CRANFIELD UNIVERSITY

Hamid ALTURBEH

COLLISION AVOIDANCE SYSTEMS FOR UAS OPERATING IN CIVIL AIRSPACE

SCHOOL OF ENGINEERING

PhD THESIS

CRANFIELD UNIVERSITY

SCHOOL OF ENGINEERING

PhD THESIS

Academic Year 2013-2014

Hamid ALTURBEH

COLLISION AVOIDANCE SYSTEMS FOR UAS OPERATING IN CIVIL AIRSPACE

Supervisor: Dr. James F. Whidborne

November 2014

بِسمِ اللّٰهِ الرَّحمٰنِ الرَّحِيمِ

I would like to dedicate this thesis to my family

Acknowledgements

Firstly, I would like to express my sincere gratitude to my supervisor, Dr James Whidborne, for professional supervision and his scientific assistance, constant help, advice, guidance and good mood during the past three years. I also wish to thank all members of the Dynamics, Simulation and Control group.

I would like to acknowledge Andrew Berry at QinetiQ for his kind help, it is worth mentioning that part of this thesis is a development of part of Berry's PhD thesis at the University of Leicester.

I would also like to acknowledge and thank the pilots at the National Flying Laboratory Centre, Cranfield University, who were helpful in giving their experience and advice that were useful in achieving this thesis. In particular to Susan Szasz for the interviewing and discussions that were essential to achieve this thesis.

Finally, my biggest thanks must go to my wonderful mother, my wife Nada, and my little daughter Sarah.

Abstract

Operation of Unmanned Aerial Vehicles (UAVs) in civil airspace is restricted by the aviation authorities which require full compliance with regulations that apply for manned aircraft. This thesis proposes control algorithms for a collision avoidance system that can be used as an advisory system or a guidance system for UAVs that are flying in civil airspace under visual flight rules. An effective collision avoidance system for the UAV should be able to perform the different functionalities of the pilot in manned aircraft. Thus, it should be able to determine, generate, and perform safe avoidance manoeuvres. However, the capability to generate resolution advisories is crucial for the advisory systems. A decision making system for collision avoidance is developed based on the rules of the air. The proposed architecture of the decision making system is engineered to be implementable in both manned aircraft and UAVs to perform different tasks ranging from collision detection to a safe avoidance manoeuvre initiation. Avoidance manoeuvres that are compliant with the rules of the air are proposed based on pilot suggestions for a subset of possible collision scenarios. The avoidance manoeuvre generation algorithm is augmented with pilot experience by using fuzzy logic technique to model pilot actions in generating the avoidance manoeuvres. Hence, the generated avoidance manoeuvres mimic the avoidance manoeuvres of manned aircraft. The proposed avoidance manoeuvres are parameterized using a geometric approach. An optimal collision avoidance algorithm is developed for real-time local trajectory planning. Essentially, a finite-horizon optimal control problem is periodically solved in real-time hence updating the aircraft trajectory to avoid obstacles and track a predefined trajectory. The optimal control problem is formulated in output space, and parameterised by using B-splines. Then the optimal designed outputs are mapped into control inputs of the system by using the inverse dynamics of a fixed wing aircraft.

Contents

List of Figures

List of Tables

Nomenclature

List of Symbols

α	Decay rate of the Yukawa potential function
B	Discretised basis functions matrix
B_{ls}	Curve fit matrix
C	Curve coefficient matrix matrix
$\dot{\phi}$	Roll rate
$\dot{\psi}$	Heading rate
γ	Flight path angle
λ	Scaling factor
ϕ	Bank angle
ψ	Heading angle
ρ	Air density
τ	Curve parameter
B_i	i^{th} order basis function
C_D	Drag coefficient
C_i	Coefficient for each of the i^{th} order basis functions
C_L	Lift coefficient
C_{D0}	Minimum drag coefficient

D	Drag
d_c	Clearance distance
g	Gravity acceleration
J	Cost function
L	Aircraft lift
m	Aircraft total mass
n	Load factor
$N_{i,p}$	The p^{th} degree B-spline basis functions
$P(\tau)$	Polynomial curve description as a function of τ
R	Turn radius
$R_{i,p}$	The i^{th} order rational basis function of degree p
S	Wing area
T	Thrust
t	Time
T_c	Time to collision
t_h	Time horizon
T_m	Manoeuvre time
T_s	Sampling time
U	Knot vector: Control vector
u	UAV speed on x-axis in earth reference frame
V	Aircraft speed
v	UAV speed on y-axis in earth reference frame
$V_{stall_{turn}}$	Stall speed in the turn

V_{stall}	Stall speed
w	UAV speed on z-axis in earth reference frame
w_i	Weights
x, y, z	Aircraft center of gravity coordinates in earth axis

List of Acronyms

ADR	Advisory Routes
ADS	Advisory System
ADS-B	Automatic Dependent Surveillance-Broadcast
AF	Alert Flag
AFRL	Air Force Research Laboratory
AMG	Avoidance Manoeuvre Generator
AOUAV	Autonomously Operating Unmanned Aerial Vehicle
ASAS	Airborne Separation Assistance System
ASTRAEA	Autonomous System Technology Related Airborne Evaluation and Assessment
ATC	Air Traffic Controller
ATZ	Aerodrome Traffic Zones
CA	Collision Assessment
CAA	Civil Aviation Authority
CDP	Collision Detection and Prioritizing
CDR	Conflict Detection and Resolution
CF	Collision Flag
CTR	Aerodrome Control Zone
DCM	Directional Cosine Matrix
ELOS	Equivalent Level of Safety
FAA	Federal Aviation Administration
FCM	Flight Control Modes

FLC	Fuzzy Logic Control
GPS	Global Positioning System
GPWS	Ground Proximity Warning System
GUI	Graphic User Interface
HMI	Human Machine Interface
IFR	Instrument Flight Rules
LTP	Local Trajectory Planner
MIDCAS	Mid Air Collision Avoidance System
MPC	Model Predictive Control
MPC	Model Predictive Control
NAS	National Airspace System
NEU	North, East, Up
NF	Nine O'clock Flag
NURBS	Non-uniform Rational B-Spline
OODA	Observe, Orient, Decide, and Act
PCAS	Portable Collision Avoidance System
PZ	Protected Zone
RA	Resolution Advisories
RF	Collision Resolved Flag
RHC	Receding Horizon Control
RS-LS	Right-Straight then Left-Straight
RSL	Right-Straight-Left
SAA	Sense and Avoid

SAAFT	Sense and Avoid Flight Tests
TCAS	Traffic Alert and Collision Avoidance System
TMA	Terminal Control Area
UAS	Unmanned Aircraft Systems
UAV	Unmanned Aerial Vehicle
VFR	Visual Flight Rules
VMC	Visual Meteorological Conditions

Chapter 1

Introduction

1.1 Motivation

Unmanned Aircraft Systems (UAS) are of increasing importance in the aerospace industry for both civilian and military applications due to their ability to complete dull, dirty and dangerous missions [1]. However, operation of Unmanned Aerial Vehicles (UAV's) in civil/non-segregated airspace is restricted by the policies of aviation authorities (e.g. Civil Aviation Authority (CAA) in the UK, Federal Aviation Administration (FAA) in the USA), which require full compliance with rules and obligations that apply for manned aircraft [2–4].

The development of a good Sense and Avoid (SAA) system is one of the most important issues that a UAV designer must deal with to give the UAV the ability to avoid conflict situations as required for manned aircraft. SAA capability must provide for collision avoidance protection between a UAS and other aircraft analogous to the see and avoid operation of manned aircraft that meets an acceptable level of safety [2]. Much research is being undertaken to achieve the civil aviation authorities requirements for SAA system, and hence enable the routine use of UAV's in all classes of airspace without the need for restrictive or specialized conditions of operation [3, 51**?**]. However, significant progress is only expected in the mid-term (between 2015-2020), and a standardised SAA system is expected to be achieved after 2020. These expectations have been made by the FAA (Integration of Civil Unmanned Aircraft Systems (UAS) in the National Airspace System (NAS) Roadmap) due to the complexity of SAA concepts, and the immature development of SAA technology [2].

The lack of a SAA system to avoid collisions with other aircraft is a major barrier to

UAS operations in non-segregated and civil airspace [5]. The operation of a SAA system can be considered in three levels [3]:

1. Strategic SAA: Conflict to be detected at long range, so the system can maintain separation distance by adjusting the UAS trajectory. Hence, collisions will not happen.

2. Conflict Resolution Advisories (RA): These advisories are issued to the UAV pilot (UAVp) to avoid collisions based on the rules of the air. The RA must be accepted by the UAVp before a manoeuvre is executed.

3. Autonomous collision avoidance: UAV avoids the collision autonomously.

In a manned aircraft the pilot in command has the ultimate responsibility for achieving the collision avoidance manoeuvre using the see and avoid principle. The pilot's decision process during the conflict can be broken down using the Observe, Orient, Decide, and Act (OODA) loop [3, 6]:

- Observe: A pilot visually scans for collision threats. Information that is offered by Air Traffic Controller (ATC), or the aircraft avionics helps the pilot to detect potential collision threats.

- Orient: Pilot uses his/her knowledge and experience to evaluate what is seen. Thus, the range, speed, and bearing of the threats can be estimated based on apparent size growth, and the assumptions about the threat type.

- Decide: The pilot should determine a safe avoidance manoeuvre that will be performed in order to avoid the collision safely. The avoidance manoeuvre should be compliant with the rules of the air.

- Act: Finally, the determined safe manoeuvre is performed by the pilot.

The required time for a pilot to recognise an approaching aircraft and initiate an avoidance manoeuvre is 12.5 seconds in total [7]. Most of this time is spent on collision recognition and decision making (See Section 4.8.4). However, this time may be greater because pilots differ in their response time [8]. Hence, a Decision Making System (DMS) that could be used as an advisory system will effectively save time and help both the on-board pilot in manned aircraft, and the UAV ground-based pilot to avoid the conflicts safely. In an Autonomously Operating UAV (AOUAV) the DMS could be used to initiate avoidance manoeuvres.

The motivation of this thesis is to develop a collision avoidance system that is able to issue the resolution advisories, and generate and track safe avoidance manoeuvres. These manoeuvres should be similar to those performed by a pilot in manned aircraft which are compliant with the rules of the air.

1.2 Aims and Objectives

This thesis aims to develop a control algorithm for collision avoidance system for aircraft that fly under Visual Flight Rules (VFR) conditions. This algorithm could be used as an advisory system for manned aircraft, or a guidance system for UAV in order to enable flight in civil airspace. According to the policies of civil aviation authorities around the world the UAV operating in civil airspace must satisfy the safety and operational conditions at least as manned aircraft [9]. So a UAV that behaves the same as manned aircraft in all conditions means an aircraft that is in full compliance with air traffic rules.

The manoeuvre during conflict resolution is a very important issue in Collision Avoidance Systems (CAS). Hence, this thesis investigates trajectory optimisation during the maneuvers, as well as the air traffic rules satisfaction for a subset of the possible conflict scenarios.

The capability of RA generation, and a safe avoidance manoeuvre determination necessitates a type of decision making algorithm that is able to perform the different functionalities of the pilot in manned aircraft. The aim here is to develop an architecture of the DMS that can be implementable in manned and unmanned aircraft, taking into consideration the civil aviation authorities requirements.

Furthermore, it is sometimes difficult to model pilot behaviour using deterministic models. However, the Fuzzy Logic Control (FLC) technique provides a tool for modelling human behaviour. The aim is to use the fuzzy logic technique to design fuzzy logic pilot models that express human centered rules in order to generate avoidance manoeuvres based on the pilot experience.

These aims are achieved by the following proposed objectives:

1. Review the work carried out on CAS for the airspace application.

2. Review the aviation authorities requirements and obligation for UAV integration in civil airspace.

3. Develop control algorithms for local trajectory planning. The local trajectory must track a predefined global trajectory and avoid any pop-up obstacles. This will include:

   (a) Orthogonal basis function local trajectory generation methods (B-spline curve methods).

   (b) Avoidance manoeuvre optimisation.

4. Develop a generic Decision Making System (DMS) for collision avoidance system based on the rules of the air in visual flight rules (VFR) conditions, and the civil aviation authorities requirements. The developed DMS should be able to perform the following tasks:

   (a) Conflicts detection and prioritizing.

   (b) Conflicts evaluation and assessment.

   (c) Issue warning alerts and conflict resolution advisories.

   (d) Initiate corespondent safe avoidance manoeuvres.

5. Propose and generate collision avoidance manoeuvres that should be similar to the avoidance manoeuvres of manned aircraft (this could be carried out for a subset of the all possible conflict scenarios). This includes:

   (a) Specify the type of the avoidance manoeuvres.

   (b) Find the characteristics of the avoidance manoeuvres.

   (c) Parameterize the avoidance manoeuvres.

6. Augment the avoidance manoeuvre generation process, so that the algorithm mimics pilot behaviour by using fuzzy logic techniques. This can be achieved by determining the inputs and outputs of the fuzzy logic system, and generating the fuzzy logic rules based on pilot behaviour during the conflict.

7. Test algorithms in simulation using MATLAB/Simulink.

1.3 Thesis Outline

This thesis consists of seven chapters including the introduction chapter. The remainder of this thesis is structured as follows:

- Chapter 2 provides a literature review of collision avoidance systems for aircraft.
- Chapter 3 presents a local trajectory planning algorithm that is used for the collision avoidance system for a fixed-wing aircraft.
- Chapter 4 proposes a decision making system (DMS) algorithm for the collision avoidance systems (CAS).
- Chapter 5 discusses the avoidance manoeuvres generating process for different conflict scenarios in which the UAV should change direction (right/left turn) in horizontal plane.
- Chapter 6 discusses how pilot experience can be used in the collision avoidance manoeuvre generation process for the UAV.
- The conclusions, limitations of the proposed algorithms, and recommendations for future work are given in Chapter 7.

1.4 Contributions to Knowledge

The contributions to knowledge which have been made as part of this work are summarized below:

- Develop a real-time local trajectory planning algorithm for a fixed-wing UAV using B-spline and MPC. The developed method is an extension of a previous method that was proposed for a quad-rotor UAV [10]. The developed method uses the differential flatness property of the fixed-wing aircraft to develop an inverse dynamic model. Hence, mapping the generated trajectory into the UAV's control commands is achieved. A method that helps the optimisation solver to avoid a local minimum is proposed.
- Develop a decision making system (DMS) architecture for collision avoidance system in VFR conditions. The proposed DMS architecture mimics the pilot decision making process during a conflict scenario. Thus, it could be used at different levels of aircraft autonomy (e.g. manned aircraft, remotely piloted UAV, or autonomously operating UAV).

- A graphical user interface (GUI) is proposed for algorithm test and simulation purposes with short comparison with currently used commercial Portable Collision Avoidance System (PCAS). The proposed GUI layout is designed based on the intruder priority.
- Propose, construct, and parameterise collision avoidance manoeuvres for a set of conflict scenarios:

  1. Head-on/overtaking conflict scenarios.
  2. Approaching conflict scenarios.

  The collision avoidance manoeuvres are proposed based on pilot suggestions[1]. Hence, the shapes of the manoeuvres are similar to the manoeuvres that are performed by manned aircraft.
- A pilot behavioural model is augmented in collision avoidance manoeuvres by using fuzzy logic technique to model the pilot reaction.
- A geometric approach is proposed to parameterise the generated collision avoidance manoeuvres. Thus the construction and generation the avoidance manoeuvres are simplified, hence the computational time for avoidance manoeuvre generation is reduced.

[1]Extended interviews and discussions about the problem have been carried out with a pilot at National Flying Lab, Cranfield University (See Appendix B)

Chapter 2

Collision Avoidance Systems (CAS): Literature Review

2.1 Introduction

Many studies have been carried out to solve the collision avoidance problem in airspace to improve the performance of both Conflict Detection and Resolution (CDR) for manned and unmanned vehicles. For manned aircraft, the human element is key in the collision avoidance process in many methods due to the pilots ability to make decisions according to the collected information [11]. However, humans are prone to errors and hence automated systems seem to be an alternative solution. Automated Collision Avoidance Systems (CAS) are being used as advisory systems (e.g. Traffic Alert and Collision Avoidance System (TCAS)) in piloted aircraft but they can also be embedded with guidance systems in UAV's [12]. More than sixty different conflict detection and resolution methods have been addressed in the literature, not only in aerospace applications, but also for ground vehicles, and robotics [11]. In the recent years, the development in sensors technology and powerful processing units has led to a significant enhancement in both detection and resolving conflict scenarios [12].

2.2 Collision Detection and Avoidance Process

A collision or conflict can be defined as an event when there is loss in separation distance between two or more vehicles [13]. The distance of separation has different values depen-dening on the air traffic rules. The minimum separation distance according to traffic rules

in civil aerospace is 5 nmi and 1000 ft for horizontal and vertical distances respectively (Instrument Flight Rules (IFR)) [11]. These separation distances can be represented as a volume that surrounds each aircraft, called the Protected Zone (PZ). Hence, a conflict occurs, if there is any interference between these protected zones [14]. For example, in tactical collision alerting systems the PZ is taken as a sphere of 500 ft in diameter, or it can be represented in terms of time instead of distance [11]. However, the functions of CDR systems are the same in all cases, which are conflict prediction, provide the information to the pilot (in case of manned aircraft), or to the guidance system (in the case of autonomous UAVs), and, in some cases, evaluate the avoidance action.

Figure 2.1 shows a block diagram that simply illustrates the collision avoidance process:

1. The first step in the collision avoidance process is to monitor the traffic environment and identify the current traffic situation. The *state estimator* estimates the current traffic situation by collecting appropriate current states that are measured by sensors and communication equipment. However, uncertainty in measured states may occur due to sensor errors and/or limitation of the update rate [12].

2. In order to predict a conflict in the future, a *dynamic trajectory model* is required as shown in Figure 2.1; this block projects the current states into the future. This projection may be based solely on current state information. For example, a straight-line extrapolation of the current velocity vector, or may be based on additional, procedural information such as a flight plan [11]. Furthermore, due to model uncertainty, there is also a mismatch between the estimated trajectory and actual one.

3. The current and estimated states are combined to calculate some *metrics* to make traffic management decisions. Some examples of metrics include predicted minimum separation, and the estimated time to closest point of approach. In the traffic environment, the current and projected states can be calculated separately for each aircraft. However, the metric aggregates information from the aircraft that share the same airspace in order to manage the traffic situation.

4. The *Conflict detection block* then uses the conflict metrics to decide whether alarm should be issued and whether action is needed to avoid the conflict. However, in many cases of piloted aircraft, the pilot determines the appropriate actions in order to avoid collision. So in this case the function of the CDR system is just to issue the notifications of conflict [11].

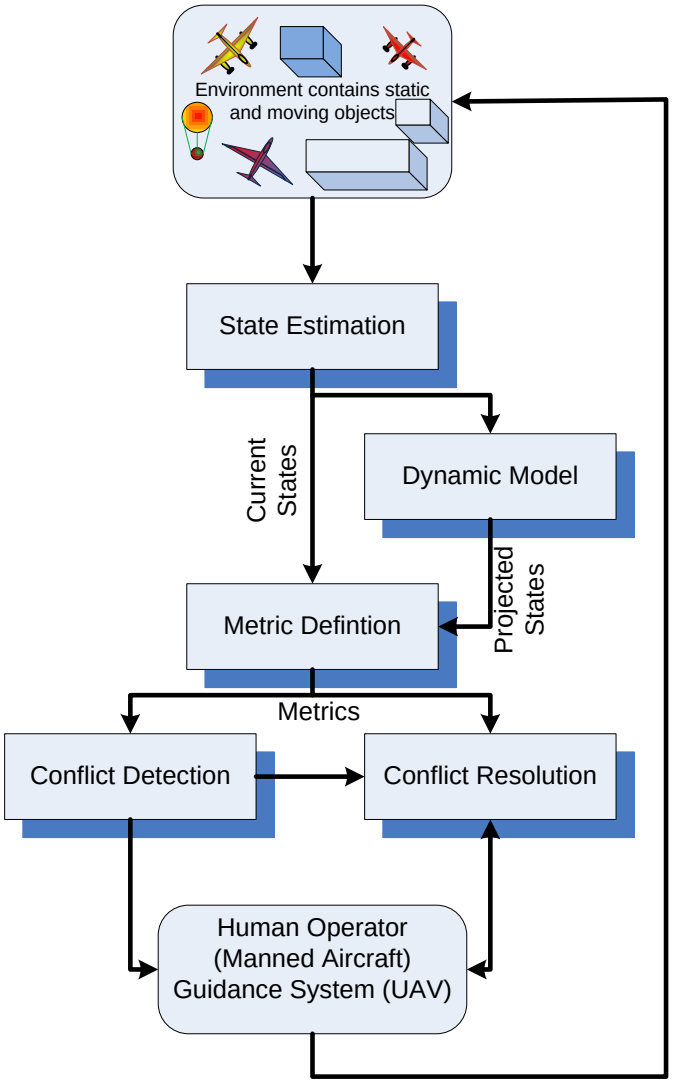

Figure 2.1 Collision avoidance process

The CDR systems in a UAV has different functions. In addition to conflict prediction it also resolves the conflict by using collision avoidance algorithms. It is worth mentioning that the notification or action will not be issued for all predicted conflicts because there are some predicted conflicts that are far into the future or too uncertain [11]. The resolution stage operates when the action to avoid the collision becomes necessary. The function of the conflict resolution stage can be expressed either as an advisory system such as TCAS, which gives the pilot appropriate commands that are needed to avoid the conflict, or as a feedback system that gives the ability to the operator to monitor if their action will resolve the conflict or not. This kind of system can be categorised as a passive system [15].

The conflict resolution block in Figure 2.1 has an independent state estimator, model of manoeuvre trajectory, and decision criteria. Either or both collision detection and collision avoidance may be automated or may be handled manually through procedures. For instance, the pilot is responsible for conflict detection and resolution in Visual Flight Rules (VFR), so the pilot must scan traffic visually (conflict detection) and take suitable action (based on the rules of the air) if there is a threat of collision (conflict avoidance) [16]. However, Instrument Flight Rules (IFR) place responsibility for monitoring the traffic separation on the Air Traffic Controller (ATC) who is using radar to detect the traffic separation, and issues resolutions to aircraft if a conflict threat is detected. In case of the aircraft that is equipped with an airborne CAS (e.g. TCAS), if operators fail to resolve the collision, additional guidance information is issued by TCAS [15]. One reason that makes the CDR system challenging and interesting is that there is interdependence between conflict detection and conflict resolution. As it is not easy to isolate conflict detection from conflict resolution and vice versa, there are many feasible design solutions. For example, deciding when action is required to resolve the conflict may depend on the action type, and similarly the type of required action may depend on how early that action begins [11].

2.3 Categorization Collision Avoidance Approaches

To provide insight into different CDR approaches, all different proposed methods must be categorized. The categorization should be built on fundamental factors that can express and identify the differences between each method. A good illustration of the design factors is given in [13] and [11] that are:

- Sensing tools,
- Encounter sensing dimension,
- Encounter current states projection,
- Collision threat assessment,
- Avoidance trajectories calculation and,
- Manoeuvre realization, and other design factors.

The next subsections give further detail of each design factor.

2.3.1 Sensing Tools

The traffic environment information around aircraft is collected by using sensors. The collected information by the sensors is used by automated systems to predict the conflict scenario which may use this prediction in the guidance algorithms to avoid conflict. The sensors that are used in CDR system can be divided into two main categories: cooperative and non-cooperative traffic sensors.

Cooperative traffic sensors enable aircraft to share information such as speed, heading, and position with other aircraft and airspace traffic control (ATC). Examples of cooperative traffic sensors are: Airborne Separation Assistance System (ASAS), and Automatic Dependant Surveillance Broadcast (ADS-B). These transfer aircraft information to ATC and other agents.

Aircraft that are not equipped with cooperative traffic sensors get information about surrounding airspace by using non-cooperative traffic sensors. There are different types of sensors that can sense the surrounding environment and collect information about other aircraft in shared airspace. Laser range finder, Electro-Optical/Infra-Red (EO/IR), radar system, stereo camera pairs, and moving single camera are some examples of sensors that are used in non-cooperative traffic systems[17–19]. However, non-cooperative sensor systems have their limitations. For example, the laser range finders, which are effective in detecting scanned obstacles, have limited capabilities to detect the environment and they are relatively expensive. Although radar systems can detect moving and stationary obstacles effectively, their weight and size mean their use in a small UAV is limited. The recent advances of digital signal processors have enabled the use of cameras as passive sensors that can provide information about the surrounding environment. Much research has already been conducted to use cameras in CAS systems [20, 21]. However experimental research carried out by [22], has shown a poor result for camera systems compared with humans in terms of detecting the intruder aircraft. The data accuracy that is provided by sensors is limited and depends on sensor type. The density of received data and data update rate add further limitations of data processing and uncertainty. One way to avoid failures associated with these limitations is increasing the safety distance between aircraft [12].

2.3.2 Encounter Sensing Dimension

The surrounding environment can be described either in a two dimensional plane (2D), or a three dimensional space (3D). The two dimensional approach can be either two dimensional horizontal plane (2D-H), or two dimensional vertical plane (2D-V). Most CAS can be categorized under 3D, or 2D-H approaches. However, the Ground Proximity Warning System (GPWS) uses a 2D-V approach [23].

2.3.3 Encounter Current State projection

Future prediction is one of the main componants of CAS by specifying how to project the current states of UAV and encounter into the near future, so the conflict threat can be assessed. Figure 2.2 shows four different methods for prediction which are [13]:

1. Straight projection (2.2 A): In this method the states are projected into the future along a single straight line trajectory without direct consideration of the uncertainties. This method is simple, but it can be only used in aircraft that have very predictable trajectories, and it assumes that the encounter will not do any manoeuvring in predicted time.

2. Worst case projection (2.2 B): An aircraft is assumed to perform any range of manoeuvres, so there is a range of trajectories. If any one of these trajectories is in conflict risk, then a conflict is predicted. Due to the extensive computational effort that is needed to evaluate the conflict the period projection time should be shortened.

3. Probabilistic method (2.2 C): Possible future trajectories could be developed by modeling the trajectory uncertainties. So the risk variation in aircraft future trajectory can be described in order to get a complete set of future trajectories, each trajectory of this set is weighted by a probability of occurring, producing a probability density function. This method gives the ability to make decisions according to the fundamental likelihood of conflict, and it also gives a direct assessment of a safety and false alarm rate. However, it is not easy to get an appropriate model for the probability of future trajectories.

4. Path plane sharing (2.2D): This method is based on sharing information of aircraft (flight plan segment, position, heading, and velocity) with all other aircraft in shared airspace and to ground stations for monitoring. By this way all aircraft will have a 3D illustration of neighboring aircraft movements, so accurate projection of encounter

could be extracted and consequently conflict parameters can be identified precisely. The Automatic Dependent Surveillance-Broadcast (ADS-B) is a good example of this method, ADS-B is proposed to be fully deployed in aircraft by the year 2020 to support free flight capability [24]. However, the complexity of this type of system increases as the amount of data that needs to be exchanged increases.

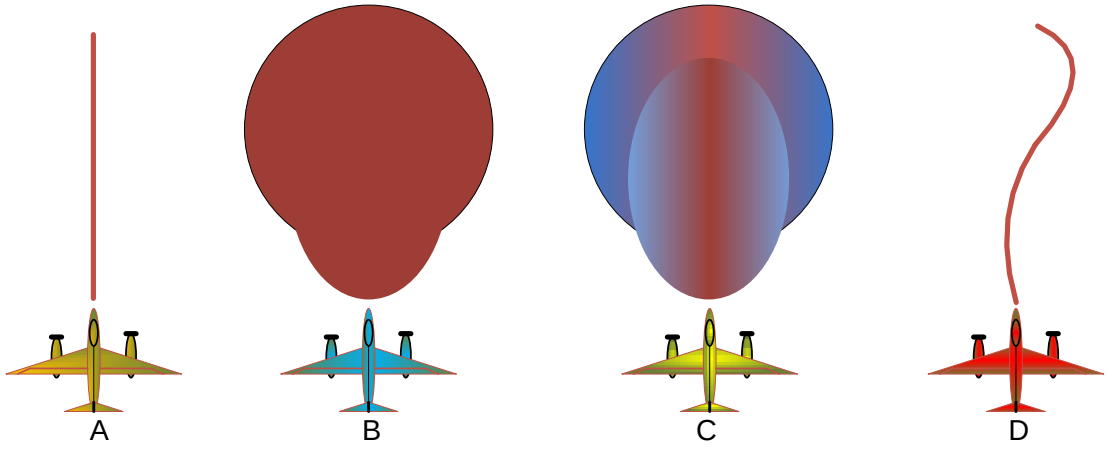

Figure 2.2 Current state projection methods: Straight projection (A); Worst case projection (B); Probabilistic method (C); Path plane sharing (D)

2.3.4 Collision Threat Assessment

Assessment of collision threat is a very important issue in CAS design process and it has received considerable attention [25–27]. Some approaches use a very simple criterion to determine when a collision exists. For example, concept of range information or concept of threat detection zones which surround each aircraft and determine a manoeuvre that ensures adequate separation between aircraft even if one aircraft does not make any manoeuvre. So a safe separation could be provided even if there is a failure in link to one aircraft. Other approaches may use complex thresholds or sets of logic [13].

2.3.5 Avoidance Trajectories Calculation

Many methods for generating trajectories that guarantee collision avoidance have been proposed in literature. For example: Predefined, Protocol Based [28], E-filed [29], Geometric [30],automotive [31], and hybrid systems [32] .

2.3.6 Manoeuvre Realization

A manoeuvre is the result of combination of actions by all aircraft in the vicinity [13]. To avoid a predicted conflict one aircraft, at least, must change its flight plan. In other words, one manoeuvre must be performed by at least one aircraft of those that involved in the conflict. The performed manoeuvre could take different type and different dimensions such as:

- Manoeuvres in horizontal plane (i.e. turn left, turn right).
- Manoeuvres in vertical plane (i.e. climb, dive).
- And/or speed-up, slowdown manoeuvres.

Depending on CAS approach, the manoeuvres can be a single dimension manoeuvre (i.e. change of only one dimension) or a combined manoeuvre. For example, a combination between change in speed with a change in vertical or horizontal plane, this combination can be performed simultaneously or in sequence. Also manoeuvres can be expressed as coordinated or uncoordinated. In a coordinated manoeuvre, the CAS can select one of two versions of manoeuvres. For example, in TCAS in which the preferred manoeuvre might be for aircraft A to climb while aircraft B descends [33]. Uncoordinated manoeuvre refers to the worst case scenario, in this case just one aircraft performs all manoeuvre actions, while the other aircraft does not respond [15].

2.3.7 Other Design Factors

There are many factors other should be taken in consideration during CAS design process. One factor is the computational time that is required for resolving the conflict. A good CAS approach should find a solution of the conflict in real time, so an effective and robust CAS should be reasonably simple to satisfy the time criterion. Another design factor is the ability of the CAS system to deal with multiple conflict scenarios. There are two approaches for this case:

- Single conflict management methods in which the aircraft handles the multiple intruders sequentially in pairs.
- Multiple conflict management methods in which the aircraft handles the situation at the same time.

However, some factors should be taken into consideration in the multiple CAS systems such as type of aircraft, separation criteria, and maximum packing density where manoeuvres no longer work [13].

2.4 Collision Avoidance Approaches

Many approaches have been proposed to find an adequate solution for the collision avoidance problem. This section gives some examples of CAS methods and discusses some advantages and disadvantages of each method.

2.4.1 Predefined Collision Avoidance

In this method an escape trajectory generated for collision avoidance is determined according to predefined rules without any additional on-line computation. This means that the response time required to avoid a conflict will be minimized, but the performed manoeuvres may lose effectiveness and optimality. That is because the commanded maneuvers cannot be modified even if there are unexpected events. For instance, a standard climb warning is issued by GPWS if there is a conflict threat with terrain [23].

2.4.2 Protocol Based Decentralized Collision Avoidance

This approach gives a suitable collision avoidance method for swarm navigation systems. For this each aircraft shares its information (e.g. position, velocity, way-points, and heading) with other teams' members. The decisions that are made by swarms' members are decentralized and based on a set of rules which are predefined. Although this method is highly scalable and guarantees safety the long trajectories that could be produced is one of its limitations. References [28, 34–36] are examples that use this kind of collision avoidance approach.

2.4.3 Optimized Escape Trajectory Approaches

A kinematic model of aircraft can be produced with a set of constraints and an optimal control problem can be formulated, so the collision avoidance problem could be handled. According to this methodology, an optimal escape trajectory for conflict resolution can be computed based on most desirable optimization constraint. For example, the TCAS uses a set of climb or dive manoeuvres and selects the least aggressive manoeuvre which provides

adequate protection [33].

Some approaches for generating optimized escape Trajectories have been proposed, but they do not appear to be in practical use at present. For example, a game theory approach has been proposed by Tomilin [32]. In this work avoiding simple moving obstacles was successfully achieved by using a controller that was designed depend on game theory approach. Fox [37] has used the dynamic window approach to determine the optimal and safe control action. This approach uses the dynamic model and kinematic constraints of the aircraft. In fact, the dynamic window approach has been used firstly as a safe navigation technique in robotics.

Shim and Sastry [38], have proposed a Model Predictive Control (MPC) approach to generate a conflict free trajectory while considering the aircraft limitations and its maneuverability. The MPC module is also used as a safeguard interface between path generator and the vehicle control system. Other examples of using MPC approach can be found in [39, 40].

Many other optimized collision avoidance approaches have been proposed in the literature such as expert system, genetic algorithms, and fuzzy logic technique [41]. However, complexity and the need to cover all scenarios lead to a very complex optimal control problem that increases the computational time.

Pre-mission path planning is often formulated as an optimization problem and many different optimization problems can be applied [13]. There are many reasons that make path planning design for UAV very difficult such as the UAV constraints (e.g. turning radius, speed, and climb/dive rate) and flying environment which may have non-flying zones or/and static/moving obstacles. Generally speaking, CAS optimal algorithms try to select the best solution from the set of all possible solutions [13].

