenarios are shown in Figure 3.8, Figure 3.9 and Figure 3.10. From Figure 3.8 and Figure 3.10, its show the scenario for case 1 and case 3 where both circles of moving UAV and target will not overlap during the time frame. The accuracy of circle overlapping test in Table 3.3 for case 1 and case 3 are examined.

From Figure 3.9, it shows the scenario for case 2 where both circles of moving UAV and target starts to overlap at time t = 85.85sec until time t = 114.14sec. Therefore, collision detection algorithm must detect the collision in near future. As a result, the accuracy of collected data in Table 3.3 for case 2 is also satisfied.

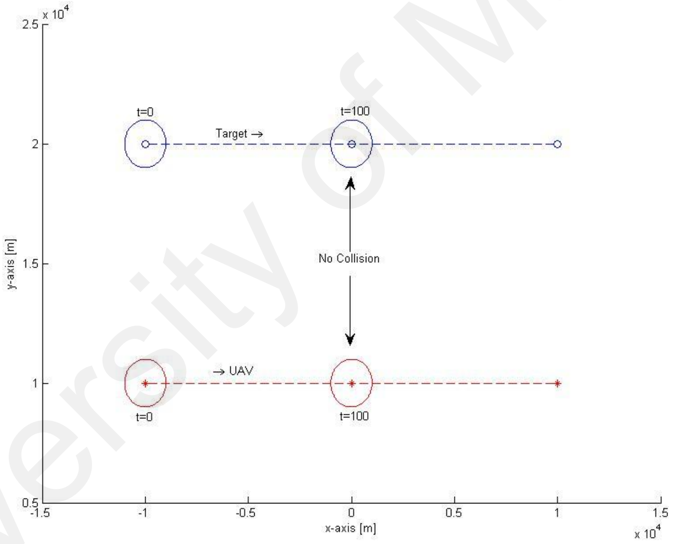


Figure 3.8: Future scenario for case 1

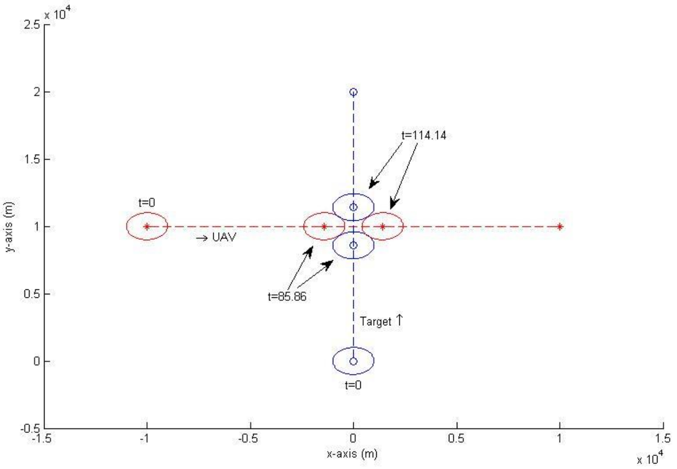


Figure 3.9: Future scenario for case 2

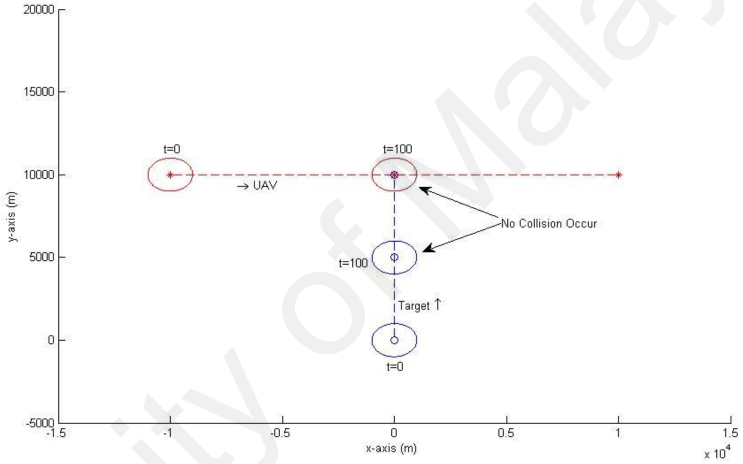


Figure 3.10: Future scenario for case 3

Furthermore, the potential of collision in near future can be determined by investigating on separation distance graph as shown in Figures 3.11-3.13. The red line represents the total radius of UAV and target. While, blue line represents the separation distance between UAV and target versus time. Collision or circle overlapping is detected when the blue line is located below than the red line as shown in Figure 3.12. According to Figure 3.12, the separation distance is zero at time $t = 100sec$ for case 2, which explaining the location of UAV and target is located at the intercept point.

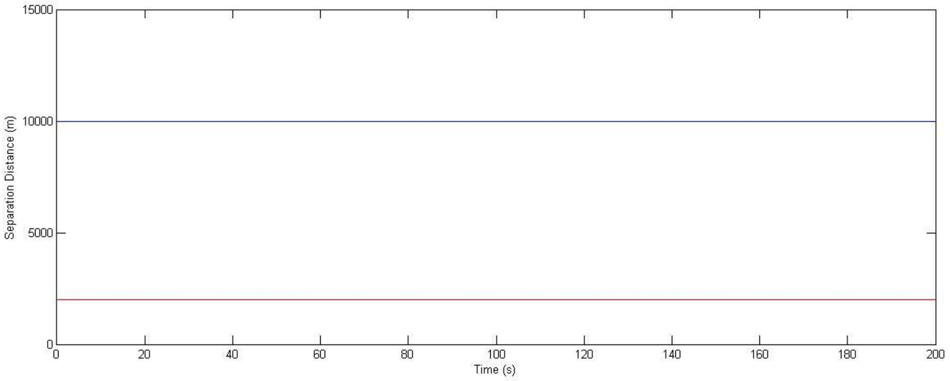


Figure 3.11: Separation distance for case 1

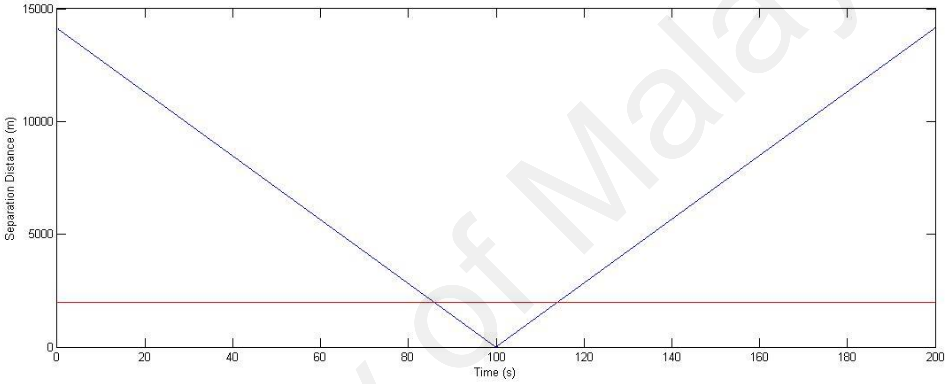


Figure 3.12: Separation distance for case 2

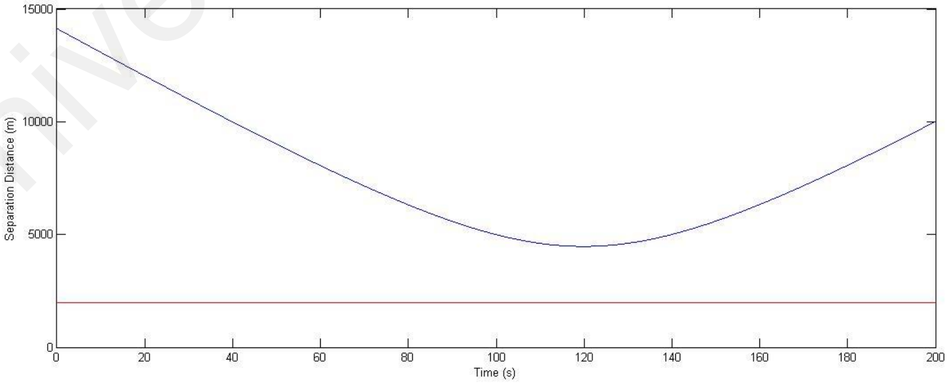


Figure 3.13: Separation distance for case 3

3.4 Comparison between PT and PTCOT Methods

In this section, the comparison and similarity between PT method and PTCOT method are discussed. The best method between them will be chosen to apply in order to design collision avoidance algorithm in the next chapter.

The comparisons in designing collision detection for both methods are explained in several factors such as:

i. Collision time range:

In PT method, collision time range is not provided during designing the collision detection algorithm. It is only depending on the intercept point time to check the collision in near future. This method is not very robust since the UAV and target start to collide at time, t_a which earlier than intercept time, t_{ip} depending on their protective zone radius. Besides, the collision can be happened without the existence of intercept point. It can be analysed when the parameters of UAV and target are shown in Table 3.4. PT method produces both values of u_a and u_b are infinity, which no collision is detected. However, Figure 3.14 shows the collision time range, 91.34 sec $\leq t_{collision} \leq 108.7$ sec when PTCOT method is applied.

Table 3.4: Parameters for UAV and Target

	Initial Position (km)	Final Position (km)	Velocity (m/s)	Heading Angle (degree)	Radius (m)
UAV	(-10,10)	(10,10)	100	0	1000
Target	(10,9)	(-10,9)	100	-180	1000

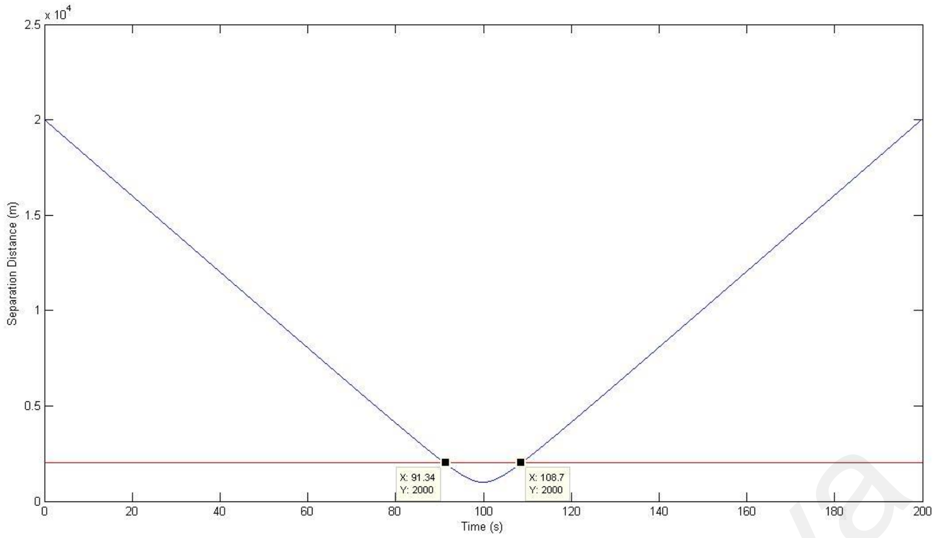


Figure 3.14: Separation distance of moving UAV and target

ii. Circumstances involved:

For PT method, it only can be applied in situation where UAV and target move in constant velocity. Meanwhile, for circle overlapping method can be applied in situation where UAV and target move in constant velocity or constant acceleration/deceleration with both flying in a straight flight path but not in changing acceleration/deceleration.

iii. Initial and final destination during time frame, $t_i < t < t_f$:

If PT method is using, each aircraft must have an idea about the initial and the final destination of each another during time frame to build (3.10). For the meantime, only the initial position at time $t = t_i$ is needed for PTCOT with the intention of produce (3.20).

The similarities in designing collision detection for both methods are explained below:

i. Collision time range:

Both methods are not applicable for obstacles. In other words, both methods cannot be applied for collision between UAV and obstacle/static object.

ii. Straight line path:

Both methods are only applicable for both aircrafts fly in a straight line path during time frame, $t_i < t < t_f$.

As a conclusion, PTCOT method is more robust than the PT method to be applied in collision detection design. Therefore, PTCOT method is chosen for collision avoidance design which will be discussed in the next chapter. However, it holds for the assumption that both UAV and target fly in a straight line path during the time frame, $t_i < t < t_f$.

CHAPTER 4

Collision Avoidance Algorithm

Collision avoidance is a process of preventing a UAV from colliding with any other moving aircraft. The collision avoidance method being employed in this research is based on the PTCOT method approach. UAV and target are two cooperative aircrafts and assumed cooperative behaviour in which communication of position, velocity, heading angle, and proposed trajectories are allowed in this research. Besides, both aircrafts are assumed to fly in the same altitude.

In this chapter, a collision avoidance maneuver is presented. The idea of collision avoidance maneuver is based on the collision avoidance (clearance trajectories) together with collision detection. Two different commands for collision avoidance are proposed in two sub-chapters below which are change in position command and change in speed command.

4.1 Introduction

Collision avoidance maneuver involves determining what the appropriate action for aircraft should be performed to avoid the collision in near future. Once collision in near future is detected, the escape trajectory is estimated by collision avoidance algorithm and the maneuvering to avoid the detected collision is realized. A sequence of predefined maneuvers is determined when the UAV get involves in the potential collision and appropriate action according to a set of rules is taken. However, this will result on the operation of time.

Various collision avoidance methods had been proposed for both manned and unmanned aircraft. For example, Kuwata propose a real-time trajectory design using receding horizon control for UAV (Kuwata, 2003), Call employs a genetic or evolutionary algorithm for UAV path planning to avoid collision (Call, 2006), and Albaker come out another method for unmanned aircraft collision avoidance system using cooperative agent-based negotiation approach (B. M. Albaker & Rahim, 2010).

Let each aircraft flies on a horizontal plane within a fixed altitude layer for this research. Only change in position or change in speed is considered as a suitable maneuvering realization command for both computing aircrafts. If change in position is chosen, then the command will provide a new flight path segment to the computing aircraft as soon as possible. The possible maneuver dimensions include turn left or right which is based on the relative angle of the UAV to the target aircraft. In the meantime, if change in speed is chosen, the resulting change in the speedup or slowdown is suggested depending on the limitation of minimum and maximum velocity of the UAV.

4.2 Position Change Command

The idea is to present the trajectory without collision which regarded as a series of flight plan segment by connecting the generated of control points. The UAV has to find a collision-free trajectory during the time frame period, $t_i < t < t_f$ as discussed in Figure 3.7. The control points for which connection were made determines the form of the avoidance trajectory. UAV requires the capability to plan a new maneuver to avoid collision in near future.

The inspiration of collision avoidance design in proposing a new flight path for this research is studied thoroughly from the demonstration maneuver realization process proposed by Albaker (B. M. Albaker & Rahim, 2010). Figure 4.1 shows the two different predefined maneuvers for UAV from top view image.

The direction of change command is depending on the relative angle, θ_{rel} between UAV and target aircraft as shown in Figure 4.2. If relative angle is positive value (which is target coming from in front of left side of UAV), then UAV will turn its trajectory to the left to avoid the predicted collision. Meanwhile, if relative angle is negative value (which is target coming from in front of right side of UAV), then UAV will turn its trajectory to the right. Determination factor of the turning of UAV trajectory will be discussed in detail in section 5.2. The results in determination of the avoidance heading angle are shown in Figure 4.1 and Figure 4.2. The mathematical equation for relative angle, θ_{rel} is shown below:

$$\theta_{rel} = \tan^{-1}\left(\frac{Y_T - Y_U}{X_T - X_U}\right)$$

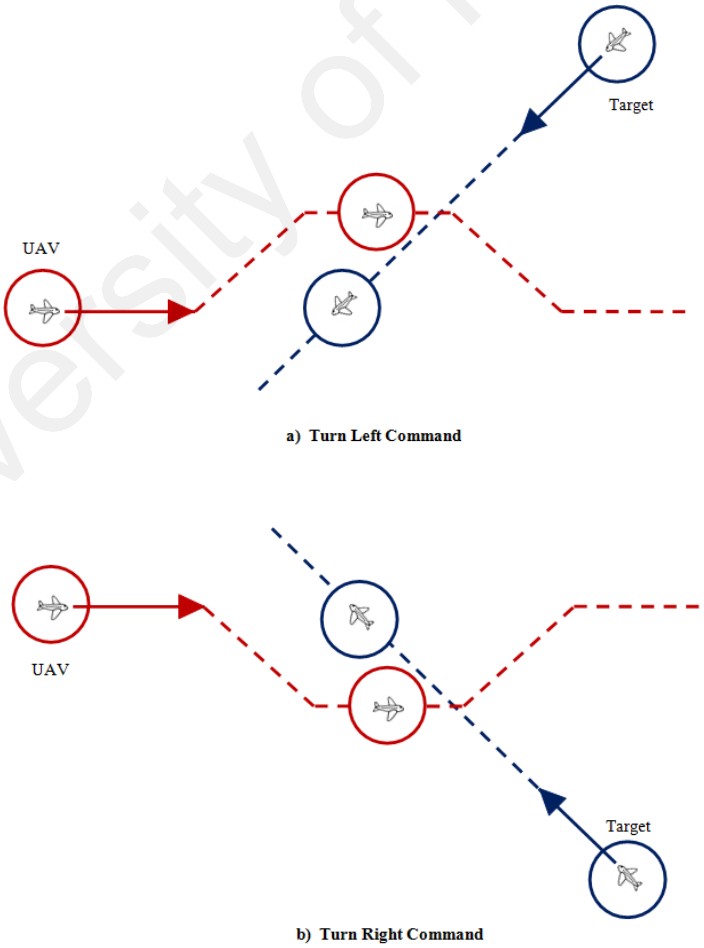
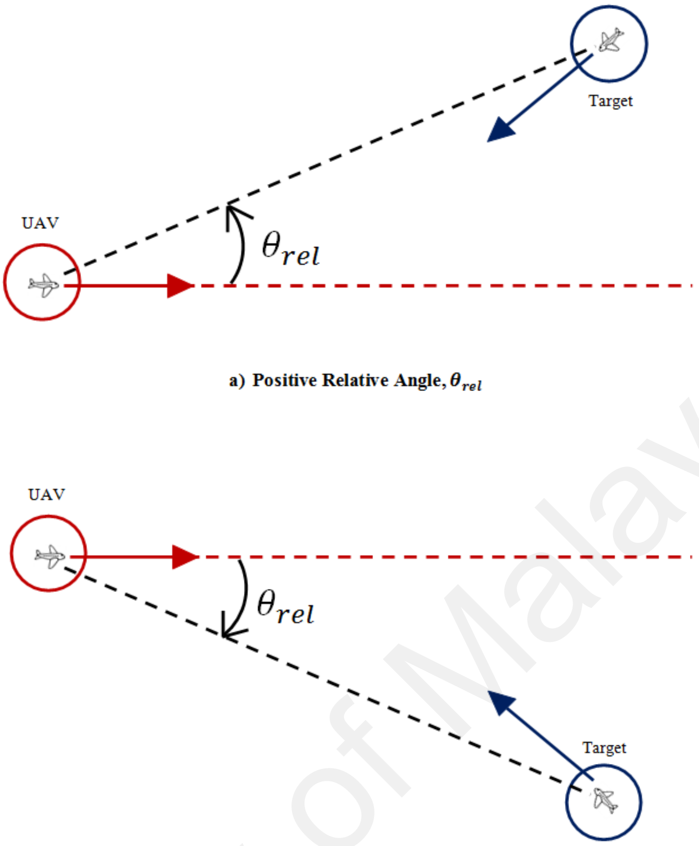


Figure 4.1: New flight path after applying collision avoidance algorithm



a) Positive Relative Angle, θ_{rel}

b) Negative Relative Angle, θ_{rel}

Figure 4.2: Relative angle between UAV and target

4.2.1 Control Points of Collision Avoidance Maneuver

In this work, an algorithm is developed to modify slightly the flight path using linear connection trajectory. Modifying in collision avoidance is described in Figure 4.3. The movements of UAV and target at specific time are described to show the collision avoidance path of UAV. Let the point of U_2', U_3', U_4', and U_5' are new control points of UAV after applying collision avoidance algorithm. While, U_2 is a point of UAV at time t_2 before apply the algorithm. The collision avoidance trajectory can be generated from the information of new control points.

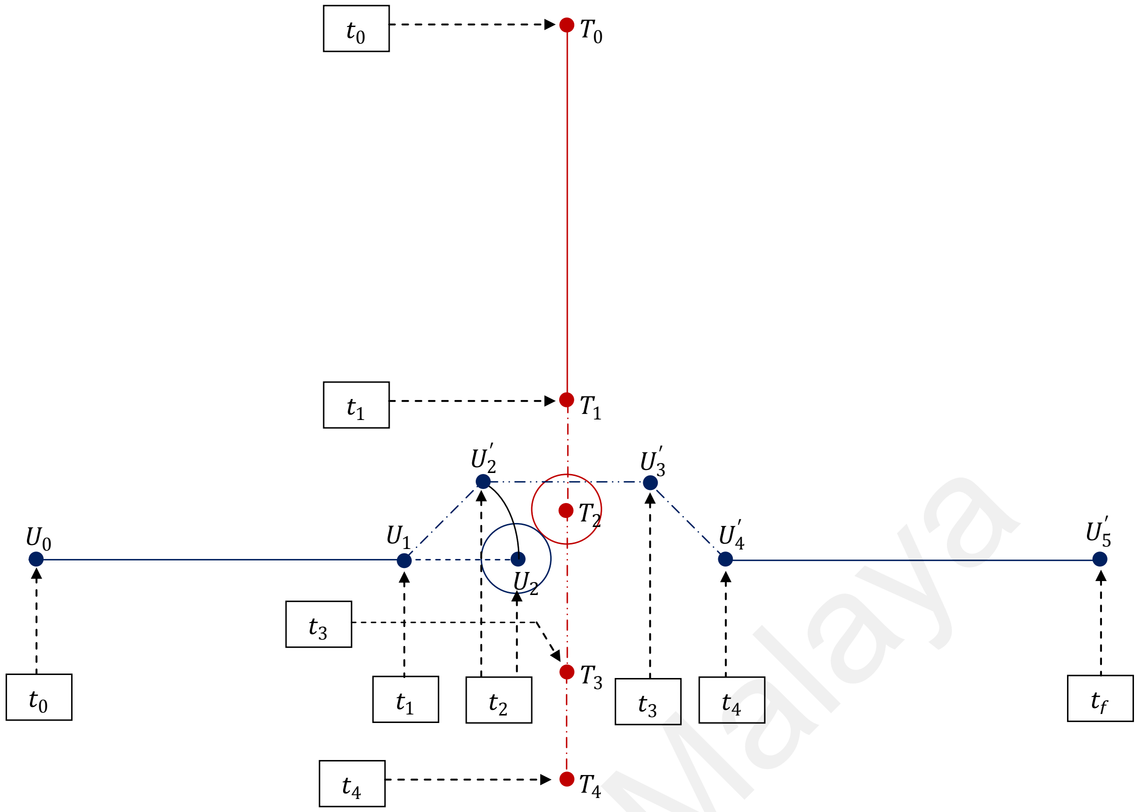


Figure 4.3: Path in form of a series of linear segments

From Figure 4.3,

T_0, T_1, T_2, T_3, T_4 = Points of target at time t_0, t_1, t_2, t_3, t_4 , respectively.

$U_0, U_1, U_2', U_3', U_4', U_5'$ = New control points of UAV at time $t_0, t_1, t_2, t_3, t_4, t_f$, respectively.

By applying the collision avoidance algorithm, the equation of motion for UAV can be obtained as

$$U(t) = \begin{cases} U_{0\rightarrow1} & t_0 \leq t \leq t_1 \\ U_{1\rightarrow2} & t_1 \leq t \leq t_2 \\ U_{2\rightarrow3} & t_2 \leq t \leq t_3 \\ U_{3\rightarrow4} & t_3 \leq t \leq t_4 \\ U_{4\rightarrow5} & t_4 \leq t \leq t_f \end{cases} \tag{4.1}$$

where,

$U_{0\to1}$ = Equation of motion from point U_0 to point U_1

$U_{1\to2}$ = Equation of motion from point U_1 to point U_2'

$U_{2\to3}$ = Equation of motion from point U_2' to point U_3'

$U_{3\to4}$ = Equation of motion from point U_3' to point U_4'

$U_{4\to5}$ = Equation of motion from point U_4' to point U_5'

4.2.2 Turn Maneuver Realization

In this section, the new equations of motion for each segment are derived. It is to determine the control points of U_1, U_2', U_3', U_4' and U_5' which it can constituting the linear collision avoidance trajectory. The control points define a trajectory for UAV without collision during the time frame. Each control points can be defined by examine the overlapping between the new trajectory path and the target using the circle overlapping test as described in (3.23).

4.2.2.1 Finding the value of point U_1 and U_2'

The value of point U_1 and U_2' (as shown in Figure 4.4) can be defined by checking the existence of overlapping circle between UAV and target during the movement of UAV from point U_{1_n} to U_{2_n}' and the movement of target from point T_{1_n} to T_2 by investigating the linear motion segments of UAVs in Figure 4.5. The value of t_2 in Figure 4.3 is equal to t_a in (3.21), as described in Figure 3.7. Let the changes of heading angle is fixed for position change command is $\Delta\theta = 45°$ for the case of Figure 4.2(a) and $\Delta\theta = -45°$ for the case of Figure 4.2(b). Note that the length of U_{1_n} to U_2 is equal to the length of U_{1_n} to U_2'.