2.4.4 Potential Field Methods

This method was first presented by Khatib [29] for robotics. It expresses the way-points as attractive forces and the obstacles as repulsive forces. By using simple electrostatic equations, a safe trajectory can be generated and then the trajectory with a low flux density is selected as the preferred path. This approach is appropriate for distributed and local collision avoidance where state information is available from all aircraft and when the number of

vehicles are small [42]. However, many difficulties arise in practical systems such as saddle points and local minima that may occur when generating a dynamic potential field and this may lead to aircraft loss of control or collision threat.

Another problem that may be faced in a practical implementation of this method is that the dynamic limitations of the aircraft are not considered. Hence, the aircraft may not be able to fly the generated trajectory. Finally, it is worth mentioning that the availability of state information is an essential factor for potential field method. So any deficiency in the state information may produce an improper field formation and then generate an aggressive control command that may be beyond the aircraft performance [43].

2.4.5 Geometric Methods

Geometric methods use the geometric properties of aircraft trajectories and utilize position and velocity vectors of all or some of the aircraft involved in the encounter. In order to predict a conflict geometric methods compare velocity vectors of aircraft with those of obstacles. Geometric methods provide information about the geometry of conflict to the guidance algorithm, so it can be used in conflict resolution strategies [44].

The collision cone approach is one example of a geometric method, it has been proposed originally for mobile robots and tested for static and dynamic environments with no constraints on vehicle shape or size [30]. This method uses the concept of a collision region, and if the vehicle velocity vector lies in this region the conflict prediction will be issued to the guidance system. The experimental results that were presented in [45] show that the collision cone approach can successfully be used for indoor mobile robot navigation in a dynamic environment. Although this approach was proposed in 1998 there have been many attempts to develop it further, but most of them focus on pair-wise scenarios [46]. Smith et al [43] used this method for a multiple conflict scenario, and also implemented it to generate a three dimensional command simultaneously. One disadvantage of geometric methods is that there is deviation from the original trajectory. However, optimal algorithms can be used to minimize this deviation [44].

2.4.6 Other CAS Approaches

Other CAS approaches include trajectory estimators [] and hybrid CAS systems [28, 32]. Trajectory estimation filters depend on the path history of the intruder to estimate the future

path and then try to avoid any possible conflict. However, these methods assume that the intruder will not make any sudden or extreme manoeuvres. Automotive collision avoidance method which attempts to predict vehicle trajectory using forward looking sensors or historical information, is another approach of CAS systems [31]. Tomlin [28, 32] has proposed a hybrid CAS method. This method uses a combined model of the vehicle and its manoeuvre. The model is a combination of continuous and discrete states hence the observed states can be filtered based on safety specification to get a safe subset of reach set. The control commands are then calculated by using the Hamilton-Jacobi equation, thus guaranteeing that the UAV will remain in its safe set. However, this method has a poor performance for large UAVs [13].

2.5 UAS Integration in the Civil Airspace

Before UAVs are allowed to fly normally in civil airspace some requirements must be satisfied to meet an Equivalent Level of Safety (ELOS); comparable to an aircraft with a pilot on board. ELOS refers to a combination of systems and a concept of operations that reduce the chance of midair collision to an acceptable level [47]. Two groups are leading the development of standards for safe and transparent UAS integration into non-segregated airspace: EUROCAE WG-73 in Europe and RTCA SC-203 in the US [48]. Reference [48] makes a comparative study for these groups' activities. This research focuses on UAV operation under VFR, so the (see-and-avoid) requirements discussed below.

2.5.1 See-and-Avoid Requirements

CAA has published document CAP-722 (Unmanned Aircraft System Operations in UK Airspace-Guidance) [49] which gives general requirements for UAV operation in UK civil airspace. FAA also has published a road map for the integration of civil UAS in the National Airspace System (NAS) [2]. Reference [47] establishes the requirements for a sense-and-avoid system for a Remotely Piloted Aircraft (ROA) that fulfills the intent of collision avoidance contained in the United States Federal Aviation Regulations (FAR) and the convention on international aviation rules of the air. The see and avoid systems requirements can be summarized as follows:

1. Take into consideration onboard sensor, beacons, transponder, air traffic control, concept of operation, and reliability.

2. Capability to give the operator warning and alerts in the form of visual and/or audible when there is a possible conflict.

3. Ability to execute an avoidance manoeuvre allowing the aircraft to manoeuvre autonomously to avoid the conflicting traffic if the UAV does not receive a pilot/operator command input to resolve the collision.

4. Field of Regard (FOR): The onboard sensor system shall cover the field of regard of ($\pm 110^{\circ}$) horizontal with respect to the longitudinal axis of the UAV, and ($\pm 15^{\circ}$) vertical with respect to the flight path at normal cruise speed, and provide sufficient coverage to enable detection of conflicting air traffic during expected maneuvers.

5. Minimum separation distance: A conflict is defined as another aircraft that will pass less than 500 feet, horizontally or vertically, from the UAV. When the SAA system detects a conflict, an operator initiated or autonomous deconfliction manoeuvre will be performed in sufficient time so the UAV and other aircraft miss each other, preferably by at least 500 feet.

6. Participating and Non-participating Traffic: The system must detect conflict that is created by participating (squawk a discrete transponder code and maintain two way radio communication with ATC), and non-participating aircraft (not required to communicate with ATC and may not even be equipped with a transponder)

7. Search Volume: One critically important factor for any SAA system is the search volume defined by azimuth and elevation. The critical factor for a SAA system is that it provide surveillance of all of the airspace that lies within the converging angle: ($\pm 110^{\circ}$) with respect to the longitudinal axis of the UAV, and a search elevation of ($\pm 15^{\circ}$) with respect to the flight path provides adequate coverage to detect converging aircraft.

8. Detection Range:The sense-and-avoid system must detect the traffic in time to process the sensor information, determine if a conflict exists, and execute a manoeuvre according to the right-of-way rules. If pilot interaction with the system is required, transmission and decision time must also be included in the total time between initial detection and the point of minimum separation

9. Lost Link Procedures: If there is any loss in command and control (C2) link(s), the system should have the capability to execute an autonomous manoeuvre so the aircraft

can avoid other traffic and then return to its previous altitude and course once the avoidance manoeuvre is complete. If the aircraft manoeuvres to avoid traffic while the link is lost, it shall notify the air-crew of this fact upon re-establishment of the link.

10. Emergency Situations: As the right of way rules for the aircraft in distress are changed, the system should be able to provide a residual capability to avoid other traffic. However, the system capability will be dependent upon the emergency situation.

11. Integrity Management: The SAA system should have a means of indicating to the pilot/operator that the sensor, computer system, display, or autonomous avoidance capability is not fully operational.

Although most research programmes focus on the technical requirements for UAV integration in the civil airspace, recently legal and ethical questions for using UAVs in non-segregated airspace have raised. Thomas Dubot has proposed the first set of laws that should be applicable to Unmanned Aircraft Operating Autonomously (UAOA) [50]:

1. A UAOA must not operate in such a way it could injure a human being or let a human being injured without activating controls or functions identified as means to avoid or attenuate this type of incident.

2. A UAOA should always maintain a continuous communication with predefined interfaces to obey orders of authorized personnel (UAS operator, ATS, Network Manager) except if such actions conflict with first rule.

3. A UAOA must operate in such a way it could protect its own existence and any other human property, on ground or in the air, including other UAS, except if such operations conflict with first or second rule.

4. A UAOA must always have a predictable behaviour, based on its route but also alternative pre-programmed scenarios, except if all forecast options conflict with first, second or third rule.

5. A UAOA interacts with surrounding traffic (separation, communication) according to requirements of the operating airspace, general priority rules and emergency and interception procedures except if such actions conflict the first, the second or the third rule.

6. As any airspace user, a UAOA should not operate in a way that could decrease significantly the global performance of ATM system in terms of safety, security, environment, cost-effectiveness, capacity and quality of service (efficiency, flexibility and predictability), except if such operation is required by first, second or third law.

7. A UAOA must ensure a complete traceability of all its actions.

2.5.2 Related Previous and Current Research Programs

Much research and many projects have been/being conducted to achieve the civil aviation authorities' requirements for UAS integration in all classes of the civil airspace, some of these projects are:

1. Mid Air Collision Avoidance System (MIDCAS) (2009-2014)[1]: MIDCAS is a 4 year long European project funded by five European countries. MIDCAS goal is to demonstrate the baseline of acceptable solutions for the critical UAS self separation and midair collision avoidance functions to contribute to the UAS integration in civilian airspace [4].

2. Autonomous System Technology Related Airborne Evaluation and Assessment (ASTRAEA)[2]: Is a UK industry-led consortium focusing on the technologies, systems, facilities, procedures and regulations that will allow autonomous vehicles to operate safely and routinely in civil airspace over the United Kingdom [3].

3. Sense and Avoid Flight Tests (SAAFT): By Air Force Research Laboratory (AFRL) and Defense Research Associates, Inc. (DRA) in USA. AFRL established the SAAFT program to demonstrate autonomous collision avoidance capabilities in both cooperative and noncooperative air traffic. The intent of the Sense-and-Avoid (SAA) program is to equip UAVs with collision avoidance capabilities and thus allow them the same access to national and international airspace that manned aircraft have [51].

[1] http://www.midcas.org/
[2] http://astraea.aero/

Chapter 3

Trajectory Planning

3.1 Introduction

A path planner can be categorized as one of two types [52]: a global planner which requires a good knowledge about the environment that the aircraft is going to fly in, and a local trajectory planner which is an algorithm that is running continuously in order to allow the aircraft to deal with events that may happen during the flight. Figure 3.1 shows a simple flight scenario where the aircraft mission is to fly from point **A** to **B** with the existence of both a pre-known obstacle and an intruder which is unknown till the sensing devices detect it during the journey.

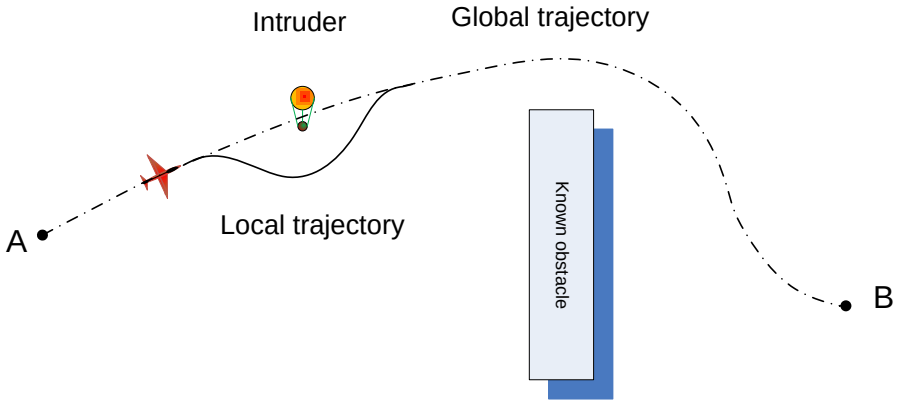

Figure 3.1 Simple flight scenario: Global and local trajectories

In order to complete this mission successfully, the global planner will calculate the optimal trajectory for whole journey from **A** to **B** taking into consideration all known obstacles. The local trajectory planner will be responsible for avoiding the detected obstacle. After

resolving the conflict the aircraft will continue tracking the global trajectory heading to the destination point **B**.

This chapter presents an approach for generating collision avoidance trajectories based on B-spline curves.

3.2 Collision Avoidance trajectories Generation Methodology

A finite-horizon optimal control problem is periodically solved in real-time that updates the aircraft trajectory to avoid obstacles and drive the aircraft to its global path. The proposed approach can be summarized as follows:

1. Given a global trajectory that the aircraft is required to follow, solve the following optimal control problem:

$$\min_{U(t)\in\mathscr{U}} J(U(t)) \tag{3.1}$$

where U is the control, $\mathscr{U}$ is the feasible space of control, and J is a cost measured over a finite time horizon, $t \in [t_0, t_f]$, that drives the local trajectory to the global trajectory. Subject to the aircraft dynamics constraints pair, state constraint given by:

$$\dot{X} = f(X,U) \tag{3.2}$$

where the state $X(t) \in \mathscr{X}$, and aircraft trajectory obstacles constraint given by:

$$Y = g(X) \tag{3.3}$$

where the output $Y(t) \in \mathscr{Y}$. Where $\mathscr{X}$ and $\mathscr{Y}$ are the feasible space of the state and the output respectively.

2. The problem is solved by a direct method by inverting the dynamics, so the optimization is performed in the output space $Y(t) \in \mathscr{Y}$, and parameterizing the trajectory by a spline function. The cost is augmented to maintain the constraints.

3. The generated local trajectory allows the UAV to track the global trajectory while avoiding any intruder or conflict scenarios that may occur. The local trajectory optimization is periodically solved on-line in a receding horizon approach to account for system uncertainties and obstacle changes.

This approach has been proposed by [10] for local trajectory planning for a quad-rotor UAVs. In this thesis this approach is applied for a fixed-wing UAV which has very different dynamics from a quad-rotor. The inverse dynamic approach is introduced for mapping trajectory profiles into UAV controls. Similar approaches has been proposed for trajectory planning for fixed-wing UAV [53], [54], and [55].

3.3 Guidance and Control Systems Architecture

The architecture of the vehicle guidance and control functionality is illustrated in Figure 3.2 [56]. This architecture was proposed for a small UAV operation within complex obstacle rich environments. The main components of this architecture can be briefly discussed:

Mission/global based reasoning This specifies the global goal of the aircraft mission and determine its mission requirements. The tasks of this unit depends on the level of autonomy of the aircraft. For example, in a low autonomy system this unit could be just an operator interface.

Sensing unit This is an onboard sensing unit that is responsible for real-time sensing tasks, such as; vehicle location detection by using, for example, Global Positioning System (GPS), en route obstacle detection by using suitable sensors, measurements of vehicle states, and wind vector detection which is very important for small UAVs, particularly operating in urban environment. More details about sensing tools can be found in section 2.3.1.

Obstacle/ environment modelling Generate a real-time 4D model for the environment that can be used by global reasoning and motion planning units. The environment model will contain the static obstacles and the moving one (the current and the future predicted positions of the moving obstacles).

Motion planning and control This unit consists of four levels:

1. Level-1: Global planning: Many techniques for global planning are presented in the literature such as, A* [57], Dubin's path [58], differential geometry [59], probabilistic roadmaps [60]. This level is not discussed this thesis.

2. Level-2: Local planning: Receding Horizon Control (RHC) is used for local trajectory planning. The proposed RHC approach in this thesis is performed in

the output space instead of traditionally used control space, this can be done because of the flatness of the system, see Section 3.7.1.

3. Level-3: Outer loop control: It is called also autopilot which traditionally is responsible for tracking the speed, altitude and heading demands. also it is responsible for handling the nonlinearity and control coupling in aircraft.

4. Level-4: Inner loop control: Vehicle stabilizer, which is usually traditional feedback controllers such as PID. For more robustness, advanced techniques such as H_∞ can be used.

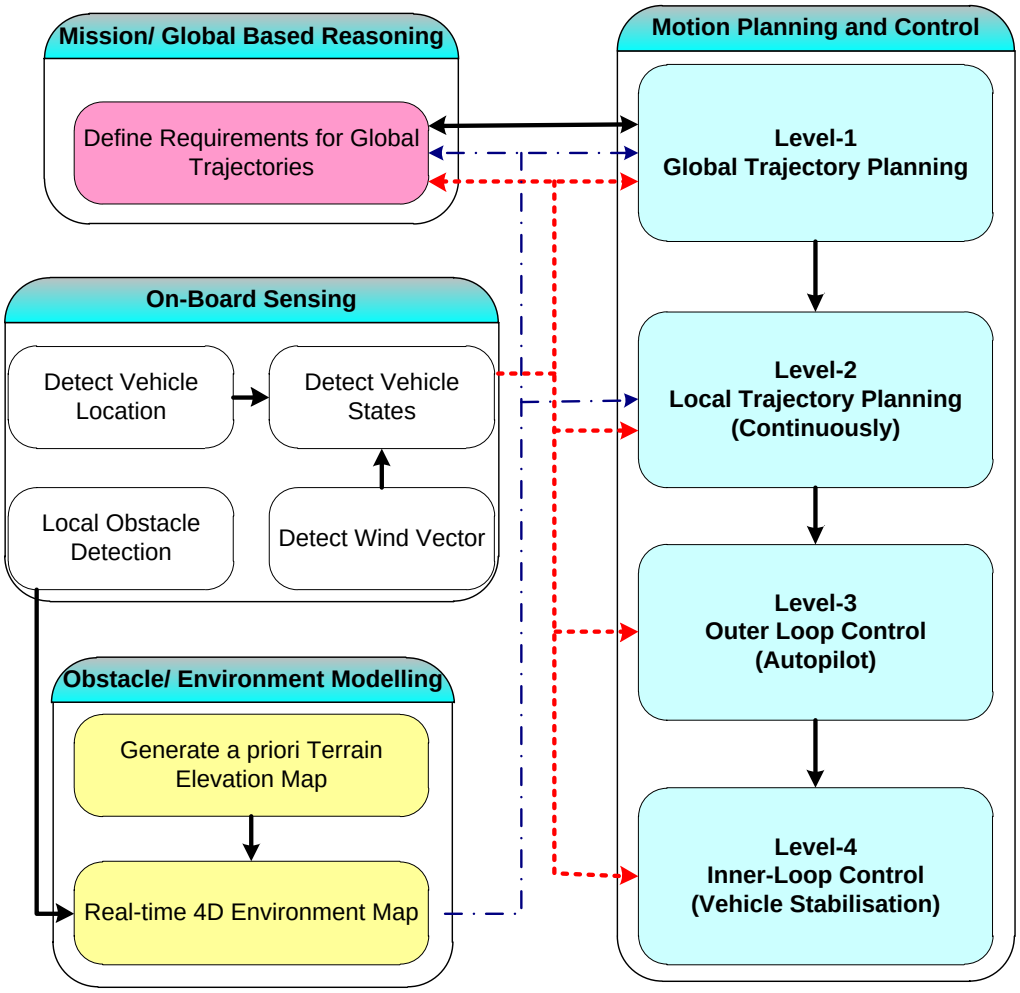

Figure 3.2 Architecture of vehicle guidance and control functionality

This architecture was built with the following considerations:

- Motion planning has been divided into global and local layers. Therefore, en route obstacles can be avoided without re-designing the global trajectory. This will reduce the computational time for the whole algorithm.

- Vehicle performance interface between outer-loop control (level-3) and local motion planning (level-2), so the lower level layer provides vehicle performance specifications to upper level layer (level-2). This will allow local motion planning to be designed in the output space rather than the control space, thus the complexity of the problem will be reduced significantly [56].

- Obstacle sensing and environment modelling were separated from motion planning and control. This also reduces the complexity of design.

- A priori obstacle map is assumed to be available, therefore, the real-time mapping will just consider the unknown obstacles.

3.4 Local Trajectory Description

There are two most common methods to represent a curve [61]:

1. Implicit equation:

$$f(x,y) = 0 \tag{3.4}$$

   where f is a function of the variables x and y. This equation describes an implicit relationship between the x and y coordinates of the point lying on the curve.

2. Parametric form: each of the coordinates of a point on the curve is represented separately as an explicit function of an independent parameter:

$$P(\tau) = (x(\tau), y(\tau)); a <= \tau <= b \tag{3.5}$$

   Thus, $P(\tau)$: is a vector-valued function of the independent variable τ. Although the interval [a,b] is arbitrary, it is usually normalized to [0,1]. For example, the first quadrant of the circle is defined by the parametric functions:

$$x(\tau) = \cos(\tau)$$
$$y(\tau) = \sin(\tau)$$

where $0 \leq \tau \leq \frac{\pi}{2}$.
Setting $u = \tan(\frac{\tau}{2})$

$$x(u) = \frac{(1-u^2)}{(1+u^2)}$$
$$y(u) = \frac{2u}{(1+u^2)}$$

where $0 \leq u \leq 1$. Thus, the parametric representation of a curve is not unique.

3.5 B-Spline Curves

Implementing a geometric modelling system need functions which [61]:

- Precisely representing of all the curves that users of the system need;
- Are easily, efficiently, and accurately processed in a computer, in particular:
    1. The computation of point and derivatives on the curves is efficient;
    2. Numerically insensitive to the rounding error of floating point;
    3. Little memory for storage requirements.
- Are simple and mathematically well understood.

However, single polynomial curves are often inadequate because [62]:

1. A large number of constraints need a polynomial with high degree;
2. Complex shapes require a high degree polynomial to be accurately represented;

The solution is to use curves which are piecewise polynomial, or piecewise rational. For example Figure 3.3 shows a curve $P(\tau)$ consisting of $(m = 5)$ segments of quadratic polynomial. $P(\tau)$ is defined on $\tau \in [0,1]$. the polynomial segments are joined at $\tau_0 = 0 < \tau_1 < \tau_2 < \tau_3 < \tau_4 < \tau_5 = 1$ which called breakpoints,or knots which map into the end point of each segments as shown in Figure 3.3. The segments are denoted by $P_i(\tau), 1 \leq i \leq m$. The curve is constructed by joining the segments with some level of continuity. Assume $P_i^{(j)}$ is the j^{th} derivative of P_i, $P(\tau)$ is said to be P^k continuous at breakpoint τ_i if $P_i^{(j)}(\tau_i) = P_{i+1}^{(j)}(\tau_i)$ for all $0 \leq j \leq k$.

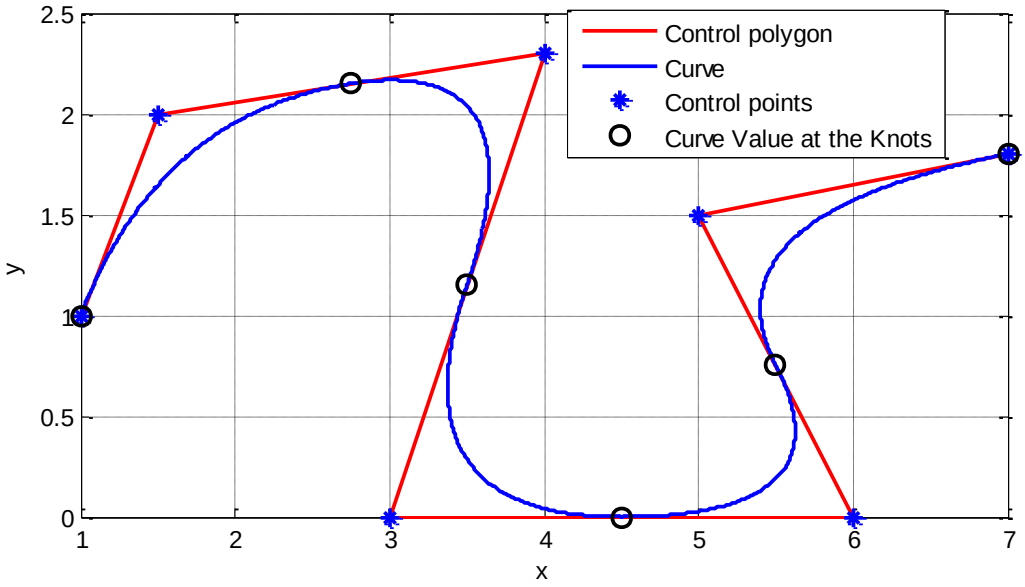

Figure 3.3 Five segments of quadratic polynomials that join at breakpoints

This thesis uses NURBS curves to describe the trajectory profiles. A NURBS curve is a vector-valued piecewise rational polynomial function. The p^{th} degree NURBS curve is given by:

$$P(\tau) = \frac{\sum_{i=0}^{n} w_i N_{i,p}(\tau) C_i}{\sum_{i=0}^{n} w_i N_{i,p}(\tau)}; \quad a \leq \tau \leq b \tag{3.6}$$

where w_i are the weights, C_i are the control points, and $N_{i,p}(\tau)$ are the p^{th} degree B-spline basis functions. There are many ways to represent B-spline basis functions. For computer implementation the recursive representation of B-spline basis functions is the most useful form [62]. Let $U = [u_0, u_1, ..., u_{m-1}, u_m]$ be a nondecreasing sequence of real numbers i.e, $u_i \leq u_{i+1}, i = 0, 1, ..., m-1$, u_i called knots or breakpoints, and U is the knot vector that contain $m+1$ knots. So the i^{th} B-spline basis function of p-degree (order $p+1$), denoted by $N_{p,i}(\tau)$ is defined as:

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

$$N_{i,p}(\tau) = \frac{\tau - u_i}{u_{i+p} - u_i} N_{i,p-1}(\tau) + \frac{u_{i+p+1} - \tau}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(\tau) \tag{3.7}$$

And $N_{i,p}(\tau) = 0$ if τ is outside $[u_i, u_{i+p+1}[$ The degree of the basis function p, number of control point $(n+1)$, and number of the knots $(m+1)$ are related by: $m = n+p+1$.

3.5.1 Knot Vector

The knot vector can be realized in different forms, but it must be a nondecreasing sequence of real numbers (i.e, $u_i \leq u_{i+1}, i = 0, 1, ..., m-1$). Equation (3.7) shows that the knot vector has a significant effect on the B-spline basis functions and hence on the resulting B-spline curve. There are two types of knot vector, periodic and open, in two flavours, uniform and nonuniform [61]. In a uniform knot vector, individual knot values are evenly spaced. For example:

$$U = [\ 0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5\ ]$$
$$U = [\ -0.3 \quad -0.2 \quad 0.1 \quad 0 \quad 0.1 \quad 0.2 \quad 0.3\ ]$$

In practice, uniform knot vectors generally begin at zero and are incremented by 1 to some maximum value, or it can be normalized in range between 0 and 1. A periodic uniform knot vectors will give periodic uniform basis functions for which:

$$N_{i,p}(\tau) = N_{i-1,p}(\tau - 1) = N_{i+1,p}(\tau + 1)$$

Thus, each basis function is a translation of the other.
In *open uniform* knot vector the end knot values has multiplicity equal to the order of the B-spline basis function $p+1$. For example,

$$p = 1; \quad U = [\ 0 \quad 0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 4\ ]$$
$$p = 2; \quad U = [\ 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 2 \quad 2 \quad 2 \quad 2\ ]$$
$$p = p; \quad U = [\ 0 \quad . \quad . \quad 0 \quad u_{p+1} \quad .. \quad u_{m-p-1} \quad 1 \quad .. \quad 1\ ] \text{(normalized form)}$$

Figure 3.4 shows B-spline basis functions that result by using open uniform knot vector:

$$U = [\ 0 \quad 0 \quad 0 \quad 0.2 \quad 0.4 \quad 0.6 \quad 0.8 \quad 1 \quad 1 \quad 1\ ]$$

The resulting open uniform basis functions yield curves that behave most nearly like Bezier curves. Section 3.6 gives more details about Bezier curve.

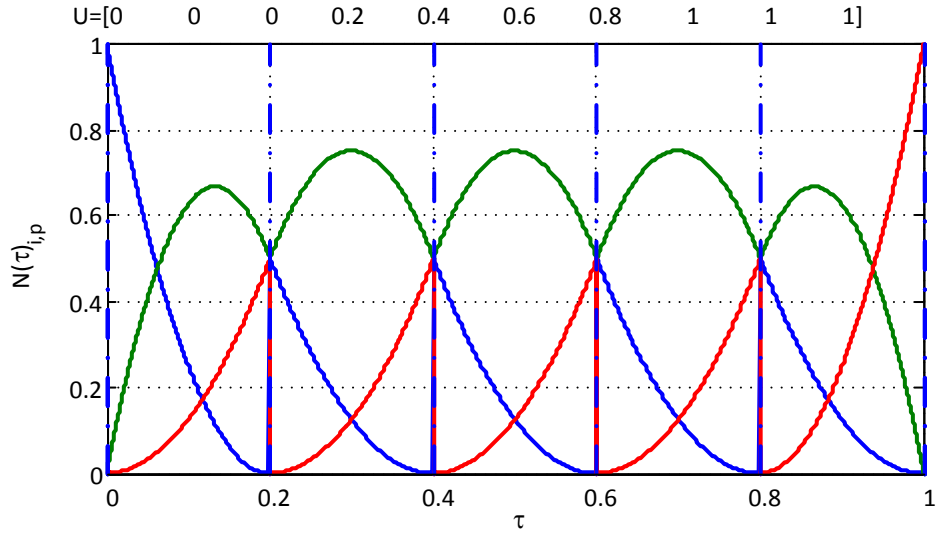

Figure 3.4 B-spline basis function p=2 with open uniform knot vector

Nonuniform knot vectors may have either unequally spaced and/or multiple internal knot values. They may be periodic or open. This research focus on the open/nonperiodic, and nonuniform knot vector which has the general form:

$$U = [a, \ldots, a, u_{p+1}, \ldots, u_{m-p-1}, b, \ldots, b] \tag{3.8}$$

where the first and end knot has a $p+1$ multiplicity. Hence for the normalized knot vector $a = 0$, $b = 1$.

3.5.2 B-spline Curves Properties

The curve that is given by (3.6) can be written in equivalent form:

$$P(\tau) = \sum_{i=0}^{n} R_{i,p}(\tau) C_i \tag{3.9}$$

$$R_{i,p}(\tau) = \frac{w_i N_{i,p}(\tau) C_i}{\sum_{i=0}^{n} w_i N_{i,p}(\tau)}; \quad a \leq \tau \leq b \tag{3.10}$$

where $R_{i,p}(\tau)$ are rational basis functions. The analytical properties of $R_{i,p}(\tau)$ determine the geometric behaviour of the curves [62].

- Generalization: If $w_i = 1$ then

$$R_{i,p}(\tau) = \begin{cases} B_{i,p}(\tau) & \text{if} \quad U = [0,0,\ldots,0,1,1,\ldots,1] \\ N_{i,p}(\tau) & \text{otherwise} \end{cases} \tag{3.11}$$

  where the zeros and ones in U are repeated with multiplicity $p+1$, and $B_{i,p}$ is the Bernstein polynomial of degree p, which are the basis of Bezeir curve.

- If $n = p$ and $U = [0,\ldots,0,1,\ldots,1]$ then $P(\tau)$ is a Bezier curve.

- End point interpolation: $P(0) = C_0$ and $P(1) = C_n$.

- Strong convex hull property: the curve is contained in the convex hull of its control polygon. If $\tau \in [u_i, u_i+1[, p \leq i < m-p-1$, then $P(\tau)$ is in the convex hull of control points $C_{i-p},\ldots,C_i$. Figure 3.5 shows quadratic ($p = 2$) B-spline curve and its B-spline basis functions. The control points are:

$$C = \{(1,1),(0.5,2),(4,2),(3,1),(2,0),(4.5,0.2)\} \tag{3.12}$$

  and the knot vector is

$$U = [0,0,0,0.25,0.5,0.75,1,1,1] \tag{3.13}$$

  It can be seen that the part of the curve that results from $\tau \in [\tau_3,\tau_4[= [0.25,0.5[$ is contained in the convex hull which is created by control points C_2, C_3, C_4. Due to the strong convex hull property it can be seen in Figure 3.5 the straight line that exists in the duration $\tau \in [\tau_4,\tau_5[= [0.5,0.75[$ is due to the co-linear control points C_3, C_4, C_5.

- Portion of unity: $\sum_i R_{i,p}(\tau) = 1$

- Differentiability: The rational basis functions are infinity continuously differentiable in the interior of a knot span [61]. But at a knot they are $p-k$ times continuously differentiable where k is the multiplicity of the knot. Due to this property cusps can occur. Figure 3.6 shows a quadratic curve $p = 2$ that is generated by using knot vector $U = [0,0,0,\frac{1}{5},\frac{2}{5},\frac{3}{5},\frac{4}{5},\frac{4}{5},1,1,1]$ it can be seen that the cusp occurs because the knot at $\frac{4}{5}$ has multiplicity $k = 2$ so the curve is $p-k = 0$ continuously differentiable at this knot.

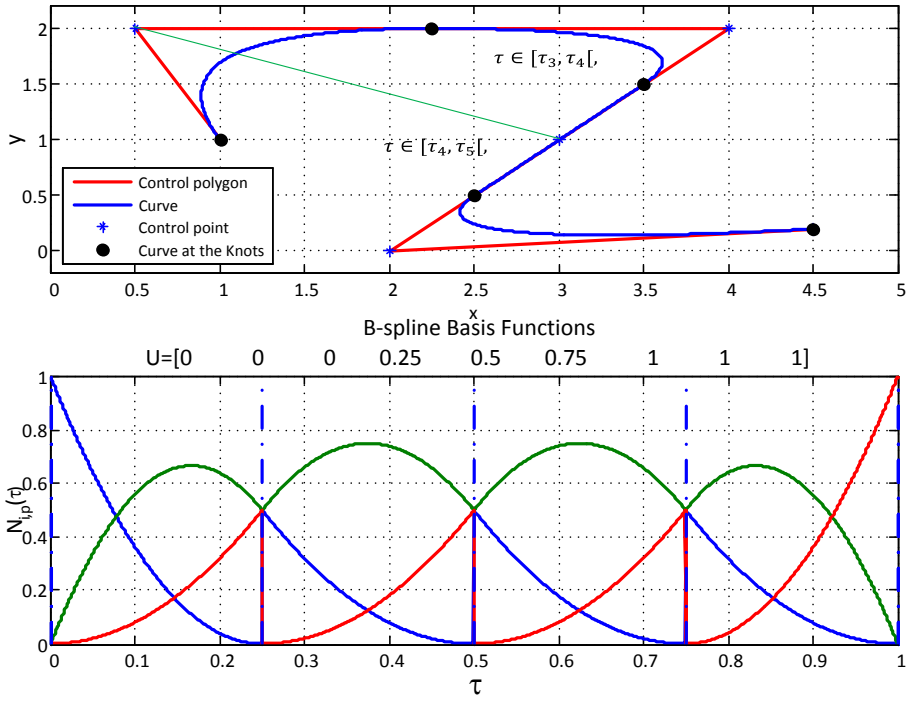

Figure 3.5 Convex hull property of B-spline curve

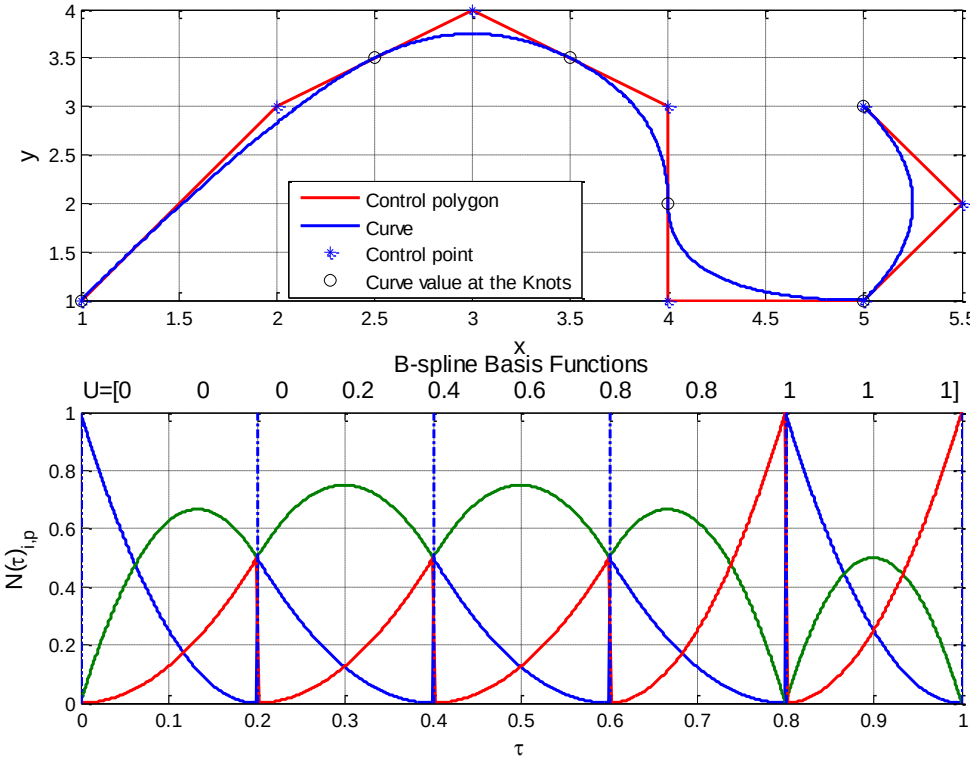

Figure 3.6 A quadratic curve $p = 2$ with cusp

It worth mentioning that the continuity can be recovered by control point modification. For example, by reallocating the control point C_6 to be co-linear with C_4 and C_5 as shown in Figure 3.7.

- Figure 3.7 shows that the moving of C_6 only modifies the curve in the interval $[\tau_6, \tau_9[$ this property is called local modification scheme. Generally speaking, moving C_i will change the B-spline curve $P(\tau)$ only in the interval $[\tau_i, \tau_{i+p+1}[$.

- The control polygon which formed by the control point is a linear approximation to the curve.

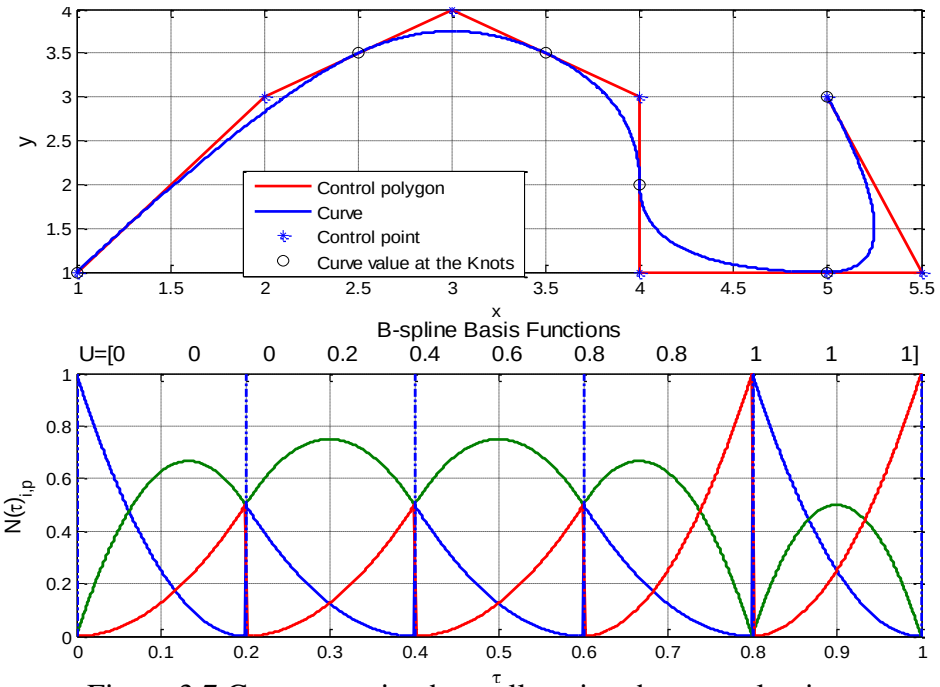

Figure 3.7 Cusp removing by reallocating the control point

3.5.3 Derivatives of B-Spline Curves

The derivatives of B-spline curves can be calculated simply by computing the derivatives of their B-spline basis functions. $P^{(k)}(\tau)$ which is the k^{th} derivative of $P(\tau)$ is given by:

$$P^{(k)}(\tau) = \sum_{i=0}^{n} N_{i,p}^{(k)}(\tau) C_i \tag{3.14}$$

where $N_{i,p}^{(k)}(\tau)$ is the k^{th} derivative of B-spline basis functions which can be calculated recursively:

$$N_{i,p}^{(k)}(\tau) = \left[\frac{N_{i,p-1}^{(k-1)}(\tau)}{u_{i+p} - u_i} - \frac{N_{i+1,p-1}^{(k-1)}(\tau)}{u_{i+p+1} - u_{i+1}} \right] p \tag{3.15}$$

The first derivative of B-spline basis function is:

$$N_{i,p}^{(1)}(\tau) = \frac{p}{u_{i+p} - u_i} N_{i,p-1}(\tau) - \frac{p}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(\tau) \tag{3.16}$$

3.6 Bezier Curve

Bezier curves are special case of NURBS when the all wights equal to unity $w_i = 1$ and the knot vector takes the following form: $U = [0,0,\ldots,0,1,1,\ldots,1]$

$$P(\tau) = \sum_{i=0}^{n} B_i(\tau) C_i \tag{3.17}$$

where $P(\tau)$ is the Bezier curve, n is the order of the polynomial, τ is the curve parameter, $B_i(\tau)$ is the i^{th} order basis function that can be calculated by (3.11), and C_i are the coefficients (control points) of i^{th} basis function.

A 6^{th} order Bezier curves are used to describe the local trajectory profiles in this thesis. Using Bezeir curves to describe local trajectories will reduce the overall dimension of the problem [10] (the optimisation problem will be to find the control points (C_i), which also called the polynomial coefficients). Sixth order curves were used in [10] for local motion planning for a small quad-rotor UAV as they are considered to provide a good compromise between dynamic flexibility over the design horizon and number of design variables. The small quad-rotor UAV has more maneuverability than the fixed wing UAV (a fixed wing UAV dose not have the ability to fly backward and hovering like a quad-rotor UAV). However, one can consider a lower order of Bezeir curve but the order must not be lower than 4 as these curves are differentiated to calculate speed, acceleration, and the rate of acceleration. Study the effect of using different order of the polynomial on the trajectory planning algorithm is out of the scope of this thesis and it can be curried out as a future work.

The basis functions that generate the Bezier curves are known as Bernstein polynomials and they are calculated by using (3.7-3.11), thus, for example, the 6^{th} order Bezier curve basis

functions are:

$$
\begin{aligned}
B_0 &= (1-\tau)^6 \\
B_1 &= 6\tau(1-\tau)^5 \\
B_2 &= 15\tau^2(1-\tau)^4 \\
B_3 &= 20\tau^3(1-\tau)^3 \\
B_4 &= 15\tau^4(1-\tau)^2 \\
B_5 &= 6\tau^5(1-\tau) \\
B_6 &= \tau^6
\end{aligned}
\tag{3.18}
$$

Figure 3.8 shows Bezier curve ($p = 6$), and its basis functions (Bernstein basis function), it can be noticed that the first basis function which is B_0 has significant effect on the start point of the curve, while B_6 controls the end point of the curve and the rest of the basis functions have no effect on the start and the end points. This is one of advantages of the Bezier curve, this property will reduce the computational time during trajectory optimisation. Substituting (3.18) in (3.17) gives a 6^{th} order polynomial (Bezier curve), after deciding the order of the polynomial that is going to be used in generating the local trajectory, the trajectory shape will vary with coefficients C_i (control points). The Bezier curve shaping in Figure 3.8 is generated by using $C = \{(1,2),(2,3),(3,2),(4,4),(5,4),(6,0),(7,3)\}$ as control points.

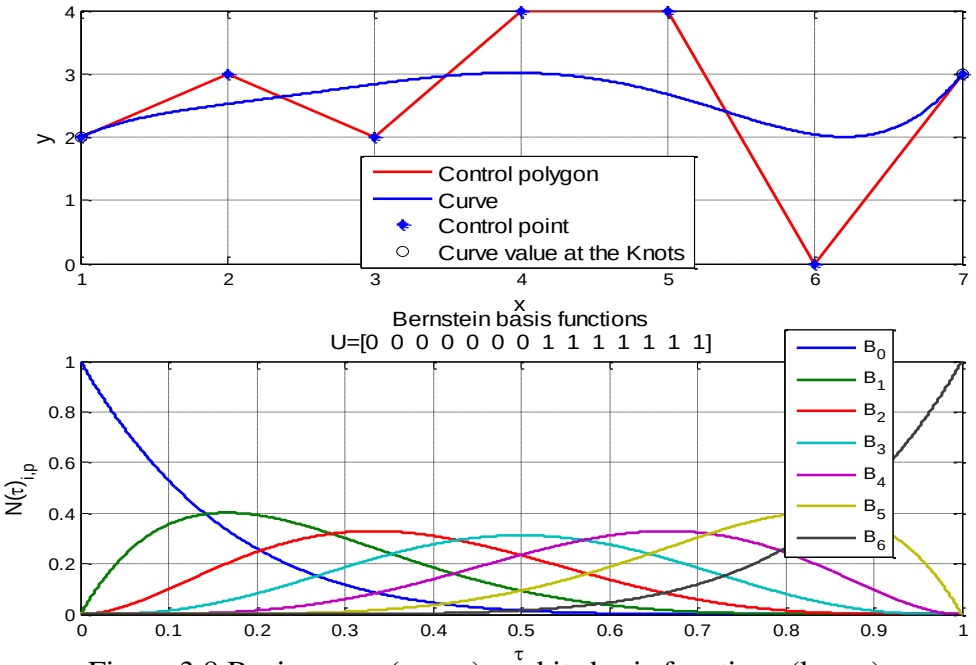

Figure 3.8 Bezier curve(upper), and its basis functions (lower)

3.6.1 Trajectory Profiles Description Using Polynomial Functions

As mentioned earlier using the polynomial functions to describe the trajectory profiles (i.e. positions, velocities, accelerations, and jerks) reduces the computational time for trajectory optimisation. The speed profiles in forward (u), lateral (v), and vertical (w) axes can be written by using polynomial functions (6^{th} order Bezier function):

$$\begin{aligned} u(\tau) &= c_0^u B_0(\tau) + c_1^u B_1(\tau) + c_2^u B_2(\tau) + \cdots + c_6^u B_6(\tau) \\ v(\tau) &= c_0^v B_0(\tau) + c_1^v B_1(\tau) + c_2^v B_2(\tau) + \cdots + c_6^v B_6(\tau) \\ w(\tau) &= c_0^w B_0(\tau) + c_1^w B_1(\tau) + c_2^w B_2(\tau) + \cdots + c_6^w B_6(\tau) \end{aligned} \tag{3.19}$$

Using sixth order polynomial functions to describe the speed profiles gives a good flexibility over the design horizon with an acceptable number of design variables (the polynomial coefficients) [56]. Calculation of acceleration, jerk, and position profiles can be done by taking the first derivative of (3.19) for the acceleration profiles, the second derivative of (3.19) for the jerk profiles, and integration of (3.19) for the position profiles. In order to do so, a relationship between the curve parameter τ and the time t must be defined. A fixed time horizon (t_h) was used so t can be represented by:

$$t = t_h.\tau \tag{3.20}$$

hence the acceleration profiles can be calculated:

$$\frac{du}{dt} = \frac{du}{d\tau}\cdot\frac{d\tau}{dt} = \frac{1}{t_h}\cdot\frac{du}{d\tau} \Rightarrow \tag{3.21}$$

$$\dot{u}(\tau) = \frac{1}{t_h}\left(c_0^u\frac{dB_0(\tau)}{d\tau} + c_1^u\frac{dB_1(\tau)}{d\tau} + c_2^u\frac{dB_2(\tau)}{d\tau} + \cdots + c_6^u\frac{dB_6(\tau)}{d\tau}\right) \tag{3.22}$$

and

$$\frac{d^2u}{dt^2} = \frac{1}{t_h^2}\cdot\frac{d^2u}{d\tau^2} \tag{3.23}$$

hence

$$\ddot{u}(\tau) = \frac{1}{t_h^2}\left(c_0^u\frac{d^2B_0(\tau)}{d\tau^2} + c_1^u\frac{d^2B_1(\tau)}{d\tau^2} + c_2^u\frac{d^2B_2(\tau)}{d\tau^2} + \cdots + c_6^u\frac{d^2B_6(\tau)}{d\tau^2}\right) \tag{3.24}$$

The acceleration and jerk profiles for the lateral and vertical axis can be calculated in a similar way. The position profiles are calculated by integrating the basis functions with

respect to the time t, this can be done by substituting $\tau = t/t_h$ in (3.18), hence:

$$\begin{aligned}
B_0^{int} &= \int_0^{t_h} B_0(t)dt = t - \frac{3t^2}{t_h} + \frac{5t^3}{t_h^2} - \frac{5t^4}{t_h^3} + \frac{3t^5}{t_h^4} - \frac{t^6}{t_h^5} + \frac{t^7}{7t_h^6} \\
B_1^{int} &= \int_0^{t_h} B_1(t)dt = \frac{3t^2}{t_h} - \frac{10t^3}{t_h^2} + \frac{15t^4}{t_h^3} - \frac{12t^5}{t_h^4} + \frac{5t^6}{t_h^5} - \frac{6t^7}{7t_h^6} \\
B_2^{int} &= \int_0^{t_h} B_2(t)dt = \frac{5t^3}{t_h^2} - \frac{15t^4}{t_h^3} + \frac{18t^5}{t_h^4} - \frac{10t^6}{t_h^5} + \frac{15t^7}{7t_h^6} \\
B_3^{int} &= \int_0^{t_h} B_3(t)dt = \frac{5t^4}{t_h^3} - \frac{12t^5}{t_h^4} + \frac{10t^6}{t_h^5} - \frac{20t^7}{7t_h^6} \\
B_4^{int} &= \int_0^{t_h} B_4(t)dt = \frac{3t^5}{t_h^4} - \frac{5t^6}{t_h^5} + \frac{15t^7}{7t_h^6} \\
B_5^{int} &= \int_0^{t_h} B_5(t)dt = \frac{t^6}{t_h^5} - \frac{6t^7}{7t_h^6} \\
B_6^{int} &= \int_0^{t_h} B_6(t)dt = \frac{t^7}{7t_h^6}
\end{aligned} \tag{3.25}$$

The receding horizon trajectory profiles are discretised into n steps within the period $0 \leq \tau \leq 1$ to evaluate the cost function at each step during the optimisation process. Discretised trajectory profiles can be calculated by discretising the basis function into n steps, so the resulted discrete basis function can be written as matrices as follow:

$$B = \begin{pmatrix} B_0(\tau_1) & B_0(\tau_2) & \cdots & B_0(\tau_n) \\ B_1(\tau_1) & B_1(\tau_2) & \cdots & B_1(\tau_n) \\ \vdots & \vdots & \cdots & \vdots \\ B_6(\tau_1) & B_6(\tau_2) & \cdots & B_6(\tau_n) \end{pmatrix} \tag{3.26}$$

$$B' = \begin{pmatrix} \frac{dB_0(\tau_1)}{d\tau} & \frac{dB_0(\tau_2)}{d\tau} & \cdots & \frac{dB_0(\tau_n)}{d\tau} \\ \frac{dB_2(\tau_1)}{d\tau} & \frac{dB_2(\tau_2)}{d\tau} & \cdots & \frac{dB_2(\tau_n)}{d\tau} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{dB_6(\tau_1)}{d\tau} & \frac{dB_6(\tau_2)}{d\tau} & \cdots & \frac{dB_6(\tau_n)}{d\tau} \end{pmatrix} \tag{3.27}$$

The same can be applied to calculate B'' and B^{int}, all these matrices can be calculated off-line, hence the on-line trajectory profiles calculation is reduced to simple matrix multiplication:

$$u = C^{u^T} B \tag{3.28}$$

$$\dot{u} = \frac{1}{t_h} C^{u^T} B' \tag{3.29}$$

$$\ddot{u} = \frac{1}{t_h^2} C^{u^T} B'' \tag{3.30}$$

$$x = x_0 + C^{u^T} B^{int} \tag{3.31}$$

where C^{u^T} is the vector of coefficients for forward axis:

$$C^{u^T} = \begin{bmatrix} c_0^u & c_1^u & \cdots & c_6^u \end{bmatrix} \tag{3.32}$$

The trajectory profiles for lateral and vertical axis can be calculated in a similar way. The optimisation problem is to find the polynomial coefficients, till now there are twenty-one coefficients that have to be calculated (seven coefficients for each axes).

3.6.2 Boundary Conditions

Aircraft states can be measured by the sensing unit, the current states can be used as boundary conditions that guarantee a smooth transition from current states to target states. Substituting $\tau = 0$ in trajectory profiles (3.19), (3.21), and (3.23) gives:

$$\begin{aligned} c_0^u &= u_0 \\ c_1^u &= \frac{t_h}{6}\dot{u}_0 + c_0^u \\ c_2^u &= \frac{t_h^2}{30}\ddot{u}_0 - c_0^u + 2c_1^u \end{aligned} \tag{3.33}$$

where:
u_0: is the initial forward speed,
$\dot{u}_0$: is the initial forward acceleration,
$\ddot{u}_0$: is the initial forward jerk.
Similarly for the lateral and vertical trajectory profiles, the first three coefficients can be found, so the number of design variables has been reduced from twenty-one into twelve variables.

3.7 Local Trajectory Optimisation

The optimal local trajectory profiles can be achieved by finding values of design variables that minimize a defined cost function and satisfy all constraints. In order to determine an optimal control trajectory for aircraft using direct methods, the optimal control problem is formulated in output space rather than control or input space. However, the output design space technique is only available when the system is differentially flat [63]. A system is differentially flat if its states and inputs can be expressed as functions of the output vector and its derivatives [63, 64]. Fortunately most fixed-wing aircraft systems can be considered as differentially flat systems. The following discussion shows that the fixed wing aircraft possesses the property of flatness.

3.7.1 Differential Flatness of the Fixed-Wing Aircraft

A fixed wing aircraft dynamics can be expressed by a three Degree of Freedom (3-DoF) point-mass model [65].

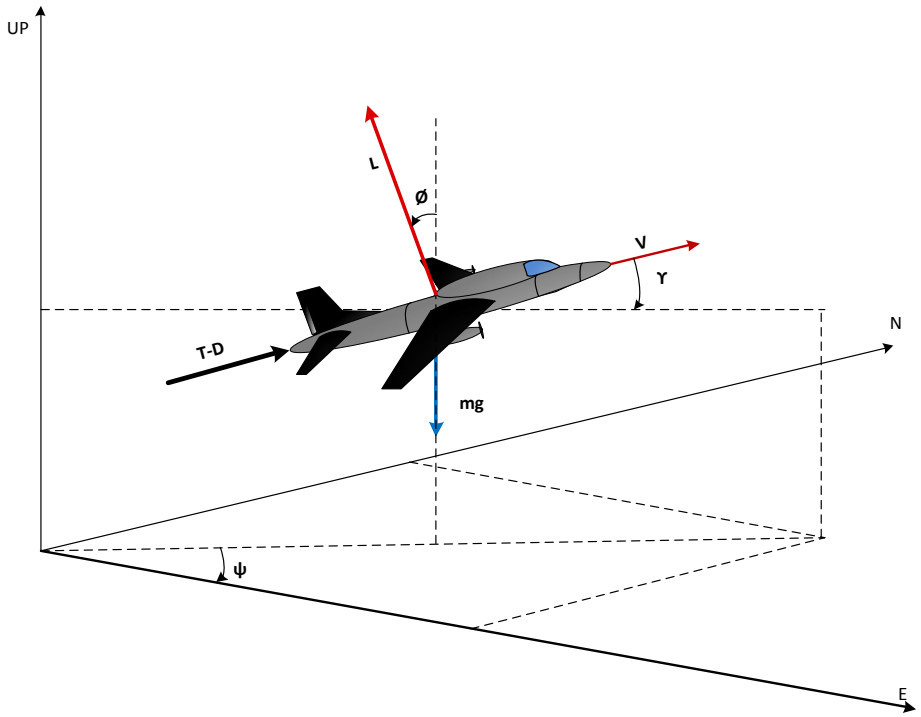

Figure 3.9 Aircraft point-mass model

Following assumptions are taken for the 3-DoF point-mass model:

1. Aircraft mass is constant.

2. Still wind.

3. Zero sideslip

4. Zero angle of attack, hence flight path angle γ equals the pitch attitude θ.

Figure 3.9 shows the coordinate system used for the derivation of the point-mass model.

Thus, the aircraft mathematical model is given by:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\gamma} \\ \dot{\psi} \\ \dot{V} \end{bmatrix} = \begin{bmatrix} V\cos\gamma\cos\psi \\ V\cos\gamma\sin\psi \\ V\sin\gamma \\ \frac{g}{V}(n\cos\phi - \cos\psi) \\ \frac{g}{V}\frac{n\sin\phi}{\cos\gamma} \\ \frac{T-D}{m} - g\sin\gamma \end{bmatrix} \tag{3.34}$$

where:
x, y, z: aircraft center of gravity coordinates in earth axis,
γ: flight path angle,
ψ : heading angle,
V: aircraft speed,
g: gravity acceleration,
ϕ: bank angle,
T: thrust,
D: drag,
m: total mass,
n: load factor which can be expressed :

$$n = \frac{L}{mg} \tag{3.35}$$

where L is the total aircraft lift given as follows:

$$L = \frac{1}{2}\rho S C_L V^2 \tag{3.36}$$

ρ: is the air density, which is modelled as a function of altitude $\rho(h)$ according to the International Standard Atmosphere (ISA).
S: is the wing area.

C_L: is the lift coefficient given by:

$$C_L = C_{L_0} + \alpha C_{L_\alpha} \tag{3.37}$$

where C_{L_0} , C_{L_α} are the zero-angle-of-attack lift coefficient, lift curve slope.
The total aerodynamic drag D is given by [66]:

$$D = \frac{1}{2}\rho S C_D V^2 \tag{3.38}$$

where C_D is the drag coefficient given by:

$$C_D = C_{D0} + kC_L^2 \tag{3.39}$$

where C_{D0} and k are the minimum drag coefficient and induced drag factor respectively.
The output vector can be expressed by:

$$Y = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{3.40}$$

The input vector can be taken as:

$$U = \begin{bmatrix} \phi \\ T \\ n \end{bmatrix} \tag{3.41}$$

By modifying (3.34) one can find:

$$V = \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2} \tag{3.42}$$

$$\gamma = \arcsin(\frac{\dot{z}}{V}) \tag{3.43}$$

$$\psi = \arcsin(\frac{\dot{y}}{V\cos\gamma}) \tag{3.44}$$

$$\phi = \arctan(\frac{\dot{\psi} V\cos\gamma}{g\cos\gamma + V\dot{\gamma}}) \tag{3.45}$$

$$n = \frac{g\cos\gamma + V\dot{\gamma}}{g\cos\phi} \tag{3.46}$$

$$T = D + m\dot{V} + mg\sin\gamma \tag{3.47}$$

It can be noticed from (3.42-3.38) that the controls and the states of the system can be expressed as functions of the output vector and its derivatives. Hence the system is differentially flat. So the optimal control problem can be formulated in the output space rather than the control space.
The aircraft and the obstacles constraints are augmented in the cost function by using a penalty function method.

3.7.2 Aircraft Constraints

Penalty functions are used to ensure that the resulting optimal trajectory will be achieved without exceeding the aircraft performance limits. Hence, the cost function is augmented with additional terms that prevent the solution approaching the limits. The Yukawa potential function is used in this thesis:

$$C^p = A_p \frac{e^{-\alpha_p d_p}}{d_p} \tag{3.48}$$

where:
C^p: aircraft performance constraint term to be added to the total cost function,
A_p: scaling factor,
α_p: decay rate,
d_p: performance margin given by:

$$d_p = 1 - 1\left(\frac{\text{current state value}}{\text{state maximum} \setminus \text{minimum value}}\right) \tag{3.49}$$

To avoid zero value of d_p a minimum performance margin value d_{min} must be defined so: if $d_p \leq d_{min}$ then $d_p = d_{min}$. The role of the penalty function can be understood from Figure 3.10 which shows the value of the potential function with respect to the value of the performance margin at different values of scaling factor A_p. It can be clearly seen that when the performance margin decreases (which means that when the current state value is close to its limit) the potential function takes a huge value. Thus the total cost function will increase significantly, so the optimisation algorithm will try to find another solution that keeps the aircraft state away from its limits.

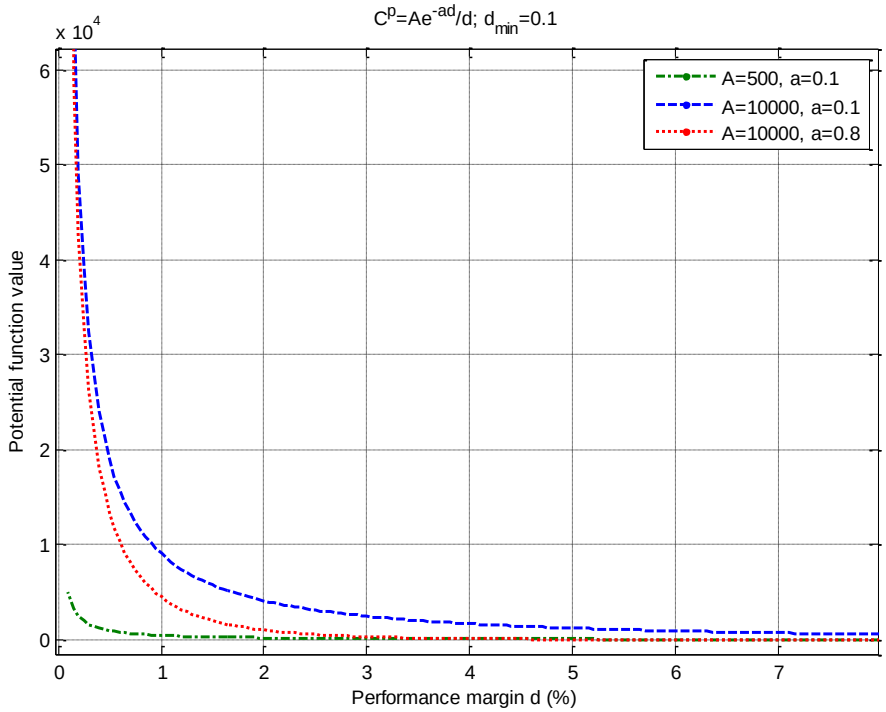

Figure 3.10 Yukawa potential function

3.7.3 Obstacle Constraints

Collision avoidance can be achieved by either putting constraints on the optimisation process or by augmenting a penalty function in the cost function. The second choice is proposed for this thesis, so the cost function can be augmented with a pilot behaviour term. Additionally this choice will simplify the search algorithm in the optimisation process. Thus the total computational time of the optimisation process will be reduced. As for performance constraints, Yukawa potential function is taken to punish the cost function if the aircraft approaches the obstacle. The Yukawa potential function used for obstacle constraints has the same form of that for performance constraints:

$$C^{ob} = A_{ob}\frac{e^{-\alpha_{ob}d_{ob}}}{d_{ob}} \tag{3.50}$$

where:
C^{ob}: penalty term that represent the obstacle constraints,
A_{ob}: scaling factor,
α_{ob}: decay rate,
d_{ob}: distance of the nearest point on the obstacle to the point of interest.

Although using a potential function to describe the obstacle constraints complicates the cost function, it simplifies the search algorithm. Another advantage of using potential function is that it handles the collision event in a manner which is close to human behaviour. For example, an avoidance manoeuvre can vary according to many factors such as aircraft speed, obstacle speed, aircraft manoeuvrability, and obstacle manoeuvrability. Additionally, due to the difficulty in generating a full 3D illustration for the obstacles that can be detected by an on-board sensor unit (this description is very important to add constraints to the optimisation process) the potential function approach does not need a 3D description of the obstacle. It just needs the distance between the aircraft and the nearest point in the obstacle [56]. However, Becerra [67] has shown that ill conditioning and difficulty in choosing a suitable value for the penalty factor are the main disadvantages of the penalty function approach.

3.7.4 Total Cost Function

The following cost function is used in the optimisation process:

$$J = \sum_{i=1}^{n} [\lambda_{position} J_i^{position} + \lambda_{speed} J_i^{speed} + \lambda_{performance} J_i^{performance} + \lambda_{ob} J_i^{ob}] + \lambda_t J^t \quad (3.51)$$

where:

$$J_i^{position} = (x_i^{demand} - x_i^{actual})^2 + (y_i^{demand} - y_i^{actual})^2 + (z_i^{demand} - z_i^{actual})^2 \quad (3.52)$$

$$J_i^{speed} = (u_i^{demand} - u_i^{actual})^2 + (v_i^{demand} - v_i^{actual})^2 + (w_i^{demand} - w_i^{actual})^2 \quad (3.53)$$

$$J^{performance} = \sum_{j=1}^{q} A_p \frac{e^{-\alpha_p d_p}}{d_p} \quad (3.54)$$

$$J_i^{ob} = \sum_{j=1}^{m} A_{ob} \frac{e^{-\alpha_{ob} d_{ob}}}{d_{ob}} \quad (3.55)$$

$$J^t = \lambda_h (\psi_n^{demand} - \psi_n^{actual})^2 + (\lambda_f (\gamma_n^{demand} - \gamma_n^{actual})^2 \quad (3.56)$$

λ: scaling factor,
n: number of points that will be evaluated across the design horizon,
q: number of performance constraints,

m: number of detected obstacles,
ψ: heading angle,
γ: flight path angle,
It can be seen that the cost function given by (3.51) makes a balance between the different terms: trajectory tracking terms ($J^{position}, J^{speed}, J^{t}$); and constraints terms (obstacle avoidance term J^{ob}, performance constraint term $J^{performance}$). This balance can be controlled by changing the scaling factors λ. The scaling factors can be constants or they may vary according to the situation. In other words, the priority of the cost function terms can be varied in order to allow the aircraft to fly safely in different flight scenarios.

3.7.5 Avoiding Local Minima

The performance constraints tend to act as an enclosing boundary around the entire search space, hence are less likely to result in local minimum. Thus, the obstacle constraints are the primary source of the local minima. When obstacles are detected this can have the impact of dividing the feasible design space into unconnected regions, therefore, reducing the effectiveness of the solver of the optimisation problem. The possibility of getting trapped in local minimum is reduced by providing a mechanism for the search to jump to the different regions of the design space. This is achieved by generating a set of candidate trajectories then comparing the cost for each candidate then selecting the one that gives the minimum cost to initiate the optimisation problem solver. The candidate trajectories are generated by applying maximum/minimum inputs to the vehicle model with the current vehicle states as initial states to ensure that the maximum performance manoeuvres in each axis are always available if required. This method does not guarantee that the chosen initial solution lies within the region of the design space that provides the best solution. In this case the input commands are:

$$\begin{aligned} \phi &= \begin{bmatrix} \phi_{min} & \phi_c & \phi_{max} \end{bmatrix} \\ T &= \begin{bmatrix} T_{min} & T_c & T_{max} \end{bmatrix} \\ n &= \begin{bmatrix} n_{min} & n_c & n_{max} \end{bmatrix} \end{aligned} \tag{3.57}$$

where ϕ_c, T_c, and n_c are the current values of the inputs, and $\phi_{min/max}$, $T_{min/max}$, and $n_{min/max}$ are the minimum and maximum values of the inputs which can be calculated from the vehicle specifications (the Aerosonde UAV [68] model and specifications are used here). This combination will produce 27 candidate trajectories.

3.8 Simulation Results

This section demonstrates simulation results of different scenarios. The global trajectory that was used in all scenarios is level flight with constant speed $v = 30m.s^{-1}$ at $1000m$ altitude, with heading $\psi = 0$ rad. Still air is assumed. The receding horizon time is $t_h = 20$ sec and sampling time $t_s = 0.2$ sec. The optimisation process is updated every 2 sec. The system is built in MATLAB/Simulink and the fminunc function is used as a solver for the optimisation problem. Scaling factors values are: $\lambda_{position} = 100$, $\lambda_{speed} = 500$, $\lambda_{performance} = 1$, $\lambda_{ob} = 1$, $\lambda_t = 1$, $\lambda_h = 10$, $\lambda_f = 1$.

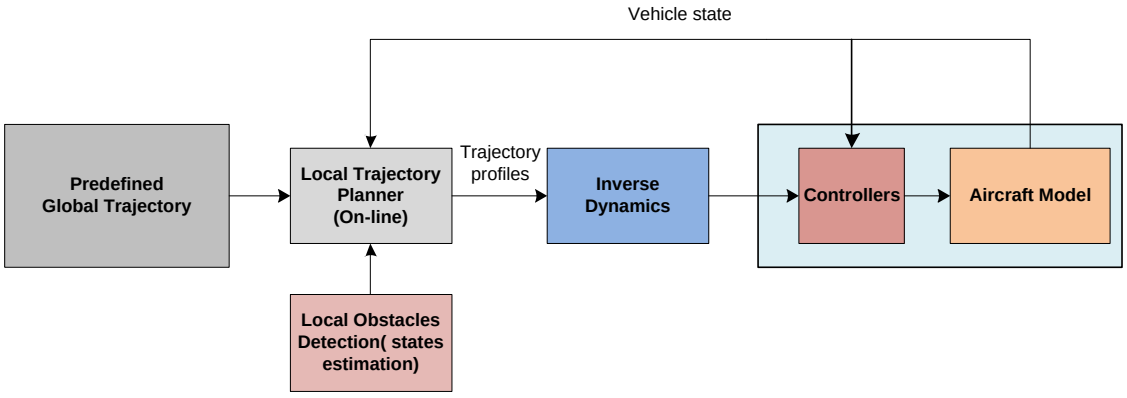

Figure 3.11 Block diagram of the proposed CAS

3.8.1 Global Trajectory Tracking with Static Obstacle Avoidance

In this scenario the initial position of the UAV is higher than the global trajectory by 200m, but the UAV is flying at the same speed and direction as the global trajectory. There is also a static obstacle that the UAV must avoid during the global trajectory tracking process. Figure 3.12 shows the simulation result of this scenario, it can be seen that the UAV is converging to the global trajectory then when the static obstacle appears in its way, the UAV performs the necessary manoeuvre in order to avoid the obstacle. Then the UAV converges again to the global trajectory after passing the static obstacle. Figure 3.13 shows time histories of some state variables of the UAV (position, speed, heading angle ψ, and flight path angle γ) during this scenario.