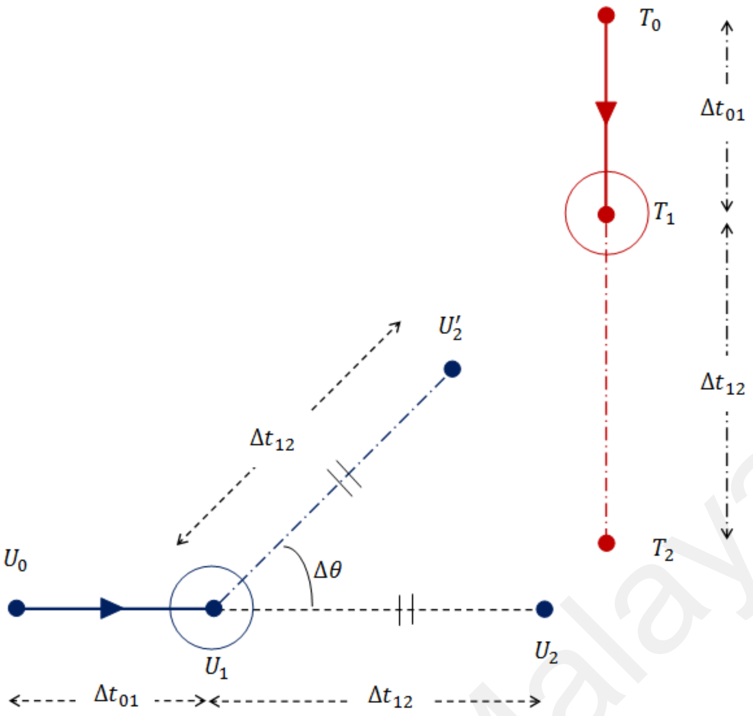


Figure 4.4: The first segment of new path for UAV

$$\vec{U}_{1_n} = \vec{U}_0 + \vec{V}_U \Delta t_{01_n} + \frac{1}{2}\vec{a}_U\left(\Delta t_{01_n}\right)^2 \tag{4.2}$$

$$\vec{T}_{1_n} = \vec{T}_0 + \vec{V}_T \Delta t_{01_n} + \frac{1}{2}\vec{a}_T\left(\Delta t_{01_n}\right)^2 \tag{4.3}$$

$$\vec{U}'_{2_n} = \vec{U}_{1_n} + \vec{V}'_U \Delta t_{12_n} + \frac{1}{2}\vec{a}'_U\left(\Delta t_{12_n}\right)^2 \tag{4.4}$$

$$\vec{T}_{2_n} = \vec{T}_{1_n} + \vec{V}_T \Delta t_{12_n} + \frac{1}{2}\vec{a}_T\left(\Delta t_{12_n}\right)^2 \tag{4.5}$$

where,

$\Delta t_{01_n} \quad = 0.0001n.$

$n \quad = 1,2,\dots,m+1.$

$$t_{1_n} \ = t_0 + \Delta t_{01_n} \tag{4.6}$$

$$\Delta t_{12_n} = t_2 - t_{1_n} \tag{4.7}$$

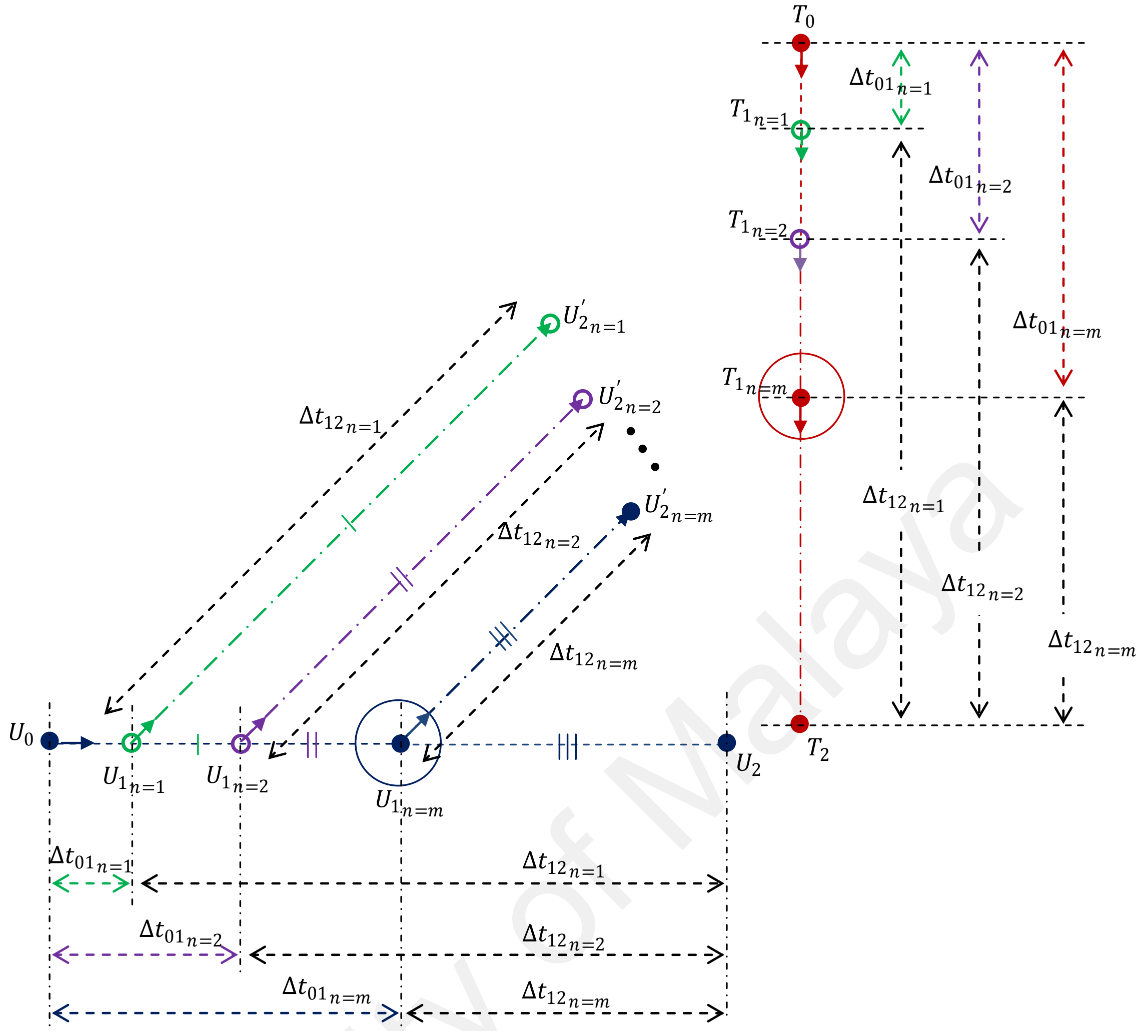


Figure 4.5: Maneuver of UAV for $\boldsymbol{n = 1, 2, \ldots, m + 1}$

Point U_1 and U'_2 can be obtained by using the PTCOT equation as expressed as (4.2) - (4.5) by rearranging (3.16) and (3.17), respectively. Δt_{01_m} is the maximum duration time that circle of UAV will not overlap with target's circle during their movement from point U_{1_m} to U'_{2_m} and from point T_1 to T_2. It is because at duration time $\Delta t_{01_{m+1}}$, PTCOT will detect the overlapping between UAV and target at point $\vec{U}_{1_{m+1}}$ and $\vec{T}_{1_{m+1}}$, respectively. The value of velocity vector, $\vec{V}_U$ and acceleration vector, $\vec{a}_U$ will change to new velocity vector, $\vec{V}'_U$ and new acceleration, $\vec{a}'_U$, , respectively as given in (4.4) because of the changes of heading angle, $\Delta\theta$.

4.2.2.2 Finding the value of point U'_3

Once again, the PTCOT is used to test out the overlapping between both circle of moving UAV and target. The idea is same as in Figure 4.5. If the prediction of circle overlapping is detected during the movement of UAV from point U'_2 to U'_3, the duration of Δt_{01_m} needs to be reduced to $\Delta t_{01_{m-n}}$ (where $n = 1,2, \ldots$), until no overlapping is detected. Note that the coordinate of U_{1_n} and U'_{2_n} will also be shifted. Consequently, it delays the time of UAV to reach the point where overlapping circles is predicted.

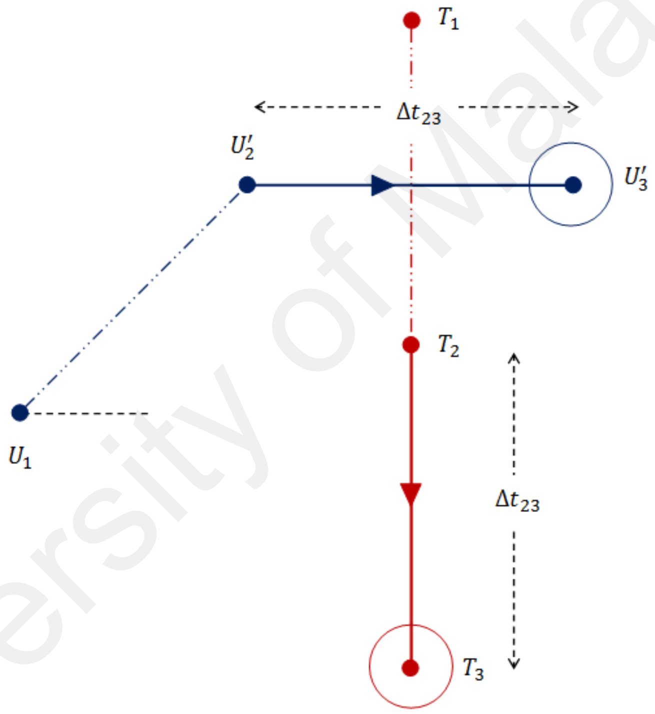


Figure 4.6: The third segment of new path for UAV

Besides, the heading angle for this segment returns to the normal heading angle. Therefore, the value of velocity and acceleration vectors also back to the previous. Finally, equation of motion for UAV and target through the length of U'_2 to U'_3 becomes

$$\vec{U}'_3 = \vec{U}'_2 + \vec{V}_U \Delta t_{23} + \frac{1}{2}\vec{a}_U(\Delta t_{23})^2 \qquad (4.7)$$

$$\vec{T}_3 = \vec{T}_2 + \vec{V}_T \Delta t_{23} + \frac{1}{2}\vec{a}_T(\Delta t_{23})^2 \qquad (4.8)$$

4.2.2.3 Finding the value of point U_4'

The calculation for finding the value of point U_4 is explained in this section. Remember that the UAV must return to its navigation flight path after the critical moments has ended as shown in Figure 4.7. Let the heading angle of UAV at this moment is $-\Delta\theta$ where the value of $\Delta\theta$ is the changes of heading angle as explained in 4.2.2.1.

Given that the PTCOT equations of UAV and target motion along this path (from U_2' to U_3') are described in (4.9) and (4.10), respectively. Whereas the period time of this waypoint is $\Delta t_{34} = \Delta t_{12}$.

$$\vec{U}_4' = \vec{U}_3' + \vec{V}_U'' \Delta t_{34} + \frac{1}{2}\vec{a}_U''(\Delta t_{34})^2 \tag{4.9}$$

$$\vec{T}_4 = \vec{T}_3 + \vec{V}_T \Delta t_{34} + \frac{1}{2}\vec{a}_T(\Delta t_{34})^2 \tag{4.10}$$

While the values of velocity vector and acceleration vector are change due to the changes of heading angle in 4.2.2.2. $\vec{V}_U''$ and $\vec{a}_U''$ represent the new velocity vector and acceleration vector, respectively.

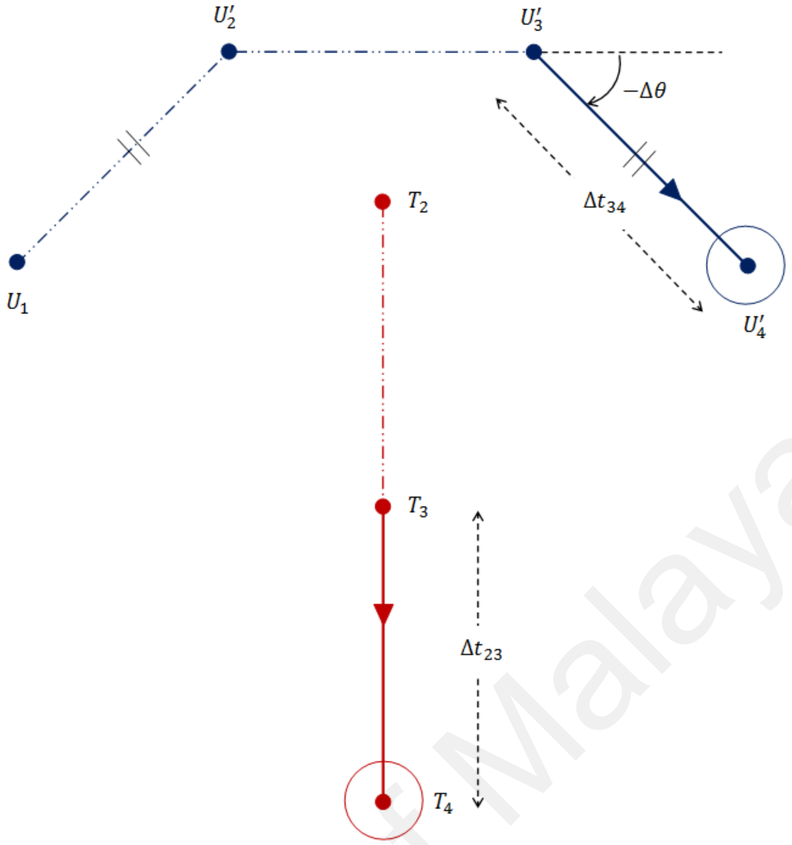


Figure 4.7: The fourth segment of new path for UAV

4.2.3 Summary of Collision Avoidance Algorithm

By applying the position change command in designing collision avoidance algorithm, the equation of UAV motion becomes:

$$U(t) = \begin{cases} \vec{U}_{0\to1} = \vec{U}_0 + \vec{V}_U t + \dfrac{1}{2}\vec{a}_U(t)^2 & t_0 \leq t \leq t_1 \\ \vec{U}_{1\to2} = \vec{U}_1 + \vec{V}'_U t + \dfrac{1}{2}\vec{a}'_U(t)^2 & t_1 \leq t \leq t_2 \\ \vec{U}_{2\to3} = \vec{U}'_2 + \vec{V}_U t + \dfrac{1}{2}\vec{a}_U(t)^2 & t_2 \leq t \leq t_3 \\ \vec{U}_{3\to4} = \vec{U}'_3 + \vec{V}''_U t + \dfrac{1}{2}\vec{a}''_U(t)^2 & t_3 \leq t \leq t_4 \\ \vec{U}_{4\to5} = \vec{U}'_4 + \vec{V}_U t + \dfrac{1}{2}\vec{a}_U(t)^2 & t_4 \leq t \leq t_f \end{cases} \tag{4.11}$$

The following assumptions are required to develop the new path of UAV in order to achieve the collision avoidance.

Assumption 4.2.3.1. The time t_0 value can be determined when the target is detected by the UAV's sensors.

Assumption 4.2.3.2. The parameter of initial position, current velocity or acceleration, heading angle, and radius of PZ are available or can be calculated from the available sensor measurements.

Assumption 4.2.3.3. If velocity remains constant, the acceleration is zero (which means: $\vec{a} = \frac{\partial \vec{V}}{\partial t} 0$).

The flow chart for the collision avoidance algorithm design for position change command is presented in Figure 4.8. Define the information of input data for UAV and target variables at time frame, t_0 is defined as follows:

1) UAV

$\vec{U}_0$ = Position vector of UAV, $\vec{a}_U$ = Acceleration vector of UAV.

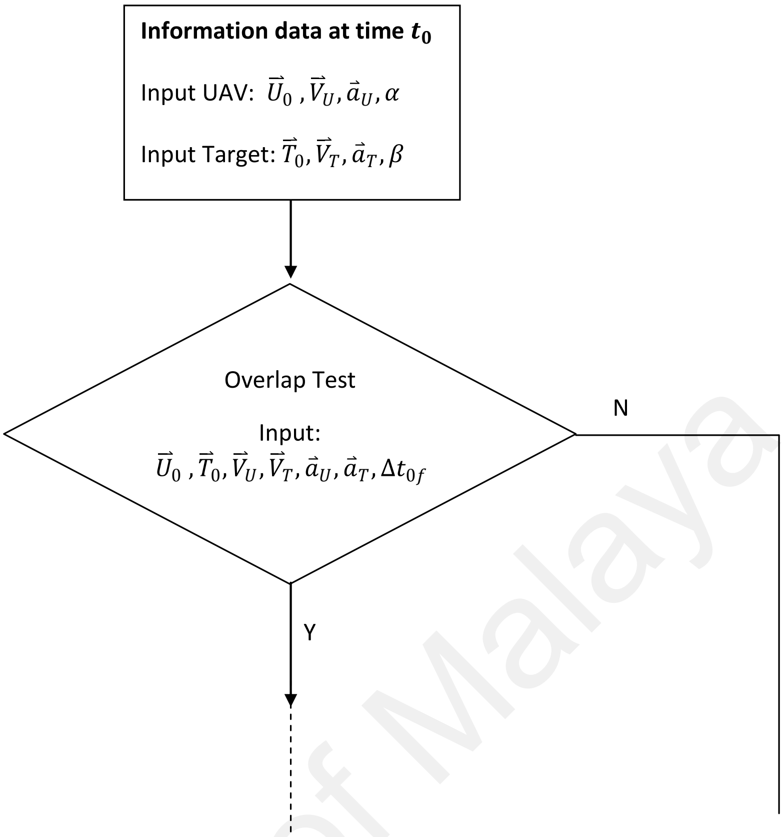
$\vec{V}_U$ = Velocity vector of UAV, α = Heading angle of UAV flight path.

2) Target

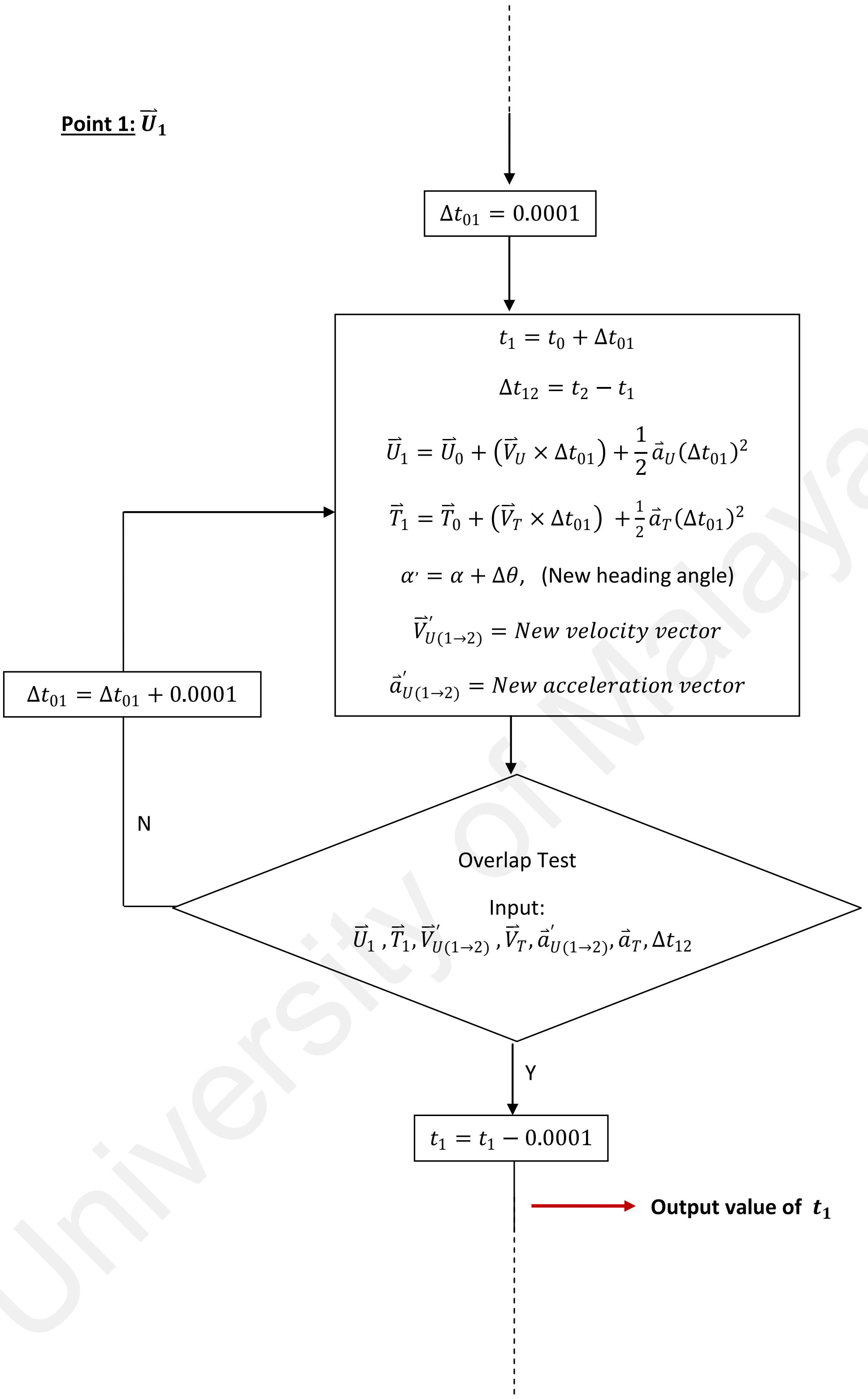
$\vec{T}_0$ = Position vector of target, $\vec{a}_T$ = Acceleration vector of target.

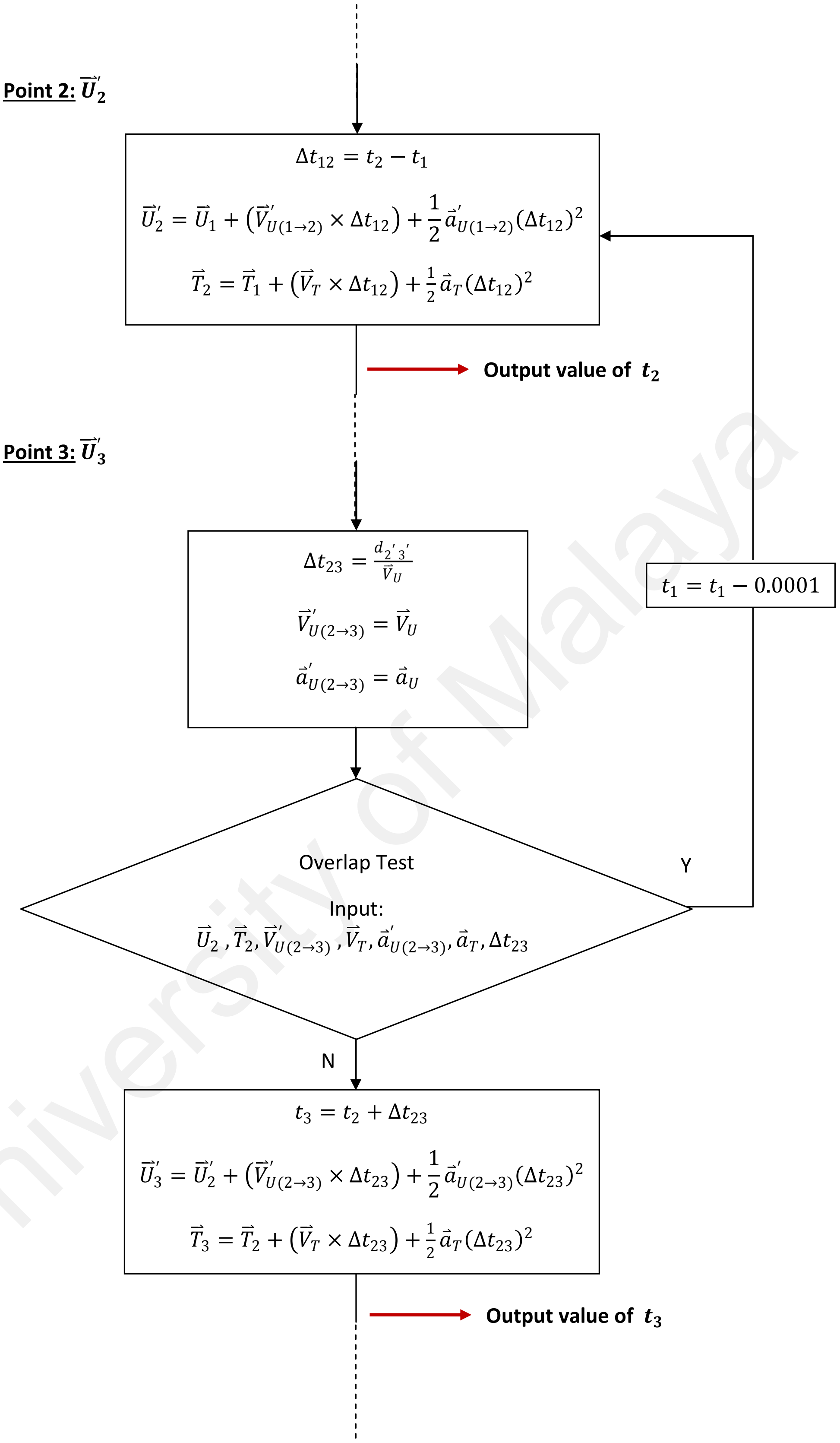
$\vec{V}_T$ = Velocity vector of target, β = Heading angle of target flight path.