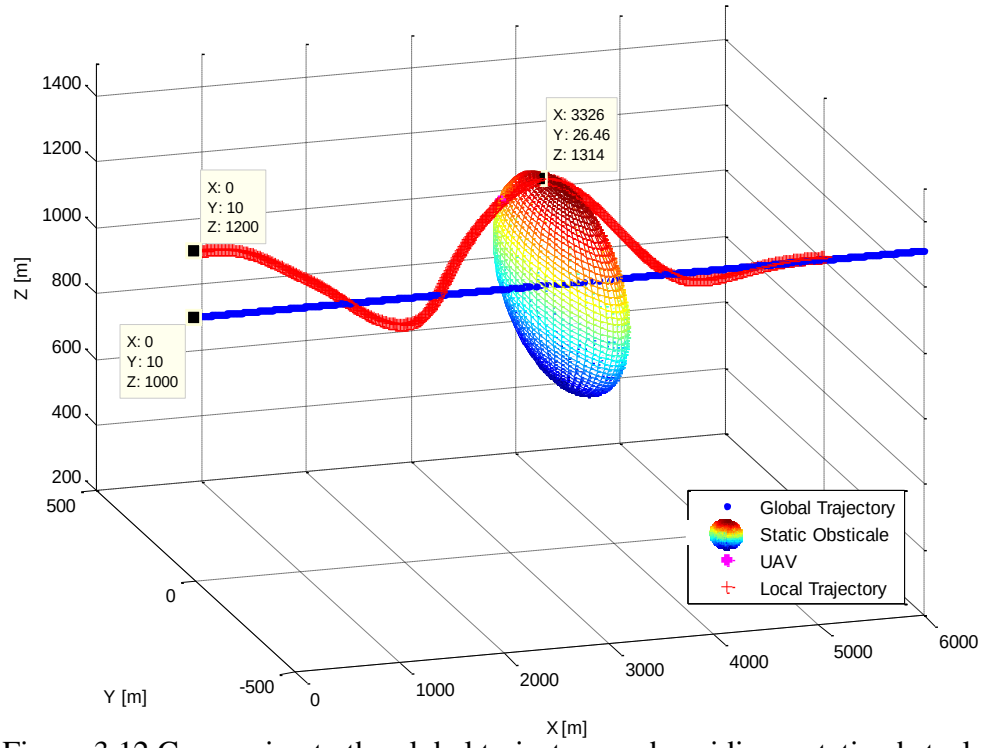

Figure 3.12 Converging to the global trajectory and avoiding a static obstacle

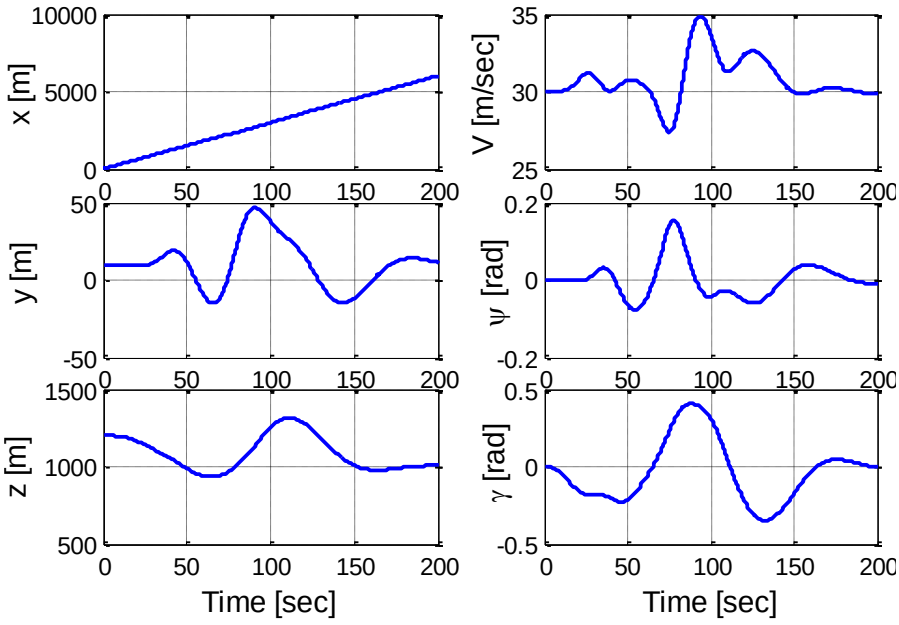

Figure 3.13 Position, speed, ψ, and γ state during the avoidance manoeuvre (static obstacle)

Figure 3.14 gives the normal load factor which is slightly deviating around one and meets the load criteria ($n \in [-1, 2.5]$) given by CS-23 document [69].

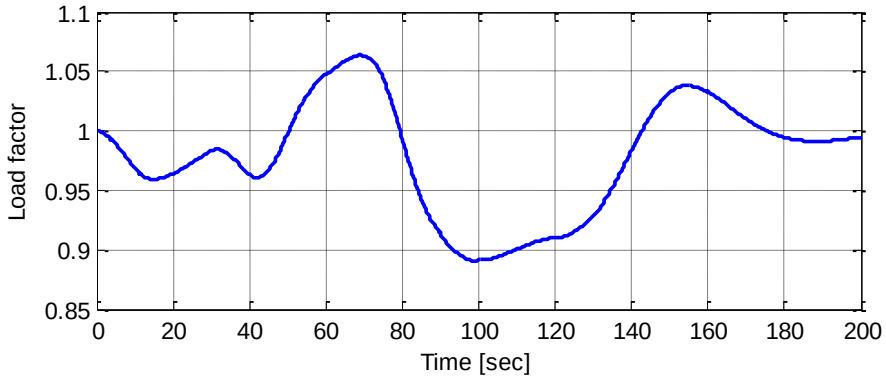

Figure 3.14 Load factor during the avoidance manoeuvre

Note that the resulting optimal trajectory is dependent on the chosen scaling factors (or weightings) of the terms in the objective cost of the optimal control problem. Hence the resulting solution may be sensitive to the choice of the scaling factors. The issue of selecting the best weightings for optimal control problems has been explored for fixed gain feedback controllers (for example see [70, 71]). For trajectory optimization problems, it appears less well-studied [72], but has been considered for model predictive control [73]. In addition, ill conditioning problems can arise when using penalty functions in the cost function and difficulties in choosing a suitable value for the penalty factor makes it difficult to tune the scaling factors [67]. In this thesis, trial and error is used for selecting the scaling factors.
Figure 3.15 and Figure 3.16 show the same simulation results ,but from different view angles, that show the sensitivity of the generated trajectory to the scaling factors values. These results show the effects of scaling factors λ on the UAV avoidance manoeuver that generated to avoid a static obstacle and track a global trajectory. It can be seen that there are obvious differences between the generated avoidance trajectories, not only in the shape of the trajectories, but also in the direction of the avoidance manoeuvrers. As it can be seen in Figure 3.15 and Figure 3.16 there are four trajectories each one is generated with different scaling factors that are given in Table 3.1:

Table 3.1 Scaling factors values

Scaling factor	$\lambda_{position}$	λ_{speed}	$\lambda_{performance}$	λ_{ob}	λ_{t}
1st avoidance manoeuvre	1	1	1	1	1
2nd avoidance manoeuvre	1	1	1	10	1
3rd avoidance manoeuvre	100	1	10	1	1
4th avoidance manoeuvre	10	1	1	1	1

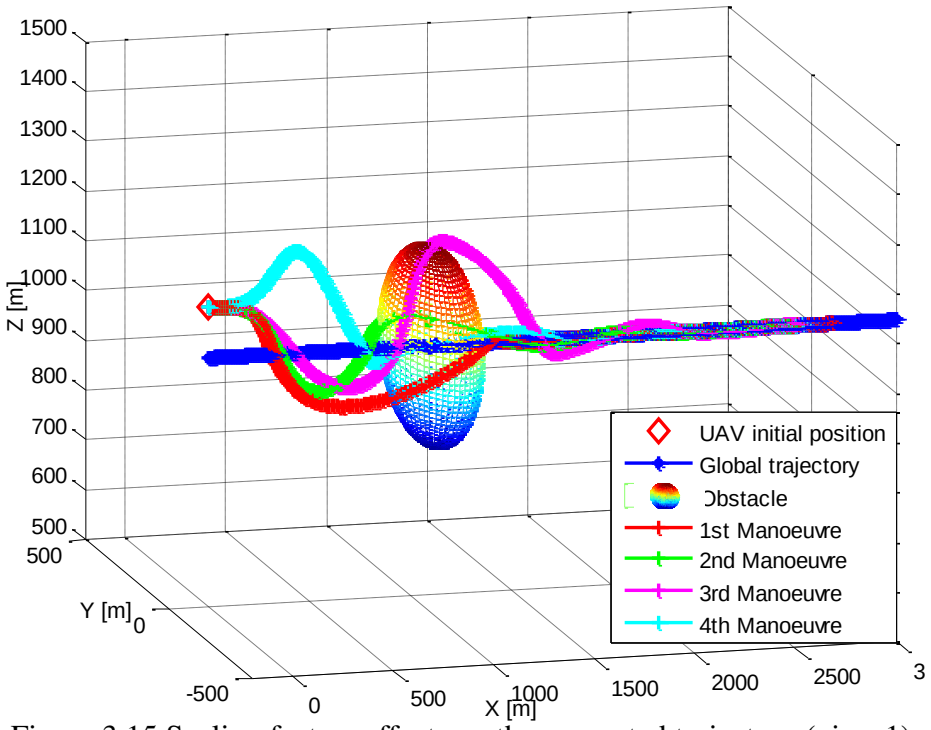

Figure 3.15 Scaling factors effects on the generated trajectory (view 1)

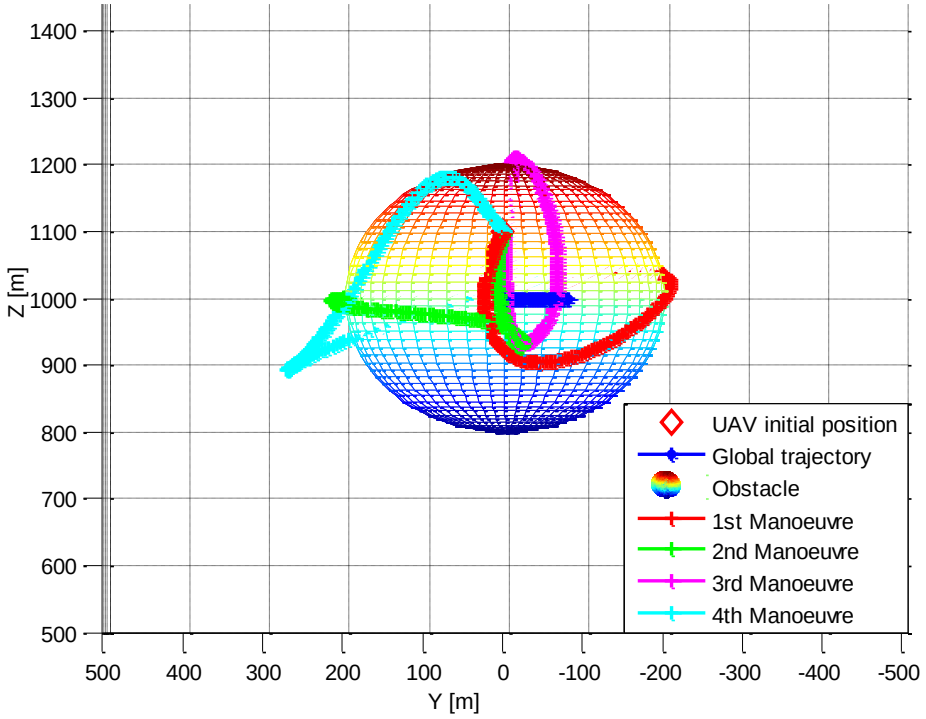

Figure 3.16 Scaling factors effects on the generated trajectory (view 2)

3.8.2 Global Trajectory Tracking with Two Moving Intruders

In this case the UAV is going to face two types of intruders so there are two collision scenarios, a head-on scenario, and an overtaking scenario.
The UAV has the following initial flight state: level flight at the initial position (0,10,1000)m, heading $\psi = 0$, and constant speed $v = 30m.s^{-1}$. The first Intruder (Intruder1) has the following initial state: level flight at initial position (2000,10,1000), heading $\psi = \pi$ rad, and constant speed v=18 $m.s^{-1}$. The second Intruder (Intruder2) has the following initial state: level flight at initial position (2100,10,1000), heading $\psi = 0$ rad, and constant speed v=15 $m.s^{-1}$. So the first intruder will make a head-on collision scenario, while the UAV is going to overtake the second intruder.
The protection zone around each intruder was chosen to be 200 m, so the distance between the UAV and the intruders should not become less than 200 m. Figure 3.17 shows the UAV trajectory, it can be seen that the UAV avoided both collision scenarios and returned to the global trajectory when it finished overtaking the second intruder.

The spheres that appear in the Figure 3.17 represent the protection zones around the intruders at their final position. Figure 3.18 gives time histories of some UAV state variables (position, speed, heading angle ψ, and flight path angle γ) during the scenarios.

As mentioned earlier the distance between the UAV and intruders must be more than 200m. This was achieved by choosing $d_{ob} = 200$ in the obstacle cost function (3.55). Figure 3.20 shows the distances between the UAV and the intruders. It can be noticed that the UAV- intruder distance does not become less than 200m.

Load factor during the avoidance manoeuvre is given in Figure 3.19. As the normal load factor is in within the range of [-1,2.5] it meets criteria that given in [69].

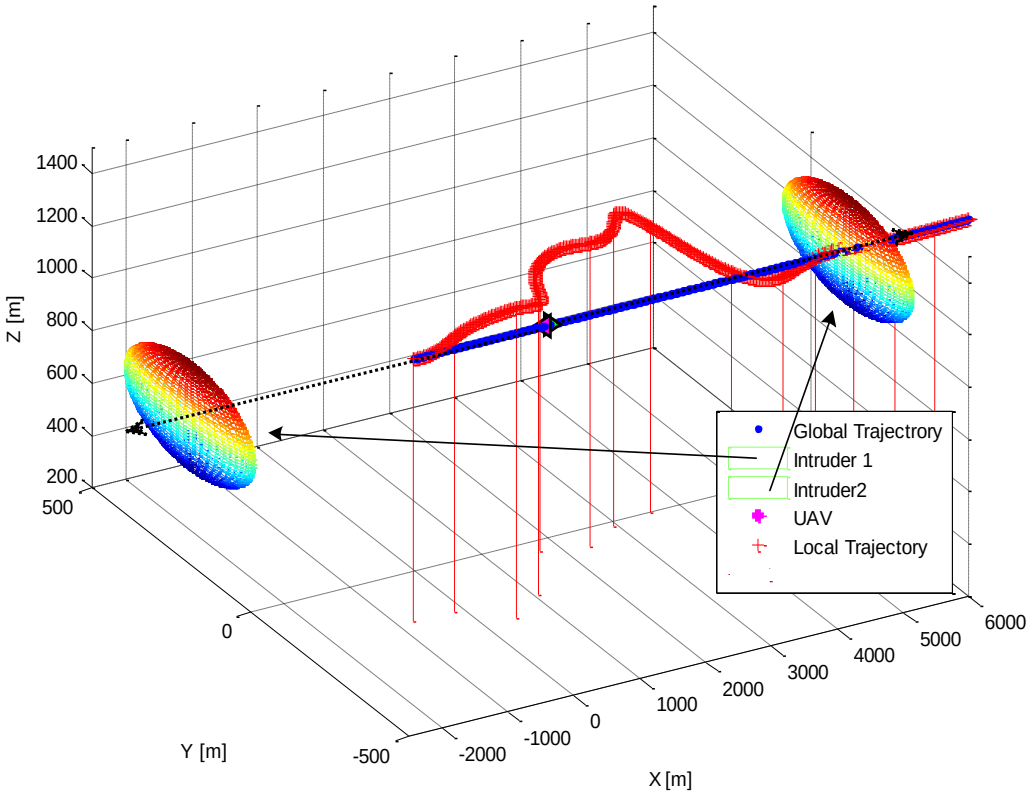

Figure 3.17 Collision avoidance of two scenarios, head-on (intruder1), overtaking (intruder2)

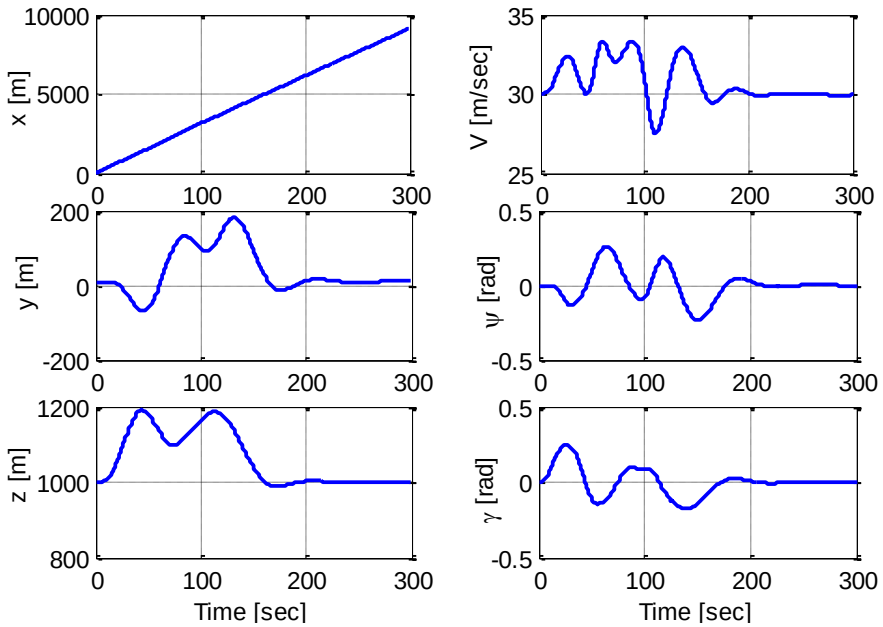

Figure 3.18 Position, speed, ψ, and γ state variables during the avoidance manoeuvre

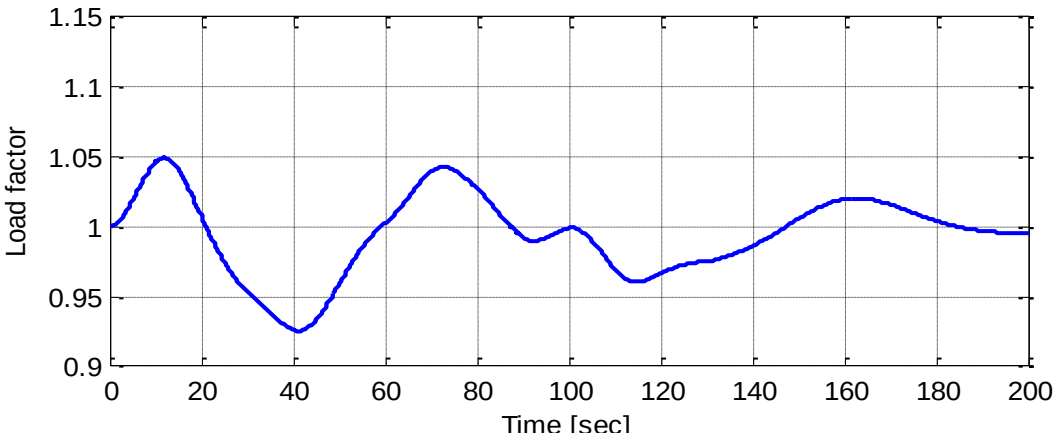

Figure 3.19 Load factor during the avoidance manoeuvre

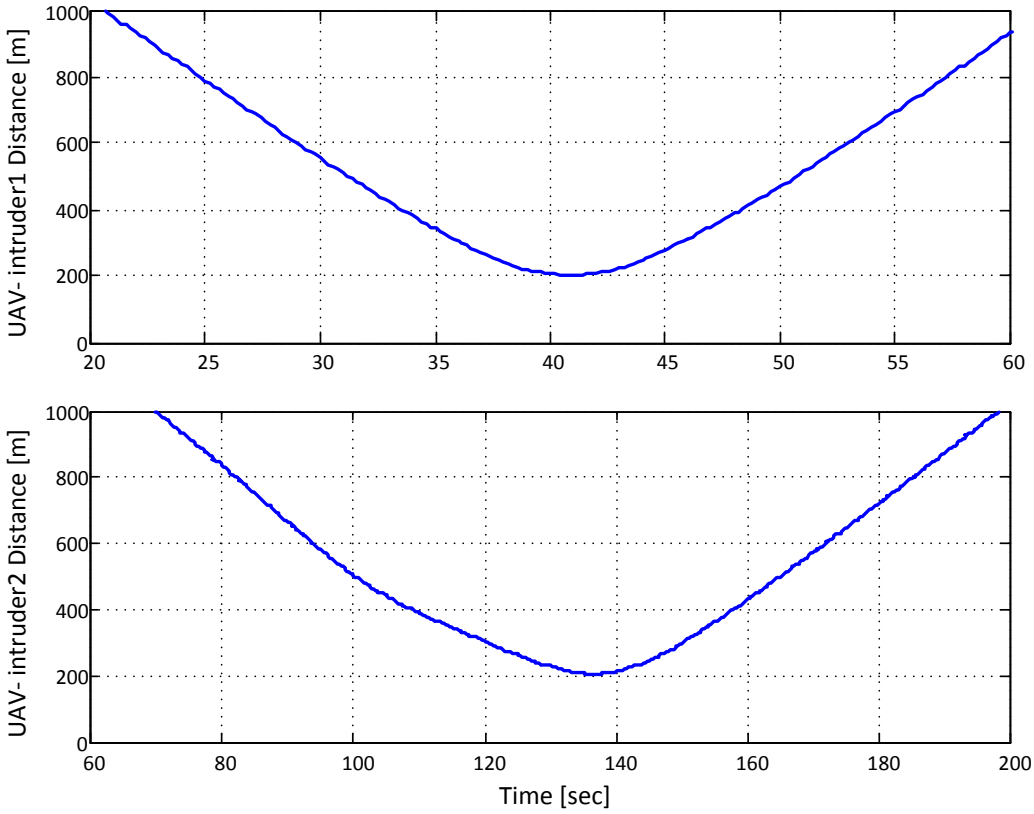

Figure 3.20 UAV-intruder1 distance (upper), UAV-intruder2 distance (lower)

3.9 Summary

This chapter presents a local trajectory planning algorithm that is used for the collision avoidance system for a fixed-wing aircraft. A real-time collision avoidance algorithm is developed based on parameterizing an optimal control problem with B-spline curves. The optimal control problem is formulated in output space rather than control or input space, which is feasible because of the differential flatness of the system for a fixed wing aircraft. The flat output trajectory is parameterized using a B-spline curve representation. In order to reduce the computational time of the optimisation problem, the aircraft and obstacle constraints are augmented in the cost function using a penalty function method. The simulation results show that the proposed approach allows the UAV to track a predefined global trajectory as well as avoiding collisions with different types of conflict scenarios in real-time.

Chapter 4

Decision Making System Based on The Rules of the Air

4.1 Introduction

Operation of Unmanned Aerial Vehicles (UAV's) in civil/non-segregated airspace is restricted by the policies of aviation authorities which require full compliance with rules and obligations that apply for manned aircraft [3]. Trajectory tracking and collision avoidance are issues that a UAV must deal with in a way that gives the UAV the ability to avoid conflict situations. Thus, any UAV that will be operated in civil/non-segregated airspace must be equipped with a collision avoidance system that has the ability to avoid conflict scenarios in full compliance with the rules of the air. UAV operations in UK airspace requirements are given in the document CAP-722 (Unmanned Aircraft System Operations in UK Airspace-Guidance) [49] which issued by Civil Aviation Authority in the UK (CAA) and emphasizes that:

> *UAV operating in the UK airspace must satisfy the safety and operational conditions exactly as manned aircraft. So the UAV must behave as same as manned aircraft in all conditions, that means the collision avoidance system must avoid a conflict according the rules of the air .*

Much research is being undertaken to enable the routine use of UAV's in all classes of airspace without the need for restrictive or specialized conditions of operation. The ASTRAEA program [3] is one example.

Firstly, this chapter gives brief descriptions of airspace classes, and the rules of the air (mainly the Visual Flight Rules (VFR)), and how aircraft should behave during collision

avoidance in different scenarios. Then a Decision Making System (DMS) for collision avoidance is developed based on the rules of the air in VFR conditions. The proposed architecture of the DMS is engineered to be implementable in manned aircraft to perform different tasks ranging from collision detection to generating avoidance Resolution Advisories (RA), and unmanned aircraft (remotely piloted, and autonomous UAV). The DMS is divided into multiple-layers, where each layer is built to perform a specific function (e.g. collision detection and thread prioritizing, collision assessment, advisory system, and manoeuvre generation). The algorithm of each layer is discussed and its flowchart included. However, the proposed DMS architecture gives flexibility for any further development layer by layer, or the whole system for research or certification purposes. A graphical user interface (GUI) is proposed for algorithm test and simulation purposes with short comparison with currently used commercial Portable Collision Avoidance System (PCAS). The proposed GUI can be used as a starting point for a future work for further development to be fully certified as a human machine interface (HMI).

4.2 Airspace Classification

The non-segregated airspace in the UK consists of two main categories, controlled, and uncontrolled airspace with different classes of airspace [21]. Controlled airspace is an airspace where all flights are conducted with Air Traffic Control (ATC). It consists of aerodrome control zone (CTR), Terminal Control Area (TMA), and airway in classes from A to D. Uncontrolled airspace contains Advisory Routes (ADR) which are in class F and Open Flight Information Regions (Open-FIR)in class G. Controlled airspace consists of five classes:

1. Class A: The busiest one. It includes all airways except where they pass through TMA, CTA, or CTR. This class is not available to flights based on VFR.

2. Class B: Upper airspace above flight level FL245 (24500 ft) so it is not usually of concern to VFR pilots. There is no class B in the UK airspace.

3. Class C: Above FL195 and also not of concern to VFR pilots.

4. Class D: Less busy airspace above 6000 ft. It includes Aerodrome Traffic Zones (ATZ). Classes from B to D require ATC clearance and flight notification.

5. Class E: Is similar to class D, but it does not require flight notification or ATC clearance. Usually, it has a reduced traffic information service from ATC.

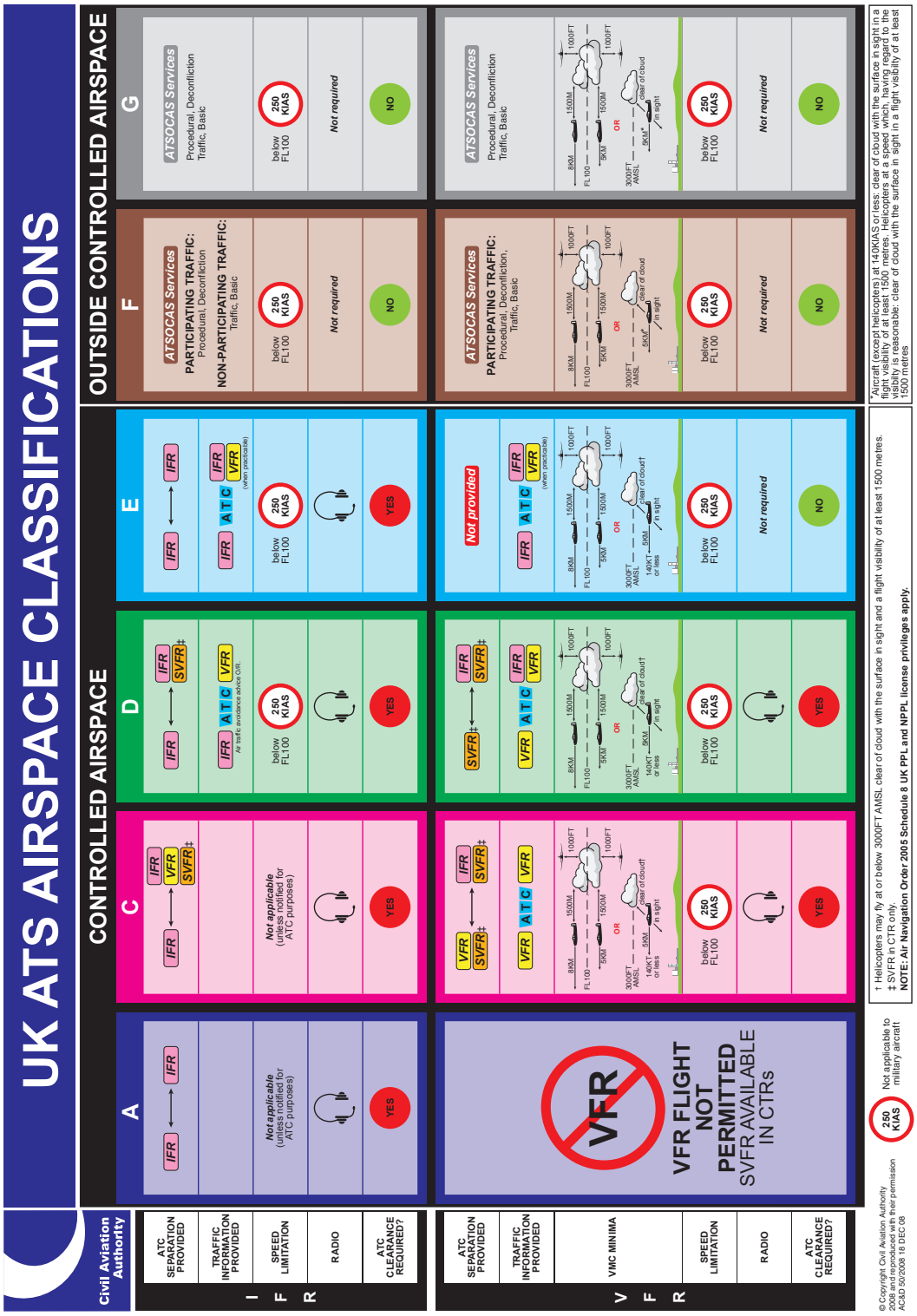

Figure 4.1 UK airspace classification

Uncontrolled airspace includes two classes:

1. Class F: Consists of advisory routes and requires flight plan lodging. Aircraft receive Air Traffic Advisory Service from ATS.

2. Class G: Open-FIR all other airspace. In this airspace flight information services and radar services are available.

Figure 4.1 shows the UK airspace classification with the aircraft requirements to fly in different classes [9]. The second row in the figure gives weather or Visual Meteorological Conditions (VMC), speed limitation, and cloud clearance for VFR in different flight level at different airspace classes.

4.3 Collision Avoidance in the Air

With many aircraft sharing the same airspace, it is often necessary to take collision avoidance action.
A collision risk exists when one aircraft is at the same level or approaching another, its range is decreasing and its relative bearing remains constant [16].
See Figure 4.2.

- Regardless of any ATC clearance, it is the duty of the commander (pilot-in-command) of an aircraft to take all possible measures to see that he does not collide with any other aircraft.

- An aircraft must not fly so close to other aircraft as to create a danger of collision.

- Aircraft must not fly in formation unless the commanders have agreed to do so.

- An aircraft which is obliged to give way to another aircraft must avoid passing over, under, or crossing ahead of, the other aircraft (unless passing well clear of it).

- An aircraft with right of way should maintain its heading and speed.

- For purposes of this rule, a glider and a machine towing it are considered to be a single aircraft under the command of the commander of the towing machine. Aeroplanes and helicopters must give way to aircraft towing gliders.

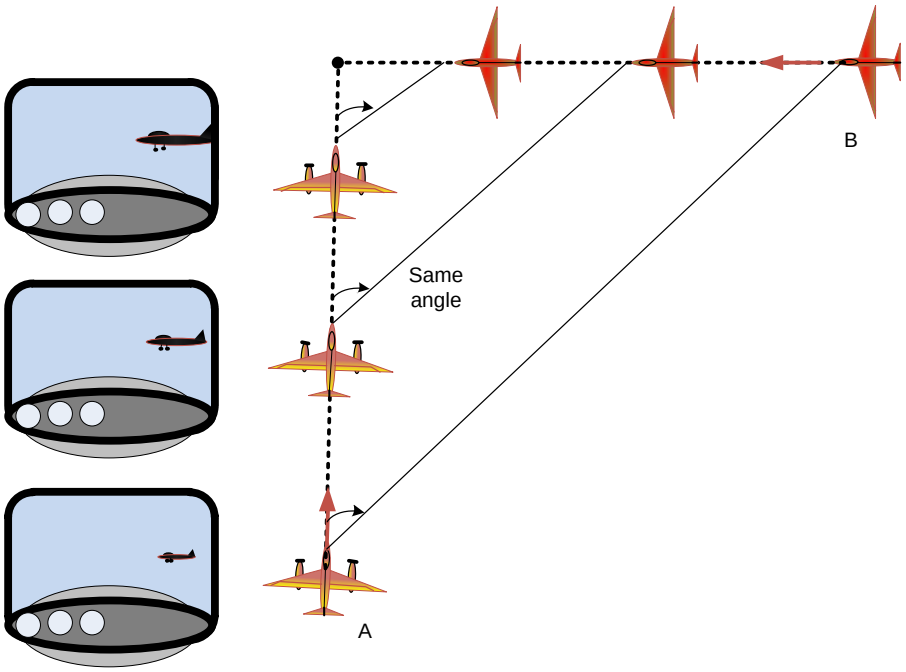

Figure 4.2 Fixed position of another aircraft in the windscreen indicates a constant relative bearing and therefore a collision risk

4.4 Right of Way Rules

The right of way for UAV and manned aircraft are the same. Figure 4.3 shows the rules of right of way around the aircraft, it can be seen that aircraft must give way to all traffic approaching from the opposite direction within 45° of aircraft's centreline and for all air traffic that are approaching from starboard (right) side of the aircraft. Balloons, gliders, airships, and aircraft towing objects always have the right of way except for cases when aircraft are being overtaking by them. An aircraft has right of way when it is overtaken by an aircraft within 70° of the aircraft's centreline [16].

4.5 The Rules of the Air for Collision Avoidance in Different Collision Scenarios

There are three main cases for collision and each one has different scenarios. The main cases are approaching head-on, converging, and overtaking. Figures 4.4, 4.5, and 4.6 show a simple illustration of these situations with the collision avoidance rule for each case. For example, while in head-on case, both aircraft should turn right to avoid collision. In the

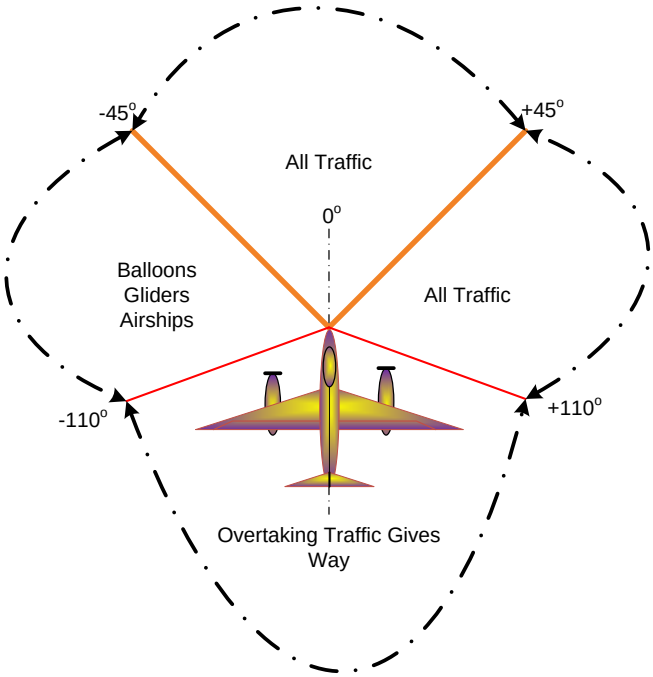

Figure 4.3 Rules of right of way around the aircraft

converging situation case, the right of way is for aircraft (B) that has the other (A) on its left, so (A) must turn right to give way to (B) then tracks behind it. However, for an overtaking situation, the overtaken aircraft will keep going on its way while the overtaking one must turn right to keep out of the way [74]. Although the above cases are the general cases there are many scenarios under each case. Thus there are different rules for each scenario. For example, head-on case contains many scenarios such as, both aircraft in level flight, both aircraft descending/climbing, and one aircraft climbing/descending while the other in level flight.

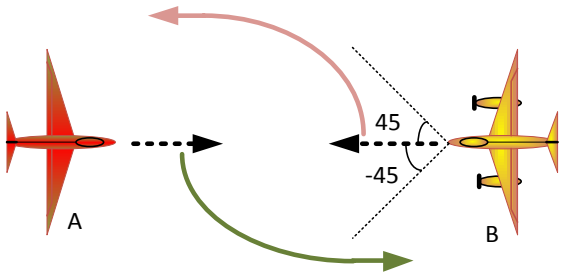

Figure 4.4 Head-on case: Both aircraft should turn right

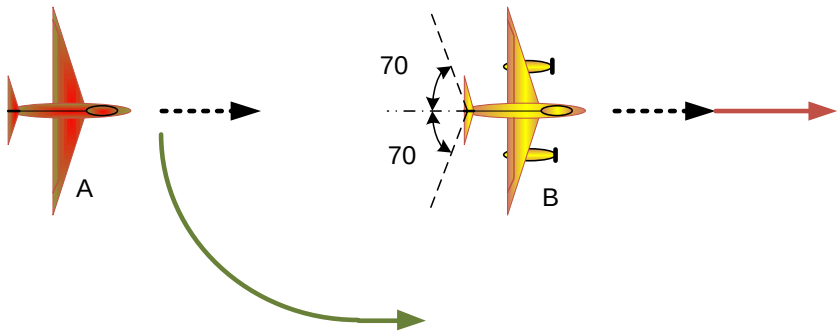

Figure 4.5 Overtaking scenario: The overtaken aircraft (B) will continue straight while the overtaking aircraft (A) must turn right

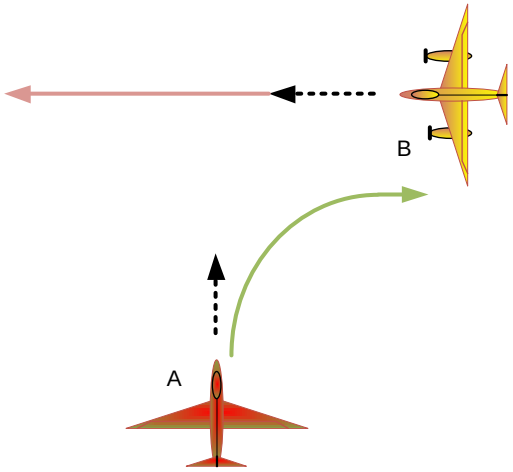

Figure 4.6 Converging case: Aircraft B has the right of way

4.5.1 Head-on Conflict Sub-scenarios

Case-1) Both aircraft are in level flight: Figure 4.7 shows two cases for head-on level flight; the first case is the general case where there is no offset between vehicles' path (or the offset is on left of each one), the second case is where the offset exists, so turning right may increase the risk of a conflict. It is clear in Figure 4.7 that if the vehicle turns right that will cause a crossing of the vehicle's path. So a left turn by both aircraft is more appropriate.

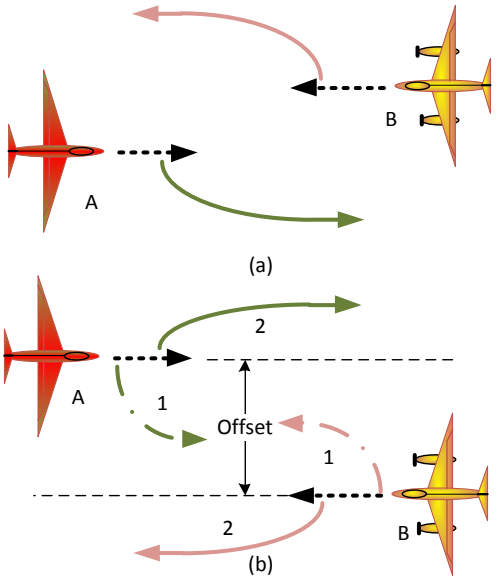

Figure 4.7 Head-on collision scenario (a) no offset; (b) offset exists

Case-2) Both aircraft are descending: In this scenario the vehicles should act in the same manner as the Case-1 (level flight). However, other resolution options may be considered as seen in Figure 4.8:

- (a) Increase the descent rate of one aircraft and level off the other.
- (b) Level off only one aircraft.

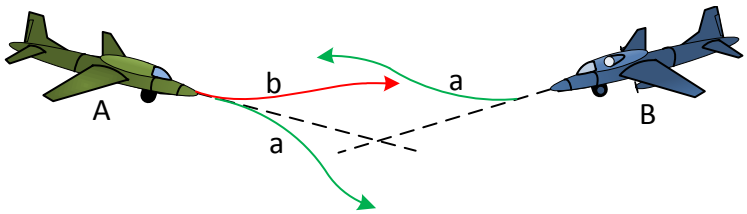

Figure 4.8 Head-on scenario in which both aircraft are descending

Case-3) Both aircraft are climbing: This scenario has more risk because of possible lack of visual sighting by pilots. In this case there are two avoidance options (Figure 4.9):

- (a) Increase the climb rate of one aircraft with reducing the climb rate or leveling off the other one.
- (b) Reducing the climbing rate for one aircraft (or levelling off).

Option (a) is restricted by the maximum climbing rate, so it is not always available because the climbing rate can not be increased if it is already at its maximum value.

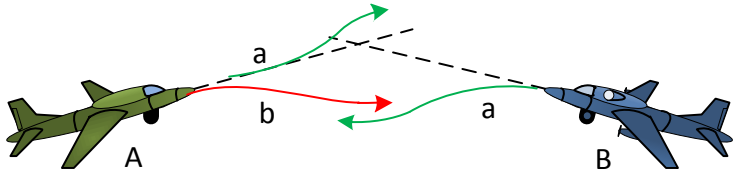

Figure 4.9 Head-on scenario where the both aircraft are climbing

Case-4) One vehicle is in level flight and the other is climbing: Conflict resolution is performed by leveling of (not at the same level as the other vehicle) or climbing rate reduction of the climbing vehicle (Figure 4.10)

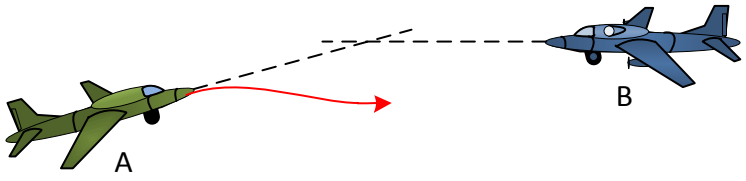

Figure 4.10 Head-on scenario: One vehicle is in level flight and the other is climbing

Case-5) One aircraft is in level flight and the other is descending: There are two resolution options both should be done by the descending aircraft (Figure 4.11):

- (a) Levelling off (not at the same altitude of the other aircraft)
- (b) Increasing the descent rate.

However, the high inertia for levelling off may make option (a) a risky choice, hence option (b) is the better solution.

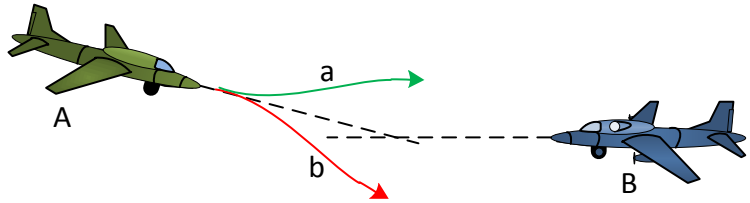

Figure 4.11 Head-on scenario: One aircraft is in level flight and the other is descending

4.5.2 Converging Conflict Sub-scenarios

Case-1) Both aircraft in level flight: The aircraft that has no right of way should turn behind the other. More efficient collision avoidance can be achieved by turning both aircraft in the same direction. Figure 4.12 shows the possible resolution, it is important to notice that any avoidance manoeuvre is not going to pass ahead of the other aircraft so options (c, and d) should not performed. In this research the turning right manoeuvre for the aircraft which has no right of way is considered.

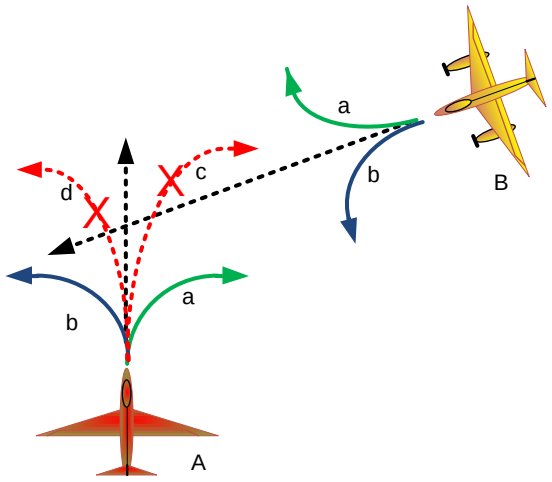

Figure 4.12 Converging conflict with resolution manoeuvres

Case-2) One or both vehicles are not in level flight: The conflict resolutions for these scenarios are as same as the resolutions for level flight Case-1.

4.5.3 Overtaking Conflict Sub-scenarios

Case-1) Both vehicles are in level flight: The overtaking aircraft should turn right if there is no offset or turn left if an offset exists. If this manoeuvre was not enough to resolve the conflict then the overtaken aircraft should also perform a manoeuvre as shown in Figure 4.13.

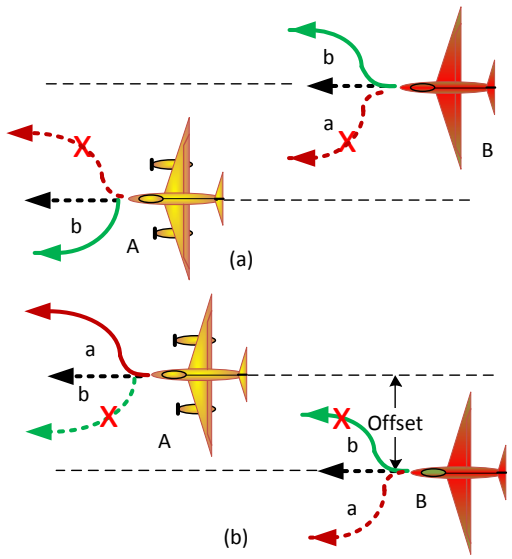

Figure 4.13 Overtaking conflict scenario: Both aircraft are in level flight, no offset (a); offset exists (b)

Case-2) Both vehicles are descending: Figure 4.14 shows the resolutions for this scenario, it can be seen that the aircraft that on a higher level should stop descending and the other should increase its descent rate. A right turn by the overtaking aircraft is not recommended due to the possibility of crossing in their paths. If the right turn is chosen then both aircraft should turn in opposite directions.

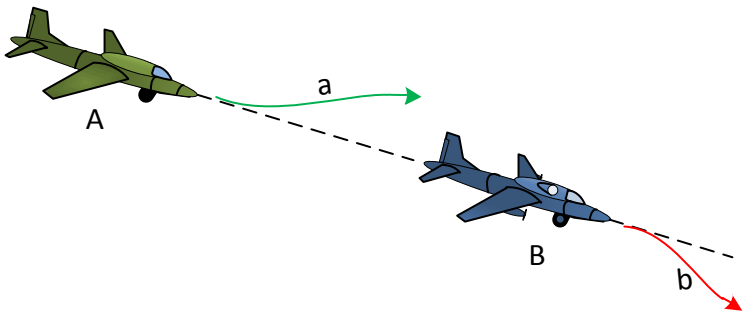

Figure 4.14 Overtaking conflict scenario: Both aircraft are descending

Case-3) Both aircraft are climbing: The recommended resolution for this conflict is performed by stopping the climbing of the lower aircraft and increasing the climbing rate of the higher aircraft slightly (without exceeding its climbing rate limit), see Figure 4.15.

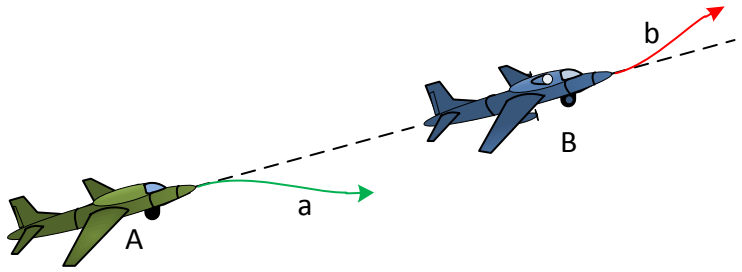

Figure 4.15 Overtaking conflict scenario: Both aircraft are climbing

Case-4) The overtaking aircraft is in level flight, and the overtaken aircraft is climbing: Resolution as in Case-2 (both aircraft are descending) should be performed with the overtaken considering leveling off, and bearing in mind the inertia. This resolution can be done when aircraft's levels are not close or have not already been crossed, see Figure 4.16.

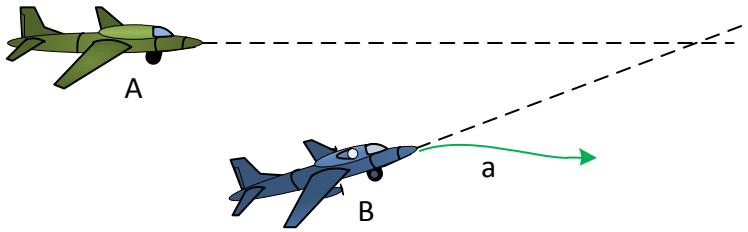

Figure 4.16 Overtaking conflict scenario:The overtaking aircraft is in level flight, and the overtaken aircraft is climbing

Case-5) The overtaking aircraft is climbing, and the overtaken aircraft is in level flight: Resolution is done by increasing the climb rate of the overtaking aircraft. This resolution is possible when the aircrafts' levels are not close or have not already crossed, see Figure 4.17.

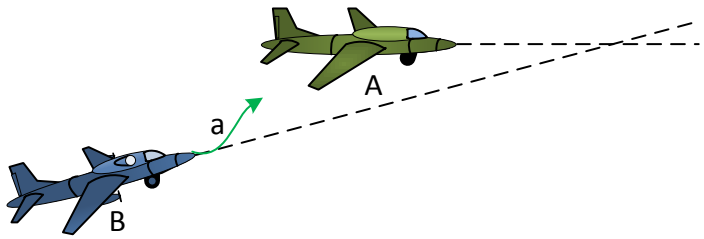

Figure 4.17 Overtaking conflict scenario: The overtaking aircraft is climbing, and the overtaken aircraft is in level flight

Case-6) The overtaking aircraft is in level flight, and the overtaken aircraft is descending:
Resolution is done by leveling off the overtaken aircraft, see Figure 4.18.

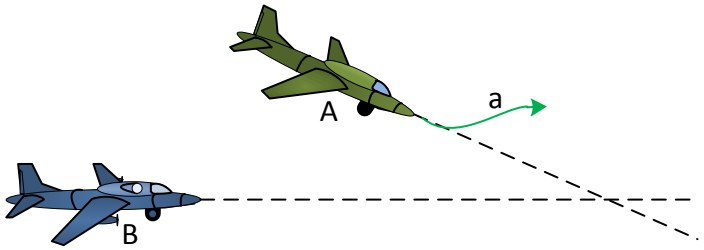

Figure 4.18 Overtaking conflict scenario: The overtaking aircraft is in level flight, and the overtaken aircraft is descending

Case-7) The overtaking aircraft is descending, and the overtaken aircraft is in level flight:
Resolution is done by leveling off the overtaking aircraft. This resolution is possible when aircrafts' levels are not close or have not already crossed, see Figure 4.19.

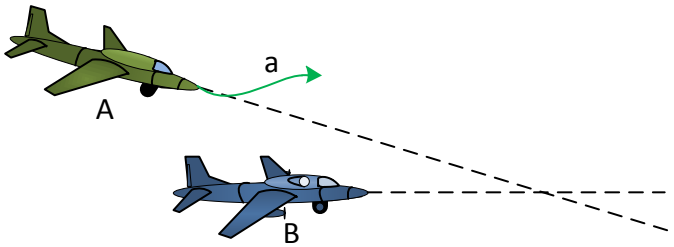

Figure 4.19 Overtaking conflict scenario:The overtaking aircraft is descending, and the overtaken aircraft is in level flight

4.6 CAA Policy on Detect and Avoid

This section is mostly taken from CAP-722 document (Unmanned Aircraft System Operations in UK Airspace Guidance) that is issued by CAA [49].
A significant increase in both civil and military UAS flying is anticipated, most of which will require access to all classes of airspace if it is to be both operationally effective and commercially viable. To achieve this, UAS will have to be able to meet all existing safety standards applicable to equivalent manned aircraft types, appropriate to the class (or classes) of airspace within which they are intended to be operated [49].

4.6.1 Separation Assurance and Collision Avoidance Elements

There are two elements to a Detect and Avoid system as follows [49]:

Separation Assurance This term is used to describe the routine procedures and actions that are applied to prevent aircraft getting into close proximity with each other. Any resolution manoeuvring conducted at this stage must be conducted in accordance with the rules of the air. When flying in airspace where the provision of separation is the responsibility of ATC, however, the Remote Pilot should only manoeuvre the aircraft after receiving ATC approval to do so, in the same fashion as is done for a manned aircraft.

Collision Avoidance. This is the final layer of conflict management and is the term used to describe any emergency manoeuvre considered necessary to avoid a collision; such a manoeuvre may contradict the rules of the air or ATC instructions. While the remote pilot would normally be responsible for initiating a collision avoidance manoeuvre, an automatic function may also be required in order to cater for collision avoidance scenarios where the Remote Pilot is unable to initiate the manoeuvre in sufficient time (due to C2 latency issues or lost link scenarios).

The separation and collision avoidance capabilities must be able to [49]:

1. Detect and avoid traffic (air and ground operations) in accordance with the rules of the air;
2. Detect and avoid all airborne objects, including gliders, hang-gliders, paragliders, microlights, balloons, parachutists etc;
3. Enable the remote pilot to determine the in-flight meteorological conditions;
4. Avoid hazardous weather;
5. Detect and avoid terrain and other obstacles; and
6. Perform equivalent functions, such as maintaining separation, spacing and sequencing that would be done visually in a manned aircraft.

4.6.2 Factors for Consideration when Developing a Detect and Avoid System for UAS

The CAA Safety Regulation Group does not define the matters to be taken into account for the design of aircraft or their systems. However, for the guidance of those engaged in the development of detect and avoid systems, some of the factors that the CAA believes may need to be considered are listed below [49].

1. Ability to comply with the rules of the air.
2. Airworthiness.
3. Control method, controllability and manoeuvrability.
4. Flight performance.
5. Communications procedures and associated links.
6. Security.
7. Emergency actions, reversionary or failure modes in the event of degradation of any part of the UAS and its associated control and/or relay stations.
8. Actions in the event of lost communications and/or failure of on-board detect and avoid equipment.
9. Ability to determine real-time meteorological conditions and type of terrain being overflown.
10. Nature of task and/or payload.
11. Autonomy of operation and control.
12. Method of sensing other airborne objects.
13. Remote pilot level of competence.
14. Communications with ATS providers, procedures and links with control station.
15. Means of launch/take-off and recovery/landing.
16. Reaction logic to other airspace objects.
17. Flight termination.
18. Description of the operation and classification of the airspace in which it is planned to be flown.
19. Transaction times (e.g. including delays introduced by satellite links).
20. Address both cooperative and non-cooperative air traffic.

This list is not exhaustive [49].

4.7 UAS Flight Control Mode

For UAS expected to be flown by Remote Pilots operating more than one type, UAS may be rated as a class rather than a specific type. In determination of the basis of class rating, CAA Licensing and Training Policy Department considers the flight control mode of the UAS to be the most appropriate means of classifying such systems. Classification of UAS according to flight control mode permits the degree of automation or autonomy of a UAS to be considered when formulating requirements for remote pilot qualification. Provisional categories of UAS Flight Control Modes (FCM) are indicated in Table 4. The provisional flight control mode categories in the table are arranged in order of increasing automation or autonomy, and decreasing requirement for traditional manned aircraft piloting competence. Each flight control mode listed in the table is based on a broad description of flight control mode capability, and gives an analogous manned aircraft autopilot mode as a comparison [49].

Table 4.1 Flight Control Mode for UAS Class Ratings

Class	Flight Control Mode Class Name:	Example Type
Class 0	Reference Class: Manned Aircraft	Airbus 320, EH101
Class 1	Direct Command: Remote Pilot	Jindivik, RMA
Class 2	Attitude Command: Control Wheel Steering	
Class 3	Flight Parameter Command: 3-Axis Autopilot	Mirach
Class 4	Stored Flight Profile Command: Autopilot + Flight Management Computer (FMC)	Global Hawk
Class 5	Sensor Command- Autopilot+FMC+Sensors	BGM-109
Class 6	Autonomous Command: Intelligent UA	AI UAV

Classes 4, 5 and 6 will require a command override intervention capability

4.8 Decision Making System (DMS) for CAS

This section explains the proposed decision making algorithm for CAS that is developed based on CAA requirements which are given in sections 4.6 and 4.7. Figure 4.20 shows its architecture for CAS which is divided into four layers:

- Layer-1 Collision Detection and Prioritizing (CDP): Consists of two subsystems; Collision Detection (CD) which detects the conflicts and generates alerts and warnings

flags; and Risks Prioritizing (RP) which defines which of the detected intruders has a greater threats than the others.

- Layer-2 Collision Assessment (CA): Detriments the collision scenario type (i.e. head-on, left/right approaching, and overtaking/overtaken).
- Layer-3 Advisory System (ADS): Gives general conflict resolution advisories (e.g. turn right, turn left, climb, descend, level, and hold speed and altitude)
- Layer-4 Avoidance Manoeuvre Generator (AMG): This layer generates avoidance manoeuvre that allows the aircraft to avoid the collision.

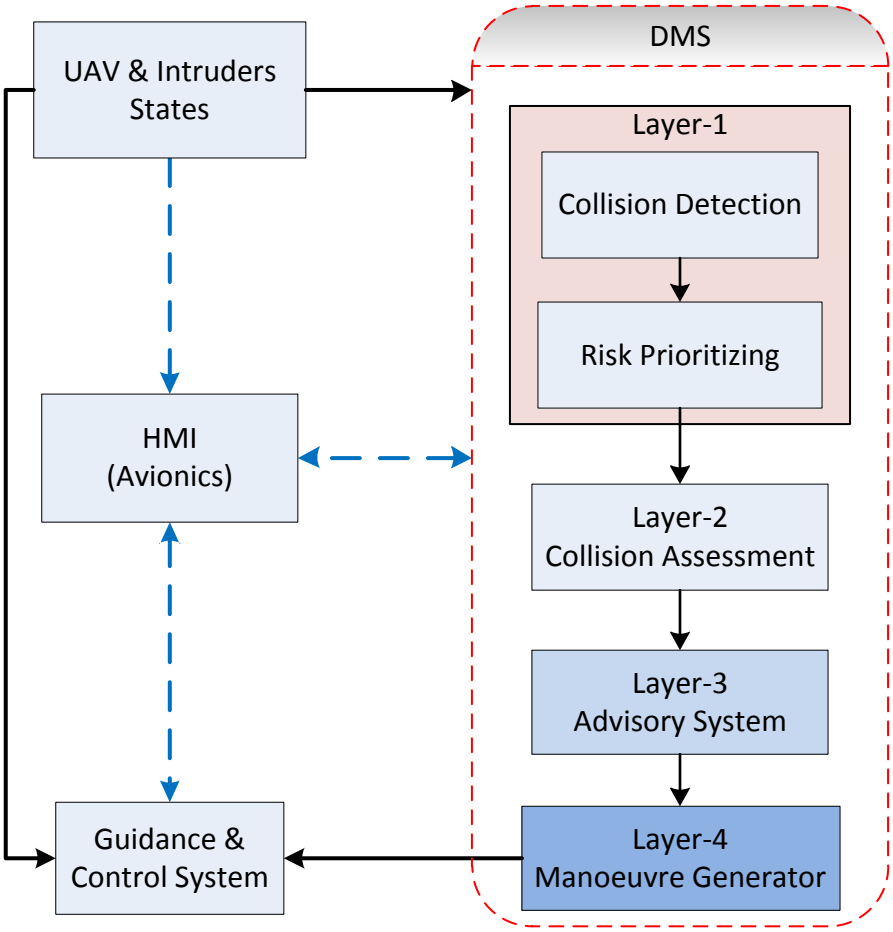

Figure 4.20 DMS architicture

The proposed DMS architecture is engineered to be implemented for different functionalities for manned aircraft, and at different level of autonomy of UAS, or at different classes of

the flight control mode that given in table 4.1. For example, in Class-0 or manned aircraft the DMS can be used to reduce the overall pilot workload by performing the flowing tasks:

- Collisions detection and risk prioritizing (Layer-1), so the pilot will know where the traffics is.
- Determines the collision scenario type (i.e. head-on, left/right approaching, and overtaking/overtaken) by using (Layer-1, and Layer-2) this will help the pilot to decide what actions are needed to avoid the conflict.
- Evaluates the collision type, and generates conflict resolution advisories (Layer-3) to help the pilot to initiate a suitable avoidance manoeuvre.

For the remotely piloted UAV (Class-1) this architecture helps the remote pilot to build a picture for the surrounding conflicts so the remote pilot could act in the same manner as an on-board pilot. Moreover, the remote pilot can switch to the highly autonomous mode by engaging Layer-4 which generates the avoidance manoeuvre. Layer-4 can be used also in flight mode classes (2-6). The next subsections give a description of the DMS layers.

4.8.1 Collision Detection Layer

This system generates Alert Flags (AF1, AF2), Collision Flag (CF), and gives general information about the collision risks, such as the intruder direction. According to the rules of the air that were discussed in Section 4.3

A collision risk exists when one aircraft is at the same level or approaching another, its range is decreasing and its relative bearing remains constant.

So the collision detection unit uses the range, the range rate, the relative altitude, and the bearing rate as inputs. The outputs are separation alert flags (AF1, AF2) that will be activated for the aircraft which their range and relative altitude are lees than specific values (loss of separation), collision flag (CF) (active when the bearing angle is constant which means a collision risk exists). Figure 4.21 shows the ranges and the relative altitudes around the aircraft. For safety reasons different ranges and relative altitude can be used to generate two types of alerts (R_1, and L_1 are used to activate AF1); (R_2, and L_2 are used to activate AF2). Having two levels of alert may help increasing the safety and it may help the pilot to prioritize the intruders in case of multi-intruder scenarios. However, this feature can be switched off simply by selecting $R_1 = R_2$, and $L_1 = L_2$. Some commercial Portable Collision

Avoidance Systems (PCAS) like XRX [75] gives the user the ability to control the values of the range and the relative altitude. For example, the range can be selected to be 6 NM, 3 NM, or 1.5 NM (1 NM = 1850 m), and the relative altitude can be ±2500 feet, ±1500 feet, or ±500 feet. Figure 4.22 shows the flowchart of the algorithm that is used in the collision detection system.

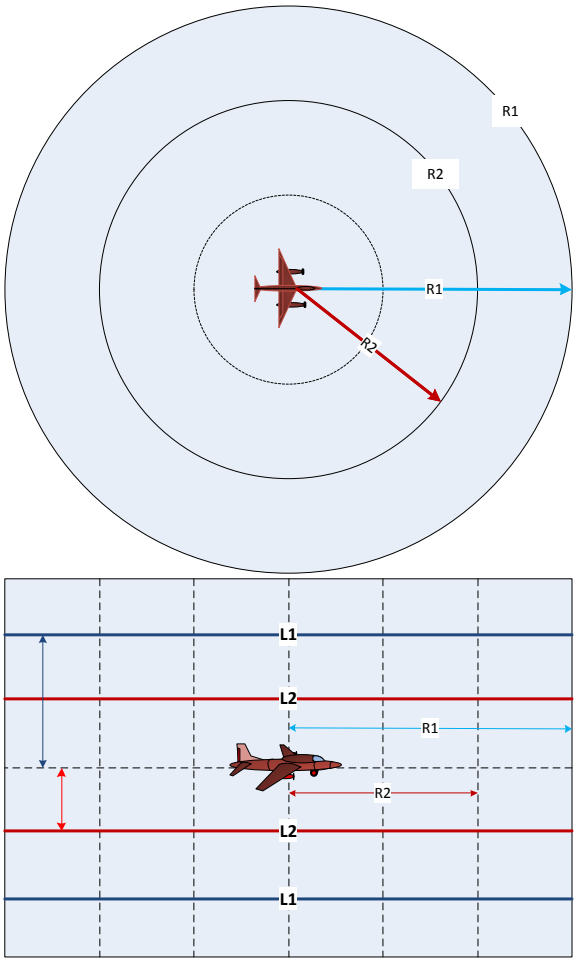

Figure 4.21 Range, and relative altitude: Horizontal plane (upper); Vertical plane (lower)

4.8.2 Prioritizing

In multi-intruder scenarios it is important to define which intruder should be given a higher priority than the others so it can be avoided first. The proposed algorithm that is used for prioritizing intruders is similar to that is used in XRX PCAS [75]. However, a commercial PCAS is designed for manned aircraft (class 0 in FCM table 4.1) so for remotely piloted UAV (class 1) a greater safety margin should be taken due to the Communication and Con-

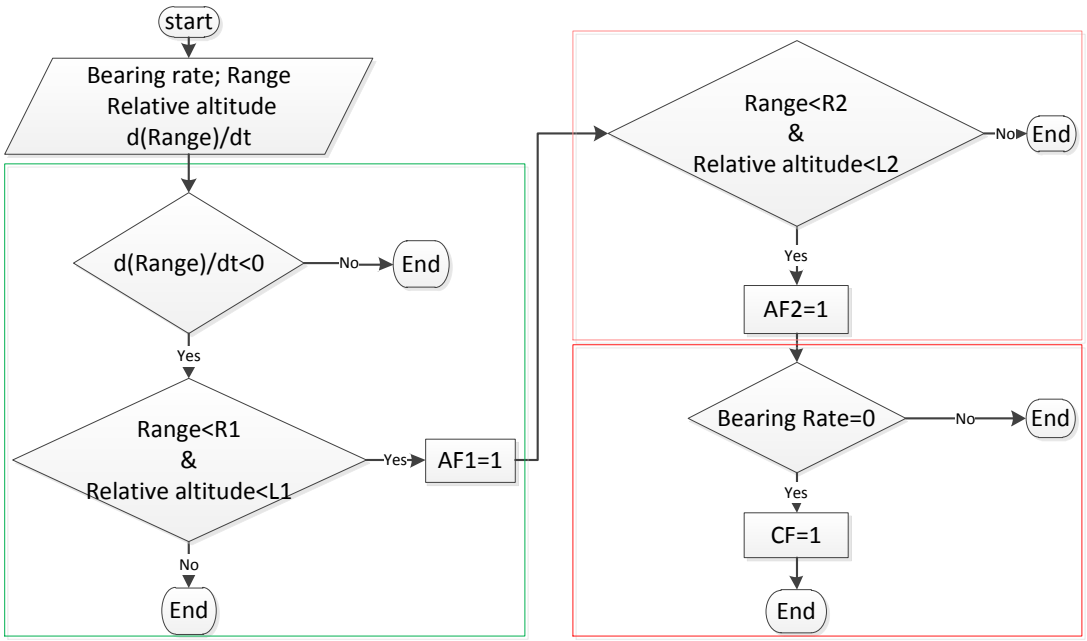

Figure 4.22 Flowchart of the collision detection system

trol (C2), and data link problems and delay [2]. In this thesis three levels of alerts and warning (AF1, AF2, CF) are proposed in order to give the remote pilot (class 1 in FCM) more time and greater safety margin so that could overcome the problems that may be associated with C2 link. The remote pilot can modify the values of ranges and relative altitudes depend on the UAV manoeuvrability and the flight environment. The prioritizing process can simply be determined by the following factors:

- Intruder range and relative altitude
- Intruder vertical speed (climbing or descending)
- Aircraft vertical speed (climbing or descending)

While the intruder with (CF) flag has the highest priority above all the intruders, the intruder with (AF1) flag has the lowest priority. If the intruders have the same flag the intruder with the least vertical separation (relative altitude) has higher priority. However, if the aircraft is descending/climbing the the relative altitude sign will be taken in consideration. For example, if there are two intruders in level flight and both are within $\pm L_2$ and both have (AF2) flag, the intruder at higher level will be given higher priority if the aircraft is climbing, but it will given lower priority if the aircraft is descending. The intruders will be given a number based on their priority, the higher the priority the smaller the number that will be given. For instance Figure 4.23 shows a conflict scenario in which the three intruders (A, B,

and C) are in different types of conflict with the aircraft. Intruder B has the highest priority and A has the lowest priority then the numbers that will be given to the intruders are (A#3, B#1, and C#2).