Initial Point: $\vec{U}_0$


Information data at time t_0
Input UAV: $\vec{U}_0$, $\vec{V}_U, \vec{a}_U, \alpha$
Input Target: $\vec{T}_0, \vec{V}_T, \vec{a}_T, \beta$
Overlap Test
Input:
$\vec{U}_0$, $\vec{T}_0, \vec{V}_U, \vec{V}_T, \vec{a}_U, \vec{a}_T, \Delta t_{0f}$
N
Y

Point 1: $\vec{U}_1$


$\Delta t_{01} = 0.0001$
$t_1 = t_0 + \Delta t_{01}$
$\Delta t_{12} = t_2 - t_1$
$\vec{U}_1 = \vec{U}_0 + (\vec{V}_U \times \Delta t_{01}) + \frac{1}{2}\vec{a}_U(\Delta t_{01})^2$
$\vec{T}_1 = \vec{T}_0 + (\vec{V}_T \times \Delta t_{01}) + \frac{1}{2}\vec{a}_T(\Delta t_{01})^2$
$\alpha' = \alpha + \Delta\theta$, (New heading angle)
$\vec{V}'_{U(1\to2)} = New\ velocity\ vector$
$\vec{a}'_{U(1\to2)} = New\ acceleration\ vector$
$\Delta t_{01} = \Delta t_{01} + 0.0001$
N
Overlap Test
Input:
$\vec{U}_1, \vec{T}_1, \vec{V}'_{U(1\to2)}, \vec{V}_T, \vec{a}'_{U(1\to2)}, \vec{a}_T, \Delta t_{12}$
Y
$t_1 = t_1 - 0.0001$
Output value of t_1


Point 2: $\vec{U}_2'$
$\Delta t_{12} = t_2 - t_1$
$\vec{U}_2' = \vec{U}_1 + \left(\vec{V}_{U(1\to2)}' \times \Delta t_{12}\right) + \frac{1}{2}\vec{a}_{U(1\to2)}'(\Delta t_{12})^2$
$\vec{T}_2 = \vec{T}_1 + \left(\vec{V}_T \times \Delta t_{12}\right) + \frac{1}{2}\vec{a}_T(\Delta t_{12})^2$
Output value of t_2
Point 3: $\vec{U}_3'$
$\Delta t_{23} = \frac{d_{2'3'}}{\vec{V}_U}$
$\vec{V}_{U(2\to3)}' = \vec{V}_U$
$\vec{a}_{U(2\to3)}' = \vec{a}_U$
$t_1 = t_1 - 0.0001$
Overlap Test
Input:
$\vec{U}_2, \vec{T}_2, \vec{V}_{U(2\to3)}', \vec{V}_T, \vec{a}_{U(2\to3)}', \vec{a}_T, \Delta t_{23}$
Y
N
$t_3 = t_2 + \Delta t_{23}$
$\vec{U}_3' = \vec{U}_2' + \left(\vec{V}_{U(2\to3)}' \times \Delta t_{23}\right) + \frac{1}{2}\vec{a}_{U(2\to3)}'(\Delta t_{23})^2$
$\vec{T}_3 = \vec{T}_2 + \left(\vec{V}_T \times \Delta t_{23}\right) + \frac{1}{2}\vec{a}_T(\Delta t_{23})^2$
Output value of t_3

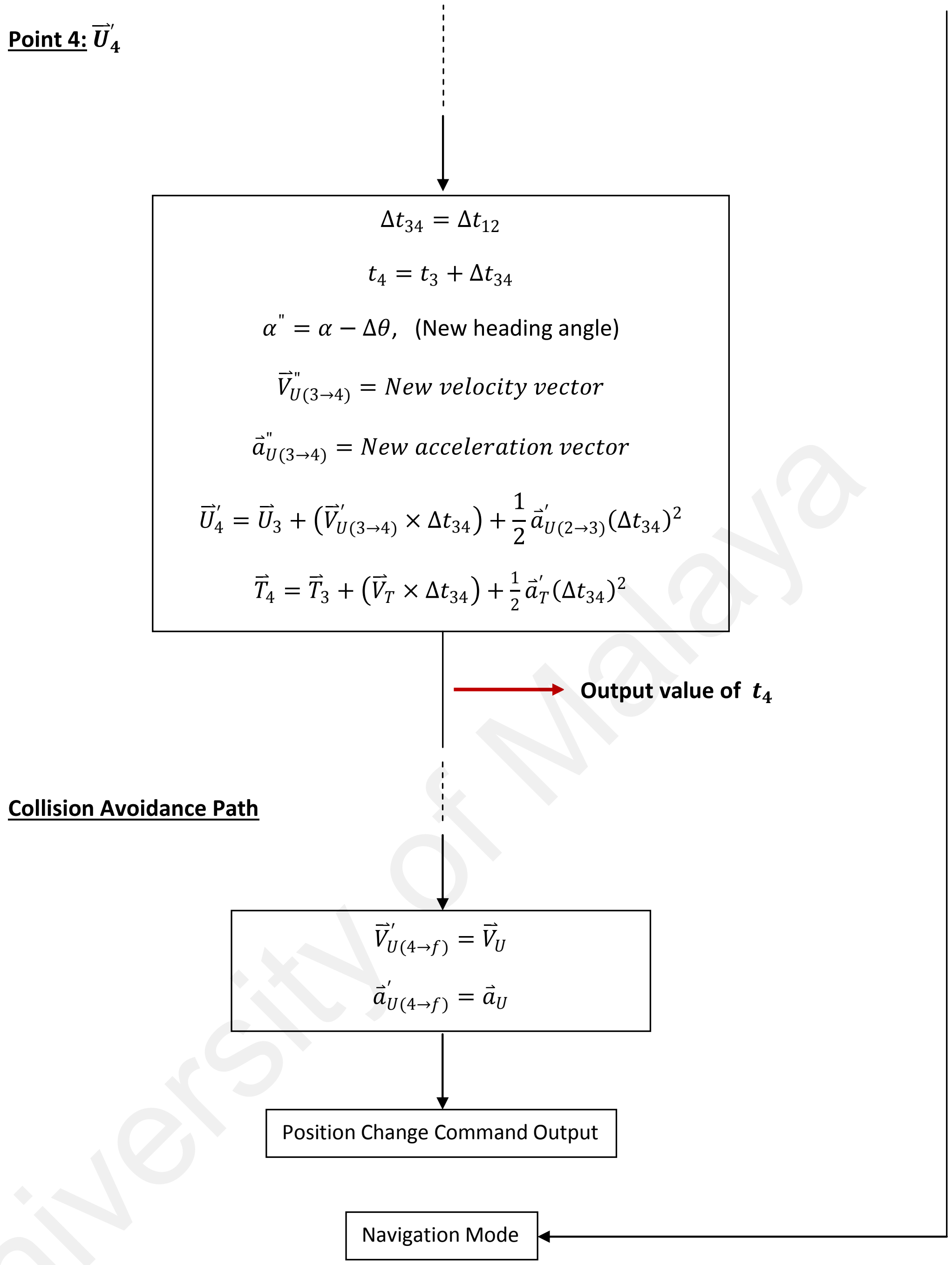

Point 4: $\vec{U}_4'$
$\Delta t_{34} = \Delta t_{12}$
$t_4 = t_3 + \Delta t_{34}$
$\alpha'' = \alpha - \Delta\theta$, (New heading angle)
$\vec{V}_{U(3\to4)}'' = New\ velocity\ vector$
$\vec{a}_{U(3\to4)}'' = New\ acceleration\ vector$
$\vec{U}_4' = \vec{U}_3 + \left(\vec{V}_{U(3\to4)}' \times \Delta t_{34}\right) + \frac{1}{2}\vec{a}_{U(2\to3)}'(\Delta t_{34})^2$
$\vec{T}_4 = \vec{T}_3 + \left(\vec{V}_T \times \Delta t_{34}\right) + \frac{1}{2}\vec{a}_T'(\Delta t_{34})^2$
Output value of t_4
Collision Avoidance Path
$\vec{V}_{U(4\to f)}' = \vec{V}_U$
$\vec{a}_{U(4\to f)}' = \vec{a}_U$
Position Change Command Output
Navigation Mode

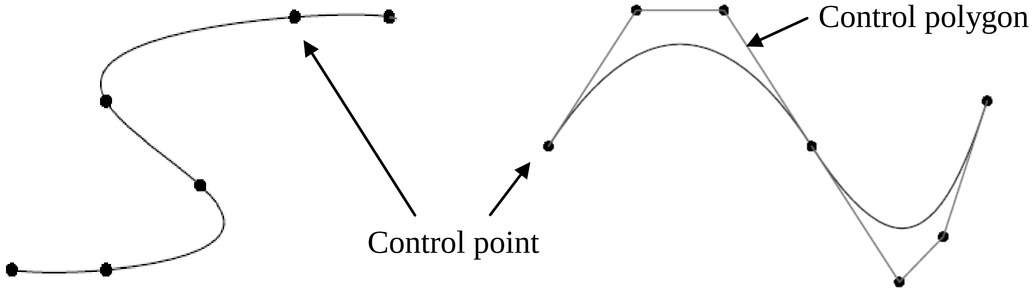
Figure 4.8: Flow chart for collision avoidance algorithm (Position Change Command)

4.3 Smooth Path of a Linear Collision Avoidance Trajectory

The further modification on the proposed linear collision avoidance trajectory which is control by the control points as discussed in previous section 4.2.1 is investigated to form a single continuous curve of collision avoidance path. In this part, the investigation on B-Spline curves method is introduced.

4.3.1 Introduction

Spline functions have been used in various fields such as computer graphics, numerical analysis, image processing, trajectory planning of robot and aircraft, and data analysis in general. A spline curve is a sequence of curve segments that are connected together to form a single continuous curve. A spline curve is a mathematical representation for which it is easy to build an interface that will allow a user to design and control the shape of complex curves and surfaces (John, 2004; H. Kano, Fujioka, H., Egerstedt, M., & Martin, C. F., 2008). The general approach is that a curve is constructed whose shape closely follows the sequence of points given. A curve that passes through each control point is called an *interpolating curve*. A curve that passes near to the control points but not necessarily through them is called an *approximating curve* (Hongxin, 2006; Kayran).



(a) Interpolating Curve (b) Approximating Curve

Figure 4.9: Spline curve (Hongxin, 2006; Kayran)

Basis splines were invented by Schoenberg and made numerically stable by Carl de Boor for the parametric representation of curves, surfaces, volumetric data, etc. (de Boor, 1978; Schoenberg, 1946). There are more commonly known as B-splines. The theory of B-Spline is well developed area of applied numerical analysis and interpolation theory, and the use of B-spline rivals that of Bezier curves in applicability to computer graphics and approximation theory (H. Kano, Egerstedt, M., Nakata, H., & Martin, C. F., 2003). On the other hand, the idea of dynamic splines was first used by Crouch and his colleagues in the determination of aircraft trajectories (Crouch, 1991).

4.3.2 B-Spline Curve Theory

Curvature continuity is an important requirement for the ship building industry, as well as for many other applications. Most shapes are simply too complicated to define using a single Bézier curve. $\boldsymbol{C}^1$ continuity is straightforward to attain using Bézier curve. However, for $\boldsymbol{C}^2$and higher continuity is cumbersome. This is where B-spline curves come in (Thomas, 2005). In practical terms, B-spline curves can be thought of as a method for defining a sequence of degree n Bézier curves that join automatically with $\boldsymbol{C}^{n-1}$ continuity, regardless of where the control points are placed (Thomas, 2005).

B-spline curve is a generalization of the Bézier curve (Thomas, 2005; Weisstein). B-splines are piecewise polynomials of degree n with $\boldsymbol{C}^{n-1}$ continuity at the common points between adjacent segments. B-splines result by mapping the elements of a knot sequence in parametric space into Cartesian space. A spline evaluated at a knot results in a junction point which is the common point shared by two adjacent segments. The B-spline curve of degree p is mathematically defined as (Bindiganavle, 2000; Robby, 2011):

$$\boldsymbol{C}(u) = \sum_{i=0}^{h} N_{i,p}(u) P_i \qquad u_{p-1} \leq u \leq u_{h+1} \tag{4.12}$$

where,

$\boldsymbol{C}(u)$ = Points along the curve as a function of parameter u.

$u \;= u_0, u_1, \ldots., u_{h+p+1}$ are the knot sequence.

$P_i \;= P_0, P_1, \ldots, P_h$ are the $h + 1$ control points.

$i \;=$ index of the control points.

$p \;=$ number of the control points that control a segment or called as spline order.

$N_{i,p}$ = ith B-spline basis function of order p.

There are two mathematical definitions of B-spline basis functions, $N_{i,p}(u)$ are described in (4.13) and (4.14) (Bindiganavle, 2000; Robby, 2011).

1. when $p = 1$:

$$N_{i,1}(u) = \begin{cases} 1 & if\ u_i \leq u < u_{i+1} \\ 0 & otherwise \end{cases} \tag{4.13}$$

2. when $p > 1$:

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p-1} - u_i} N_{i,p-1}(u) + \frac{u_{i+p} - u}{u_{i+p} - u_{i+1}} N_{i+1,p-1}(u) \tag{4.14}$$

Each point on a B-spline is a weighted combination of the local control points, which form a control polygon enclosing the curve. The number of B-spline basis functions is obviously equal to the number of control points and this number is the dimension of the function space. The number of knots needed to define this function space is equal to the dimension plus its order. A curve is constructed whose shape closely posses through each control points. For example, a $\mathbf{C}^2$ cubic B-spline curve of degree 3 shown in Figure 4.10. In general, the lower the degree, the closer a B-spline curve follows its control polylines. The impact of different degree can be shown in Figure 4.11. By changing the position of control point P_i only affects the curve $\boldsymbol{C}(u)$ as shown in Figure 4.12 (Hongxin, 2006).

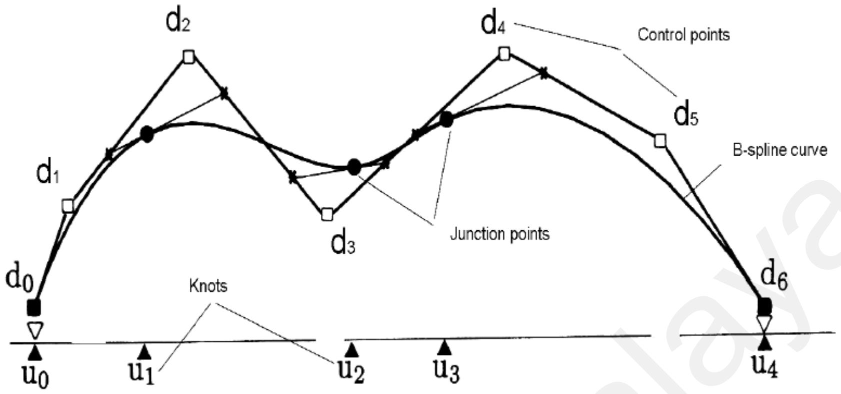


Figure 4.10: A $\mathbf{C^2}$ Cubic B-spline curve with its control polygon (Bindiganavle, 2000)

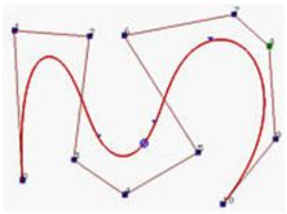
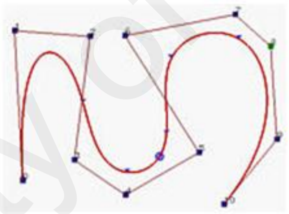
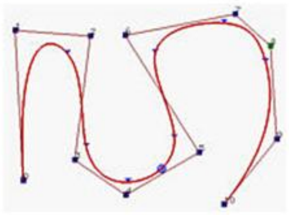
		
Degree 7	Degree 5	Degree 3

Figure 4.11: Three curves with different degree

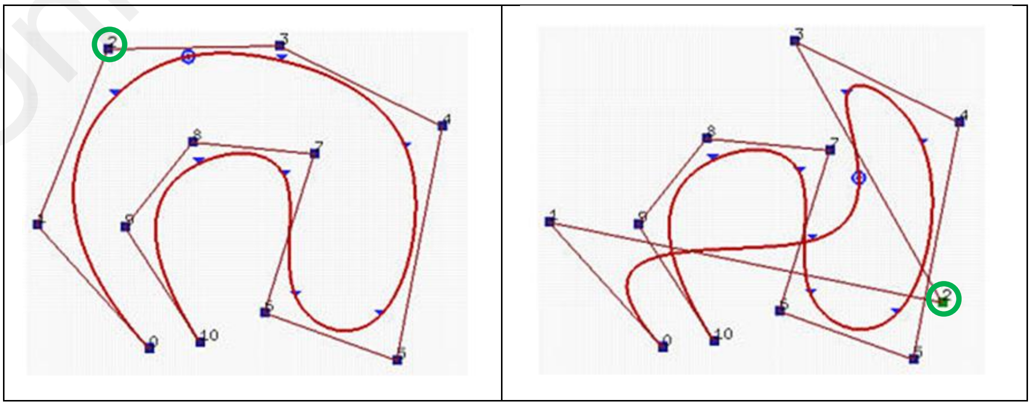


Figure 4.12: Modify the position of control point 2

Clamped B-spline curve $\boldsymbol{C}(u)$ passes through the first and last control points P_0 and P_h, respectively. The basis function $N_{0,p}(u)$ and $N_{h,p}(u)$ is the coefficient of control point P_0 and P_h, respectively. The B-spline curve equation is derived as (Hongxin, 2006):

$$\boldsymbol{C}(u) = N_{0,p}(u)P_0 + \sum_{i=1}^{h-1} N_{i,p}(u)P_i \ + N_{h,p}(u)P_h \tag{4.15}$$

4.3.3 Curves of Collision Avoidance Path

The smooth continuous collision avoidance path for UAV can be achieved with B-spline curve. To be able to do the computation of the curves as shown in Figure 4.13, all the information needed are substituted into b-spline curve equation (4.15) where;

1. B-spline order, p for quadratic = 3.
2. Knot sequence, $u \ = 0, 0, 0, 0.2, 0.3, 0.4, 0.45, 0.55, 0.6, 0.7, 0.82, 1, 1, 1.$
3. $\boldsymbol{P} \ = U_0, a, U_1, b, U_2', c, U_3', d, U_4', e, U_5'$ are the $h + 1$ avoidance control points, where $h = 10$ and the value of a, b, c, d and e are the mid-point of each line segment (or control polygon).

Each line segment (called as control polygon) is controlled by a couple of consecutive control points. Curvature continuity is an important requirement for continuous smooth curve of aircraft trajectory. Control polygon (red line) in Figure 4.13 shows the line segments of collision avoidance path by joining the control points as obtained from (4.11). Meanwhile, the blue line shows a continuous smooth path of collision avoidance for UAV which is obtained after the modification on line segments using B-spline curve.

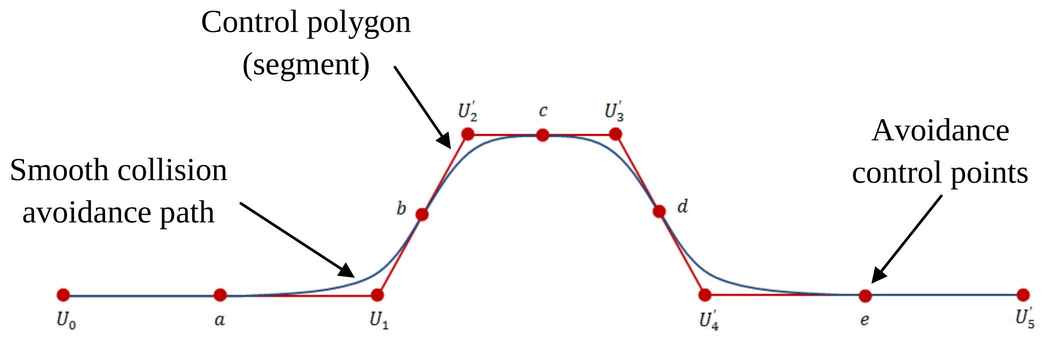


Figure 4.13: Continuous smooth collision avoidance curve of UAV

4.4 Speed Change Command

In this part, collision avoidance algorithm design is proposed using another method where the control in speed for UAV is suggested. The waypoint or flight path for UAV will not change. Only the speed of UAV will change either speeding up (accelerated) or slowing down (decelerated) at the suitable time to avoid the collision in near future that has been predicted. As a result, the time taken to reach the goal point will be longer or shorter after collision avoidance algorithm for speed command is applied. However, there is limitation during speeding up or slowing down the UAV's velocity. The increasing or decreasing value of $\vec{a}_U$ is depends on the limitation of UAV's maximum or minimum velocity as shown in (4.16) and (4.17), respectively.

$$\vec{a}_U' < \vec{a}_{U(min)} \tag{4.16}$$

$$\vec{a}_U' > \vec{a}_{U(max)} \tag{4.17}$$

Acceleration vector in equation of motion for UAV given in (3.16) play the role in designing the speed change command. If the velocity of UAV is increased, UAV will pass through the coordinate of circles-start-to-overlap, U_2 as shown in Figure 4.3 where at time t_2, the new position of UAV is U_2' as explained in Figure 4.14. If the velocity of UAV is decreased, the new position of UAV at time t_2 is U_2' as explained in Figure 4.15.

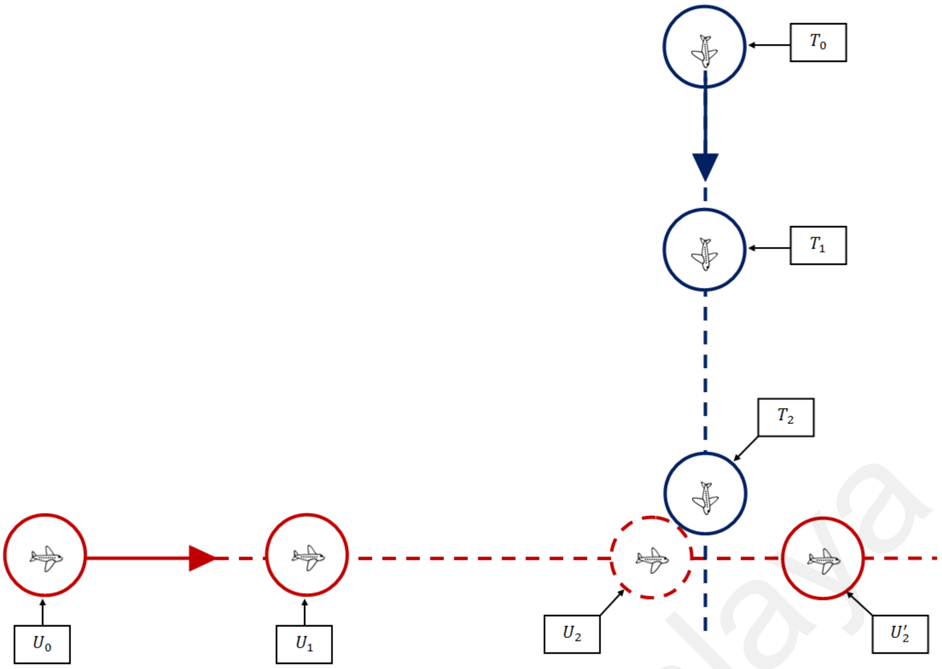


Figure 4.14: Speeding up the velocity of UAV

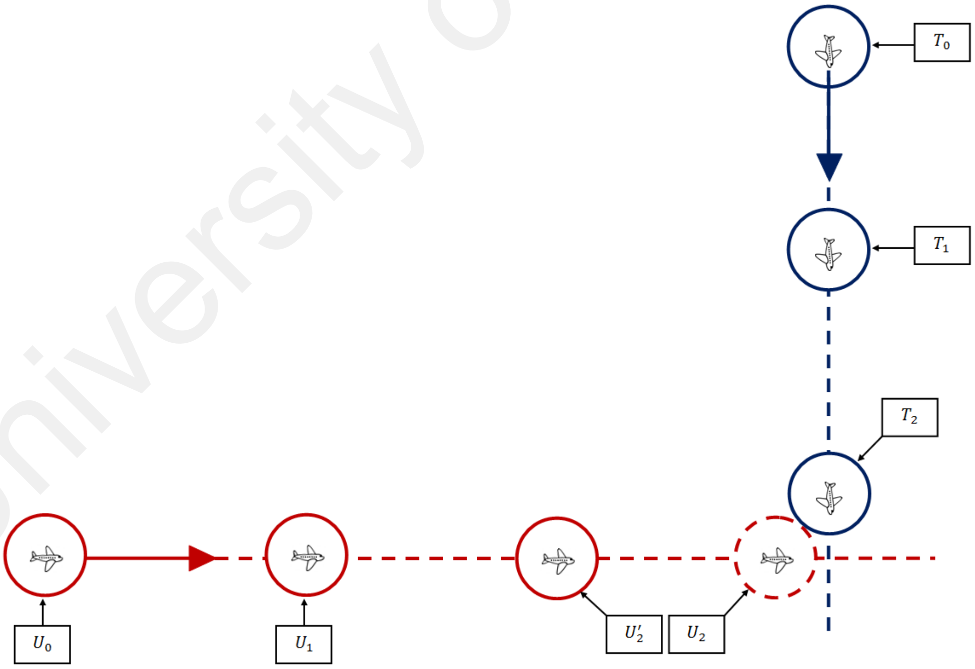


Figure 4.15: Slowing down the velocity of UAV

In Figure 4.14 - 4.15,

U_0 = Position vector of UAV at initial time frame, t_0

T_0 = Position vector of target at initial time frame, t_0

U_1 = Position vector of UAV at time, t_1

T_1 = Position vector of target at time, t_1

U_2 = Position vector of UAV at initial overlapping time, t_2 without applying collision avoidance command

U_2' = Position vector of UAV at initial overlapping time, t_2 with applying collision avoidance command

T_2 = Position vector of target at initial overlapping time, t_2.

4.4.1 Acceleration

In this section, the equation of UAV motion is examined for the case where the speed change command is accelerated velocity. The increment of acceleration for UAV is determined by testing the circle overlapping test for each increment of $1\, m/s^2$. Note that at time t_1, the speed of UAV start to increase must not less than the limitation of time-to-accelerate, t_{min} as described in Figure 3.9.

$$t_1 \geq t_{min} \tag{4.18}$$

Let t_{min} is the minimum time of UAV to initiate the speed change command. Its value depends on the process time taken to initiate the collision detection and avoidance algorithm.

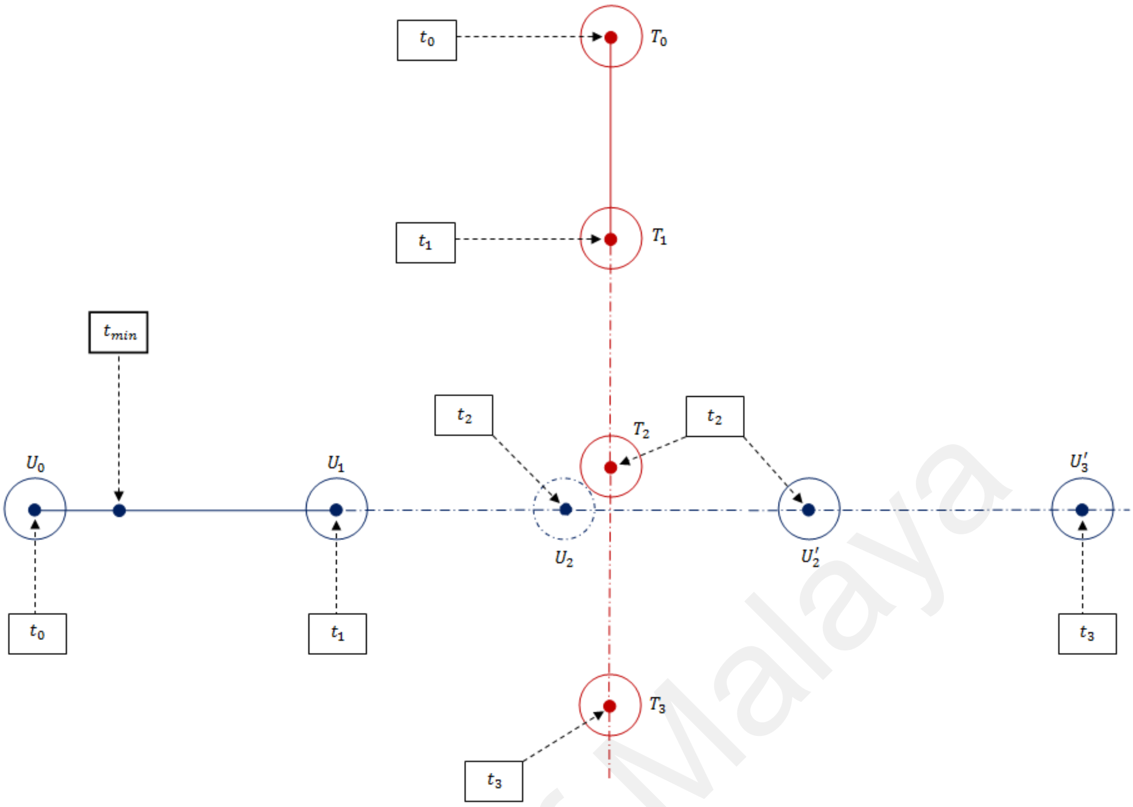


Figure 4.16: The movement of UAV and target for acceleration command

The flow chart of acceleration test is shown in Figure 4.18 (Acceleration Command part). When the speed of UAV is accelerated, the coordinate of UAV at time t_2 is shifted from $\vec{U}_2$ to $\vec{U}_2'$ as shown in Figure 4.16. In this case, UAV also has surpassed within the line of target from point of $\vec{T}_2$ to point of $\vec{T}_3$. Consequently, UAV has passed the predicted collision between target.