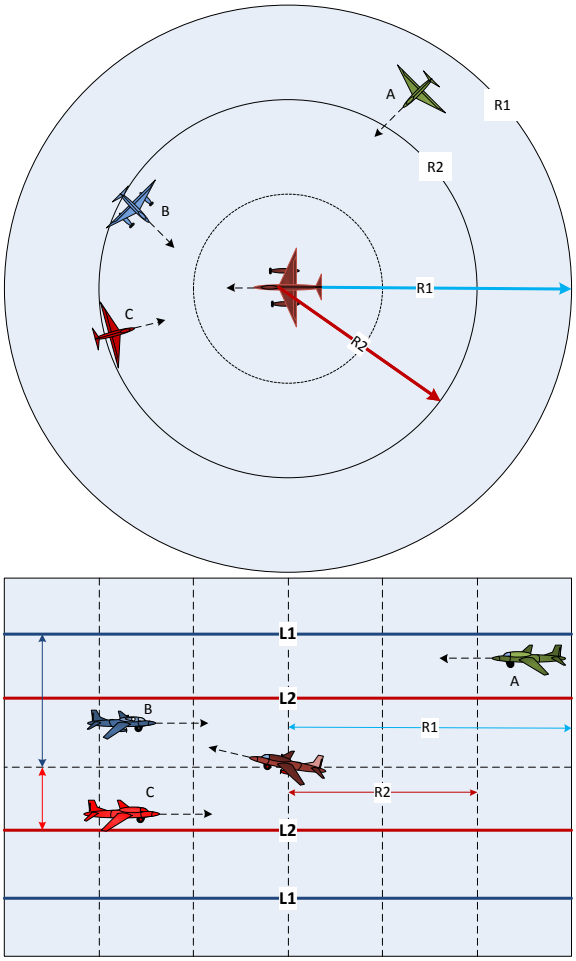

Figure 4.23 Three intruders scenario with different priorities (Horizintal plane (upper) shows the ranges; Vertical plane (lower) shows the relative altitudes

It can be seen that although intruders B and C at the same range in the horizontal plane (upper part in Figure 4.23), and they have the same absolute relative altitude, but intruder B is given higher priority than intruder C. This is because the aircraft is climbing and intruder B is at a higher altitude than intruder C.

Using Time to Collision (T_c) in The Prioritizing Process

Different intruders have different speeds and heading, hence different relative speeds with the aircraft. Thus, using the ranges alone is not enough for the prioritizing process. In this

thesis time to collision (T_c) is used in prioritizing process, the smaller the T_c, the higher the priority will be given. However, the range is implicit in the time to collision, as time to collision is given by:

$$T_c = \frac{Ra}{V_r} \tag{4.1}$$

where R_a is the range, and V_r is the relative speed or closure rate $V_r = \frac{d(R_a)}{dt}$. A conflict scenario shown in Figure 4.24 clarifies the importance of using (T_c) in the prioritizing process. In this scenario the aircraft and the intruders (A, and B) are in level flight at the same altitude. Initially the intruders' ranges and speeds are equal, with constant bearing angle for both, so both have collision flag (CF=1). It is clear that the closure rate of the intruder (B) is greater than the closure rate of the intruder (B) so ($T_{c_B} < T_{c_A}$). Hence, using range and relative altitudes for prioritizing is not enough or even it will give the wrong priority for the intruders (if the range of (A) is less than (B) range). In this example, the intruder with the smaller T_c which is the intruder (B) is given higher priority than the intruder (A).

4.8.3 Displaying Conflicts Data

Designing a display (Human machine interface (HMI), or avionics) that satisfies the CAA requirements is not in the scope of this thesis. However, a basic Graphic User Interface (GUI) is developed for the purpose of algorithm testing and simulation. Current commercial systems display the data in different ways. For example, Figure 4.25 shows the XRX version of a portable collision avoidance system [75] (PCAS, which is a trademark of Zaon Flight Systems Inc). Figure 4.26 shows the display of the XRX which can show three intruders at the same time, putting the intruder with the highest priority (Primary intruder#1) on the left side of the screen, and the intruder with the lowest priority (Secondary intruder#2) on the right side of the screen. It gives some information about the intruders such as, the direction, range, the relative altitude, and the vertical trend indicator (Climbing/Descending). Figure 4.27 shows the proposed (GUI), the layout of the GUI is designed based on the intruder priority:

- A vertical layout is chosen to represent the intruder priority (The intruder with the highest priority is at the top of the screen and the intruder with the lowest priority at the bottom of the screen).

- The size of the intruder display reflects the intruder priority, the higher the priority the bigger the size of the intruder display.

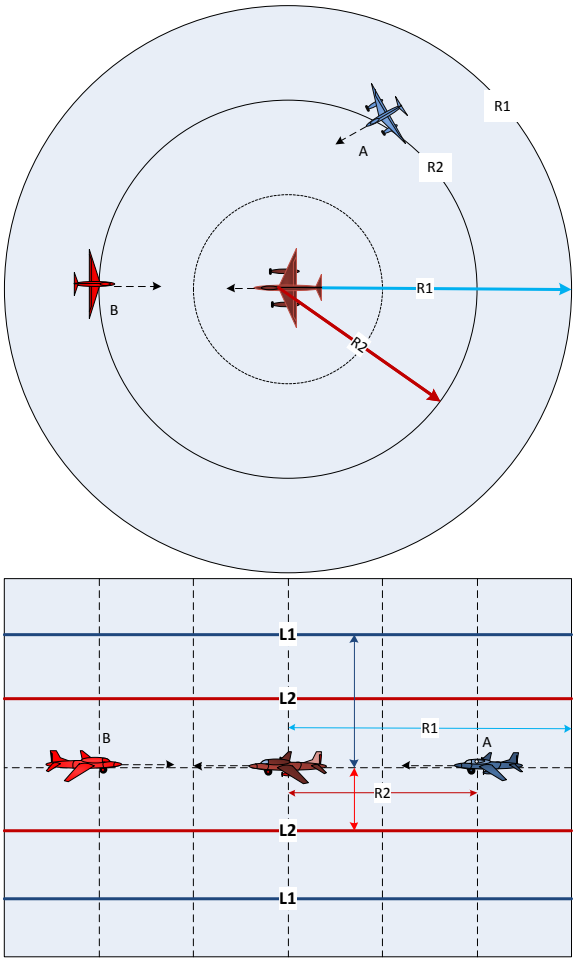

Figure 4.24 Conflict scenario clarifying the importance of the T_c in the prioritizing process

- The higher the intruder priority the more information is displayed.
- The accuracy of the bearing angle of the intruder (intruder direction) is increased as the intruder priority is increased.
- The relative altitude is represented numerically and graphically.

The proposed GUI can be a ground for a future research for further development to be a fully certified HMI.

Figure 4.28 shows how the GUI looks like (right) in case of a conflict scenario which is given on the left. Intruder C has the highest priority so its information are displayed on the to display, while intruder B has the lowest priority then it is appear on the display at the bottom. Intruder C bearing is more than 0°, but lees than 30° so two direction indicators are highlighted.

Figure 4.25 PCAS XRX

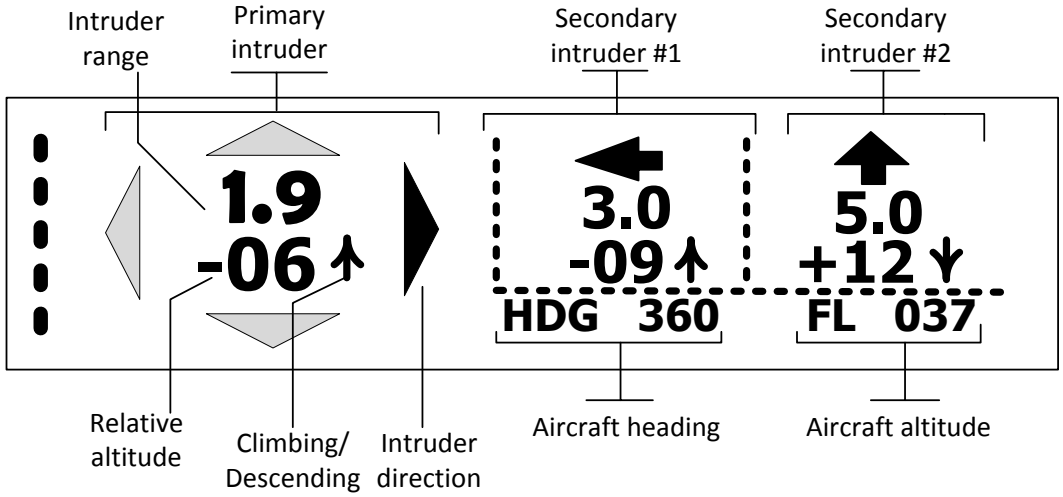

Figure 4.26 PCAS XRX display

4.8.4 Collision Assessment Layer

The collision information of the intruder with the highest priority is passed to collision assessment layer (Layer-2) and the assessment algorithm is applied in order to determine the conflict type. Heading angles of the aircraft and the intruder, and bearing angle is used in the assessment process. Sections 4.4 and 4.5 described different conflict scenarios and showed how to differentiate between them by using the bearing angle (see figure 4.3). For example, if the intruder is approaching within ($\pm 45^\circ$) it will be considered as a head-on collision. However, an offset scenario is considered just when the difference between the

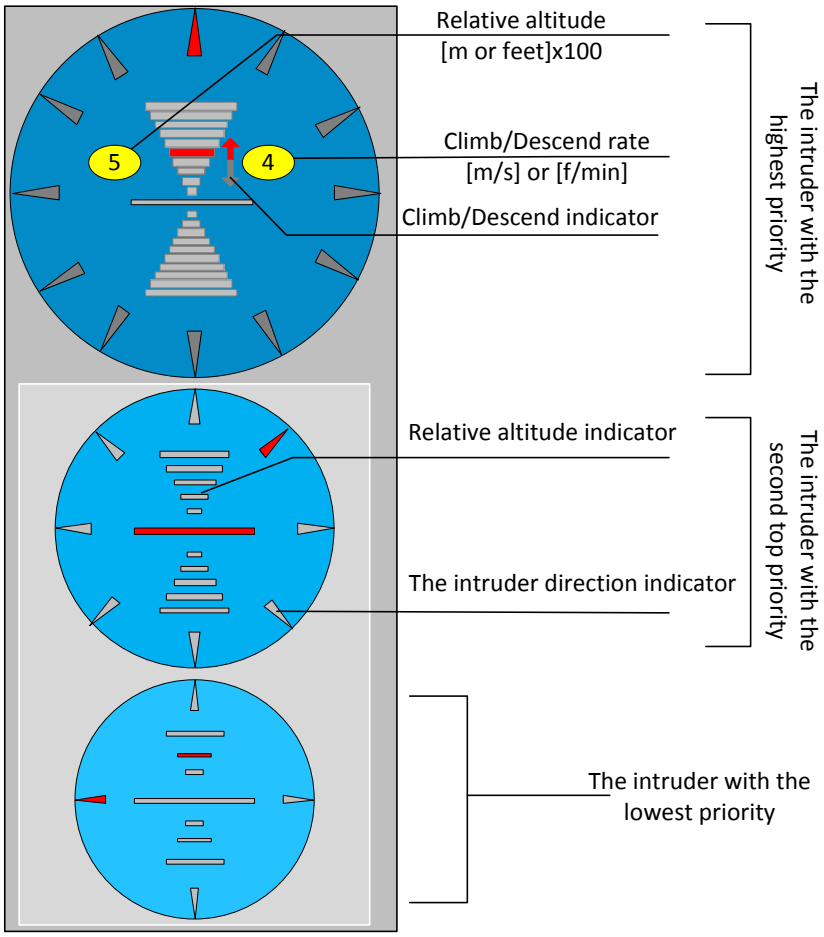

Figure 4.27 Displaying Layer-1 output on GUI/ Avionics

heading angle of the aircraft and the intruder is (180°) and the intruder is on the right of the centerline of the aircraft (see Figure 4.7). If the aircraft and the intruder position in the earth axis reference frame (North, East, Up (NEU) reference frame) are known then the intruder range and bearing can be calculated. Figure 4.29 shows the aircraft and the intruder in the NEU reference frame. The body reference frame $[o, x_b, y_b, z_b]^T$ is attached to the aircraft and is used to convert the intruder position values from the NEU reference frame into the aircraft body reference frame using the Directional Cosine Matrix (DCM) then:

$$R_a = \sqrt{{H_{range}}^2 + {V_{range}}^2} \tag{4.2}$$

Bearing is calculated in the horizontal plane as the vertical bearing can be ignored (relative altitude is small enough to be ignored comparing to the range). So if $[x_e, y_e]^T$ are the intruder position in the horizontal plane in NEU reference frame, and the aircraft heading is

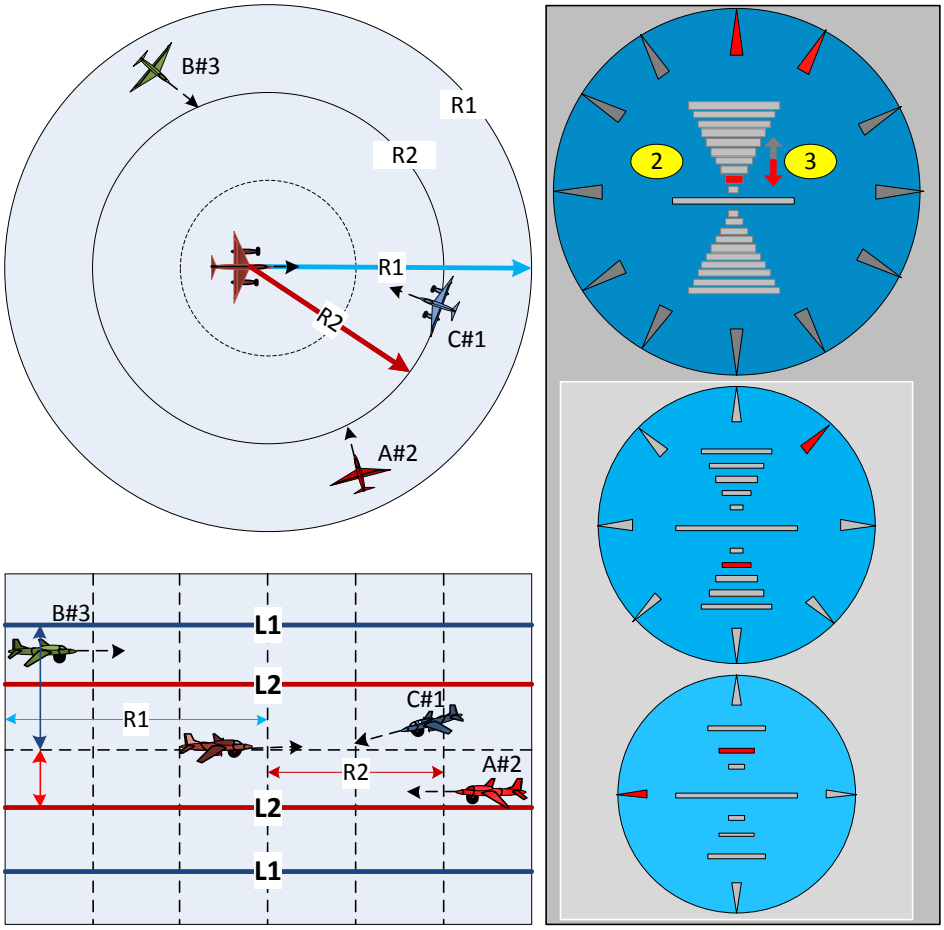

Figure 4.28 Conflict scenario (left), and its displayed information on GUI/Avoinics (right)

ψ, the intruder position in the horizontal plane in the aircraft reference frame $[x_{1b}, y_{1b}]^T$ is given as:

$$\begin{bmatrix} x_{1b} \\ y_{1b} \end{bmatrix} = DCM \begin{bmatrix} x_e \\ y_e \end{bmatrix} \tag{4.3}$$

where DCM is the directional cosine matrix:

$$DCM = \begin{pmatrix} \cos(\psi) & \sin(\psi) \\ -\sin(\psi) & \cos(\psi) \end{pmatrix} \tag{4.4}$$

So bearing is given by:

$$Br = atan2(y_{1b}, x_{1b}) \tag{4.5}$$

Figure 4.30 shows the flowchart for the proposed collision assessment algorithm. This algorithm is developed based on Figure 4.3, the bearing is used to assess the approaching

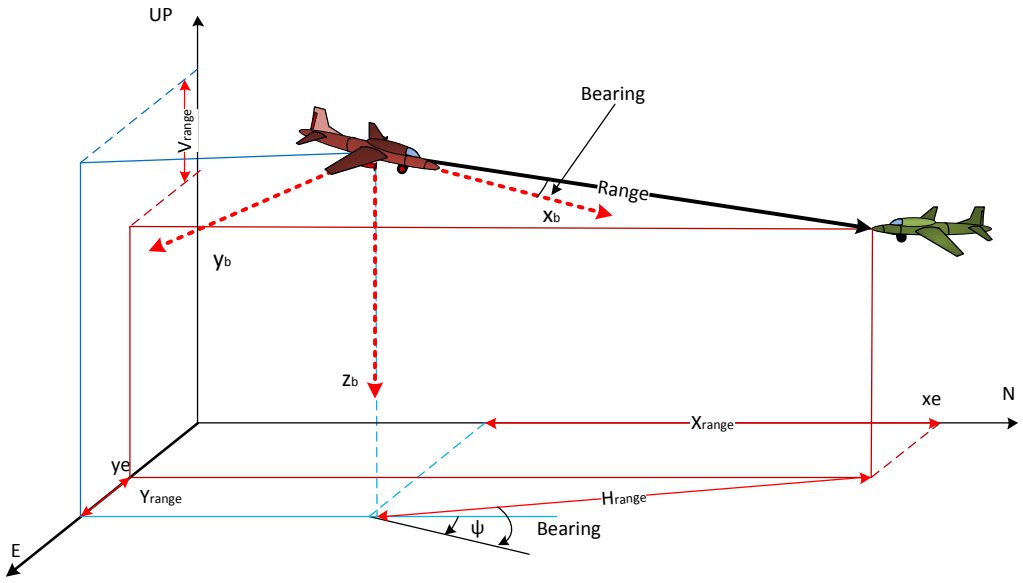

Figure 4.29 Range and bearing of the aircraft and the intruder in NEU and body reference frames

direction.

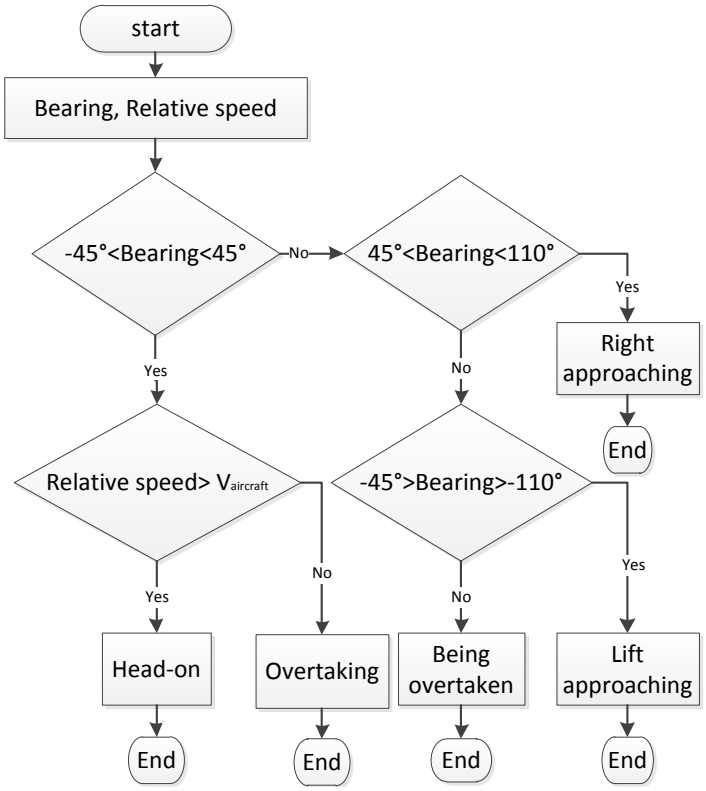

Figure 4.30 Collision assessment algorithm flowchart

However, when the bearing is ($\pm 45°$), relative speed between the aircraft and the intruder is used to differentiate between head-on and overtaking conflict scenarios:

- Head-on conflict scenario, if relative speed is greater than the aircraft speed.
- Aircraft is overtaking the intruder conflict scenario, if the relative speed is smaller than the aircraft speed.

The information that will be passed to the pilot (in manned aircraft), or the remote pilot in case of UAV are collision type (i.e. Head-on, right/left approaching, overtaking, and being overtaken), the relative speed or closure rate V_r, and the time to collide T_c. Figure 4.31 shows the proposed GUI/avionics layout for showing information that is generated by Layer-2.

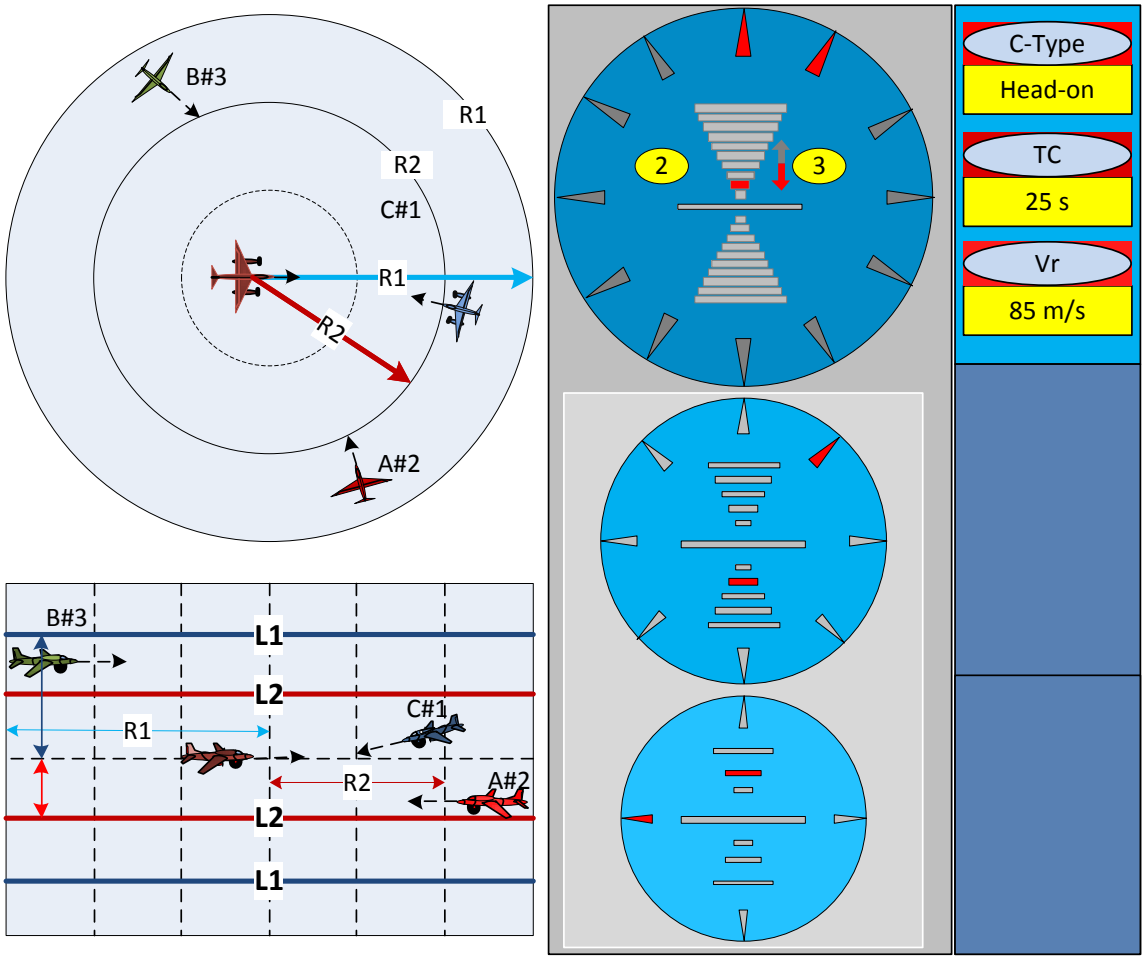

Figure 4.31 Conflict scenario (left), GUI with Layer-1, and Layer-2 information (right)

It is important for the pilot to know T_c in order to initiate the avoidance manoeuvre. The required time for a pilot to recognise an approaching aircraft and initiate an avoidance manoeuvre is divided as follows [7]:

- See object (0.1 second)
- Recognise aircraft (1 second)
- Recognise collision (5 second)
- Decision to turn right or left action (4 second)
- Muscular reaction lag time (0.4 second)
- Aircraft lag time (2 second)

Thus the total time is (12.5 second) which is the minimum time for the pilot to detect the intruder prior to the time of impact to have a chance of avoiding a collision. However, this time may increased as pilots differ in their response time, the reaction time for less experienced or older pilots is likely to be greater than (12.5 second) [8]. Pilots use their experience and the size of the intruder to determine the time to collision. Figure 4.32 represents an aircraft as it would appear to the pilot from the distances indicated in the table on the left [7]. The required time to cover these distances is given in seconds for relative speed of 600 mph and 210 mph.

Distance	TC [s]	TC [s]
	600 mph	210 mph
10 miles	60	170
5 miles	30	85
3 miles	18	56
2 miles	12	38
1 mile	6	13
0.5	3	9

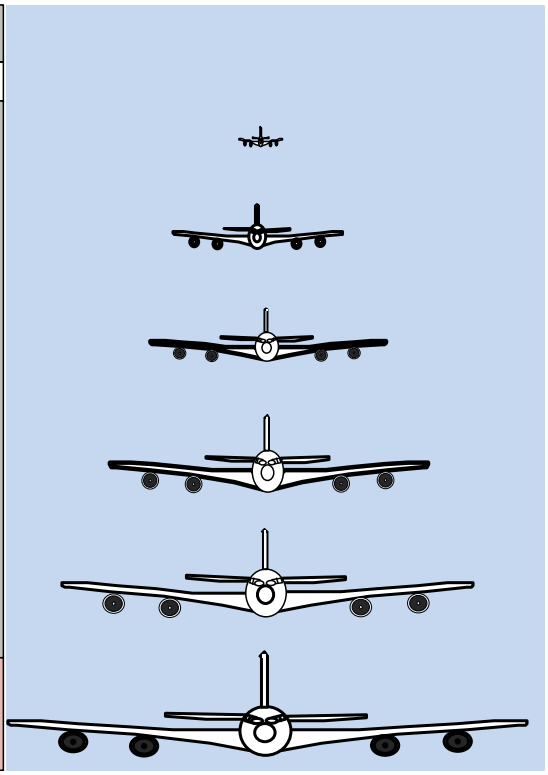

Figure 4.32 Aircraft closure rate chart

Two flags are generated by the collision assessment system:

1. The intruder at Nine O'clock Flag (NF): This flag is activated when the intruder become at at 9 o'clock position relative to the UAV. NF will be used by the avoidance manoeuvre generation system. This will be discussed in more detail in the next chapter (see Section 5.2.3). In some conflict avoidance scenarios (e.g. Approaching conflict scenarios) in manned aircraft the pilot observes the intruder position while performing the avoidance manoeuvre, and when the intruder reaches the 9 o'clock position the pilot tries to restore the initial heading angle of the aircraft. Hence, the aircraft flies parallel to its previous path.

2. Collision Resolved Flag (RF): This flag is activated when the collision is resolved. The collision is resolved if the intruder range is greater than a predefined value, and the relative speed or closure V_r is positive. RF flag is used for algorithm resetting purposes (see Sections 5.2.2, and 5.2.3).

4.8.5 Advisory System

Layer-1 and Layer-2 functions (shown in Figure 4.20) are data collection and assessment without any decision making functionality. Layer-3 is the first layer that has the decision making functionality in the DMS architecture. It uses the information that is generated by Layer-1 and Layer-2 with the rules of the air to generate the suitable Resolution Advisories (RA) which can be used by pilot or the remote pilot to avoid the conflict. In this thesis the level flight scenario is discussed so the horizontal advisories for avoidance manoeuvres are generated which are:

- Turn right manoeuvre: If AF1=1 and AF2=0 then the the RA is single (Right), but if (AF2=1 or CF=1) then the RA command is double (Right, Right) indicating that the pilot should make a greater right turn than the single (Right) command.

- Turn left manoeuvre: Same as turn right case so if AF1=1 then (Left); if AF2=1 or CF=1 then (Left,Left).

- Holding the current speed and altitude: The RA in this case is (Hold)

- Level off

- Descend or increase/decraese descending rate: (Descend) means the pilot should initiate descending manoeuvre, (Decrease/Increase Descend) for decreasing or increasing descending rate.

- Climb or increase/decrease climbing rate:(Climb) means the pilot should initiate climbing manoeuvre, (Decrease/Increase Climb) for decreasing or increasing climbing rate.

Figure 4.33 shows the flowchart for the advisory algorithm that is used in head-on collision scenarios. It can be seen that when (AF2) is not activated (which means that the intruder is far enough away to make vertical manoeuvre) if the aircraft is climbing or descending the RA command is to level off the aircraft. But if the aircraft is in level flight then the RA is (Right) in case there is no offset, and (Left) in the offset case. However, if (AF2 or CF) are activated then the vertical resolution advisories will not be taken into consideration for safety reasons due to the aircraft inertia in vertical manoeuvre. So the advisory system will just generate the horizontal resolution advisories which are (Right, Right) when no offset exists and (Left, Left) in the case of offset.

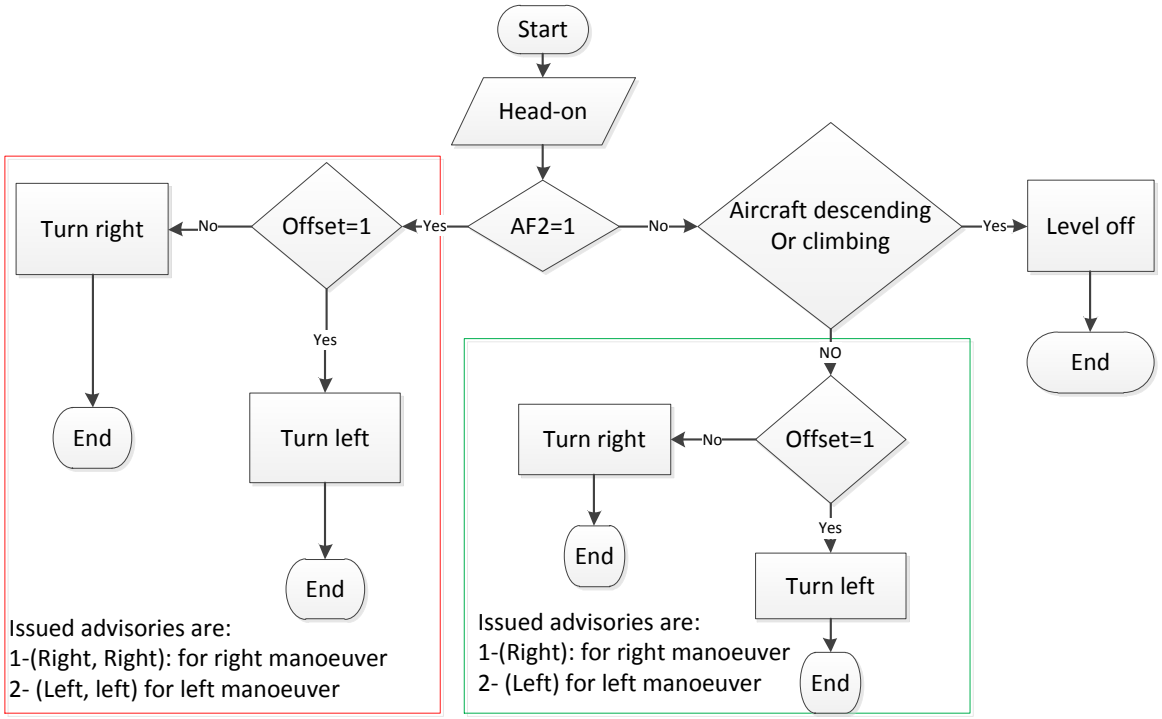

Figure 4.33 Flowchart for advisory generating during head-on collision

The purpose of using a single or double word when generating the resolution advisories is to give the pilot an indication of the level of the risk, hence the pilot takes this into account when initial an avoidance manoeuvre. For example, the turn rate (in case of a horizontal avoidance manoeuvre), when double word is issued by the advisory system, should be greater than the turn rate in case of single word command. Figure 4.34 gives the flowchart for the advisory algorithm that generates the resolution advisories in (overtaking, and being

overtaken) scenarios. In this flowchart (R_{alt}) refers to the relative altitude, where the altitude of the overtaking aircraft is the reference. For the same reason that has been discussed in head-on scenarios the vertical resolution advisories are taken into consideration just in case (AF1=1, and AF2=0).

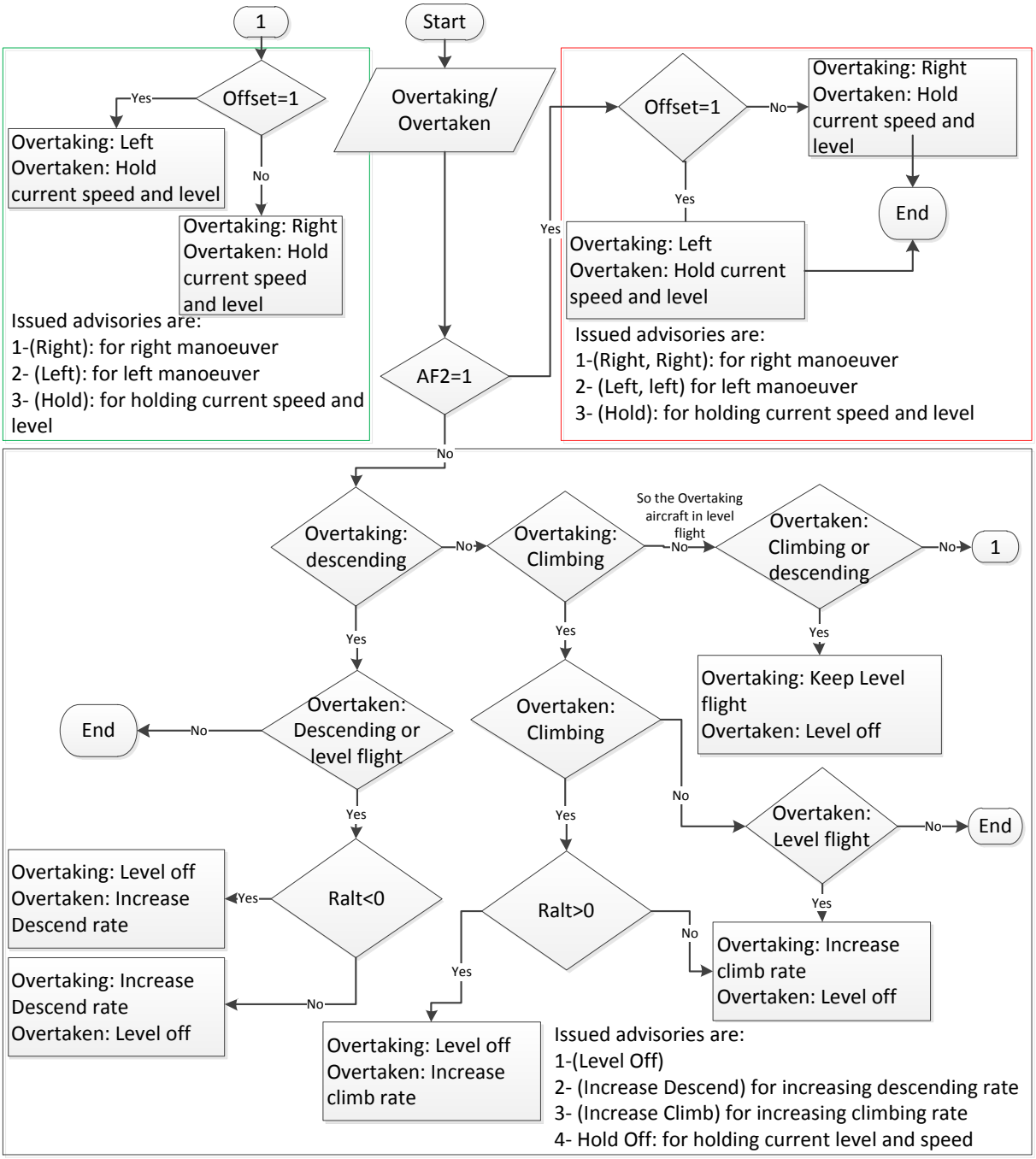

Figure 4.34 Flowchart for RA generation for overtaking/overtaken conflict scenarios

The conflict resolution advisories can be passed to the GUI unit in case of the remotely piloted UAV, or can be shown on the avionics for manned aircraft. Figure 4.35 shows the proposed layout for the GUI which displays the advisories as a text (e.g. Right; Left; Right, Right; Descend/Climb; Hold; Level Off), the resolution advisories are shown graphically using arrows to indicate the avoidance manoeuvre direction (Right/Left arrows), and the vertical advisories (Up arrow for climbing, and down arrow for descending) the length of arrows represents the value of turning, climbing, and descending that should be initiated by the pilot. The Auto/Manual command bottom gives the remote pilot two options:

- Manual: The pilot is responsible for performing the avoidance manoeuvre.
- Auto: The UAV is operating in highly autonomous mode where the avoidance manoeuvre is generated by CAS.

The manoeuvre generation layer (Layer-4) gives the proposed DMS the (Auto) functionality which will be discussed in more detail in the chapters 5 and 6.

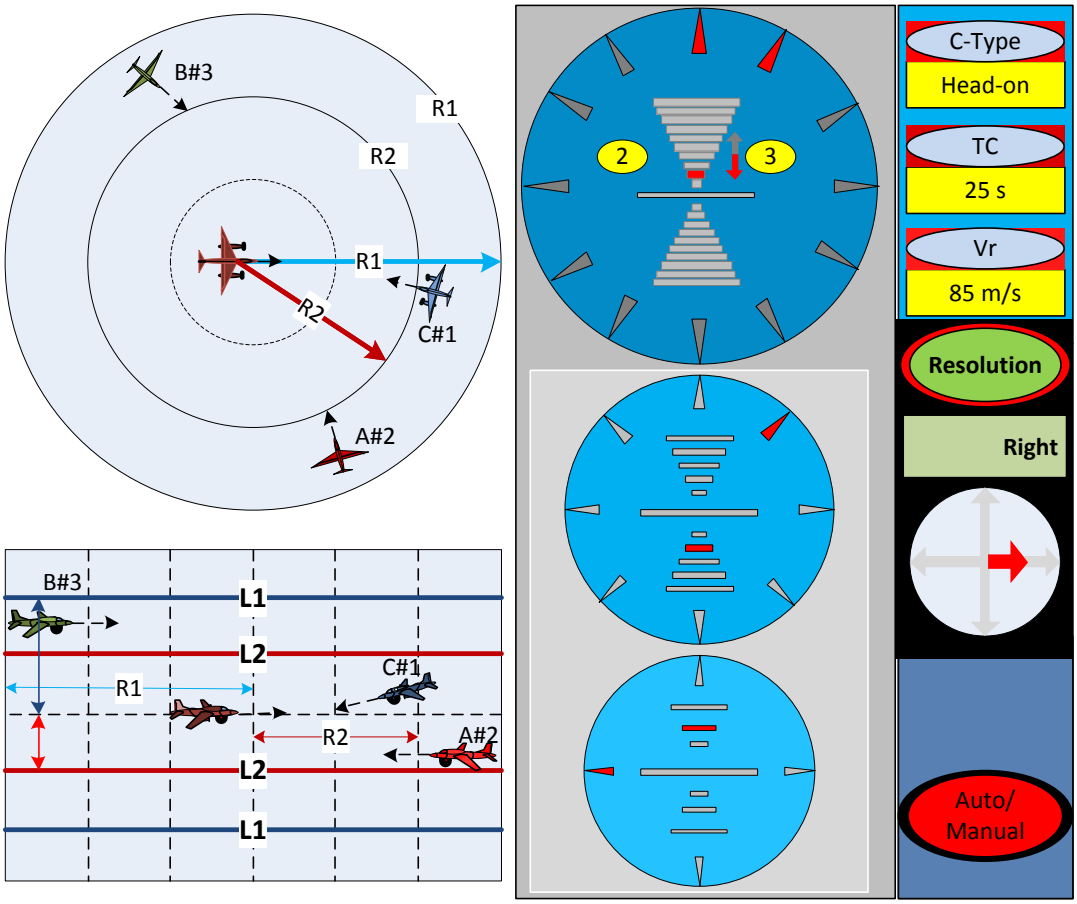

Figure 4.35 GUI/Avionics layout including the conflict resolution advisories

4.8.6 Avoidance Manoeuvre Generation

Avoidance manoeuvre generation is the final step in the proposed DMS. In order to make the UAV behave the same as a manned aircraft during the conflict scenarios, the generated avoidance manoeuvre must be compliant with the rules of the air. The avoidance manoeuvre generation layer uses the information from the upper layers in the DMS, and the rules of the air to generate the avoidance manoeuvre trajectory profiles. A predefined manoeuvre shape is used for generating the avoidance manoeuvre trajectory profiles. The predefined manoeuvre parameters are calculated based on the UAV and the intruder initial states, and the UAV dynamics constraints. The next chapter discusses in more detail the avoidance manoeuvres generation process for different conflict scenarios.

4.9 Summary

This chapter proposes a decision making system (DMS) algorithm for the collision avoidance systems (CAS). Firstly, a brief description of airspace classes, the rules of the air, and how aircraft should behave during collision avoidance in different conflict scenarios is provided. Then a decision making system (DMS) for collision avoidance is developed based on the rules of the air in VFR conditions and CAA requirements. The DMS is divided into multiple-layers, each layer is built to perform a specific function. The different functions that should be performed by the DMS layers are presented (e.g. conflict detections, prioritizing, and assessment). A graphical user interface (GUI) is proposed for algorithm testing and simulation purposes with a short comparison with currently used commercial Portable Collision Avoidance System (PCAS).

Chapter 5

Avoidance Manoeuvre Generation

5.1 Introduction

Avoidance manoeuvre generation is the final step in the proposed Decision Making System (DMS) that discussed in chapter 4. In order to make the UAV behave the same as a manned aircraft during the conflict scenarios the generated avoidance manoeuvre must be compliant with the rules of the air. This chapter discusses the avoidance manoeuvres generation process for different conflict scenarios in which the UAV changes direction (right/left turn) in a horizontal manoeuvre. Two conflict scenarios are discussed head-on/overtaking, and right approaching. Although the rules of the air give general instructions for avoiding the different conflict scenarios, there are no specific procedures for performing the avoidance manoeuvres. However, pilots use their experience to comply with the rules of the air.

The proposed avoidance manoeuvre generation process uses the information from the upper layers in the DMS and the rules of the air to generate the avoidance manoeuvre trajectory profiles. A predefined manoeuvre shape is used for generating the avoidance manoeuvre trajectory profiles. The predefined manoeuvre parameters are calculated based on the UAV and the intruder initial states together with the UAV dynamics constraints. For example, a combination of level flight and a coordinated turn manoeuvre with constant speed and altitude is selected for the horizontal avoidance manoeuvres. The coordinated turn manoeuvre is defined by the heading rate values during a period of time. Thus, the avoidance manoeuvre time is divided into periods of time each one associated with a specific value of heading rate (while the heading rate of coordinated turn has a non-zero constant value, it is zero in level flight). As the time periods are specified with their associated heading rate values the overall heading rate signal can be constructed. The avoidance manoeuvre trajectory

profiles are generated by applying the calculated heading rate signal to the UAV dynamic model. The generated avoidance manoeuvre trajectory profiles are, then, either passed to the inverse dynamics to generate the UAV control inputs, or used as global trajectory profiles for the Local Trajectory Planning (LTP) algorithm that was discussed in Chapter 3. The second option is considered in this thesis (see Section 5.4.1).

5.2 Avoidance Manoeuvre Trajectory Generation

The avoidance manoeuvre trajectory profile generation process can be divided into the following steps:

1. Specify the avoidance manoeuvre type and find its characteristics. The coordinated turn with a constant speed and altitude is selected to be the manoeuvre for the turn part of the avoidance manoeuvre and the level flight manoeuvre is selected for the straight flight part of the avoidance manoeuvre.

2. Define the heading rate of the coordinated turn according to the UAV current states and dynamic constraints (the heading rate for the level flight is zero). Then depending on the conflict scenario the manoeuvre time periods with the associated heading rate can be calculated.

3. Construct the heading rate signal and apply it to the UAV dynamic model to generate the avoidance manoeuvre trajectory profiles (position, and speed profiles).

4. The generated avoidance manoeuvre trajectory profiles can be used as global trajectory profiles for the Local Trajectory Planner (LTP) algorithm, or they can be passed directly to the inverse dynamic algorithm to generate the UAV command inputs.

5.2.1 Coordinated Turn with Constant Speed and Altitude

The coordinated turn with a constant speed and altitude is proposed for the turning part of the avoidance manoeuvre (selected based on pilot suggestions). Pilots prefer this type of manoeuvre for several reasons [65]:

- The constant speed means the coordinated turn is desirable for passenger comfort.

- It allows the pilot to function more effectively.

- It maintains maximum aerodynamic efficiency by minimizing sideslip and also minimizing undesirable aerodynamic loading of the structure.

- In the case of a remotely piloted UAV the turn coordination is useful for performing video surveillance or targeting.

- In a coordinated turn, level or otherwise, the aircraft maintains the same pitch angle and roll attitude with respect to the reference coordinate, but its heading angle changes continuously at constant rate. Therefore, the Euler-angle rates $\dot{\phi}$ and $\dot{\theta}$ are identically zero, and the turn rate $\dot{\psi}$ is a non-zero constant value.

Figure 5.1 shows the forces acting on the aircraft when it is flying in a steady level coordinated turn at a constant speed V.

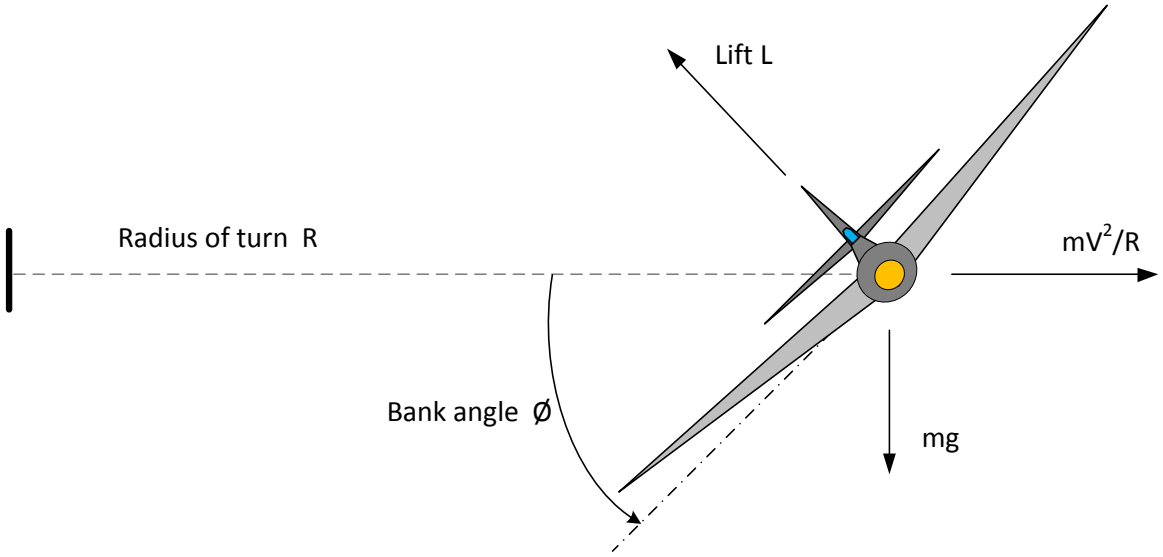

Figure 5.1 Forces balance in equilibrium state of turning aircraft

At an equilibrium state the acting forces on the aircraft are balanced. So the projection of the lift force on the horizontal axis balances the centrifugal force:

$$L\sin(\phi) = \frac{mV^2}{R} \tag{5.1}$$

and the projection of the lift force on the vertical axis balances aircraft wight force:

$$L\cos(\phi) = mg \tag{5.2}$$

hence, the turn radius can be calculated from 5.1 and 5.2:

$$R = \frac{V^2}{g\tan(\phi)} \tag{5.3}$$

The time to complete one turn is:

$$t = \frac{2\pi R}{V} = \frac{2\pi V}{g\tan(\phi)} \tag{5.4}$$

Heading rate or rate of turn is given by:

$$\dot{\psi} = \frac{2\pi}{t} = g\frac{\tan(\phi)}{V} \tag{5.5}$$

at a specific speed value the maximum heading rate is:

$$\dot{\psi}_{max} = g\frac{\tan(\phi_{max})}{V} \tag{5.6}$$

where ϕ_{max} is the maximum roll angle. In order to maintain constant altitude and speed during turning, lift and thrust must be increased to compensate the vertical lift reduction and the increase in the drag. The normal load factor is:

$$n = \frac{L}{mg} \tag{5.7}$$

The normal load factor during the turn can be derived from (5.2), and (5.7):

$$n = \frac{1}{\cos(\phi)} \tag{5.8}$$

So the bank angle is limited by the maximum normal load factor:

$$n_{max} = \frac{\frac{1}{2}\rho V^2 S C_{L_{max}}}{mg} = \frac{1}{\cos(\phi_{max})} \tag{5.9}$$

In addition, the stall speed must be taken into consideration in the generation of avoidance manoeuvre, because it increases during the turn:

$$V_{stall_{turn}} = V_{stall}\sqrt{n} = \frac{V_{stall}}{\sqrt{\cos(\phi)}} \tag{5.10}$$

where $V_{stall_{turn}}$ is the stall speed in the turn, V_{stall} is the stall speed in the level flight. Equation 5.10 shows that as the bank angle is increased the stall speed is increased. Also the load factor must be increased to maintain the aircraft altitude.

5.2.2 Avoidance Manoeuvre for Head-on/Overtaking Conflict Scenarios

The rules of the air for head-on and overtaking conflict scenarios are discussed in Chapter 4. Depending on the conflict scenario the UAV should avoid the conflict by turning (right/left). The direction of turn is determined by the upper layers of the DMS. The same manoeuvre type is proposed for the head-on and the overtaking conflict scenarios. Figure 5.2 shows the avoidance manoeuvre for a head-on conflict scenario (where the UAV should turn right). The proposed manoeuvre for the head-on/overtaking conflict scenarios are performed as follows:

1. Turn right: Use a coordinated turn with constant speed V and constant altitude to change heading angle by a predefined value $\Delta\psi_{T1}$. The constant heading rate during this time period (T_1) is $\dot{\psi}_1$.

2. Fly straight: Level flight with constant speed V, then when the achieved lateral distance is greater than a predefined clearance distance d_c.

3. Turn left: Using again the coordinated turn with the same turn rate $\dot{\psi}_3 = -\dot{\psi}_1$ and the same total change of heading angle $\Delta\psi_{T3} = -\Delta\psi_{T_1}$.

4. Fly straight: Keep flying parallel to the global trajectory till the collision resolution flag RF is issued by the collision assessment layer.

The avoidance manoeuvre that is shown in Figure 5.2, can be generated by using the heading rate that is shown in Figure 5.3 as command signal and holding the speed and altitude at constant values.

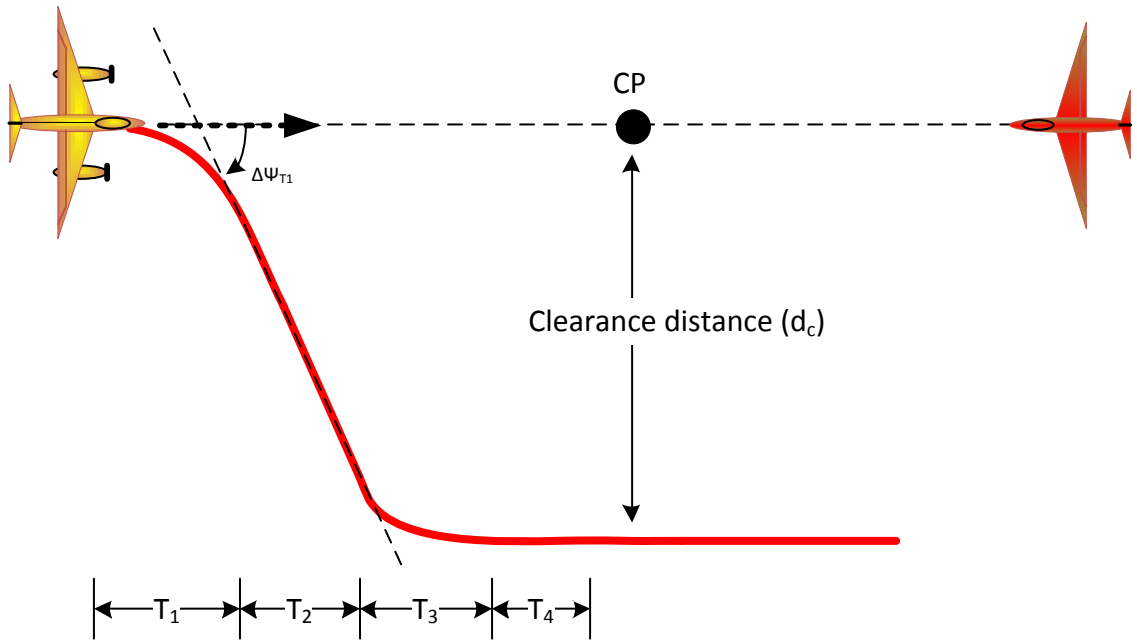

Figure 5.2 The avoidance manoeuvre for the head-on conflict scenario

Heading rate integration gives the heading angle for the selected manoeuvre as shown in Figure 5.4.

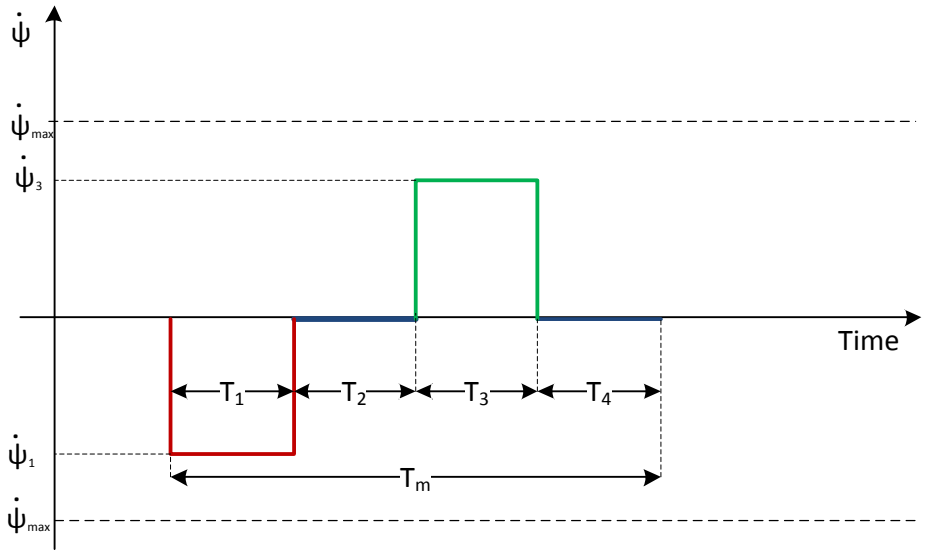

Figure 5.3 The heading rate of the proposed head-on collision avoidance manoeuvre

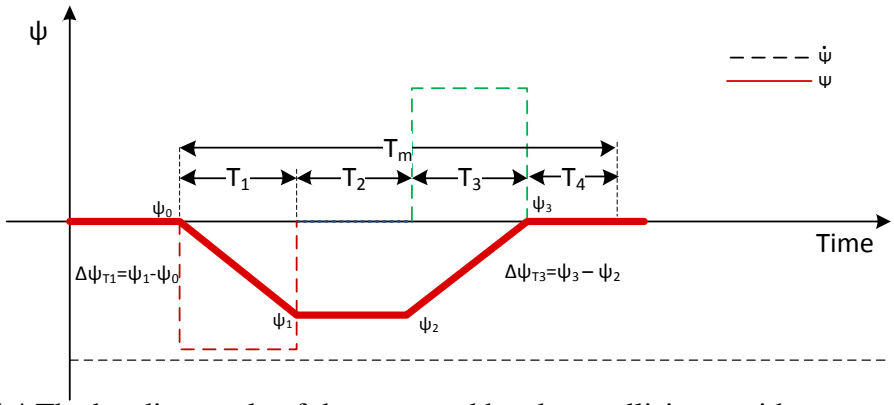

Figure 5.4 The heading angle of the proposed head-on collision avoidance manoeuvre

The proposed avoidance manoeuvre is governed by the selected heading rates ($\dot{\psi}_1$, $\dot{\psi}_3$), and the predefined total heading angle changes ($\Delta\psi_1$, $\Delta\psi_3$). The time periods (T_m, T_1, T_2, T_3, T_4) can be calculated depending on the heading rates and the total heading angle changes which have to satisfy the following conditions:

1. The heading difference $\Delta\psi_{T_1} < \frac{\pi}{2}$: To prevent the UAV from heading backward during and after resolving the conflict. This also will guarantee that the intruder will be in the Field of Regard (FOR) while the avoidance manoeuvre is performed (the requirement for the onboard sensor system for UAV is to cover the field of regard of ($\pm 110^{\circ}$) horizontal with respect to the longitudinal axis of the UAV, see Section 2.5.1)

2. The UAV heading should be the same at the start and end of the manoeuvre ($\psi_0 = \psi_3$): It can be noted that the change in heading angle occurs during (T_1) and (T_3), so this condition can be achieved by:

$$\Delta\psi_{T3} = -\Delta\psi_{T_1} \tag{5.11}$$

   where $\Delta\psi_{T_1} = T_1\dot{\psi}_1$, and $\Delta\psi_{T_3} = T_3\dot{\psi}_3$ are the total heading changes during the time periods T_1, and T_3 respectively so:

$$T_3\dot{\psi}_3 = -T_1\dot{\psi}_1 \Rightarrow \frac{\dot{\psi}_1}{\dot{\psi}_3} = -\frac{T_3}{T_1} \tag{5.12}$$

   in this research $\dot{\psi}_3 = -\dot{\psi}_1$, so time periods T_1, and T_3 are identical.

3. The second time period T_2 is selected to allow the UAV to achieve enough clearance distance from the expected Collision Point (CP), so the clearance distance should be greater or equal to a predefined value d_c.

Figure 5.5 shows a suggested right manoeuvre in which the achieved clearance distance, which is the summation of the distances (D_1, D_2, D_3) that achieved during the time periods (T_1, T_2, T_3) respectively, is equal to predefined value d_c, i.e.

$$D_1 + D_2 + D_3 = d_c \tag{5.13}$$

Now $\dot{\psi}_1 = -\dot{\psi}_3$, so $D_1 = D_3$, so from geometry

$$D_1 = R(1 - \cos(\Delta\psi_{T_1})) \tag{5.14}$$

where R is the turn radius given by (5.3), hence

$$D_2 = d_c - 2R(1 - \cos(\Delta\psi_{T_1})) \tag{5.15}$$

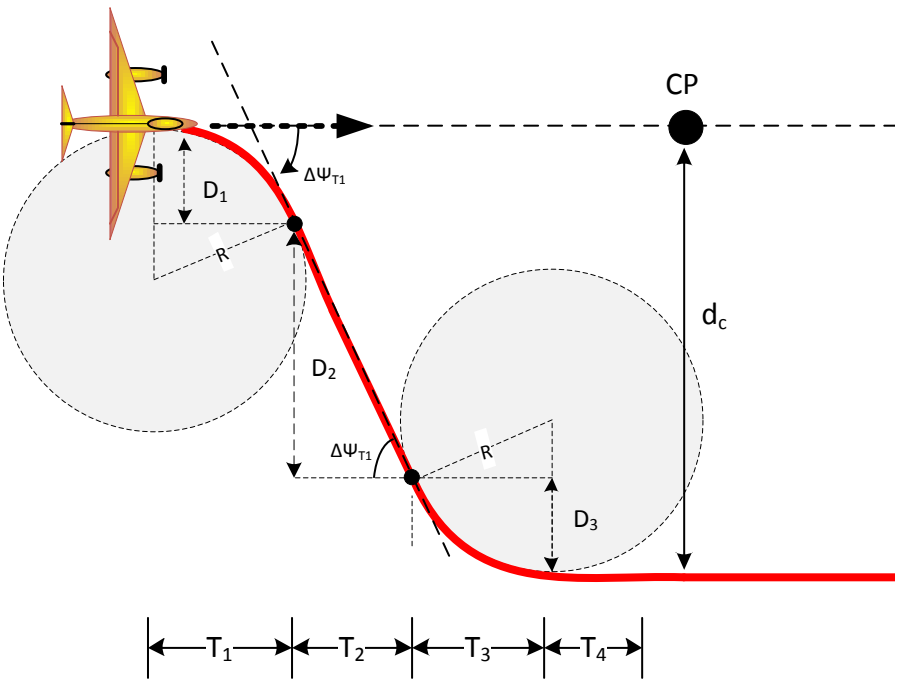

Figure 5.5 The geometric representation of the proposed head-on collision avoidance manoeuvre

The relationship between the time period T_2 and the distance D_2, at a constant speed V is:

$$T_2 = \frac{D_2}{V\sin(\Delta\psi_{T_1})} = \frac{d_c - 2R(1 - \cos(\Delta\psi_{T_1}))}{V\sin(\Delta\psi_{T_1})} \tag{5.16}$$

where $0 < \Delta\psi_{T_1} < \frac{\pi}{2}$.

4. Calculating T_4, and T_m: The manoeuvre time T_m is going to be used in the next steps, for instance, in the trajectory profiles discretisation, and in the speed profiles integration for position profile generation. Hence, it must be predefined as a constant value. The constant value of T_m must be enough to perform the avoidance manoeuvre and resolve the collision. The time period T_4 can be calculated:

$$T_4 = T_m - (T_1 + T_2 + T_3) \tag{5.17}$$

It worth mentioning that it is not necessary for the avoidance manoeuvre to take the whole manoeuvre time T_m, because it may be interrupted when the collision resolution flag RF is generated by collision assessment unit as seen in Figure 5.6. The collision resolution flag RF is generated when the conflict is resolved as discussed in Section 4.8.4.

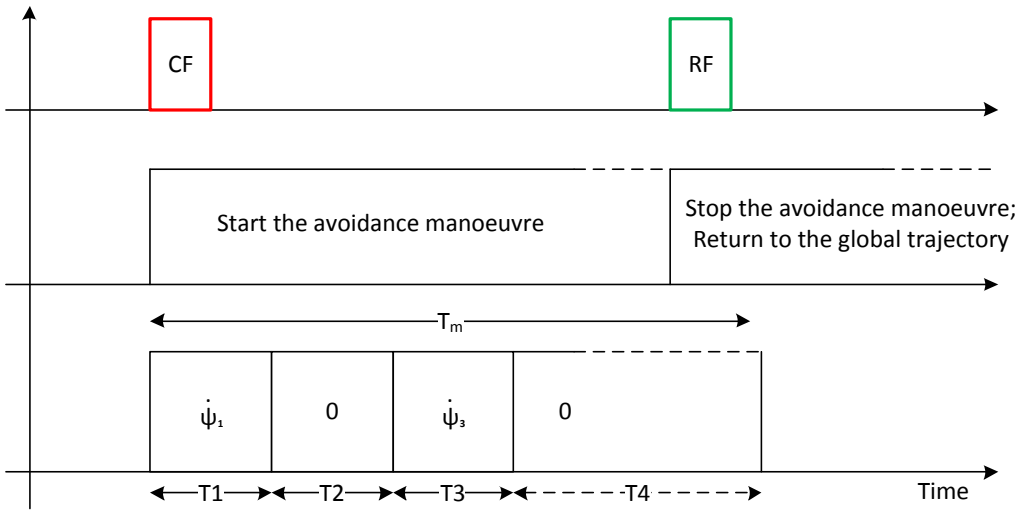

Figure 5.6 Avoidance manoeuvre initialization and interruption by CF, and RF

Two variations of the avoidance manoeuvre are proposed: average manoeuvre and exaggerated manoeuvre. The average manoeuvre is used when the time to collision T_c is within a predefined range value (e.g. greater than 20 sec). The exaggerated manoeuvre is used in the critical situation when the time to collision is small. In manned aircraft, the pilot uses an exaggerated turn manoeuvre to make the other pilot aware of the collision risk and to performe the avoidance manoeuvre. Table 5.1 gives the quantities that are used for each manoeuvre. Although the avoidance manoeuvres are proposed based on pilot suggestions, the proposed algorithm is not sufficient to allow the UAV to behave the same as a manned aircraft by generating different manoeuvres for different conflict situations. In a manned

Table 5.1 Average and exaggerated manoeuvre quantities

Quantity	Average manoeuvre	Exaggerated manoeuvre
T_c	$> 20s$	$\leq 20s$
$\Delta\psi$	$\frac{\pi}{4}$	$\frac{\pi}{3}$
$\dot{\psi}_o$	$[50\% - 75\%]\psi_{max}$	$[75\% - 100\%]\psi_{max}$

aircraft the pilot uses his/her experience to initiate a safe avoidance manoeuvre. Thus, a further development is proposed to use the pilot experience in avoidance manoeuvre generating process. A fuzzy logic technique is proposed to define the roll rate of the avoidance manoeuvre based on the pilot behaviour during different conflict scenarios. The proposal is discussed in more details in the Chapter 6.

5.2.3 Avoidance Manoeuvre for Approaching Scenarios

According to the rules of the air the UAV gives the way to the traffic that is approaching from the right side by turning right and tracking behind the approached intruder.

An aircraft which is obliged to give way to another aircraft must avoid passing over or under, or crossing ahead of, the other aircraft unless passing well clear of it [16] (see Chapter 4).

The proposal in this research is to use the coordinated turn manoeuvre at constant speed and altitude to generate the avoidance manoeuvre for the right approaching conflict scenarios. Three types of avoidance manoeuvre are proposed:

1. Right-Straight-Left (RSL) manoeuvre: The UAV turns right by $\frac{\pi}{2}$rad, then travels straight until the intruder is at 9 o'clock relative to the UAV (usually pilots use a clock position to give the relative direction of an object), at this point the UAV turns left by $\frac{\pi}{2}$rad, so the heading angle will be as same as the UAV when the avoidance manoeuvre is initiated. Thus, the UAV will be parallel with the global trajectory that was being tracked before the avoidance manoeuvre was initiated. If the advisory system command are issued to turn left first then this manoeuvre is simply becomes a Left-Strait-Right manoeuvre (LSR)

2. Right-Straight then Left-Straight (RS-LS) manoeuvre: The RSL avoidance manoeuvre is highly dependent on the intruder states, hence, a developed version of RSL avoidance manoeuvre is proposed to overcome this drawback. The RS-LS avoidance

manoeuvre is divided into two parts: Right-Straight (RS) then Left-Straight (LS). While the RS part is initiated by the collision flag CF, the LS part is initiated by the nine o'clock flag NF flag (the flag that is activated when the intruder is at 9 o'clock position relative to the UAV, see Section 4.8.4).