4.4.2 Deceleration

In this section, the equation of UAV motion is examined for the case where the speed change command is decelerated velocity. The decrement of acceleration for UAV is also determined by testing the circle overlapping test for each decrement of $1\, m/s^2$. The speed of UAV start to decrease must not less than the limitation of time-to-decelerate, t_{min} as described in (4.18).

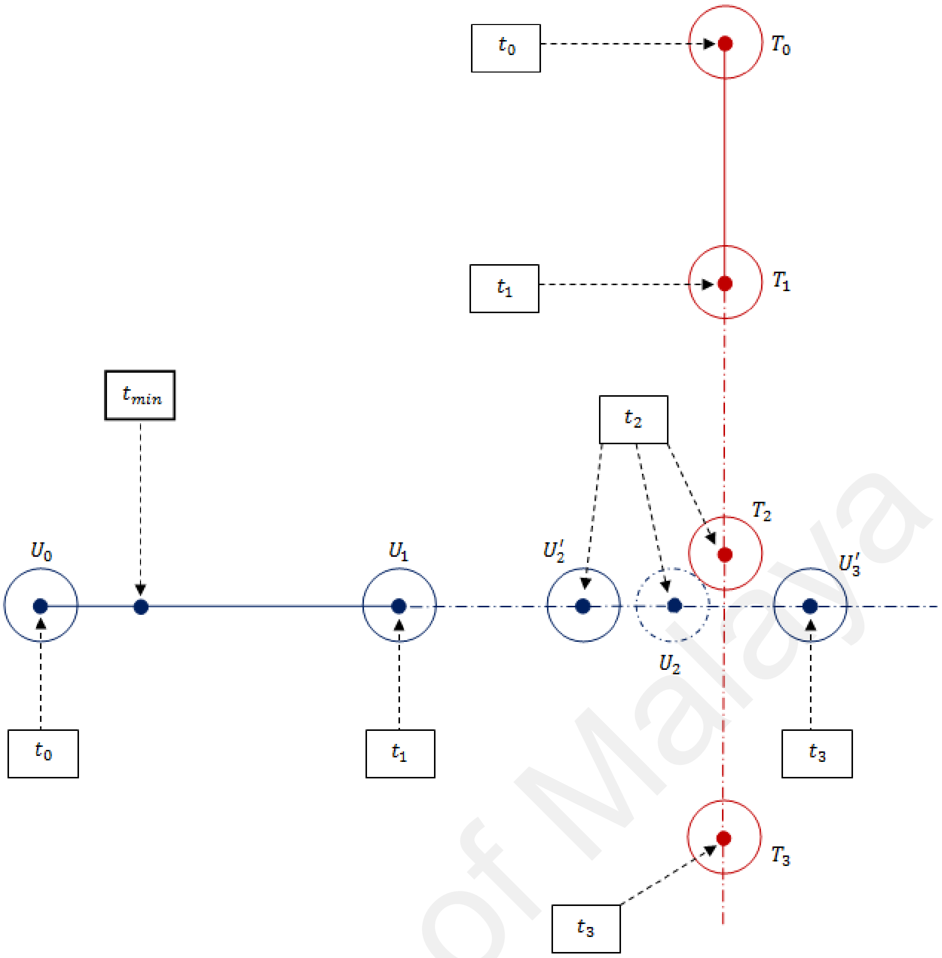
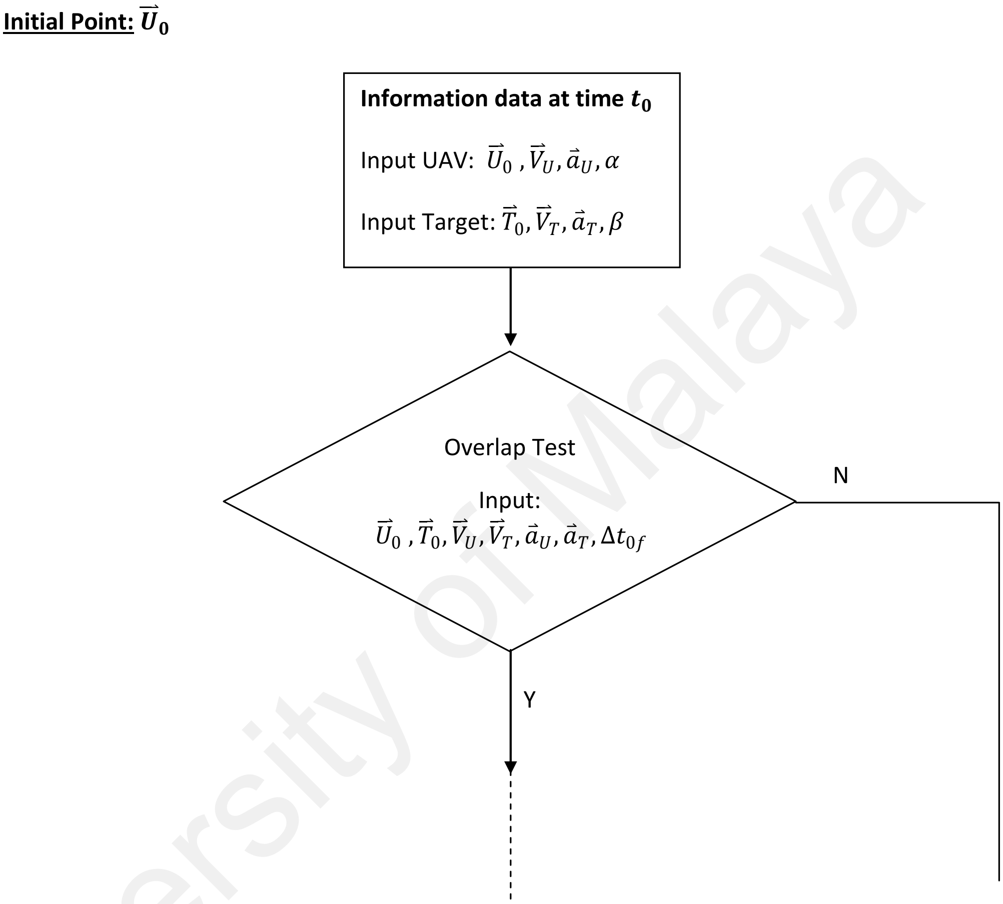


Figure 4.17: The movement of UAV and target for deceleration command

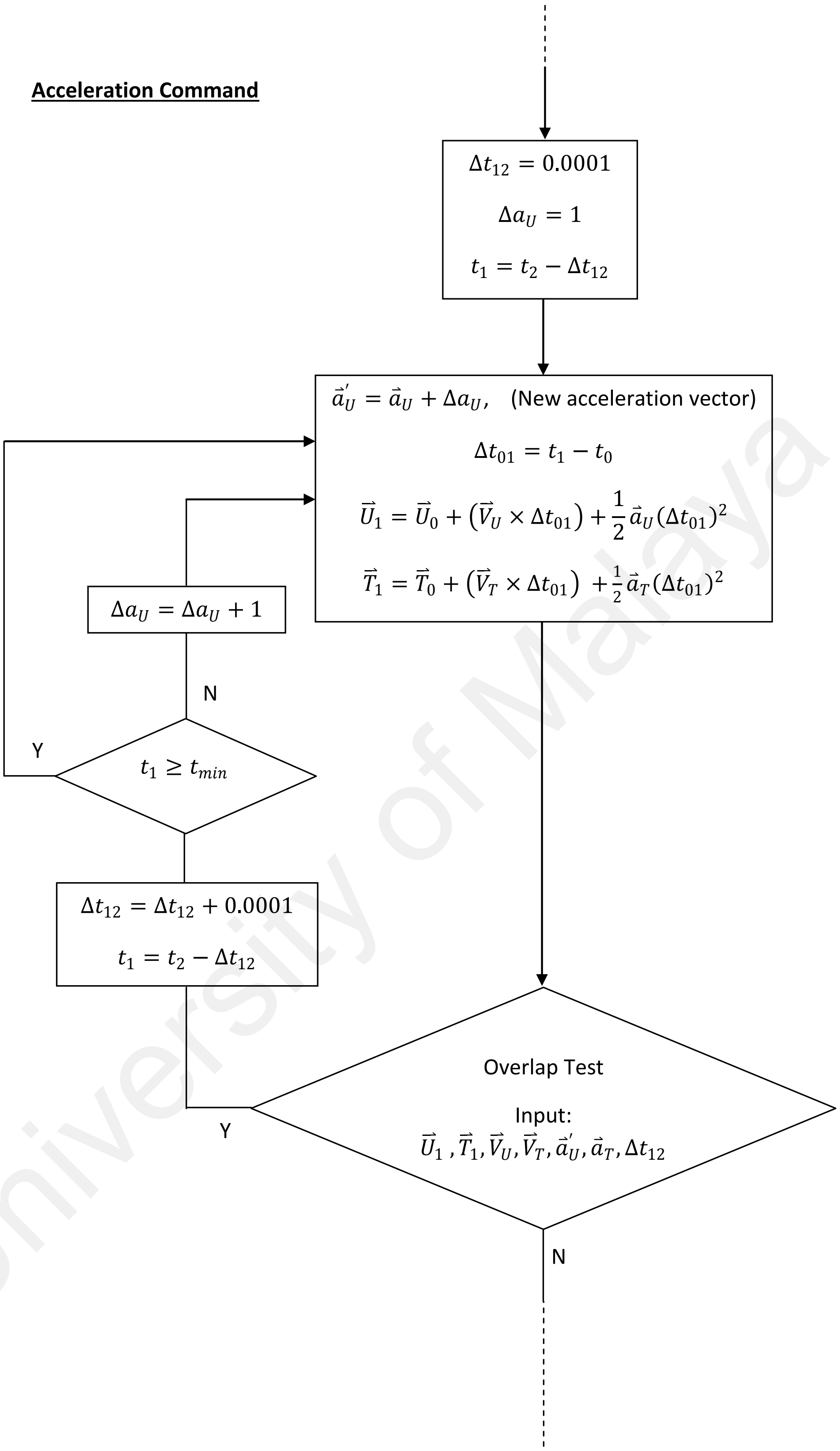
When the speed of UAV is decelerated, the coordinate of UAV at time t_2 is shifted from $\vec{U}_2$ to $\vec{U}_2'$ as shown in Figure 4.17. Unfortunately, the UAV has not surpassed yet the target path from point of $\vec{T}_2$ to point of $\vec{T}_3$. Probability of both circles of UAV and target to overlap still exists. Therefore, circle overlapping test must be done twice where the first test is from point t_1 to t_2, and the second test is from point t_2 to t_3. The flow chart of deceleration test is shown in Figure 4.18 (Deceleration command part).

4.4.3 Summary of Collision Avoidance Algorithm

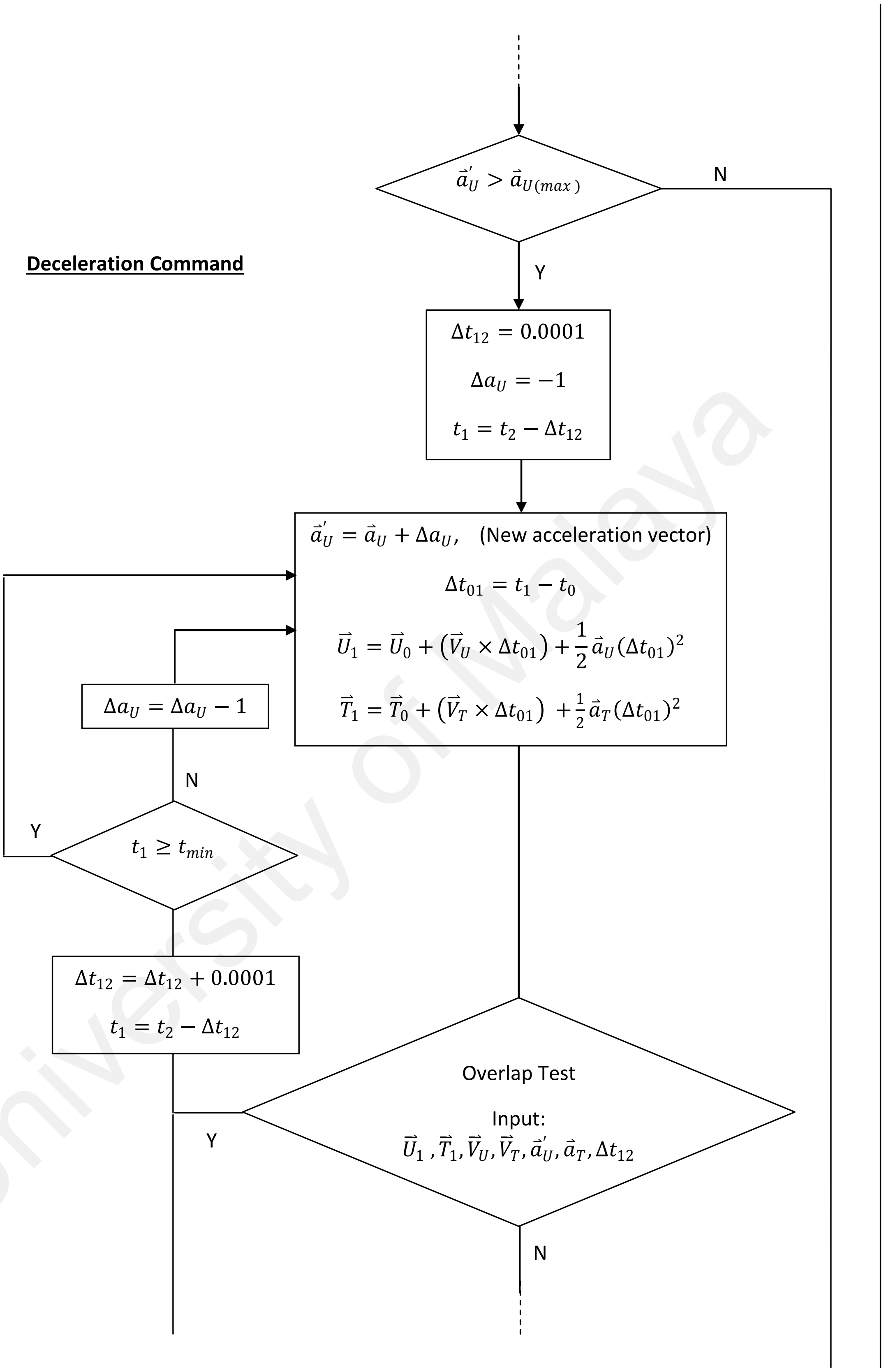
The flow chart for collision avoidance algorithm for speed change command is presented in Figure 4.18.


Initial Point: $\vec{U}_0$
Information data at time t_0
Input UAV: $\vec{U}_0, \vec{V}_U, \vec{a}_U, \alpha$
Input Target: $\vec{T}_0, \vec{V}_T, \vec{a}_T, \beta$
Overlap Test
Input:
$\vec{U}_0, \vec{T}_0, \vec{V}_U, \vec{V}_T, \vec{a}_U, \vec{a}_T, \Delta t_{0f}$
N
Y

Acceleration Command


$\Delta t_{12} = 0.0001$
$\Delta a_U = 1$
$t_1 = t_2 - \Delta t_{12}$
$\vec{a}_U' = \vec{a}_U + \Delta a_U,$ (New acceleration vector)
$\Delta t_{01} = t_1 - t_0$
$\vec{U}_1 = \vec{U}_0 + (\vec{V}_U \times \Delta t_{01}) + \frac{1}{2}\vec{a}_U(\Delta t_{01})^2$
$\vec{T}_1 = \vec{T}_0 + (\vec{V}_T \times \Delta t_{01}) + \frac{1}{2}\vec{a}_T(\Delta t_{01})^2$
$\Delta a_U = \Delta a_U + 1$
N
Y
$t_1 \geq t_{min}$
$\Delta t_{12} = \Delta t_{12} + 0.0001$
$t_1 = t_2 - \Delta t_{12}$
Y
Overlap Test
Input:
$\vec{U}_1, \vec{T}_1, \vec{V}_U, \vec{V}_T, \vec{a}_U', \vec{a}_T, \Delta t_{12}$
N

Deceleration Command


$\vec{a}_U' > \vec{a}_{U(max)}$
N
Y
$\Delta t_{12} = 0.0001$
$\Delta a_U = -1$
$t_1 = t_2 - \Delta t_{12}$
$\vec{a}_U' = \vec{a}_U + \Delta a_U$, (New acceleration vector)
$\Delta t_{01} = t_1 - t_0$
$\vec{U}_1 = \vec{U}_0 + (\vec{V}_U \times \Delta t_{01}) + \frac{1}{2}\vec{a}_U(\Delta t_{01})^2$
$\vec{T}_1 = \vec{T}_0 + (\vec{V}_T \times \Delta t_{01}) + \frac{1}{2}\vec{a}_T(\Delta t_{01})^2$
$\Delta a_U = \Delta a_U - 1$
N
Y
$t_1 \geq t_{min}$
$\Delta t_{12} = \Delta t_{12} + 0.0001$
$t_1 = t_2 - \Delta t_{12}$
Y
Overlap Test
Input:
$\vec{U}_1, \vec{T}_1, \vec{V}_U, \vec{V}_T, \vec{a}_U', \vec{a}_T, \Delta t_{12}$
N

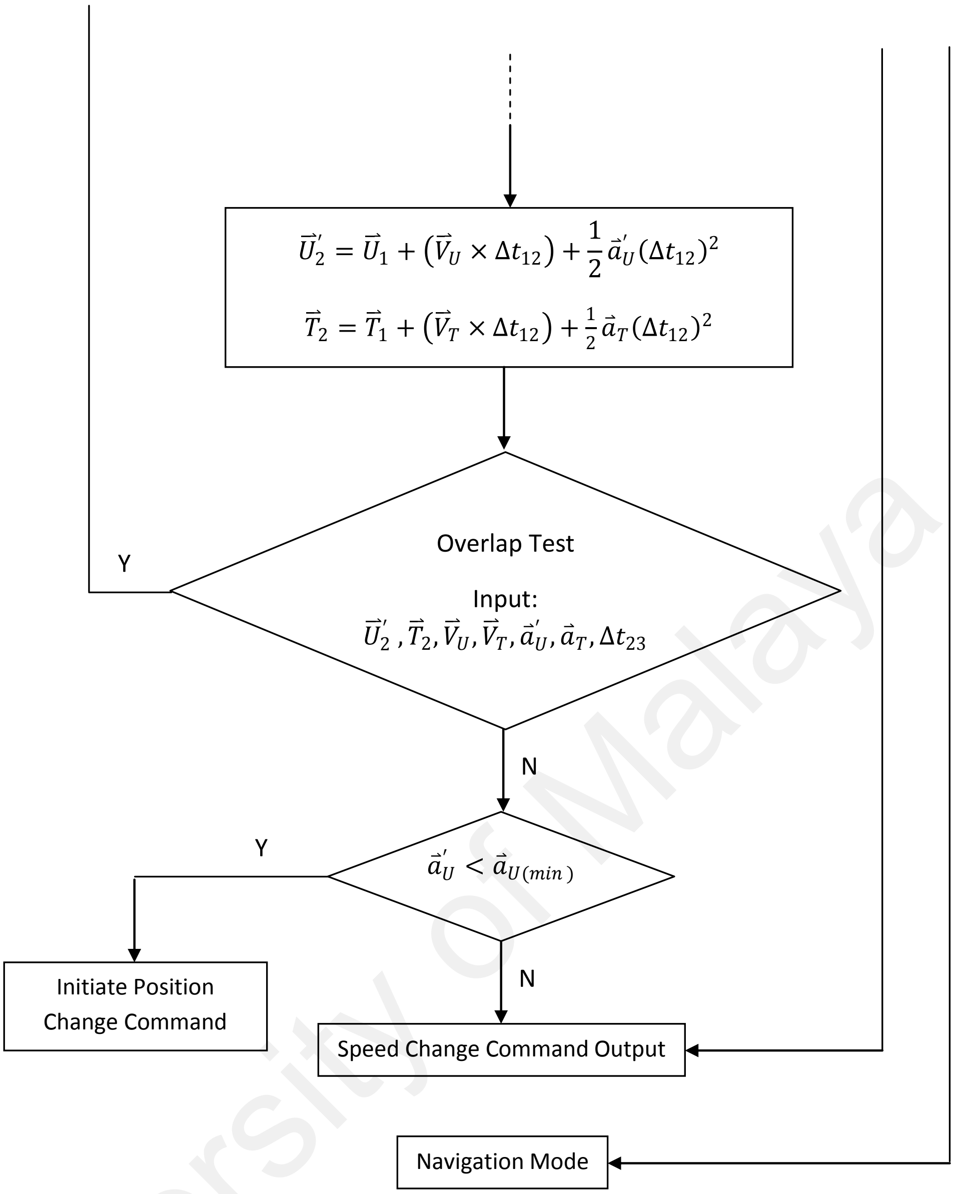

$\vec{U}_2' = \vec{U}_1 + (\vec{V}_U \times \Delta t_{12}) + \frac{1}{2}\vec{a}_U'(\Delta t_{12})^2$
$\vec{T}_2 = \vec{T}_1 + (\vec{V}_T \times \Delta t_{12}) + \frac{1}{2}\vec{a}_T(\Delta t_{12})^2$
Overlap Test
Input:
$\vec{U}_2', \vec{T}_2, \vec{V}_U, \vec{V}_T, \vec{a}_U', \vec{a}_T, \Delta t_{23}$
Y
N
$\vec{a}_U' < \vec{a}_{U(min)}$
Y
N
Initiate Position Change Command
Speed Change Command Output
Navigation Mode

Figure 4.18: Flow chart for collision avoidance algorithm (Speed Change Command)

Speeding up the velocity will be examined first for collision avoidance. If it is not suitable to choose, then slowing down the velocity will be examined. However, if both of acceleration and deceleration of UAV's speed are not satisfied with the limitation of minimum and maximum velocity for UAV, then collision avoidance algorithm for position change command needs to be initiated.

CHAPTER 5

Simulation Results

This chapter discusses the performance of collision avoidance algorithm using circle overlapping test approach. The simulation results of two avoidance commands are demonstrated in MATLAB for different cases of conflict scenarios. Conflict scenarios consist of several problem between cooperative UAV and target by different heading angle, size of protective zone (it is depending on the size of aircraft), and current velocity. This algorithm is able to describe the behaviour of the future events, given a specific set of input conditions, as well as predict the effects of changes in control system variables (e.g. heading angle or velocity).

5.1 Simulation Performance without CA Algorithm

Simulation is carried out to demonstrate performance of the proposed collision avoidance algorithm using circle overlapping circle approach. Both avoidance commands are examined in the MATLAB simulation. The aircrafts (UAV and target) are modelled as a particle in a two-dimensional plane. The simulation parameters for different conditions of conflict scenario are listed in Table 5.1 below. These conditions are used to examine the effectiveness of collision avoidance algorithm in position change command and speed change command. The simulation conditions between UAV and target are illustrated as follows:

1. Case 1 – Same velocity, same size and 90°collision scenario.
2. Case 2 – Same velocity, different size and 90°collision scenario.
3. Case 3 – Different velocity, same size and 90°collision scenario.
4. Case 4 – Different velocity, different size and 90°collision scenario.
5. Case 5 – Same velocity, same size and not 90°collision scenario.

Table 5.1: Simulation parameters for UAV and target

CASE	Initial Point (km)		Final Point (km)		Velocity (ms^{-1})		Radius (m)		Heading Angle, (°)	
	$\vec{U}_i$	$\vec{T}_i$	$\vec{U}_f$	$\vec{T}_f$	$\vec{V}_U$	$\vec{V}_T$	r_U	r_T	UAV	Target
1	$(-10,10)$	$(0,20)$	$(10,10)$	$(0,0)$	100	100	1000	1000	0°	-90°
2	$(-10,10)$	$(0,20)$	$(10,10)$	$(0,0)$	100	100	700	1000	0°	-90°
3	$(-10,10)$	$(0,20)$	$(10,10)$	$(0,0)$	100	80	1000	1000	0°	-90°
4	$(-10,10)$	$(0,20)$	$(10,10)$	$(0,0)$	100	80	700	1000	0°	-90°
5	$(-8,0)$	$(0,20)$	$(10,20)$	$(0,0)$	100	100	1000	1000	48°	-90°

Table 5.2: Collision in-near-future data timing

		Case 1	**Case 2**	**Case 3**	**Case 4**	**Case 5**
Δt_1 (sec)		85.86	87.98	100	104.51	104.76
Δt_2 (sec)		114.14	112.02	119.51	115	125.94
Position at Δt_1, (km)	**UAV**	$(-1.41,10)$	$(-1.20,10)$	$(0,10)$	$(0.45,10)$	$(-0.99,7.78)$
	Target	$(0,11.41)$	$(0,11.20)$	$(0,12)$	$(0,11.64)$	$(0,9.52)$
Position at Δt_2, (km)	**UAV**	$(1.41,10)$	$(1.20,10)$	$(1.95,10)$	$(1.5,10)$	$(0.42,9.36)$
	Target	$(0,8.58)$	$(0,8.79)$	$(0,10.44)$	$(0,10.8)$	$(0,7.41)$

Future collision scenario for each case is analysed to see the collision in-near-future without collision avoidance inside the UAV system. Table 5.2 represents the summary of collision in-near-future data for all cases. Meanwhile, Figures 5.1-5.5 illustrate the future collision scenario for all cases.

The variable Δt_1 and Δt_2 represent the duration time of both circle to start and finish overlapping after the obstacle is detected by UAV's sensor, respectively as shown in Figure 3.7. The value of t_a and t_b can be obtained in (3.21) and (3.22), respectively. Let the line of minimum separation represents the total radius of SBC for both UAV and target.

For case 1, at time $t = 85.86sec$ and $t = 114.1sec$, the separation distance between UAV and target is equal to the minimum separation line as shown in Figure 5.1(a). It can be explained by looking at point 1 and point 2 on Figure 5.1(b), respectively. At that time, the separation distance is equal to their total radius of SBC. As a conclusion, the collision time range for case 1 can be determined by $85.86sec \leq t_c \leq 114.1sec$ after the target is detected by UAV's sensor, where $\Delta t_1 = 85.86sec$ and $\Delta t_2 = 114.1sec$.

The explanation for case 2, 3, 4 and 5 are similar to case 1 except the value of their collision time range are different depending on their total radius of SBC. It can be concluded as follows for case 2, 3, 4 and 5:

- Collision time range for case 2 (shown in Figure 5.2): $87.98sec \leq t_c \leq 112sec$
- Collision time range for case 3 (shown in Figure 5.3): $100sec \leq t_c \leq 119.5sec$
- Collision time range for case 4 (shown in Figure 5.4): $104.5sec \leq t_c \leq 115sec$
- Collision time range for case 5 (shown in Figure 5.5): $104.76sec \leq t_c \leq 125.94sec$

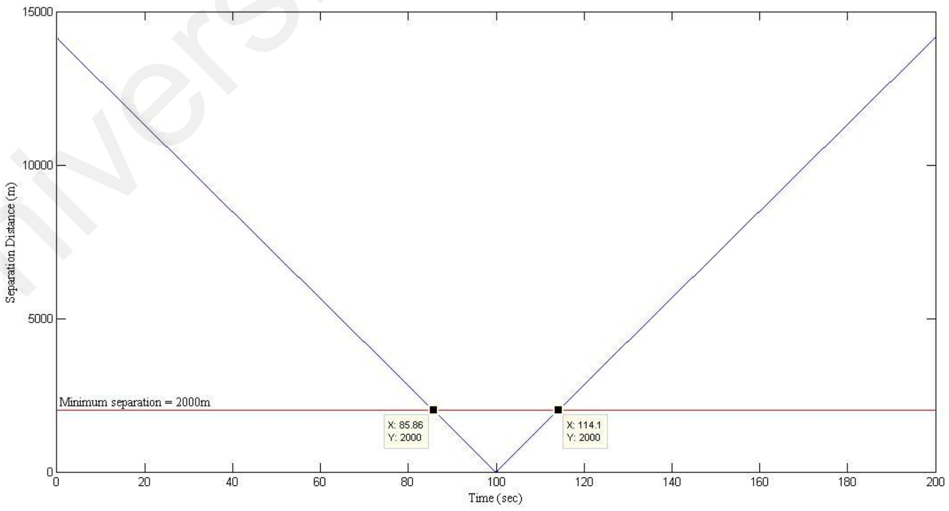


Figure 5.1(a): Separation distance of moving UAV and target for case 1

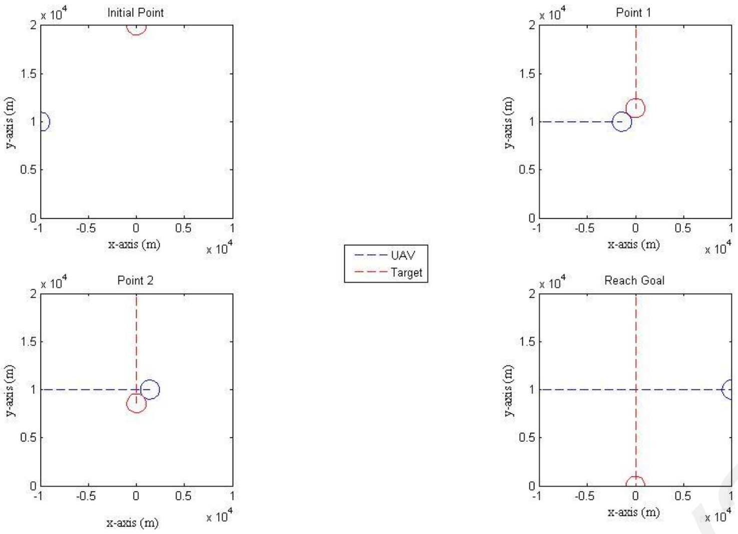


Figure 5.1(b): Collision maneuver realization process for case 1

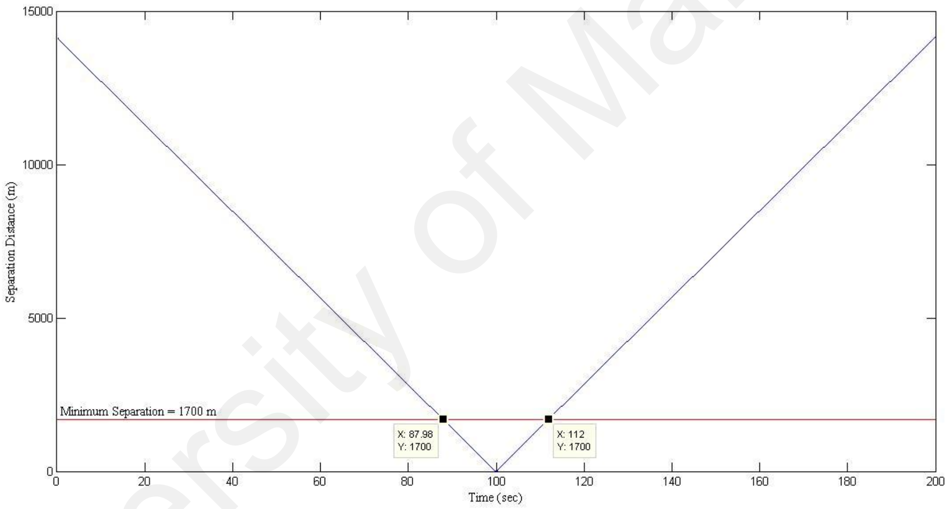


Figure 5.2(a): Separation distance of moving UAV and target for case 2

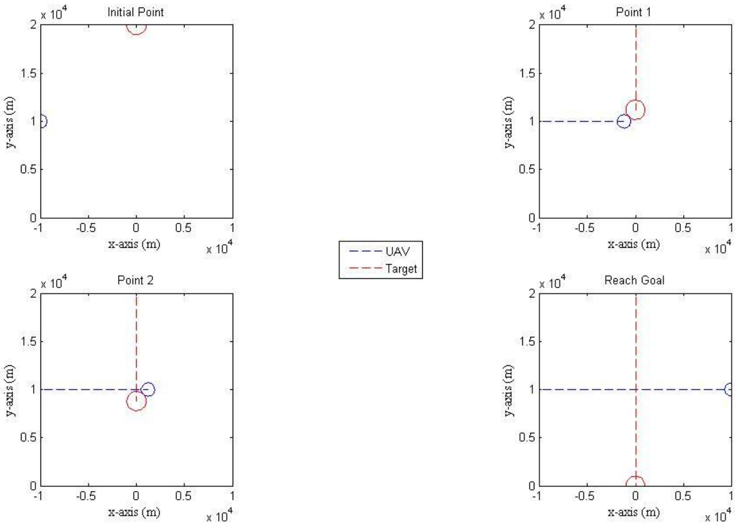


Figure 5.2(b): Collision maneuver realization process for case 2

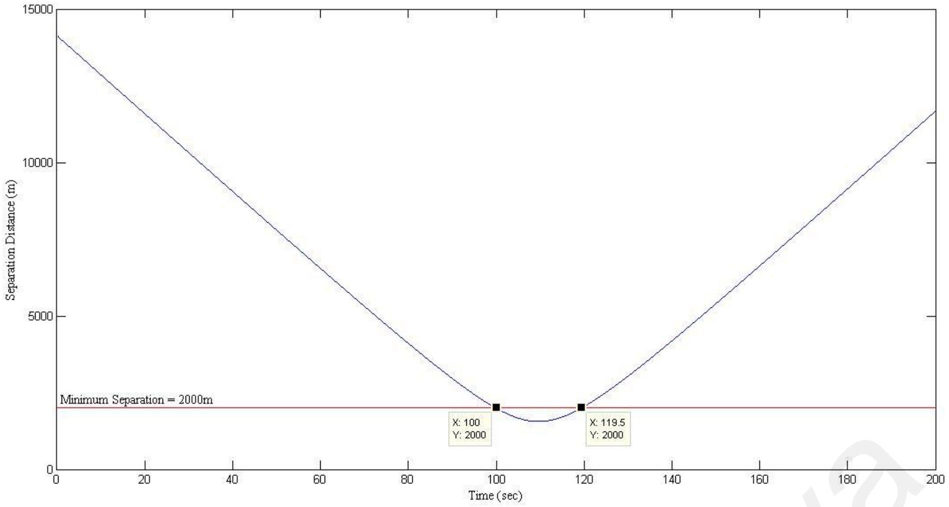


Figure 5.3(a): Separation distance of moving UAV and target for case 3

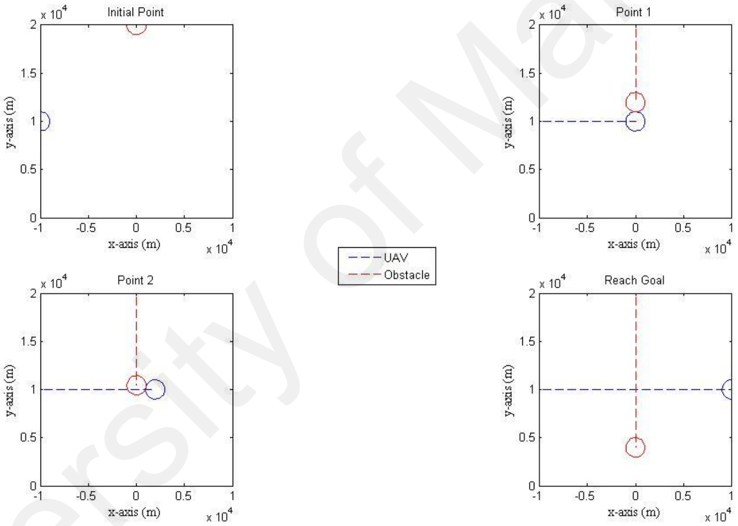


Figure 5.3(b): Collision maneuver realization process for case 3

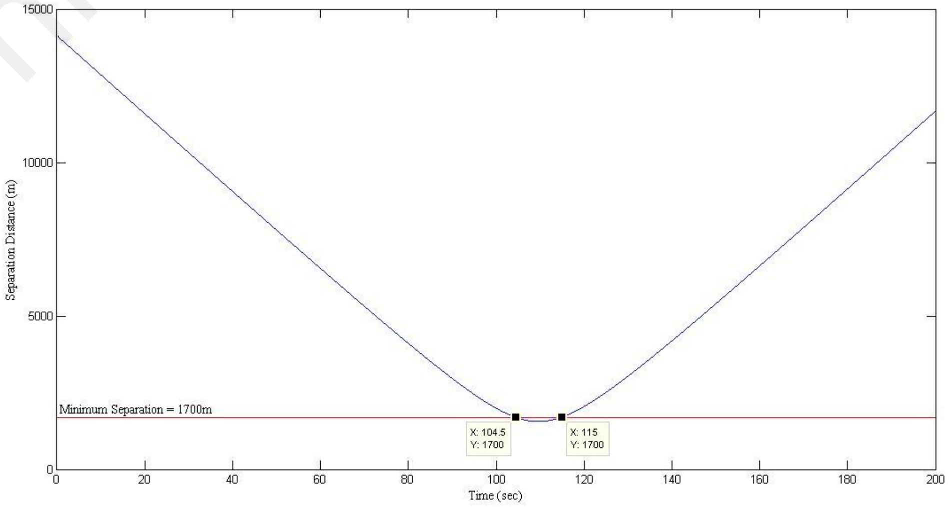


Figure 5.4(a): Separation distance of moving UAV and target for case 4

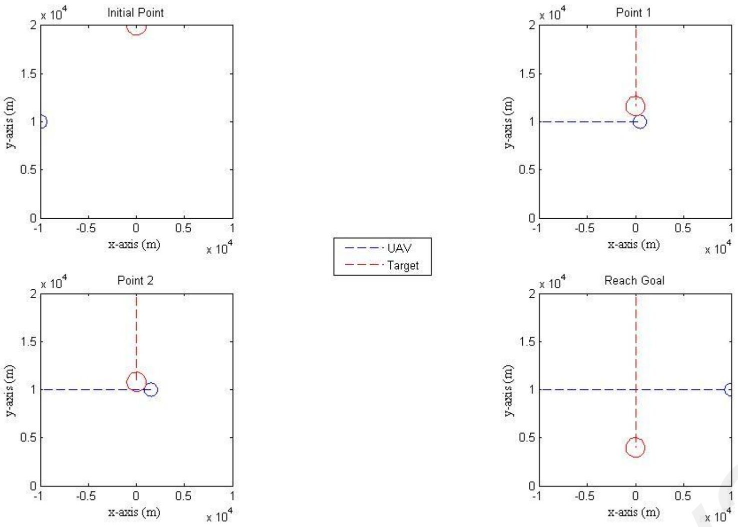


Figure 5.4(b): Collision maneuver realization process for case 4

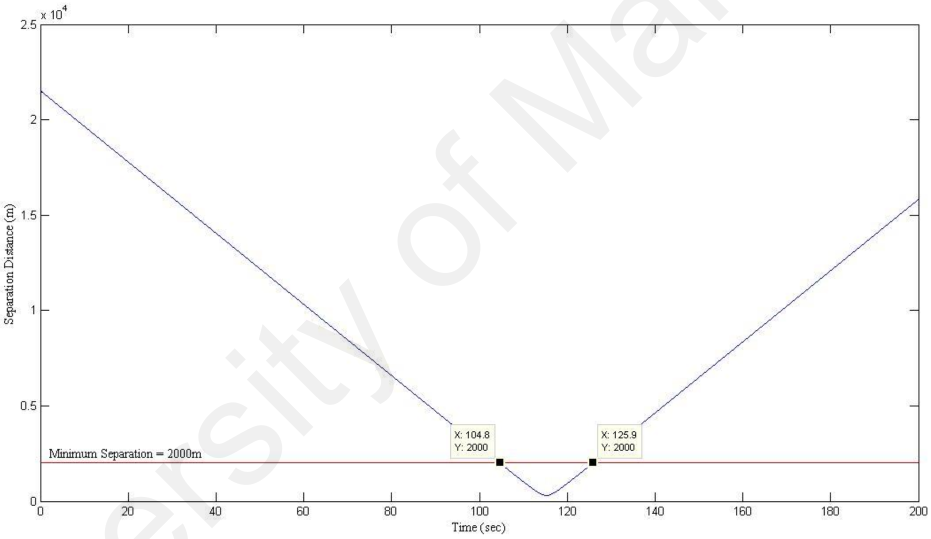


Figure 5.5(a): Separation distance of moving UAV and target for case 5

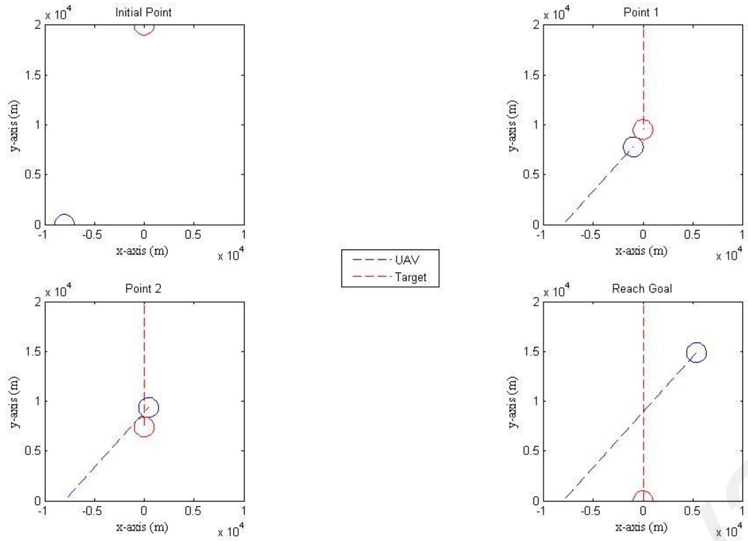


Figure 5.5(b): Collision maneuver realization process for case 5

5.2 With CA Algorithm: Position Change Command Results

The goal of the algorithm design is to guide a UAV to achieve the safe trajectory towards its destination. This command is the process of generating a safe trajectory from the current UAV location at t_i to a goal at t_f based on the current map (workspace). In this case, it is very important to have an accurate map and reliable localization from sensor readings.

The determination factor of the turning direction of UAV trajectory is investigated. The target direction at time t_i determines the value of the changes of heading angle, $\Delta\theta$ for collision avoidance. Analysis on collision avoidance scenario with different avoidance heading angles for case 1 is illustrated in Figure 5.6.

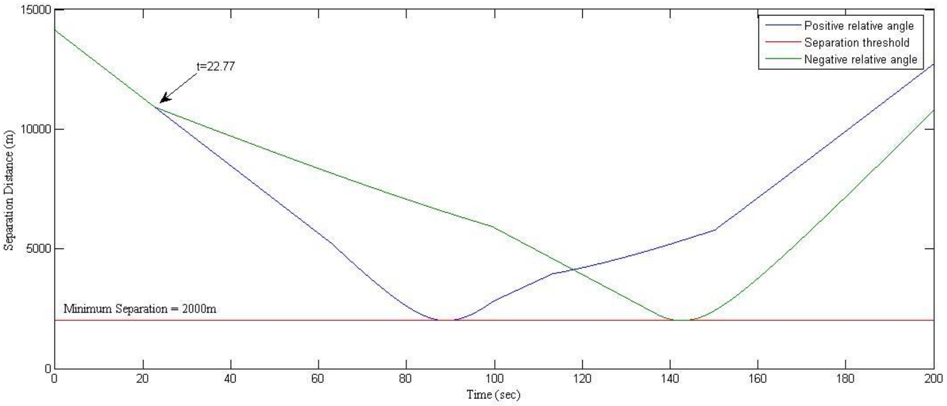


Figure 5.6: Separation distance at different changes in heading angle

The duration time of UAV to avoid the collision is longer if the avoidance heading angle is negative value for case 1 problem. Furthermore, UAV starts to turn its direction at time $t = 22.77\,sec$ earlier than the other one. Therefore, the avoidance heading angle is determined by the initial relative angle between UAV and target as explained in Figure 4.2.

5.2.1 Avoidance Trajectories Path Planning

After the collision time range is determined as discussed in previous chapter 3, the CA algorithm starts to initiate. UAV will turn its direction or path referring to the new path given by CAS system. A new path is formed by defining the control points (U_1, U_2', U_3', U_4') as explained in Figure 4.3. The path in form of a series of linear segments connected by control points is described as follows:

1. Initial Point – Target is detected by the UAV's sensors and collision avoidance algorithm starts to initiate.
2. Point 1 – The first line segment of $U_0 \rightarrow U_1$
3. Point 2 – The second line segment of $U_1 \rightarrow U_2$
4. Point 3 – The third line segment of $U_2 \rightarrow U_3$

5. Point 4 – The fourth line segment of $U_3 \rightarrow U_4$
6. Reach Goal – UAV returns to its normal path.

The performance of simulation results for each different initial condition as given in Table 5.1 are presented in Figures 5.7 – 5.11. Table 5.3 represents the summary of the coordinate of each control points for every case at their specific time after applying collision avoidance algorithm as explained in Figure 4.3.

Table 5.3: Control points data at specific time

	Case 1	**Case 2**	**Case 3**	**Case 4**	**Case 5**
$\boldsymbol{t_1}$ **(sec)**	63.04	68.58	49.61	54.91	82.59
$\boldsymbol{U_1}$ $(\boldsymbol{km})$	$(-3.69,10)$	$(-3.14,10)$	$(-5.04,10)$	$(-4.51,10)$	$(-2.47,6.14)$
$\boldsymbol{t_2}$ **(sec)**	100	100	100	104.51	104.76
$\boldsymbol{U'_2}$ $(\boldsymbol{km})$	$(-1.08,12.61)$	$(-0.92,12.22)$	$(-1.48,13.57)$	$(-1.01,13.51)$	$(-2.59,8.35)$
$\boldsymbol{t_3}$ **(sec)**	113.37	111.36	129.52	133.57	117.75
$\boldsymbol{U'_3}$ $(\boldsymbol{km})$	$(0.25,12.61)$	$(0.22,12.22)$	$(1.48,13.57)$	$(1.90,13.51)$	$(-1.72,9.32)$
$\boldsymbol{t_4}$ **(sec)**	150.33	142.78	179.91	183.17	139.92
$\boldsymbol{U'_4}$ $(\boldsymbol{km})$	$(2.87,10)$	$(2.44,10)$	$(5.04,10)$	$(5.41,10)$	$(0.49,9.44)$

It was found that the separation distance for all cases are located at the above of minimum separation line as shown in Figure 5.7(a)-5.11(a) after the CA algorithm is applied. Meanwhile, Figure 5.7(b)-5.11(b) show the movement of UAV at different time as demonstrated in Figure 4.3: t_0 (*Initial Point*), t_1 (*Point 1*), t_2 (*Point 2*), t_3 (*Point 3*), t_4 (*Point 4*), t_f (*Reach Point*).

For case 1 (see Figure 5.7(b)), at time $t_1 = 63.04sec$ (*Point 1*), UAV starts to turn its direction (by new heading angle,α') to the left since the obstacle is detected on the left side. It can be proven by looking at *Point 2*, where UAV turns its direction during time t_1 to t_2 ($63.04sec \leq t \leq 100sec$) to avoid the collision with obstacle. After

UAV pass the overlapping period, at time t_2, it will change its direction by normal heading angle until time $t_3 = 113.37sec$ as shown in *Point 3*. After that, once again it changes its direction (by the second new heading angle,α'') until time $t_4 = 150.33sec$ so that it can return back to its original path as shown in *Point 4* and *Point 5*.

The discussion for case 2, 3, 4 and 5 are similar to case 1 except the value of their position changing time,(t_1, t_2, t_3, and t_4) and coordinate are different as given in Table 5.3. Found that the starting point of UAV, (point U_1) to turn its direction and time t_1 for every case are different even though their initial position, $\vec{U_i}$ during obstacle detected are located at same coordinate, $(-10,10)$ at same time t_i as shown in Table 5.1. It is because of different radius of SBC and different velocity between UAV and target. The bigger the size of target (increasing in radius of SBC), gives the value of t_1 for UAV to turn its direction increasing. The faster the velocity of the UAV, gives the value of t_1 for UAV to turn its direction increasing. It can be explained in Figure 5.12b. This algorithm also can be used for different angle of collision scenario as explained in case 5.

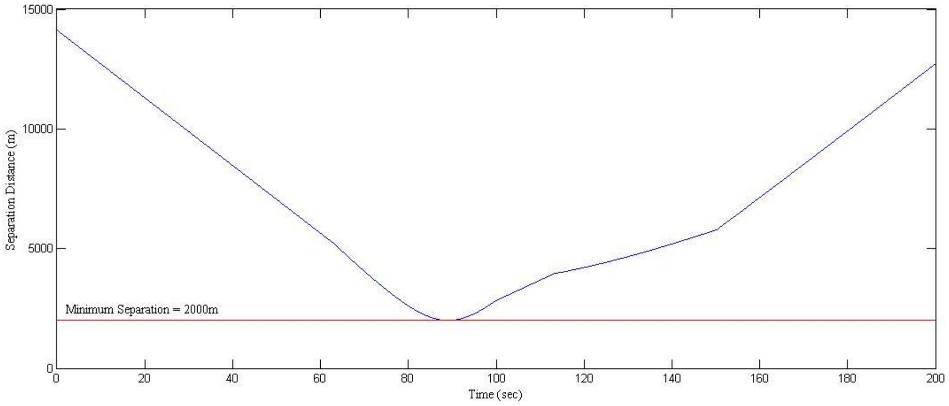


Figure 5.7(a): Avoidance separation distance of moving UAV and target for case 1

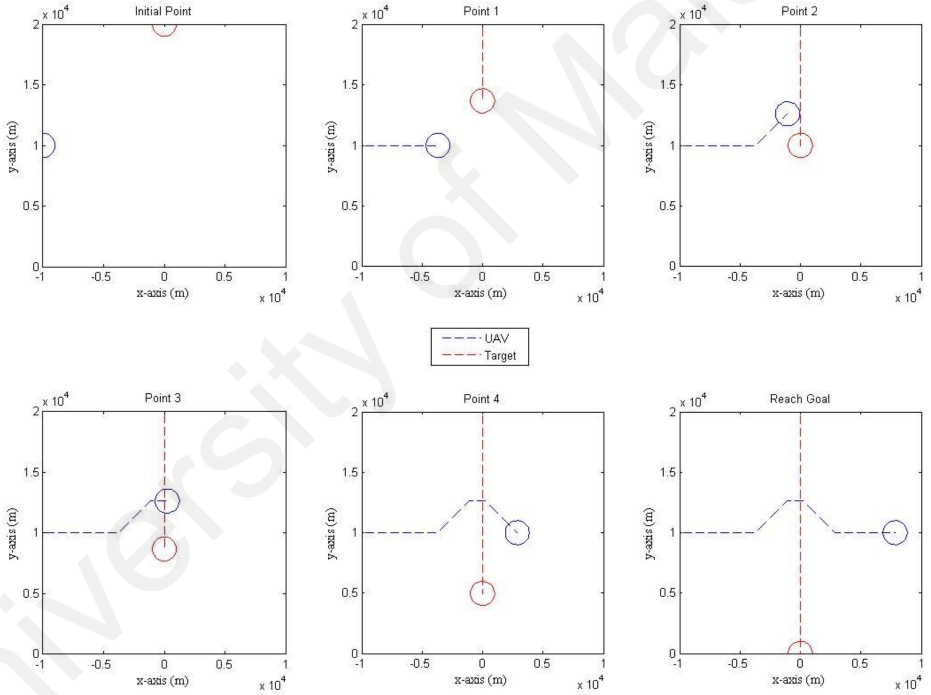


Figure 5.7(b): Avoidance maneuver realization process for case 1

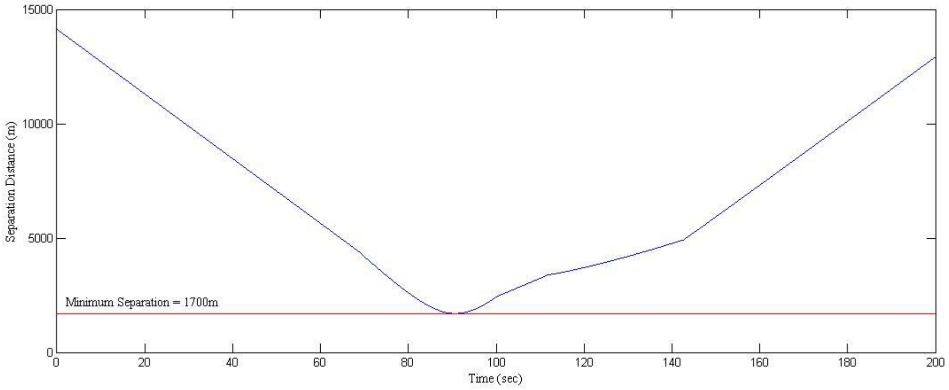


Figure 5.8(a): Avoidance separation distance of moving UAV and target for case 2

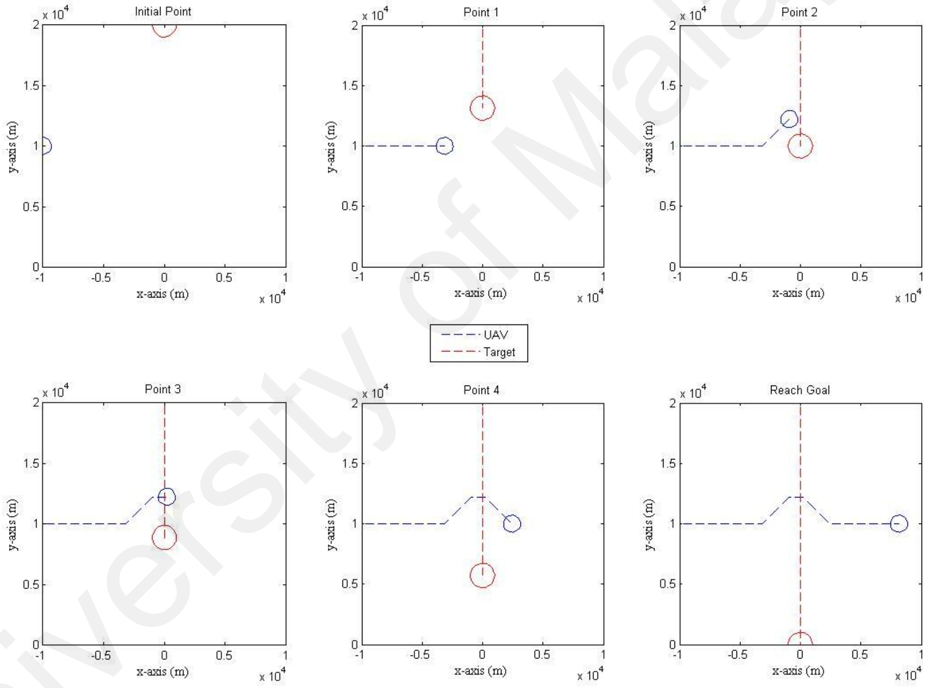


Figure 5.8(b): Avoidance maneuver realization process for case 2

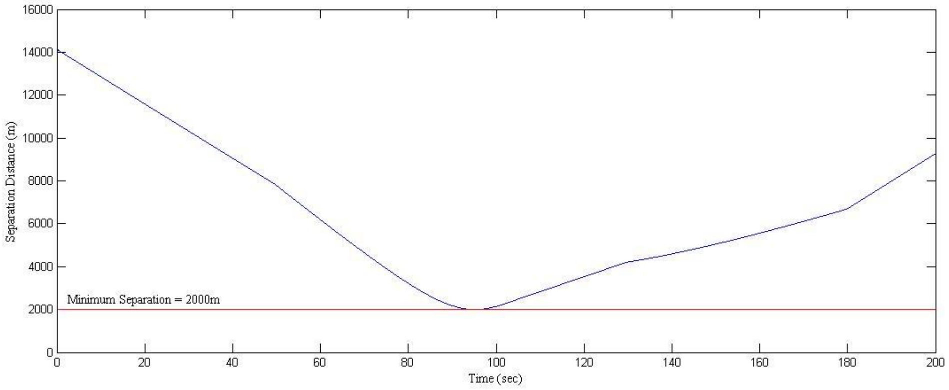


Figure 5.9(a): Avoidance separation distance of moving UAV and target for case 3

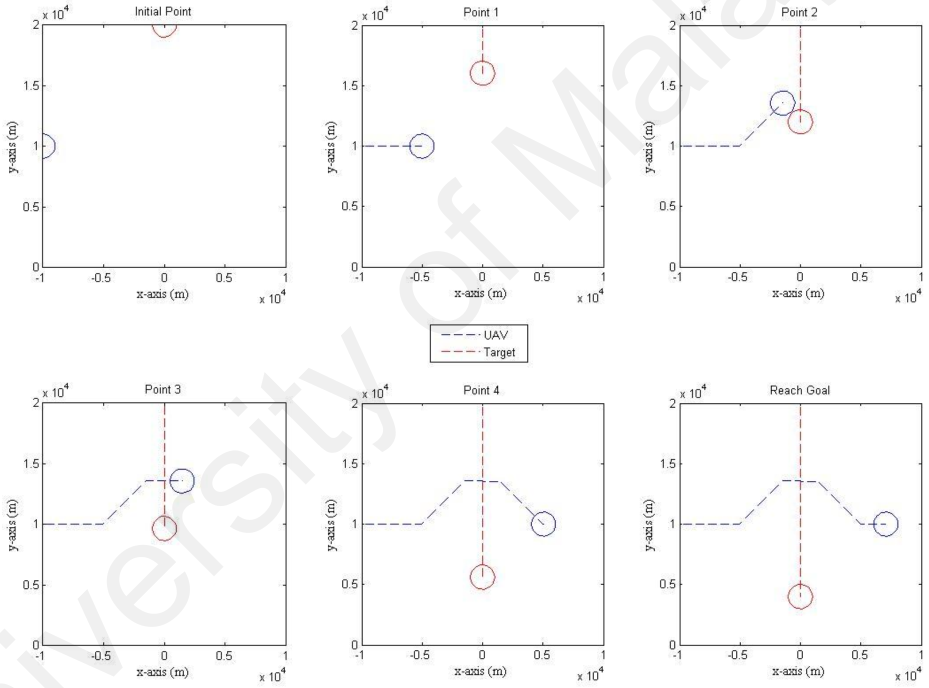


Figure 5.9(b): Avoidance maneuver realization process for case 3

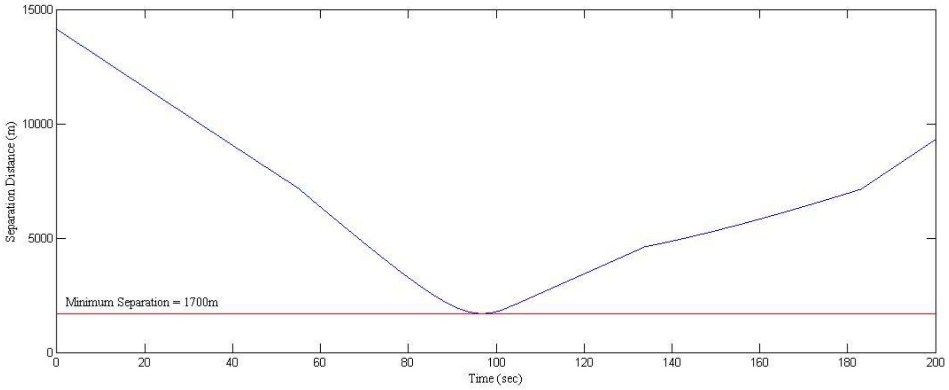


Figure 5.10(a): Avoidance separation distance of moving UAV and target for case 4

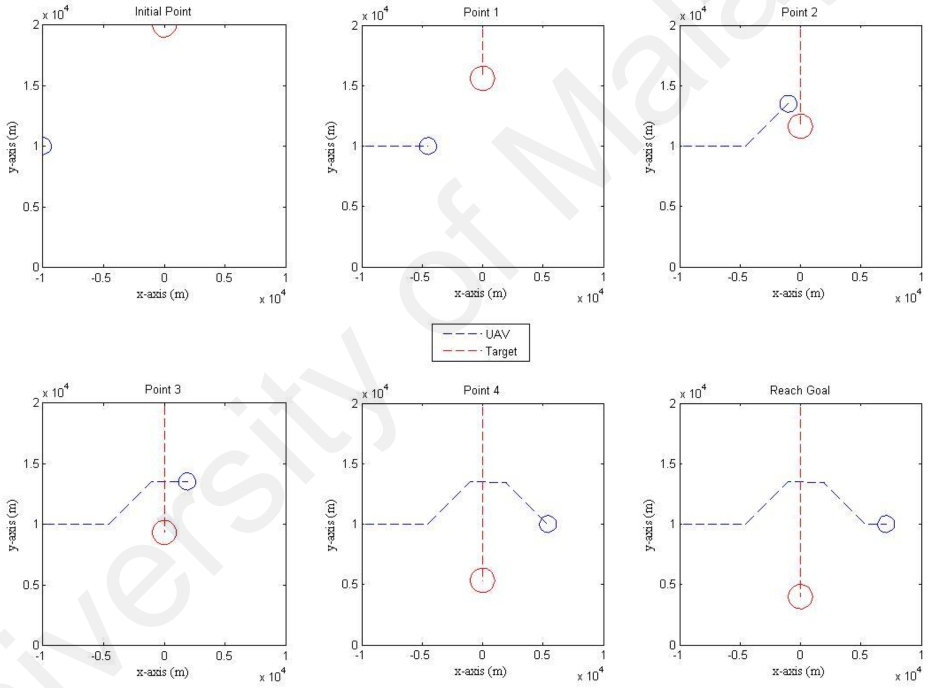


Figure 5.10(b): Avoidance maneuver realization process for case 4

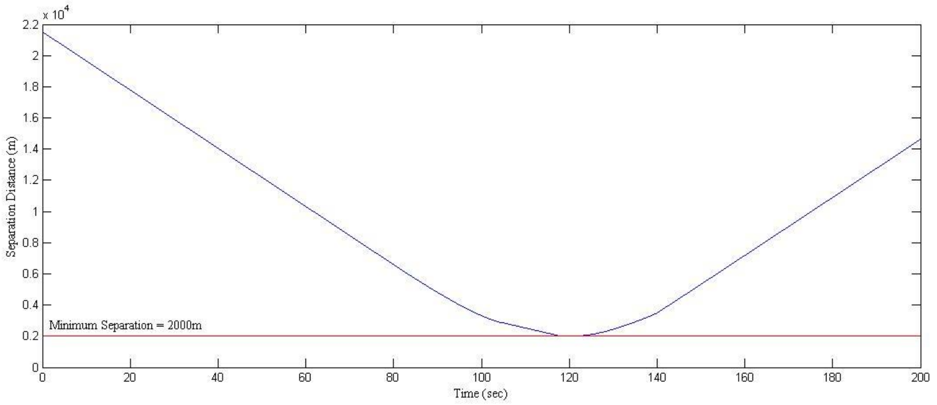


Figure 5.11(a): Avoidance separation distance of moving UAV and target for case 5

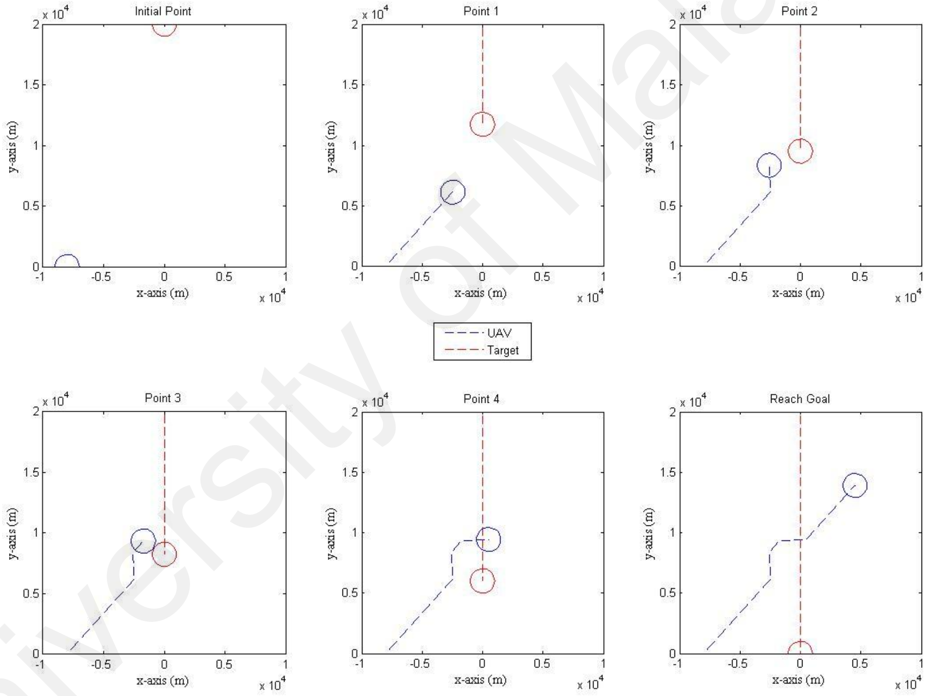
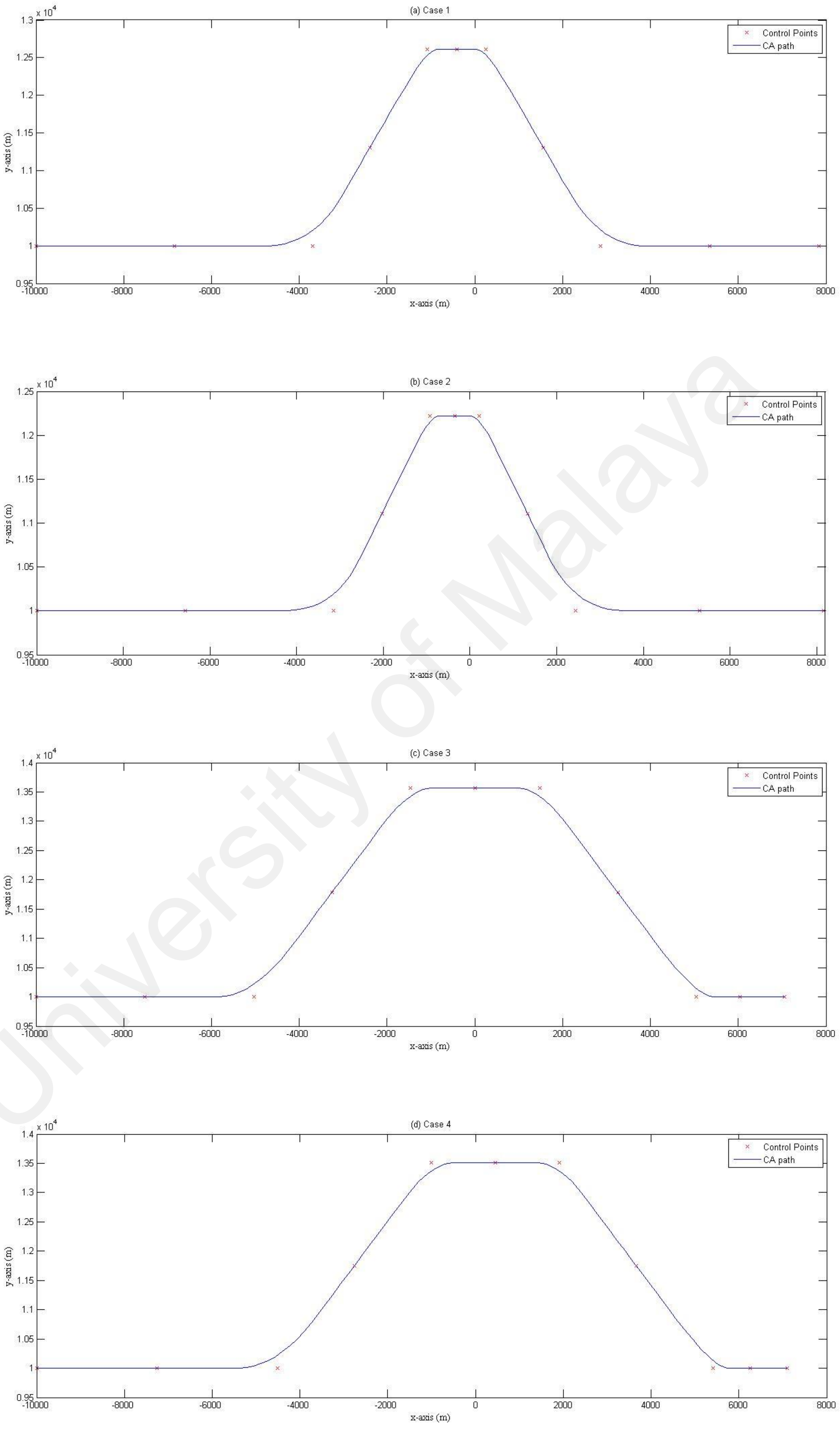


Figure 5.11(b): Avoidance maneuver realization process for case 5

5.2.2 Smooth Continuous Avoidance Path

The performances of smoothing collision avoidance path using B-splines curve for each cases is shown in Figures 5.12a and for all case 1-4 are shown in Figure 5.12b. The blue line shows the smooth avoidance maneuver for UAV after the position change command is activated and all control points are generated.


(a) Case 1
(b) Case 2
(c) Case 3
(d) Case 4
x 10^4
y-axis (m)
x-axis (m)
Control Points
CA path
1.3
1.25
1.2
1.15
1.1
1.05
1
0.95
1.4
1.35
-10000
-8000
-6000
-4000
-2000
0
2000
4000
6000
8000

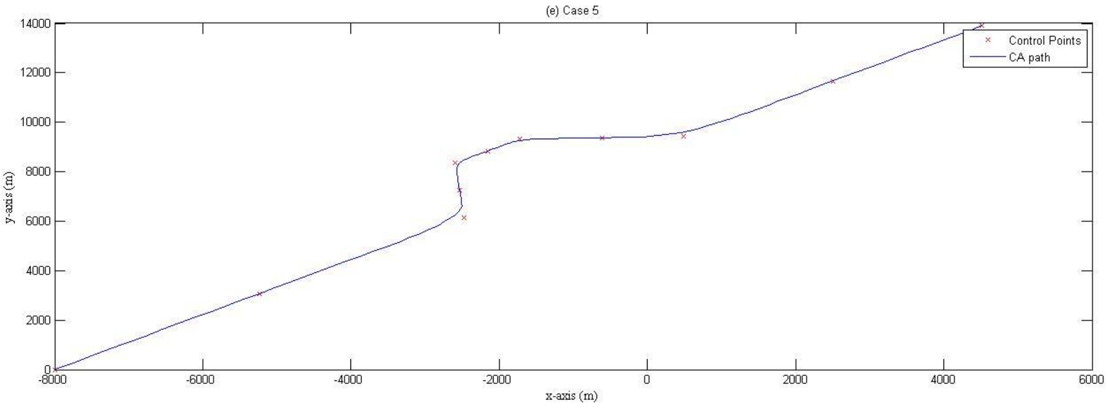


Figure 5.12a: The smooth continuous avoidance path UAV for:
(a) Case 1, (b) Case 2, (c) Case 3, (d) Case 4 and (e) Case5

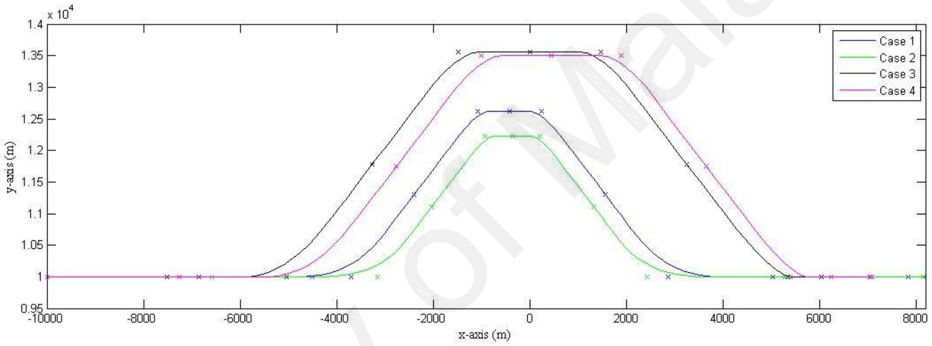


Figure 5.12b: The smooth continuous avoidance path UAV for all cases

5.3 With CA Algorithm: Speed Change Command Results

In this section, the simulation results for both speed command (acceleration and deceleration) are investigated.

5.3.1 Acceleration Command

Found that, the separation distance graphs for all cases are located above the minimum separation line after the CA algorithm is initiated as shown in Figures 5.13(a)-5.17(a). It can be proven by investigating the movement of UAV (as shown in Figures 5.13(b)-5.17(b)) at three different times: *Initial Point*, *Point 1* and *Point 2*. Therefore, the collision avoidance mission is achieved.

As discussed in previous section 5.1, Δt_1 represents the initial time of both circles start to overlap as given in Table 5.2. Before collision avoidance algorithm for speed command is initiated, the position of UAV and target at time Δt_1 is investigated as shown in figure at *Point 1* (as shown in Figures 5.1(b)-5.5(b)). It shows that collision starts to occur between UAV and target.

The collision avoidance is achieved for all cases after CA algorithm for speed command is initiated. It can be proven by looking at the position of UAV and target at time Δt_1 as shown in figure at *Point 2* (as shown in Figures 5.13(b)-5.17(b)). Meanwhile, Table 5.4 shows the minimum increment of UAV velocity to avoid the collision for all cases.

Table 5.4: New velocity for UAV

	$\boldsymbol{v_{old}}\ (m/s^{-1})$	$\boldsymbol{v_{new}}\ (m/s^{-1})$
Case 1	100	155.55
Case 2	100	150.94
Case 3	100	122.49
Case 4	100	111.78
Case 5	100	180.94

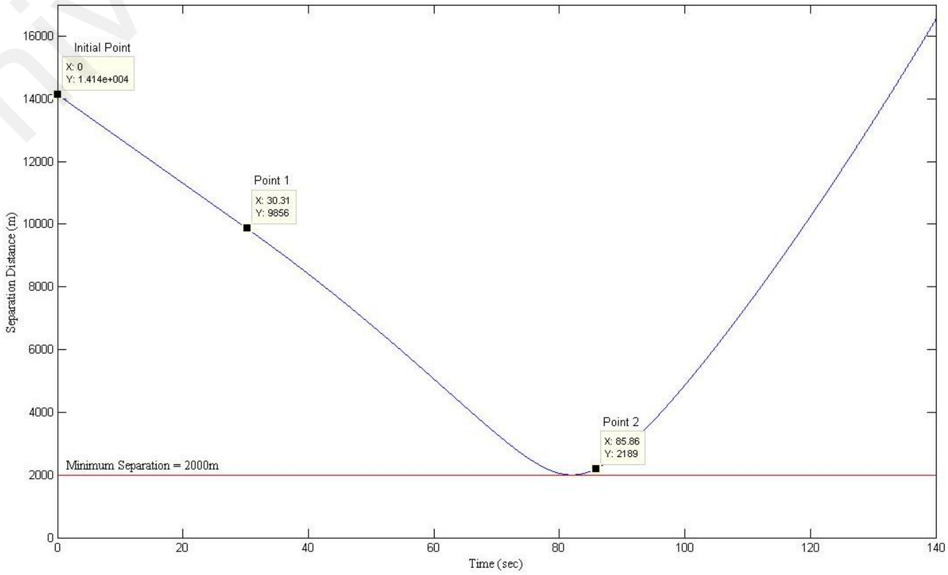


Figure 5.13(a): Avoidance separation distance of moving UAV and target for case 1

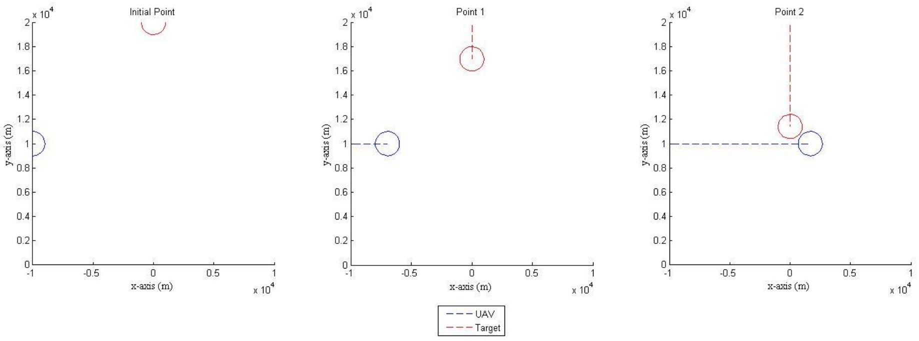


Figure 5.13(b): Avoidance maneuver realization process for case 1

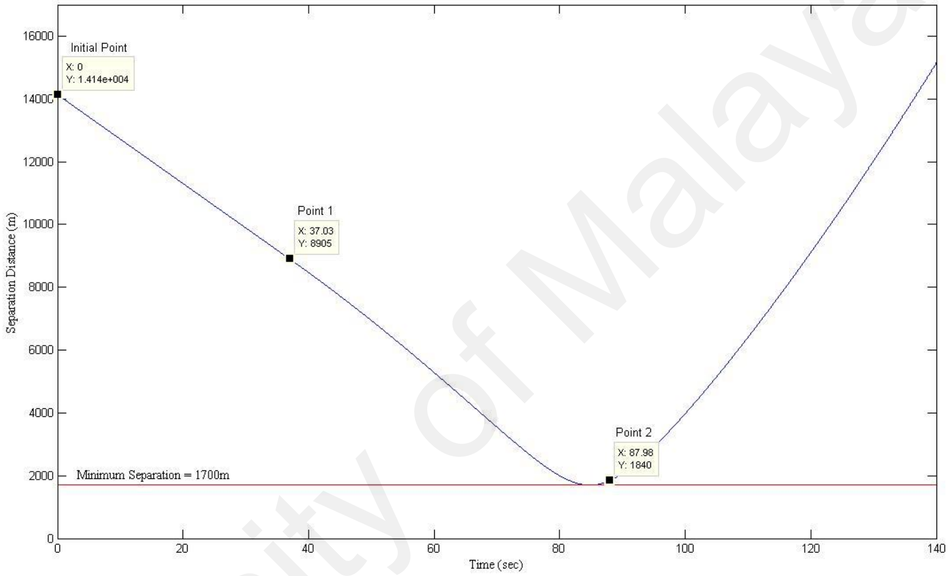


Figure 5.14(a): Avoidance separation distance of moving UAV and target for case 2

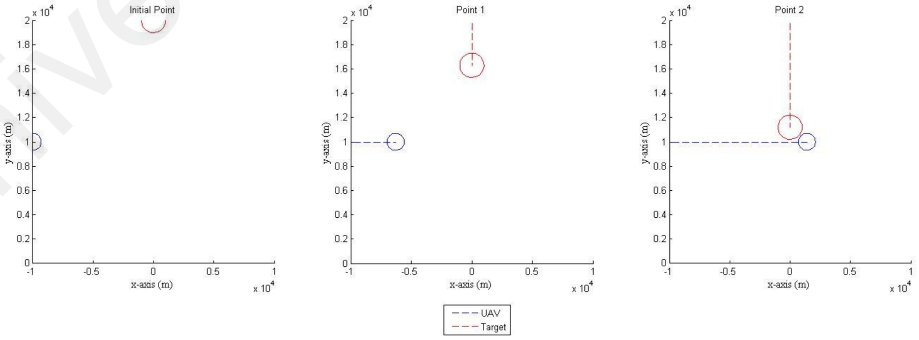


Figure 5.14(b): Avoidance maneuver realization process for case 2

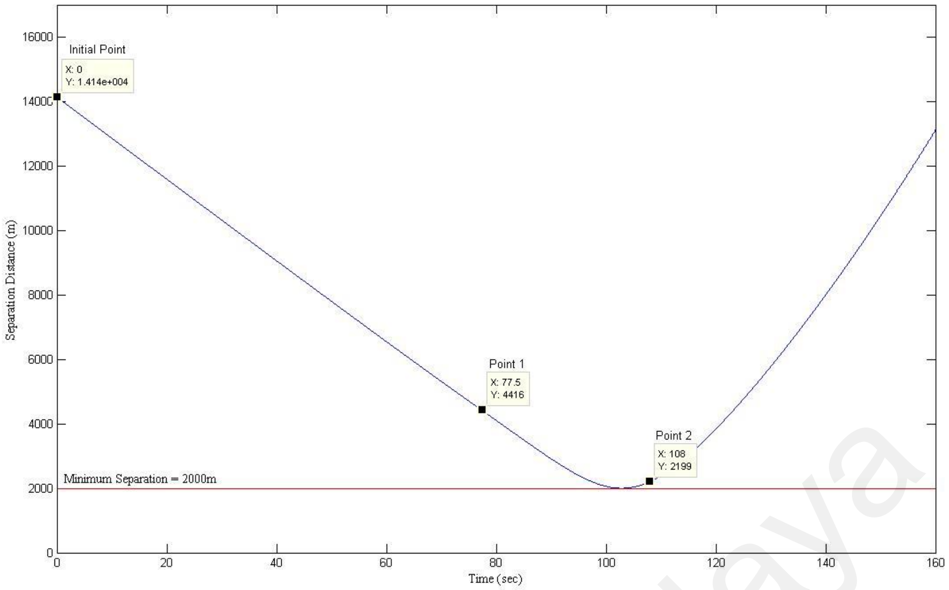


Figure 5.15(a): Avoidance separation distance of moving UAV and target for case 3

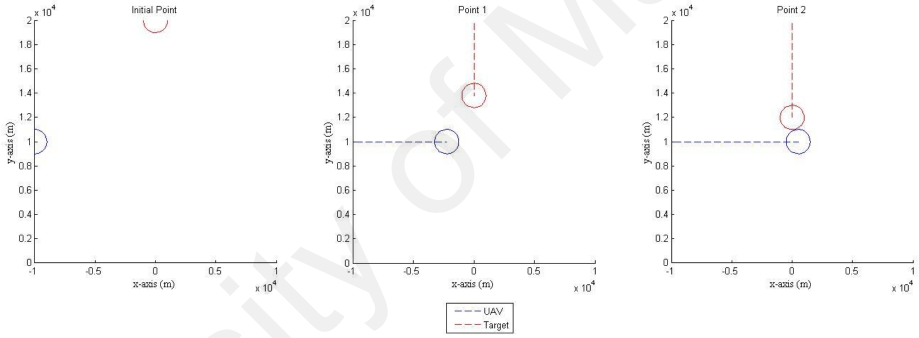


Figure 5.15(b): Avoidance maneuver realization process for case 3

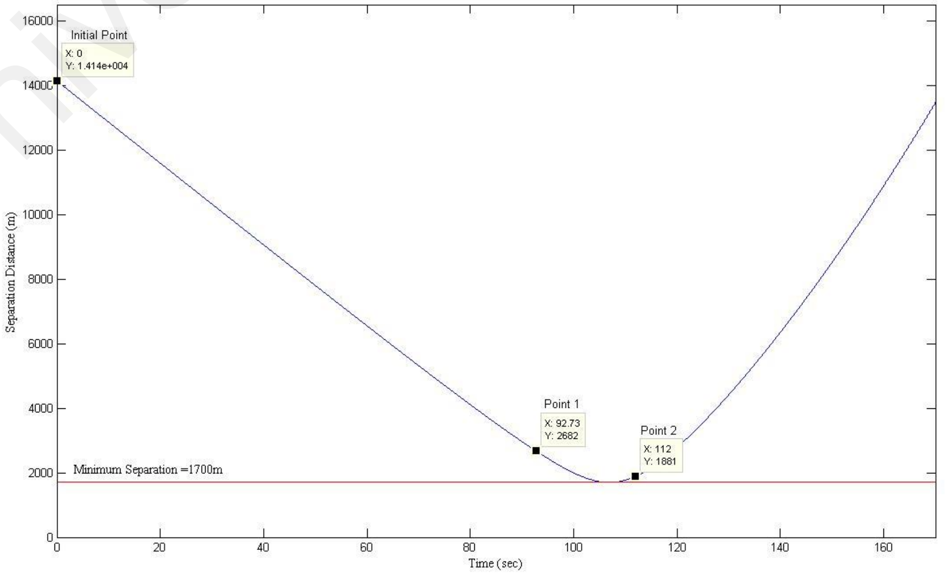


Figure 5.16(a): Avoidance separation distance of moving UAV and target for case 4

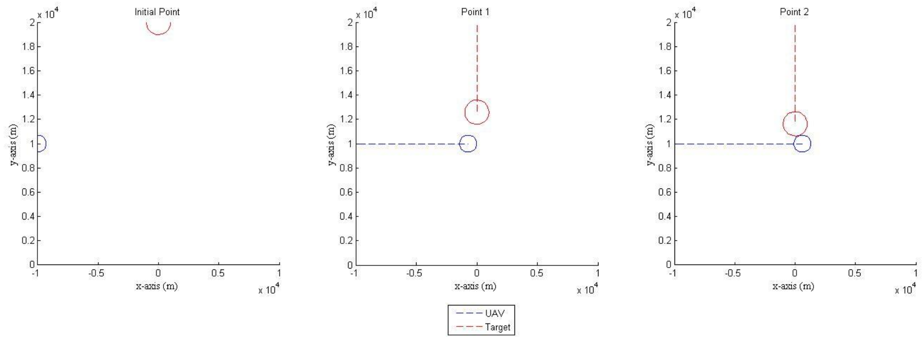


Figure 5.16(b): Avoidance maneuver realization process for case 4

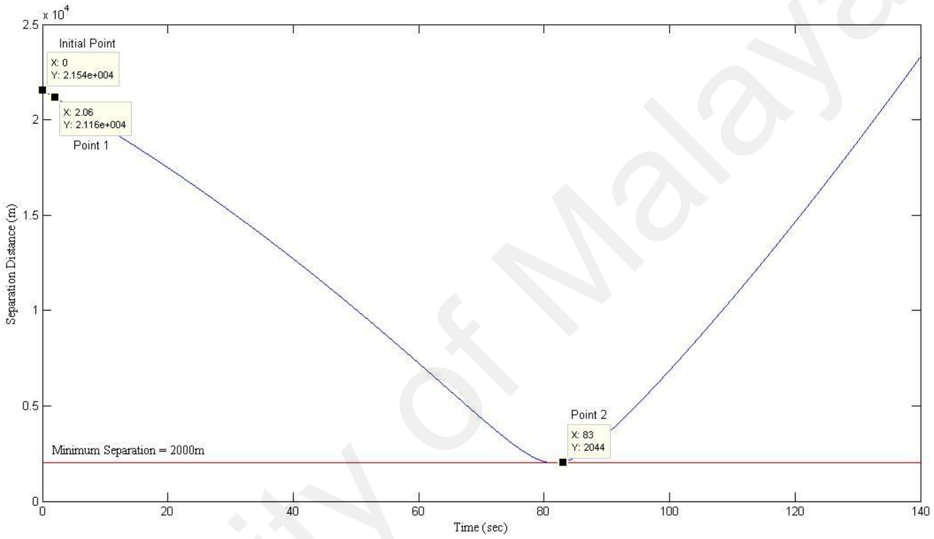


Figure 5.17(a): Avoidance separation distance of moving UAV and target for case 5

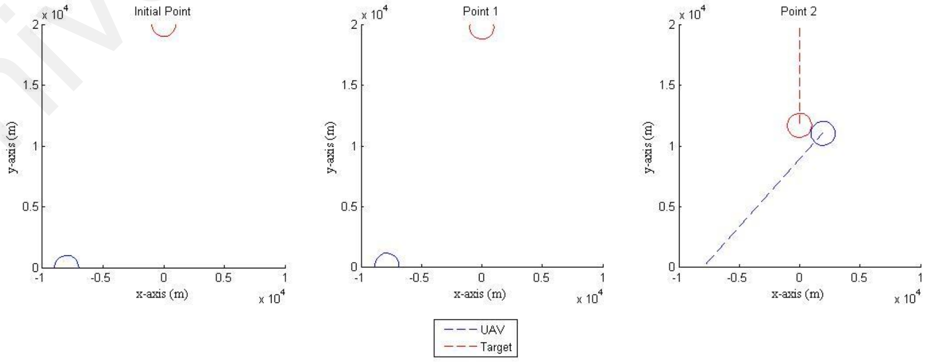


Figure 5.17(b): Avoidance maneuver realization process for case 5

5.3.2 Deceleration Command

Deceleration command is not suitable to apply into the problem cases given in Table 5.1. It is because of UAV's velocity given is not enough to perform the collision avoidance since the decreasing of UAV's velocity at per second. Therefore, the new simulation parameters with same condition as previous for UAV and target that suitable for deceleration command to be used is examined as described in Table 5.5.

Table 5.5: Simulation conditions for UAV and target

CASE	Initial Point (km)		Final Point (km)		Velocity (ms^{-1})		Radius (m)	
	$\vec{U}_i$	$\vec{T}_i$	$\vec{U}_f$	$\vec{T}_f$	$\vec{V}_U$	$\vec{V}_T$	r_U	r_T
1	$(-12,10)$	$(0,20)$	$(10,10)$	$(0,0)$	200	200	1000	1000
2	$(-12,10)$	$(0,20)$	$(10,10)$	$(0,0)$	200	200	700	1000
3	$(-12,10)$	$(0,20)$	$(10,10)$	$(0,0)$	200	150	1000	1000
4	$(-12,10)$	$(0,20)$	$(10,10)$	$(0,0)$	200	150	700	1000
5	$(-9.8,-2)$	$(0,20)$	$(10,20)$	$(0,0)$	200	200	1000	1000

Table 5.6: Collision in-near-future data timing

		Case 1	**Case 2**	**Case 3**	**Case 4**	**Case 5**
Δt_1 **(sec)**		50	51.66	55.07	56.40	60.28
Δt_2 **(sec)**		60	58.34	69.73	68.40	68.54
Position at Δt_1, (km)	**UAV**	$(-2.00,10)$	$(-1.67,10)$	$(-0.99,10)$	$(-0.72,10)$	$(-1.74,6.96)$
	Target	$(0,10)$	$(0,9.67)$	$(0,11.74)$	$(0,11.54)$	$(0,7.94)$
Position at Δt_2, (km)	**UAV**	$(0,10)$	$(-0.33,10)$	$(1.95,10)$	$(1.68,10)$	$(-0.64,8.19)$
	Target	$(0,8.00)$	$(0,8.33)$	$(0,9.54)$	$(0,9.74)$	$(-0,6.29)$

Table 5.6 and Figures 5.18(a)-5.22(a) represent the summary of collision in-near-future data for all cases. The variable Δt_1 and Δt_2 represent the duration time of both circle to start and finish overlapping after the target is detected by UAV's sensor, respectively as shown in Figure 3.7. Let the red line of minimum separation represents the total radius of SBC for both UAV and target. The separation distance graphs for all cases before CA algorithm for speed (decelerate) command is activated are shown in Figures 5.18(a)-5.22(a). The collision can be predicted by examined the graphs located below the red line. Besides, the collision time range in (3.23) is $\Delta t_1 \leq t_{collision} \leq \Delta t_2$.

The separation distance graphs for all cases as shown in Figures 5.18(b)-5.22(b) are located above the minimum separation line after the CA algorithm is initiated, which describe the collision avoidance is achieved as shown in Figures 5.18(c)-5.22(c). Discover that both circles will never be overlapped to each other for all cases since target is moving faster than UAV and cross the intersection point earlier. Meanwhile, Table 5.7 shows the minimum decrement of UAV velocity to avoid the collision for all cases.

Table 5.7: New velocity for UAV

	$\boldsymbol{v_{old}}\ (m/s^{-1})$	$\boldsymbol{v_{new}}\ (m/s^{-1})$
Case 1	200	167.70
Case 2	200	175.56
Case 3	200	126.94
Case 4	200	131.55
Case 5	200	152.08

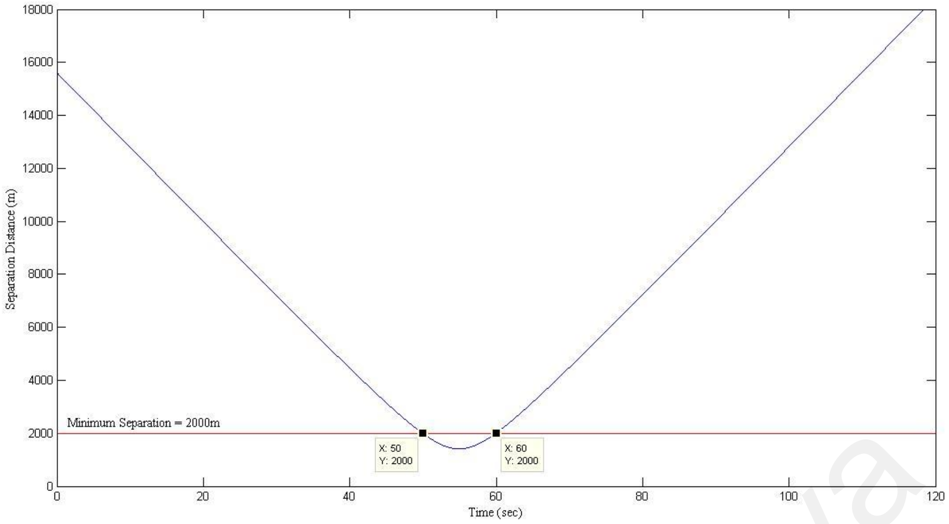


Figure 5.18(a): Separation distance before applying CA for case 1

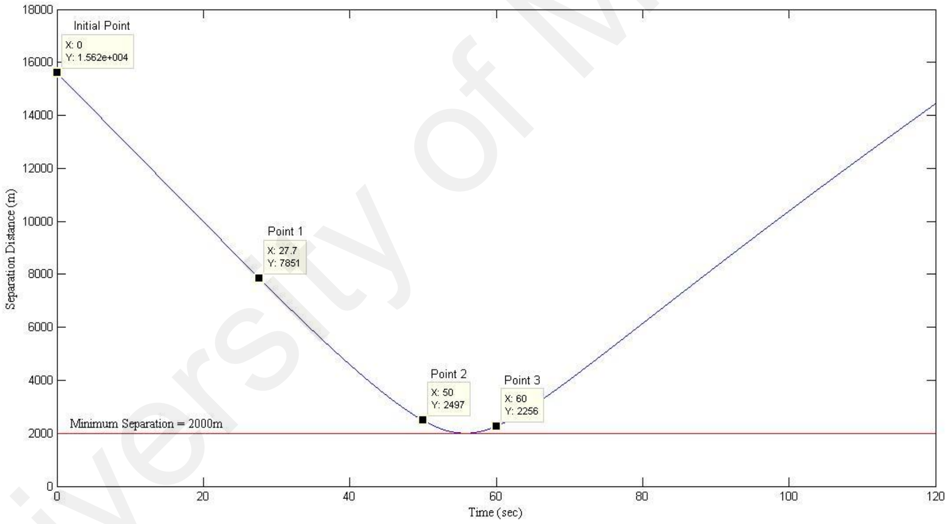


Figure 5.18(b): Separation distance after applying CA for case 1

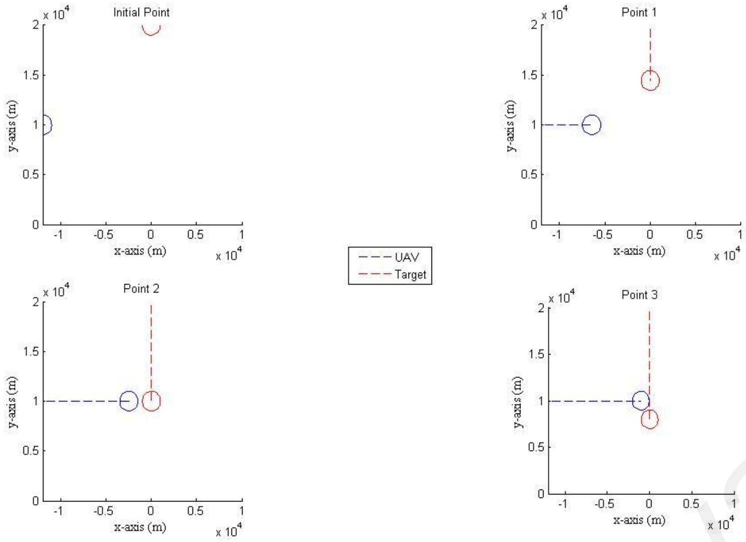


Figure 5.18(c): Maneuver realization after applying CA for case 1

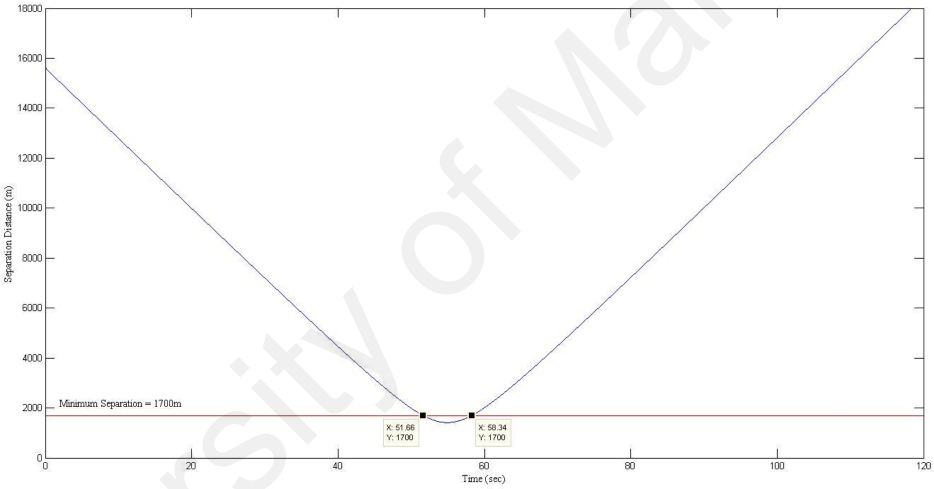


Figure 5.19(a): Separation distance before applying CA for case 2

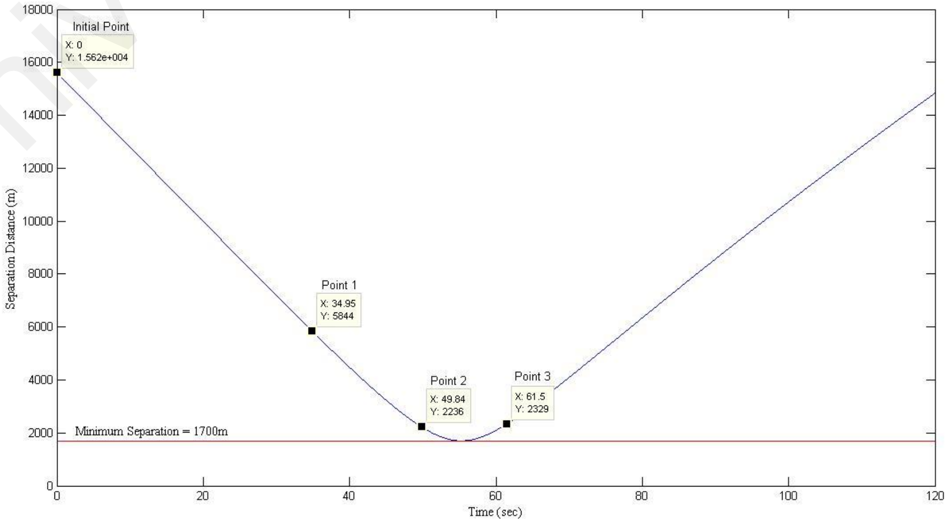


Figure 5.19(b): Separation distance after applying CA for case 2

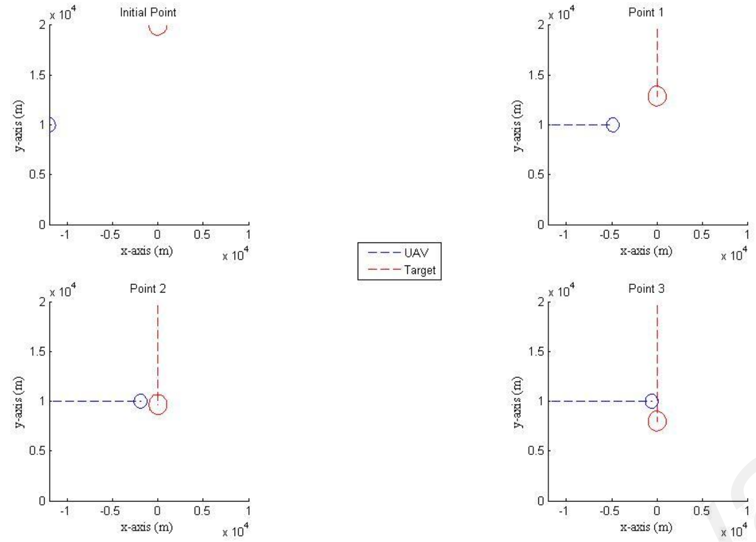


Figure 5.19(c): Maneuver realization after applying CA for case 2

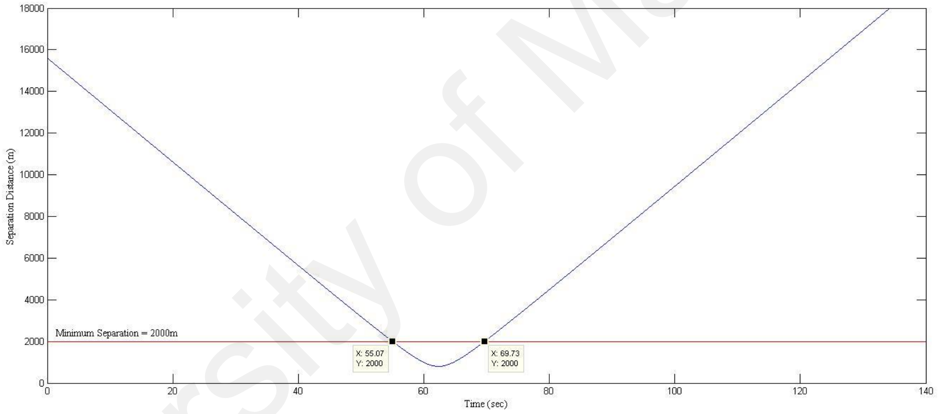


Figure 5.20(a): Separation distance before applying CA for case 3

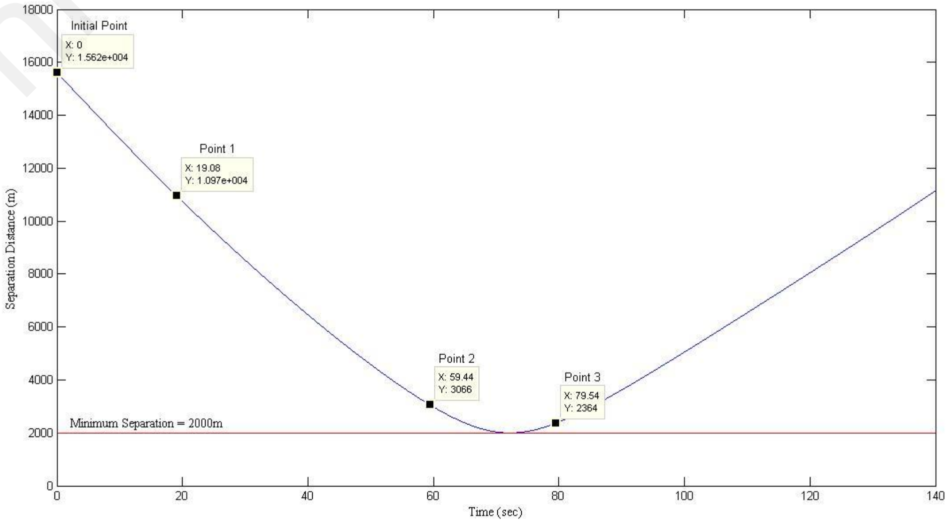


Figure 5.20(b): Separation distance after applying CA for case 3

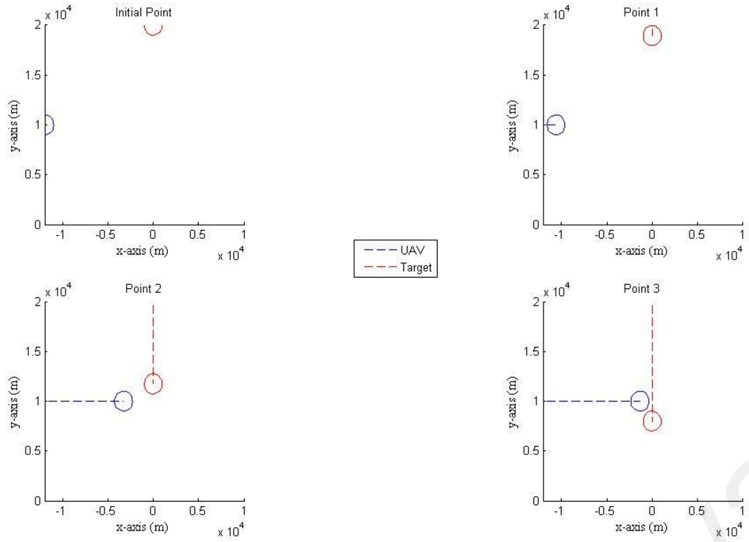


Figure 5.20(c): Maneuver realization after applying CA for case 3

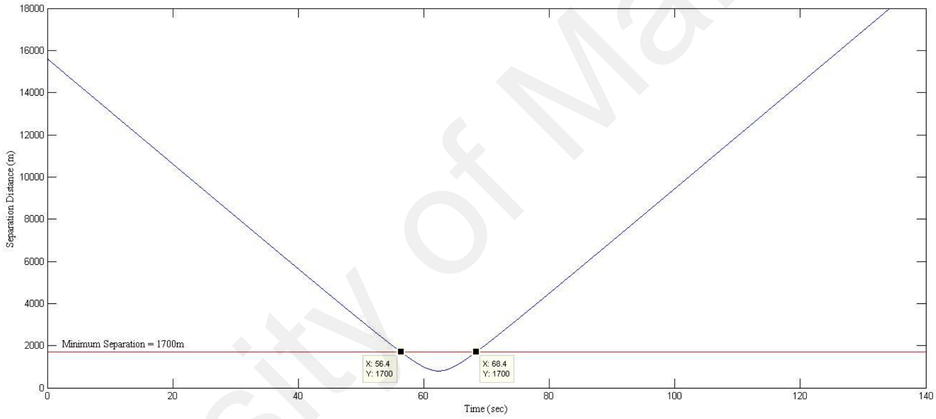


Figure 5.21(a): Separation distance before applying CA for case 4

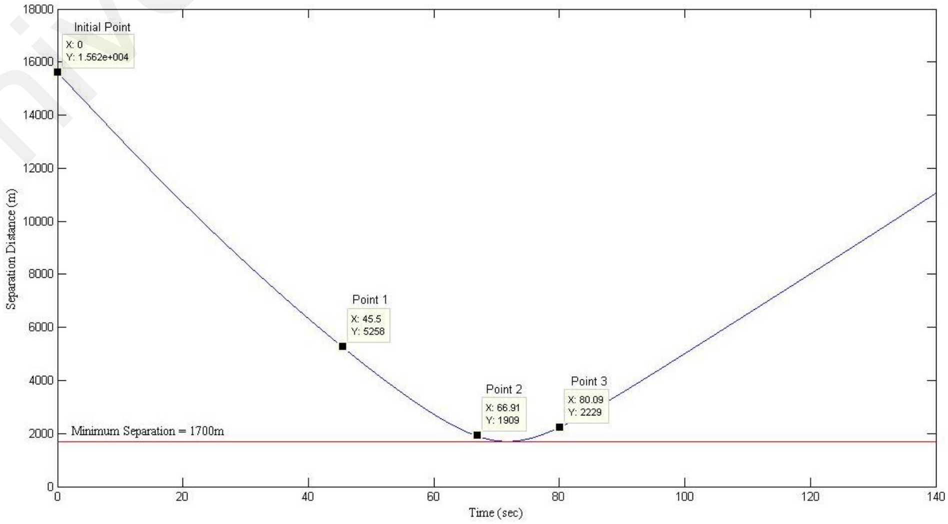


Figure 5.21(b): Separation distance after applying CA for case 4

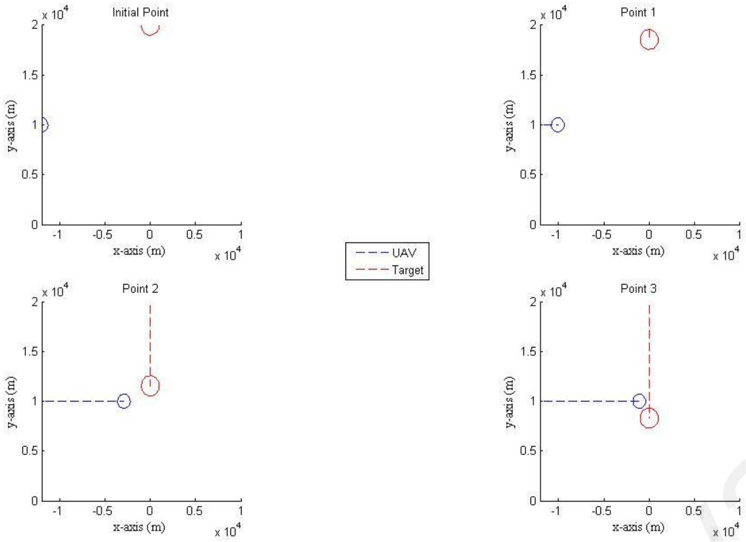


Figure 5.21(c): Maneuver realization after applying CA for case 4

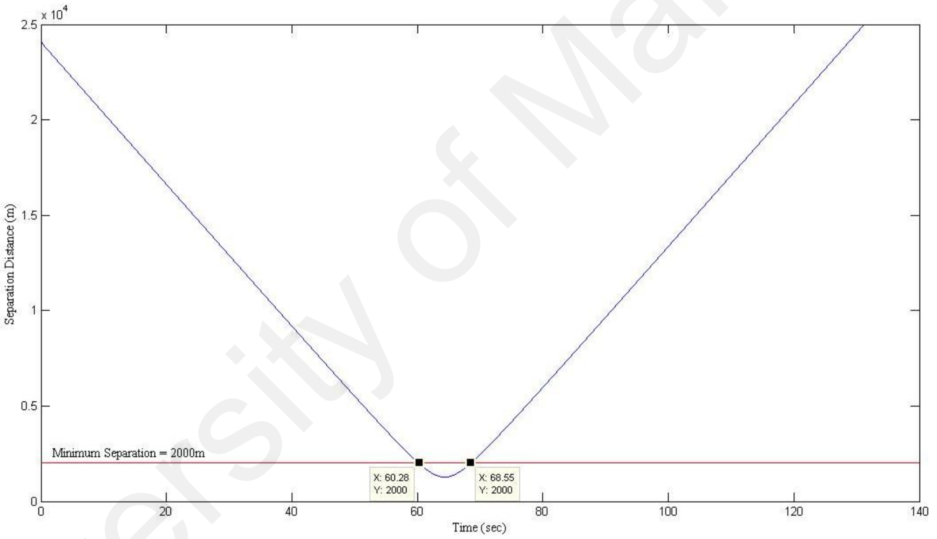


Figure 5.22(a): Separation distance before applying CA for case 5

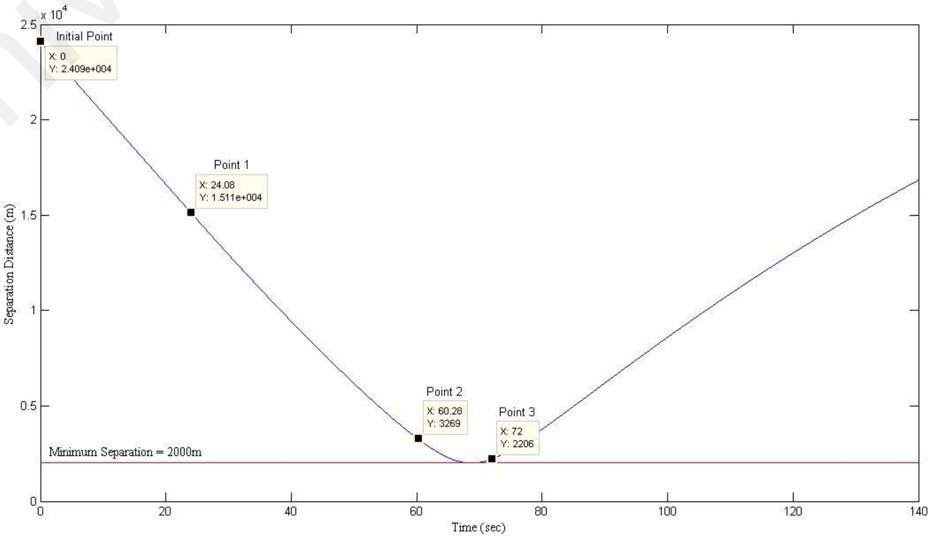


Figure 5.22(b): Separation distance after applying CA for case 5

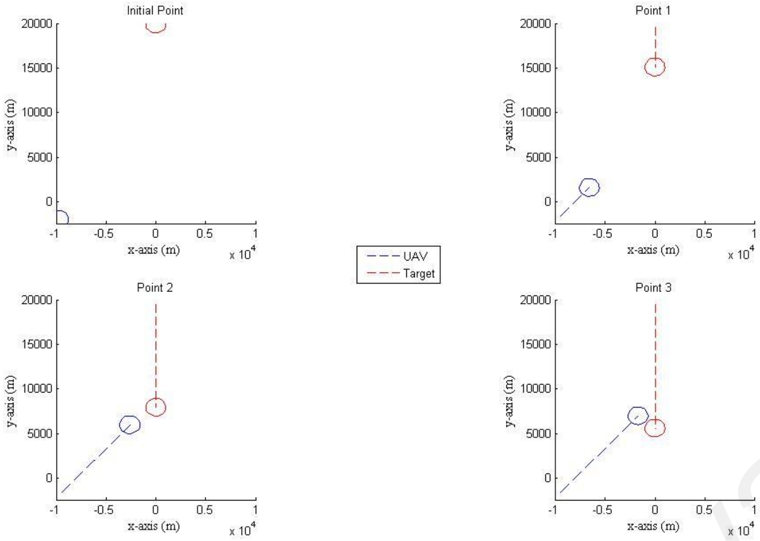


Figure 5.22(c): Maneuver realization after applying CA for case 5

CHAPTER 6

Conclusion

Chapter 6 provides a discussion of the research conclusion, improvement of algorithms development for future work, and the achievement or contribution of this research. The conclusions presented in this chapter summarize the research work on collision detection and collision avoidance algorithms.

6.1 Conclusion

This research contributes in the designing of two algorithms for cooperative UAV which are collision detection algorithm and collision avoidance algorithm. One is collision detection algorithm to detect the potential of collision in near future as described in Chapter 3. The investigation on parametric theorem and circle overlapping test (PTCOT) method for collision detection algorithm is proposed. Findings from the simulation result, the collision time range can be determined immediately once obstacle is detected. Using PTCOT equation, much information can be collected such as collision time range, position of collision in near future, the existence of interception point and separation distance graph. Besides, it also can be applied in many situations where UAV and target move in constant velocity or constant acceleration or constant deceleration. Unfortunately, it cannot be applied in changing acceleration/deceleration situation. Finally, the characteristic of collision scenario can be determined through the separation distance graph.

Second is collision avoidance algorithm design is presented in Chapter 4 based on the same method. Three avoidance commands (position change command, speed change command for acceleration, and speed change command for deceleration) are proposed in order to solve the predicted of collision in near future. This method is applicable for two cooperative UAVs and only for straight projection case. The controlling on flight dynamics and mechanism during changing the direction (autopilot control: roll, pitch, yaw and thrust) or speed (engine controls) are not discussed here because it need much time to develop a full system and do the testing. Therefore, only collision avoidance path for UAV is provided in this thesis.

For position change command, the proposed method generates the control points by which connections are made to determine the form of avoidance trajectories. By using B-spline method, a smooth continuous path for avoidance trajectory is discussed in section 5.2. Meanwhile, in speed change command for acceleration/deceleration provides a suitable value of increment/decrement for UAV to accelerate/decelerate and define the starting point for UAV to start speed changing. However, the new velocity after the increment/decrement must not exceed than the maximum/minimum velocity limit of the aircraft. Otherwise, this speed change command is not practical to be applied as collision avoidance.

The collision avoidance for speed change command can be investigated through investigating the separation distance before and after initiated the CA algorithm. From the simulation results, there is separation distance line located below the minimum separation line before CA algorithm for speed change command is initiated. However, after the algorithm is initiated, no separation distance line located below the minimum separation line. It shows that the separation distance between two moving aircrafts are not less than their minimum separation distance as explained in Figure 3.7. For acceleration case, when UAV's velocity is increasing, it causes UAV to fly over their interception point earlier than the target without any overlapping between both protection zone which explained in Figure 2.5. While for deceleration case, when

UAV's velocity is decreasing, it causes target to fly over their interception point earlier than the UAV without any overlapping between both protection zone.

As a conclusion, PTCOT method is a simple and easily to understand and apply in developing the collision detection and collision avoidance algorithm. The ability of this method to be used in many situations of UAV and target characteristics makes it more robust. In addition, it provides much information about collision in near future once the obstacle is detected by the sensors make it more suitable to be applied in designing CD algorithm. For CA algorithm design, it can be used in both command which is in position change command and speed change command (acceleration/deceleration). Besides, the development of collision detection and avoidance algorithm using PTCOT method for UAV is a new method in collision avoidance research.

6.2 Contribution

This research contributes a new approach for collision detection and avoidance algorithm design for UAV using the PTCOT method. In particular, the following list shows the contribution of this thesis:

- Two designs of collision detection algorithm and collision avoidance algorithm for both cooperative aircrafts are developed.
- Modification on avoidance trajectory is proposed to determine the smooth continuous avoidance path by using the B-spline method.
- Propose an avoidance trajectory and smooth continuous avoidance path to avoid the possible collision in near future between two aircrafts that flying in a share airspace.
- Present two choices of avoidance commands for collision avoidance: position change command and speed change command.

Most of the material presented in this text has been published in conferences or is currently under peer review for subsequent publication. A list of publications is presented in the following:

1- Radzi, N. F. M., Mubin, M., & Rahim, N. A. (2011). Collision Detection Algorithm for UAVs based on Parametric Theorem Approach. Paper presented at the IEEE International Conference on Computer and Communication Devices, Bali.

2- Radzi, N. F. M., Mubin, M., Rahim, N. A., Ouchi, S., Kodani, N., & Mokhtar, N. (2011). Collision Detection Algorithm using Parametric Theorem. Paper presented at the International Conference on Modeling and Simulation Technology, Tokyo, Japan.

3- Radzi, N. F. M., Mubin, M., Rahim, N. A., & Mokhtar, N. (2012). UAV Collision Detection Algorithm Design Based on Circle Overlapping Test. Applied Mechanics and Materials Journal, Vols. 229-231, pp 1487-1491, 2012.

6.3 Future Work

Multiple improvements can be made to the research presented in this thesis. Overall, this thesis proposed an avoidance trajectory for UAV so that collision avoidance is achieved. Unfortunately, the aircraft modelling and controlling on flight dynamics and engine controls are not discussed here. The additional research on this topic can be completed by investigating on how to modelling and controlling the aircraft's flight dynamics (by controlling the angle of roll, yaw, pitch and thrust) to control an aircraft's direction and the engine controls as aircraft change it current speed during flying.

Furthermore, flying multiple unmanned aircraft in the same airspace invited to collision increment. Thus, advance form of collision avoidance will be required. The algorithms can be improved by adding the ability to avoid multiple obstacles. Another target for algorithm improvement is to develop a CAS algorithm that applicable for several conflicts with non-cooperative aircrafts and obstacle such as mountains, trees, buildings and etc.

References

Ajith Kumar, B., & Ghose, D. (2001). Radar-assisted collision avoidance/guidance strategy for planar flight. *IEEE Transactions on Aerospace and Electronic Systems, 37*(1), 77-90.

Albaker, B. M., & Rahim, N. A. (2009). *Straight Projection Conflict Detection and Cooperative Avoidance for Autonomous Unmanned Aircraft Systems*. Paper presented at the IEEE Conference on Industrial Electronics and Applications, Xi'an, China.

Albaker, B. M., & Rahim, N. A. (2009a, 25-26 July 2009). *Intelligent conflict detection and awareness for UAVs.* Paper presented at the Innovative Technologies in Intelligent Systems and Industrial Applications, 2009. CITISIA 2009.

Albaker, B. M., & Rahim, N. A. (2009b, 14-15 Dec. 2009). *A survey of collision avoidance approaches for unmanned aerial vehicles.* Paper presented at the International Conference for Technical Postgraduates (TECHPOS), 2009.

Albaker, B. M., & Rahim, N. A. (2010). Unmanned Aircraft Collision Avoidance System using Cooperative Agent-Based Negotiation Approach. *International Journal of Simulation Systems, Science & Technology, 11*(5), 1-7.

Albaker, B. M., & Rahim, N. A. (2011a). Autonomous Unmanned Aircraft Collision Avoidance System based on Geometric Intersection. *International Journal of the Physical Sciences, 6(3)*, 391-401.

Albaker, B. M., & Rahim, N. A. (2011b). A Conceptual Framework and a Review of Conflict Sensing, Detection, Awareness and Escape Maneuvering Methods for UAVs. In M. Mulder (Ed.), *Aeronautics and Astronautics*: InTech.

Albaker, B. M., Rahim, N. A., & Mubin, M. (2009, October 6-7). *Collision Detection and Resolution for Cooperative UAVs based onto Peer-to-Peer Communication and Predefined Maneuvering Rules.* Paper presented at the Asia Pacific Conference on Defence & Security Technology, Kuala Lumpur, MALAYSIA.

Asmat, J., Rhodes, B., Umansky, J., Villavicencio, C., Yunas, A., Donohue, G., & Lacher, A. (2006, 28-28 April 2006). *UAS Safety: Unmanned Aerial Collision Avoidance System (UCAS).* Paper presented at the Systems and Information Engineering Design Symposium, 2006 IEEE.

Bicchi, A., & Pallottino, L. (2000). On optimal cooperative conflict resolution for air traffic management systems. *IEEE Transactions on Intelligent Transportation Systems, 1*(4), 221-231.

Bindiganavle, K. (2000). Chapter 4: Parametric B-Splines. Retrieved from http://scholar.lib.vt.edu/theses/available/etd-04192001-172731/unrestricted/chapter_4.pdf

Bourke, P. (April 1989). Intersection point of two lines (2 dimensions), from http://paulbourke.net/geometry/lineline2d/

Britt, J. J. (1994, 27 Jun-1 Jul 1994). *Case study: Applying formal methods to the Traffic Alert and Collision Avoidance System (TCAS) II.* Paper presented at the COMPASS '94, Ninth Annual Conference on Computer Assurance, Gaithersburg, MD.

Byrne, J., Cosgrove, M., & Mehra, R. (2006, 15-19 May 2006). *Stereo based obstacle detection for an unmanned air vehicle.* Paper presented at the IEEE International Conference on Robotics and Automation, Orlando, Florida.

Call, B. R. (2006). *Obstacle Avoidance for Unmanned Air Vehicles.* Master of Science, Brigham Young University.

Carbone, C., Ciniglio, U., Corraro, F., & Luongo, S. (2006, 13-15 Dec. 2006). *A Novel 3D Geometric Algorithm for Aircraft Autonomous Collision Avoidance.* Paper presented at the 45th IEEE Conference on Decision and Control.

Crouch, P., & Jackson, J. (1991). Dynamic interpolation and application to flight control. *J. Guidance, Control and Dynamics, 14*, 814-822.

de Boor, C. (1978). *A Practical Guide to Splines*: Springer-Verlag.

Dear, R. G., & Sherif, Y. S. (1991). Z-Basic algorithm for collision avoidance system. *IEEE Transactions on Systems, Man and Cybernetics, 21*(4), 915-921.

Gertler, J. (January 3,2012). U.S. Unmanned Aerial Systems: Congressional Research Service.

Hachour, O. (2009). The use of linear parametric smoothed curve path planning for mobile robots. *International Journal of Systems Applications, Engineering & Development, 3*(3), 93-104.

Han, S. C., Bang, H., & Yoo, C. S. (2009). Proportional Navigation-Based Collision Avoidance for UAVs. *International Journal of Control Automation and Systems, 7*(4), 553-565. doi: DOI 10.1007/s12555-009-0407-1

Hawkes, D. (1998). Airborne Collision Avoidance System II: CNS/ATM Steering Group.

Hill, J., Johnson, F., Archibald, J., Frost, R., & Stirling, W. (2005). *A Cooperative Multi-agent Approach to Free Flight*. Paper presented at the International Joint Conference on Autonomous Agents and Multi Agent Systems, Netherlands.

Hongxin, Z. J., F. (2006). B-Spline Interpolation and Approximation. In Z. University (Ed.).

John, H. M. (2004). Spline Curves *Numerical Methods using Matlab*.

Kano, H., Egerstedt, M., Nakata, H., & Martin, C. F. (2003). B-Splines and control theory. *Applied Mathematics and Computation, 145*(2-3), 263-288.

Kano, H., Fujioka, H., Egerstedt, M., & Martin, C. F. (2008). Optimal smoothing spline curves and contour synthesis. *Journal Automatica, 44*(1), 185-192.

Kayran, A. Chapter 14: Spline Curves. Retrieved from www.ae.metu.edu.tr/~ae464/splines.pdf.

Kuchar, J. K., & Drumm, A. C. (2007). The Traffic Alert and Collision Avoidance System. *Lincoln Laboratory Journal, 16*, 277 - 296.

Kuchar, J. K., & Yang, L. C. (2000). A review of conflict detection and resolution modeling methods. *IEEE Transactions on Intelligent Transportation Systems, 1*(4), 179-189.

Kurnaz, S., Cetin, O., & Kaynak, O. (2009). Fuzzy Logic Based Approach to Design of Flight Control and Navigation Tasks for Autonomous Unmanned Aerial Vehicles. *Journal of Intelligent & Robotic Systems, 54*, 229-244. doi: 10.1007/s10846-008-9263-0

Kuwata, Y. (2003). *Real-time Trajectory Design for Unmanned Aerial Vehicles using Receding Horizon Control.* Master of Science in Aeronautics and Astronautics, University of Tokyo.

Kwag, Y. K., Choi, M. S., Jung, C. H., & Hwang, K. Y. (2006). Collision avoidance radar for UAV. *Proceedings of 2006 CIE International Conference on Radar, Vols 1 and 2*, 488-491
1944.

Kwag, Y. K., & Chung, C. H. (2007). UAV based collision avoidance radar sensor. *Igarss: 2007 Ieee International Geoscience and Remote Sensing Symposium, Vols 1-12*, 639-642
5334.

Lee, X. (2006). Line (in the works), from http://xahlee.org/SpecialPlaneCurves_dir/Line_dir/line.html

Lindsten, F. (2008). *Angle-only based collision risk assessment for unmanned aerial vehicles.* Linkoping University.

Livadas, C., Lygeros, J., & Lynch, N. A. (2000). High-level modeling and analysis of the traffic alert and collision avoidance system (TCAS). *Proceedings of the IEEE, 88*(7), 926-948.

O'Brien, K. (2009). Automatic Dependent Surveillance-Broadcast (ADS-B): Boieng Institute of Aviation Leadership.

Pallottino, L., Feron, E. M., & Bicchi, A. (2002). Conflict resolution problems for air traffic management systems solved with mixed integer programming. *IEEE Transactions on Intelligent Transportation Systems, 3*(1), 3-11.

Park, J. W., Oh, H. D., & Tahk, M. J. (2009). UAV Conflict Detection and Resolution Based on Geometric Approach. *International Journal of Aeronautical and Space Sciences, 10*(1), 37-45.

Radzi, N. F. M., Mubin, M., & Rahim, N. A. (2011). *Collision Detection Algorithm for UAVs based on Parametric Theorem Approach.* Paper presented at the IEEE International Conference on Computer and Communication Devices, Bali.

Radzi, N. F. M., Mubin, M., Rahim, N. A., Ouchi, S., Kodani, N., & Mokhtar, N. (2011, October, 22-23). *Collision Detection Algorithm using Parametric Theorem.* Paper presented at the International Conference on Modeling and Simulation Technology, Tokyo, JAPAN.

Robby, T. T. (2011). B-Spline Curves. 1-8. Retrieved from http://www.staff.science.uu.nl/~tan00109/teaching/2012_ddm/05_bspline/05_bspline1.pdf

Saunders, J., Beard, R. W., & McLain, T. W. (2007). *Obstacle Avoidance using Circular Paths*. Paper presented at the AIAA Guidance, Navigation, and Control Conference and Exhibit, Hilton Head, SOUTH CAROLINA.

Schoenberg, I. J. (1946). Contributions to the problem of approximation of equidistant data by analytic functions, Part A: On the problem of smoothing or graduation, a first class of analytic approximation formulas. *Quarterly of Applied Mathematics, 4*, 45-99.

Sigurd, K., & How, J. (2003). *UAV Trajectory Design using Total Field Collision Avoidance.* Paper presented at the AIAA Guidance, Navigation, and Control Conference Austin, TEXAS.

Sinopoli, B., Micheli, M., Donato, G., & Koo, T. J. (2001). *Vision based navigation for an unmanned aerial vehicle.* Paper presented at the IEEE International Conference on Robotics and Automation, Seoul, KOREA.

Sislak, D., Pechoucek, M., Volf, P., Pavlicek, D., Samek, J., Marik, V., & Losiewicz, P. (2007). AGENTFLY: Towards Multi-agent Technology in Free Flight Air Traffic Control *Whitestein Series in Software Agent Technologies and Autonomic Computing* (pp. 77-100).

Sislak, D., Volf, P., Komenda, A., Samek, J., & Pechoucek, M. (2007). *Agent-based Multi-layer Collision Avoidance to Unmanned Aerial Vehicles*. Paper presented at the International Conference on Integration of Knowledge Intensive Multi-Agent Systems.

Stephen, A. C., Kenneth, J. K., Peter, P., & Linton, W. (2005). Unmanned Aircraft Systems Roadmap 2005-2030.

Thomas, W. S. (2005). An Introduction to B-Spline Curves. Retrieved from http://tom.cs.byu.edu/~455/bs.pdf

Tomlin, C., Pappas, G. J., & Sastry, S. (1998). Conflict resolution for air traffic management: a study in multiagent hybrid systems. *IEEE Transactions on Automatic Control, 43*(4), 509-521.

Wallace, E. K., Collins, R., Rapids, C., & Iowa. (1999). *Conflict Detection and Alerting for Separation Assurance Systems.* Paper presented at the Digital Avionics System Conference, St. Louis.

Wangermann, J., & Stengel, R. (1999). Optimization and Coordination of Multi Agent Systems using Principled Negotiation. *Journal of Guidance Control and Dynamics, 22*, 43-50.

Watanabe, Y. C., A.J. & Johnson. (August 2007). *Vision-Based Obstacle Avoidance for UAVs.* Paper presented at the AIAA Guidance, Navigation and Control Conference and Exhibit, Hilton Head, USA.

Weisstein, E. W. B-Spline, from http://mathworld.wolfram.com/B-Spline.html

Williams, E. (2004). *Airborne Collision Avoidance System*. Paper presented at the 9th Australian Workshop on Safety Related Programmable Systems (SCS'04), Brisbane.

Yizhen, Z., Antonsson, E. K., & Grote, K. (2006). *A new threat assessment measure for collision avoidance systems.* Paper presented at the IEEE Intelligent Transportation Systems Conference, Totonto, CANADA.

Yuan, W., & Song, D. (2009, 25-27 May 2009). *Modeling and simulation of Traffic Alert and Collision Avoidance System.* Paper presented at the 4th IEEE Conference on Industrial Electronics and Applications.