3. Circle Manoeuvre: The UAV turns right by 2πrad so the avoidance trajectory is a full circle, and the UAV will return to the global trajectory at the end of circulation, then it can continue tracking the global trajectory.

Figure 5.7 clarifies the proposed avoidance manoeuvres for avoiding the possible right approaching conflict scenarios:

- Scenario A: The UAV and the intruder initial positions are u_0, and a_0 respectively. The avoidance manoeuvre time T_m is divided into four time periods:

    1. Time period T_1: The UAV turns right by $\frac{\pi}{2}$rad during the time period T_1. At the end of T_1 the UAV position is u_1, and the intruder position is a_1.

    2. Time period T_2: During time period T_2 the UAV will travel straight from position u_1 to position u_{2a} at which the intruder position is a_2. It can be seen in Figure 5.7 that when the UAV at u_{2a} and the intruder at a_2 the intruder bearing angle is $\frac{\pi}{2}$rad (the intruder is at 9 o'clock of UAV). At this point the UAV will turn left by $\frac{\pi}{2}$rad.

    3. Time period T_3: The UAV turns left by $\frac{\pi}{2}$rad during the time period T_3, hence, the UAV heading at the end of T_3 will be as same as the UAV initial heading angle. At the end of T_3 the UAV will be at position u_{3a}, and the intruder position is a_3.

    4. Time period T_4: The UAV will travel straight during the time period T_4. Then the UAV can resume tracking the global trajectory when the conflict resolution flag RF is issued.

- Scenario B: In this scenario the UAV and the intruder initial positions are u_0, and b_0 respectively. The Avoidance manoeuvre steps are as same as the avoidance manoeuvre steps for scenario A, but the time period T_2 for scenario B is shorter than T_2 for scenario A (the time periods that shown at the bottom of Figure 5.7 are for scenario B). Conflict scenarios like A and B can be avoided by using the RSL or RS-LS avoidance

manoeuvres. However, these avoidance manoeuvres are not sufficient for avoiding all possible right approaching conflicts, for example, when the intruder position at the end of the time period T_1 is on the right of the UAV, or if it is at a head-on position such as the intruder in conflict scenario C. Therefore, the circle avoidance manoeuvre is proposed for this kind of conflict scenario.

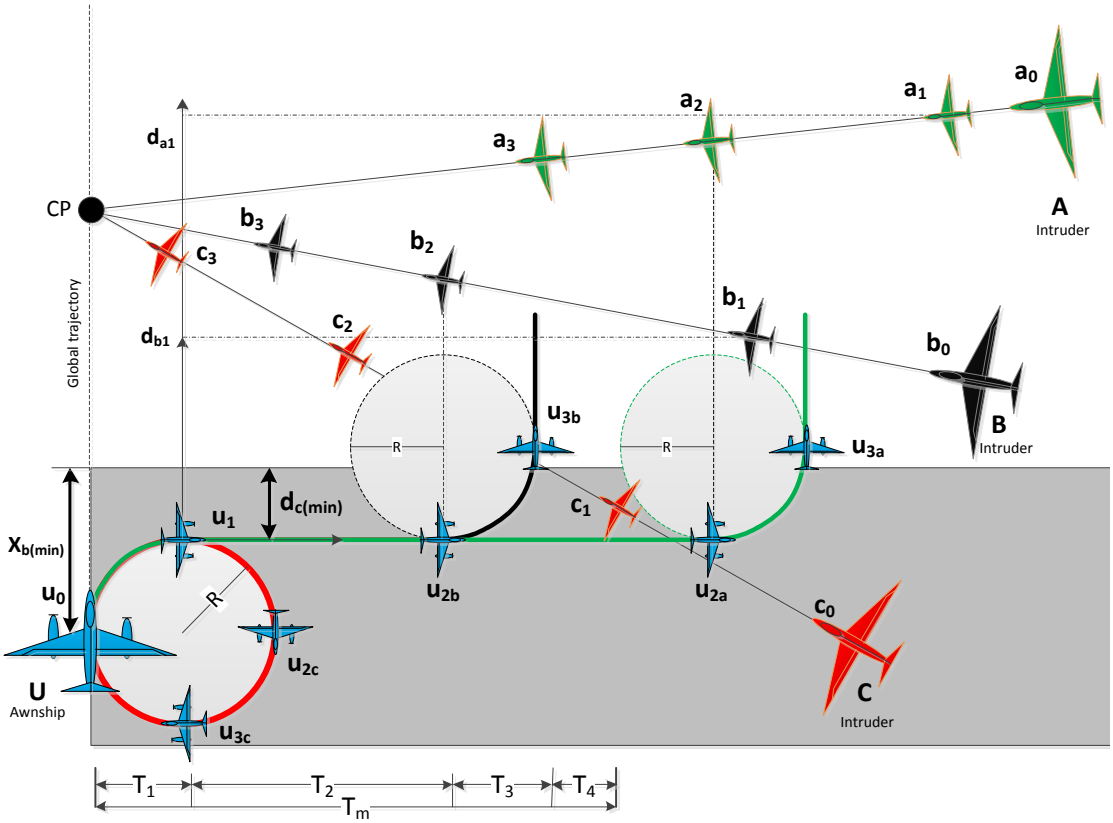

Figure 5.7 Avoidance manoeuvres for different right approaching conflict scenarios

- Scenario C: The initial position of intruder is c_0, and the UAV initial position is u_0. As can be seen in Figure 5.7, the position of the intruder at the end of time period T_1 is c_1 which is nearly at a head-on position with the UAV which at this moment is at u_1 position. So instead of going straight, the proposed manoeuvre, is to make a full circle manoeuvre. A minimum distance $d_{c(min)}$ is proposed to differentiate between the intruders that need to be avoided by using the RSL/RS-LS avoidance manoeuvres, or by using the circle type avoidance manoeuvre. The RSL/RS-LS avoidance manoeuvre are used for intruders which will be out of the shaded area, that shown in Figure 5.7, at the end of T_1 time period. The circle type avoidance manoeuvre is used for the intruders that will be inside the shaded area at the end of time period T_1. The shaded

area is determined by the predefined minimum clearance distance $d_{c(min)}$.

5.3 Avoidance Manoeuvre Parameterisation

This section discusses the proposed avoidance manoeuvres parameterisation. A geometric approach is used for parameterising the avoidance manoeuvres.

5.3.1 RSL Avoidance Manoeuvre Parameterisation

The RSL avoidance manoeuvre is used to avoid the right approaching conflict scenarios in which the intruder will not appear ahead of or to the right of the UAV when the $\frac{\pi}{2}$ rad turn is completed. Figure 5.8 shows the geometric representation of the proposed RSL avoidance manoeuvre which is divided into four time periods (T_1, T_2, T_3, and T_4). A constant speed and altitude coordinated turn manoeuvre is proposed to perform the turn parts of RSL manoeuvre during T_1, and T_3 periods, while the straight parts are achieved by a constant speed level flight.

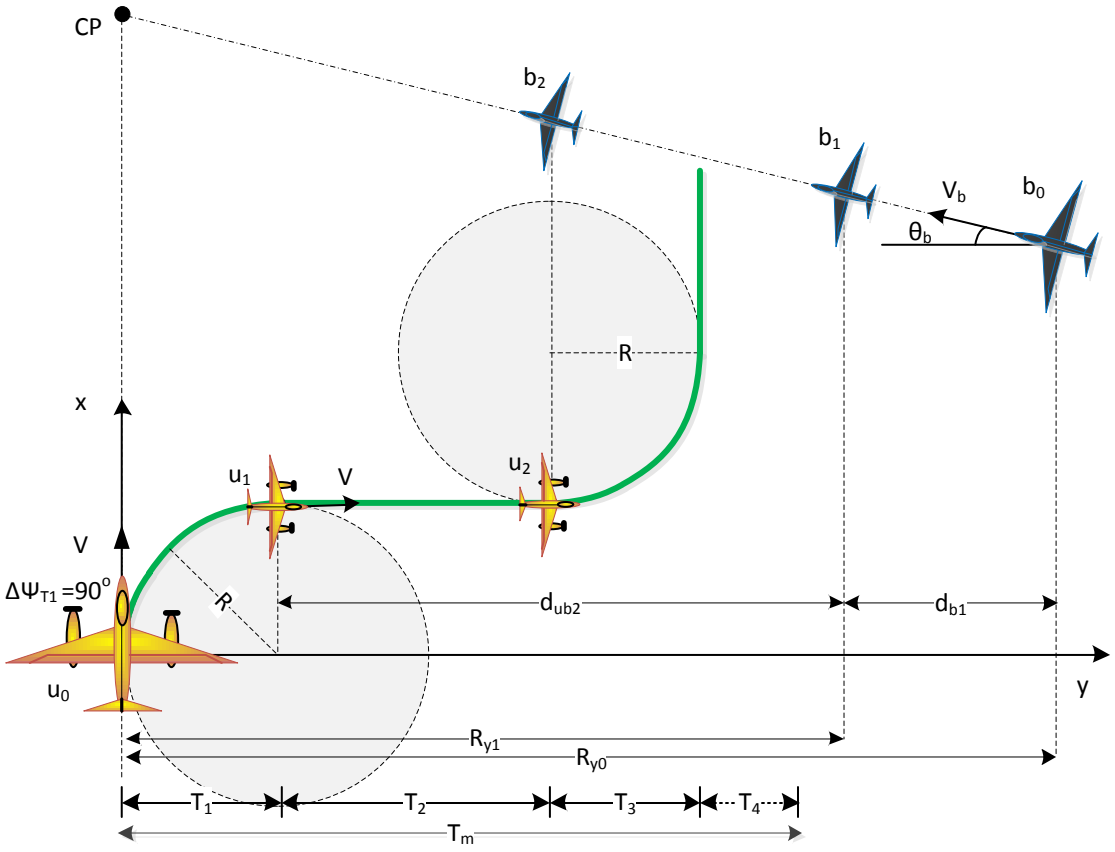

Figure 5.8 The RSL avoidance manoeuvre for right approaching conflict scenario

Figure 5.9 shows the heading rate that is used to produce the proposed RSL avoidance manoeuvre. The selected heading rate must not exceed the maximum heading rate of the UAV.

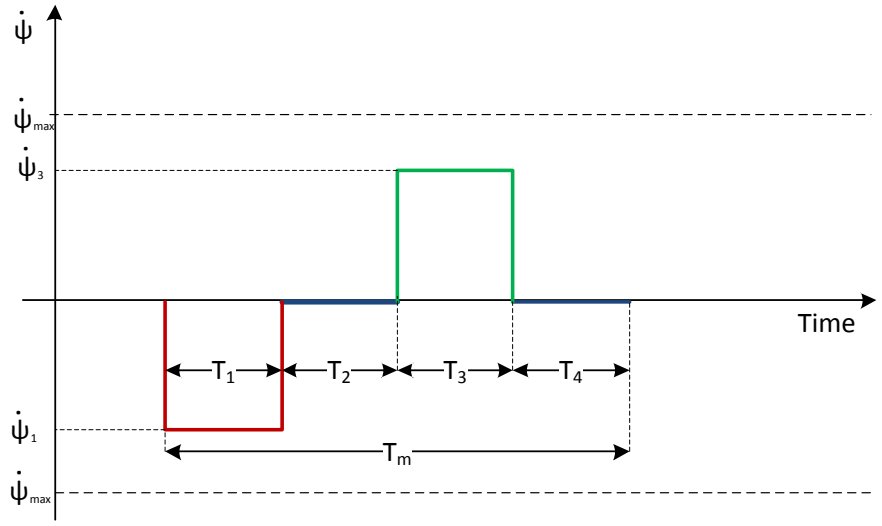

Figure 5.9 The heading rate signal that generates the proposed RSL avoidance manoeuvre

Integration of the heading rate gives the heading angle which is shown in Figure 5.10.

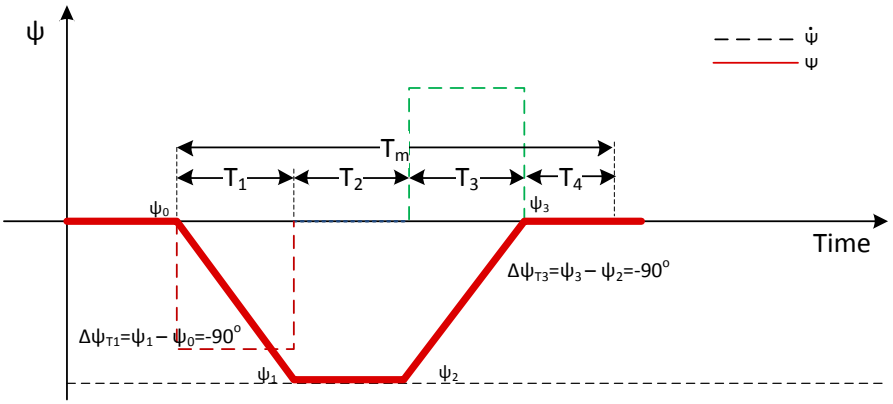

Figure 5.10 The heading angle of the RSL avoidance manoeuvre

The RSL avoidance manoeuvre parameters (T_m, T_1, T_2, T_3, and T_4) are calculated as follows:

- Define the heading rate of turn, that can be linked to the maximum heading rate by defining two type of manoeuvres (as it has done for the head-on/overtaking avoidance manoeuvre):

1. Average manoeuvre: $\dot{\psi}_1 = [50\% - 75\%]\dot{\psi}_{max}$

2. Exaggerated manoeuvre: $\dot{\psi}_1 = [75\% - 100\%]\dot{\psi}_{max}$

- Calculate the time period T_1:

$$T_1 = \frac{\Delta\psi_{T_1}}{\dot{\psi}_1} = \frac{\pi}{2\dot{\psi}_1} \tag{5.18}$$

- Calculate the time period T_2: At the beginning of time period T_2 the UAV, and the intruder are at u_1, and b_1 positions respectively. They move to positions u_2, and b_2 at the end of T_2. Hence, the time period T_2 is given by:

$$T_2 = \frac{d_{ub2}}{V + V_b\cos(\theta_b)} \tag{5.19}$$

where V_b is the intruder speed, θ_b is the angle between the intruder speed vector and the y-axis, and d_{ub2} is the projection of the distance between the UAV and the intruder at the end of the time period T_1 on the y-axis:

$$d_{ub2} = R_{y0} - (R + d_{b1}) \tag{5.20}$$

where R_{y0} is the initial intruder range projection on the y-axis, R is the turn radius given by (5.3), and d_{b1} is the projection on the y-axis of the traveled distance by the intruder during the time period T_1:

$$d_{b1} = V_b\cos(\theta_b)T_1 \tag{5.21}$$

Unlike the avoidance manoeuvre for the head-on/overtaking conflict scenarios, the RSL avoidance manoeuvre calculation depends on the intruder speed and heading (during the time period T_2).

- Calculate the time period T_3: In order to make sure that the UAV heading angle at the end of time period T_3 is the same as the UAV initial heading angle, the time period T_3 is chosen to be equal to the time period T_1, and $\dot{\psi}_3 = -\dot{\psi}_1$.

- Calculate the time period T_4: $T_4 = T_m - (T_1 + T_2 + T_3)$.

5.3.2 RS-LS Avoidance Manoeuvre Parameterisation

The time period T_2 in the RSL avoidance manoeuvre calculation depends on the intruder's initial state (heading angle, and speed). The intruder's speed and heading angle are supposed to be held during the conflict when the UAV is performing the avoidance manoeuvre (according to the rules of the air). However, mismatch between the actual intruder state and the available one (measured by the on-board sensing unit, or provided by the ground station) should be taken into consideration. Also, using the available intruder's initial state values to calculate the whole avoidance manoeuvre is not sufficient if these values are changed during the avoidance manoeuvre performing. Therefore, updated values of the intruder's state can be useful to reduce the mismatch effects and to overcome the intruder's state values changes problem. Hence, the RS-LS avoidance manoeuvre, which is a modified version of RSL avoidance manoeuvre, is proposed to overcome the RSL drawbacks. The RS-LS avoidance manoeuvre is divided into two parts: Right-Straight RS, then Left-Straight LS. While the RS part is initiated by the collision flag CF, the LS part is initiated by NF flag. The NF flag is proposed to be activated when the UAV is performing the avoidance manoeuvre, and the intruder is at the 9 o'clock position relative to the UAV. Figure 5.11 illustrates the sequence of the RS-LS avoidance manoeuvre parts with the collision assessment flags (CF, NF, and RF), and shows the heading rate values during the time periods.

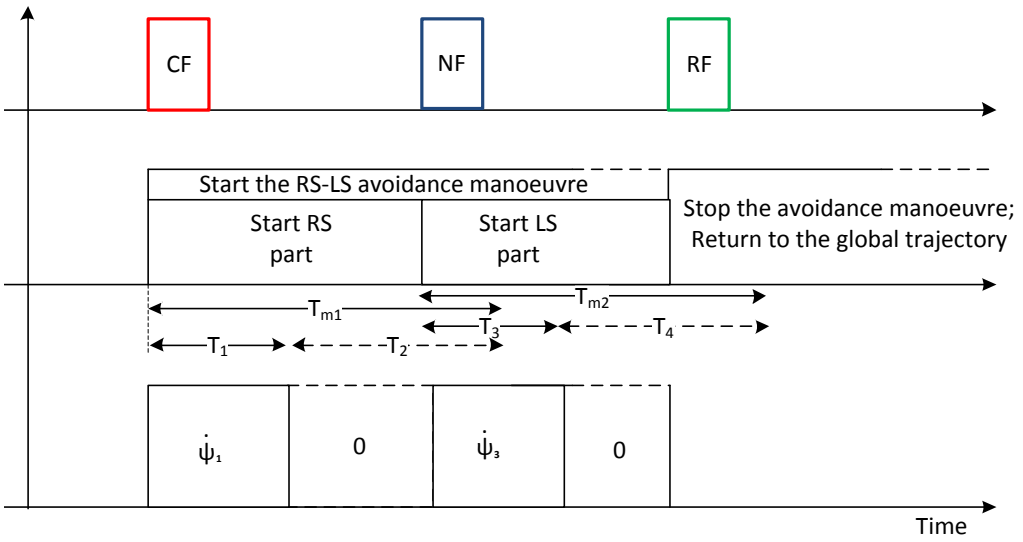

Figure 5.11 The RS-LS avoidance manoeuvre parts sequence

- Predefined time values T_{m1}, and T_{m2} are given for manoeuvre parts RS, LS parts respectively. They should be selected to be large enough to perform the avoidance manoeuvre.

- The time periods T_1, and T_3 are calculated in the same manner as in the RSL avoidance manoeuvre.

- While the time period $T_2 = T_{m1} - T_1$ is interrupted by the NF flag, the time period $T_4 = T_{m2} - T_3$ is interrupted by the RF flag.

5.3.3 Circle Avoidance Manoeuvre Parameterisation

The RSL/RS-LS avoidance manoeuvres are not suitable for some right approaching conflict scenarios. For example, conflict scenario C that is discussed in Section 5.2.3 in which the position of the intruder at the end of time period T_1 is nearly at a head-on position with the UAV. So instead of going straight, the proposed manoeuvre, is to make a full circle manoeuvre. The circle avoidance manoeuvre is proposed for the right approaching conflict scenarios if the intruder will be at the head-on position, or at the right of the UAV when it completes $\frac{\pi}{2}$ rad right turn. This will guarantee that the UAV is not going to cross the intruder flight path (this is one of the rules of the air requirements). Figure 5.12 gives a geometric representation of the circle avoidance manoeuvre.

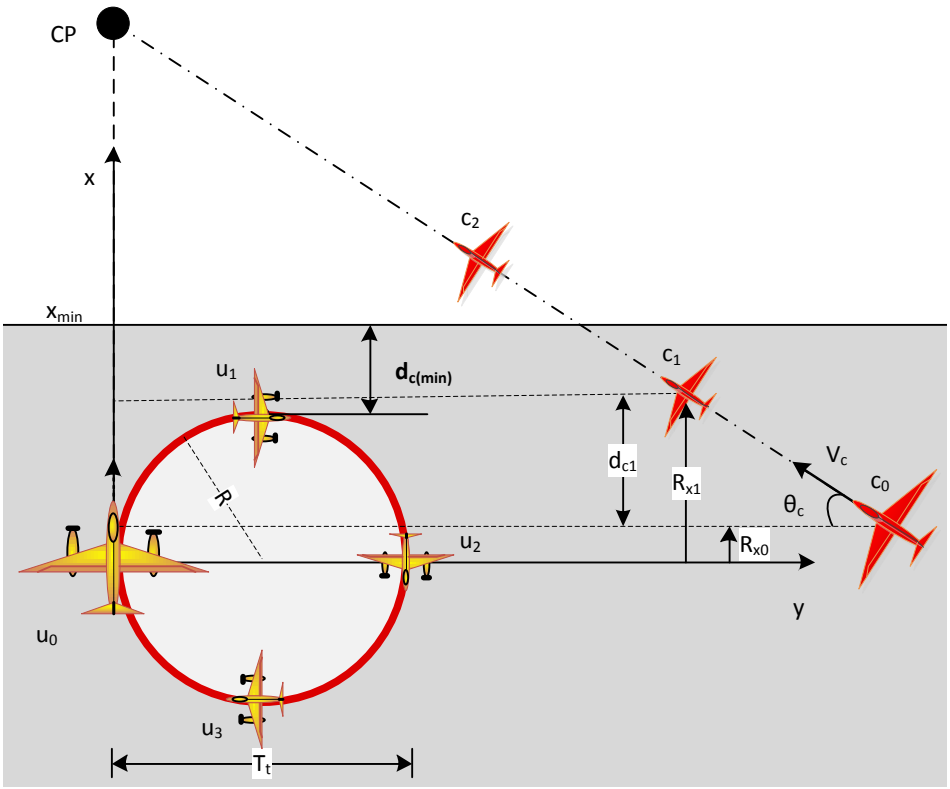

Figure 5.12 The geometric representation of the proposed circle avoidance manoeuvre

The condition to perform the circle manoeuvre instead of the RSL/RS-LS avoidance manoeuvre is:

$$R_{x1} \leq X_{min} \Rightarrow R_{x1} \leq R + d_{c(min)} \tag{5.22}$$

where R_{x1} is the intruder range projection on the x-axis, when it at the position c_1 (the intruder ranges are measured from the reference frame origin at u_0), R is the turn radius, and $d_{c(min)}$ is a predefined minimum clearance distance, measured from the UAV position u_1 (when it complete $\frac{\pi}{2}$ rad right turn). The intruder range projections R_{x1} is given by:

$$R_{x1} = R_{x0} + d_{c1} \tag{5.23}$$

where R_{x0} is the intruder range projection on the x-axis when it is at the position c_0, and d_{c1} is the projection of the intruder traveled distance from c_0 to c_1 given by:

$$d_{c1} = \frac{T_t}{4} V_c \sin(\theta_c) \tag{5.24}$$

where V_c is the intruder speed, θ_c is the angle between the y-axis and the intruder speed vector as shown in Figure 5.12, and T_t is a full circle turn time given by (5.26). Then the circle avoidance manoeuvre condition can be written:

$$R_{x0} \leq R + d_{c(min)} - \frac{T_t}{4} V_c \sin(\theta_c) \tag{5.25}$$

It can be noted that this avoidance manoeuvre is simply a circle path with a constant speed and altitude (coordinated turn). The circle radius is controlled by the selected heading rate $\dot{\psi}$ shown in Figure 5.13, and the UAV speed V. The relationship of circle radius is give by (5.3). The UAV heading angle is shown in Figure 5.14. As can be seen in Figures 5.13 and 5.14 the manoeuvre is divided into two periods:

- The circle turn time period T_t:

$$T_t = \frac{\Delta\psi_{T_t}}{\dot{\psi}_1} = \frac{2\pi}{\dot{\psi}_1} \tag{5.26}$$

  where $\dot{\psi}_1$ is the selected avoidance heading rate

- The time period T_4: As the manoeuvre time T_m is predefined, so $T_4 = T_m - T_t$

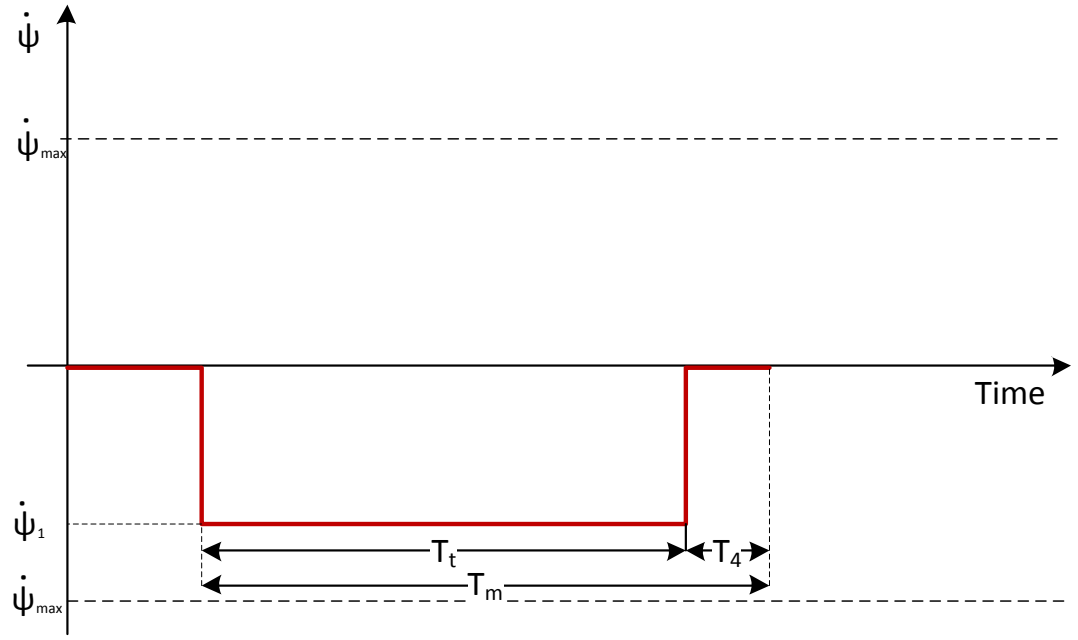

Figure 5.13 Heading rate for the circle avoidance manoeuvre

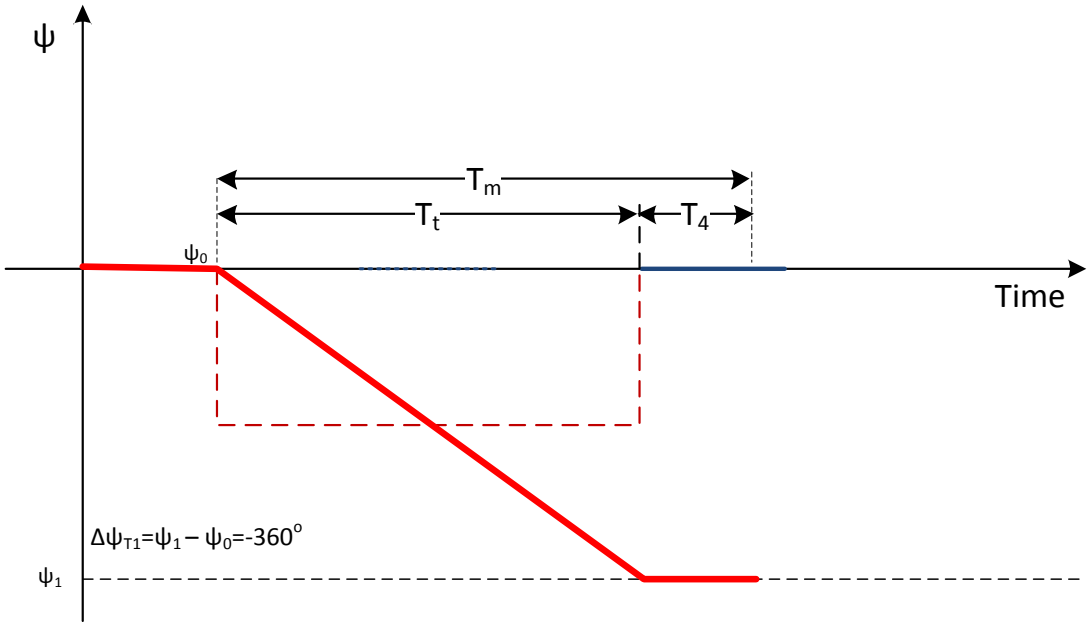

Figure 5.14 Heading angle for the circle avoidance manoeuvre

5.4 Avoidance Manoeuvre Trajectory Profiles Generation and Parameterisation

Trajectory profiles are needed to be used in the next steps of the collision avoidance system. The calculated trajectory profiles can be sent either to the local trajectory planning algorithm, which discussed in Chapter 3, or they can be curve fitted then sent directly to the inverse is dynamic algorithm which generates the UAV command signals based on the trajectory profiles. In both methods the avoidance manoeuvre trajectory profiles (speed, and position) are needed. The constructed signal of the heading rate of the avoidance manoeuvre is used as input for the UAV lateral directional model to generate the avoidance manoeuvre trajectory profiles (speed, and position). The lateral directional dynamics of the UAV under constant speed and altitude is given by [76]:

$$\begin{aligned} \dot{x} &= V\cos\psi \\ \dot{y} &= V\sin\psi \\ \dot{\psi} &= U \end{aligned} \tag{5.27}$$

where (x,y) are the UAV position, V is the UAV speed, and U is the input signal. The discrete representation of the lateral directional dynamics can be can be formed by using numerical method. Euler method is used to obtain the discrete form of the dynamic system due to its simplicity. One disadvantage of Euler method is its poor accuracy comparing to the relatively more sophisticated methods which, in general, are time consuming [77]. Accuracy problem can be overcome by reducing the step size. However, the need of higher accuracy can be reduced by increasing the clearance distance between the aircraft and the intruder.

$$\begin{aligned} \dot{x}(k) &= V\cos\psi(k) \\ \dot{y}(k) &= V\sin\psi(k) \\ \dot{\psi}(k) &= U(k) \\ x(k+1) &= x(k) + T_s\dot{x}(k) \\ y(k+1) &= y(k) + T_s\dot{y}(k) \\ \psi(k+1) &= \psi(k) + T_s\dot{\psi}(k) \end{aligned} \tag{5.28}$$

where T_s is the sampling time, and $k = 0, 1, \ldots, n$ is the discrete steps. The relationship between the time manoeuvre and sampling time is $n = \frac{T_m}{T_s}$. Hence, the constructed heading rate signal is discretised then used to generate the discrete speed and position profiles of the proposed avoidance manoeuvre.

$$\dot{\psi} = [\dot{\psi}(0), \dot{\psi}(1), \ldots, \dot{\psi}(n-1), \dot{\psi}(n)] \tag{5.29}$$

Figure 5.15 shows the flowchart of the avoidance manoeuvre trajectory profiles calculation.

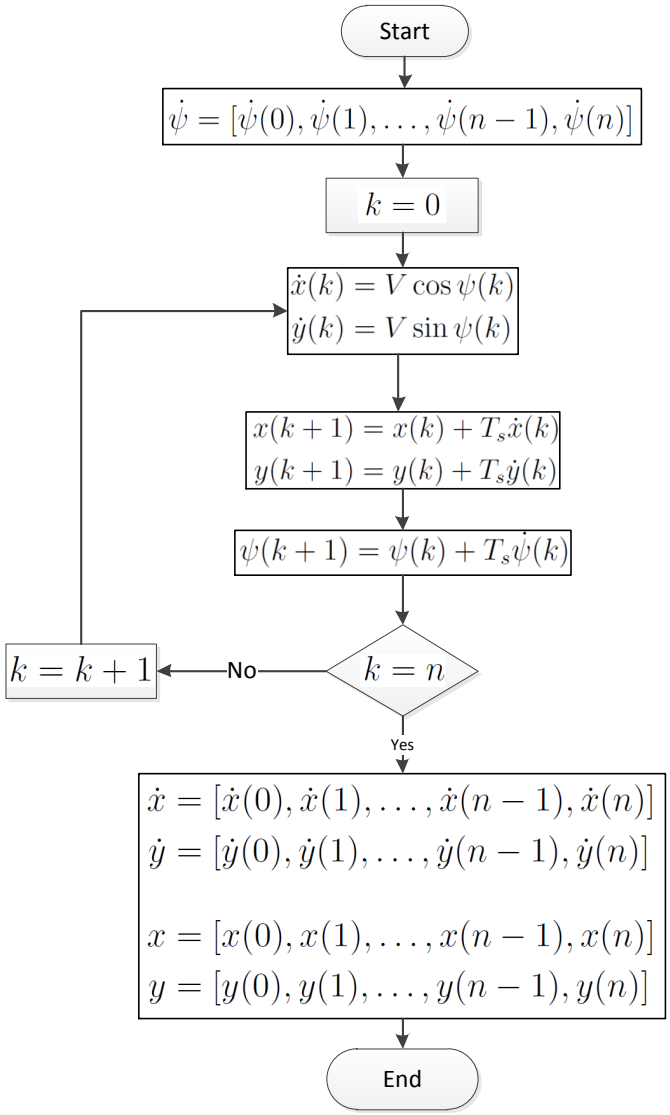

Figure 5.15 Flowchart of discrete trajectory profiles calculation

The generated trajectory profiles (position, and speed) can then be curve fitted to cal-

culate all the trajectory profiles (position, and speed, acceleration, and jerk) which can be used by the inverse dynamic algorithm to generate the command signals for the UAV control system. Alternatively it can be used as a global trajectory for the LTP algorithm.

5.4.1 Avoidance Manoeuvre Trajectory Profiles Curve Fitting

The inverse dynamic algorithm that generates the command signal for the UAV depends on the trajectory profiles (position, speed, acceleration, and rate of acceleration) in the 3D reference frame. However, the discretized lateral directional model of the UAV 5.28 with the constructed heading rate signal as input generates only the speed and position profiles in the 2D reference frame. Thus, a 6^{th} order Bezier curve is used to curve fit the generated speed profiles. Then the Bezier curve coefficients are used to generate the avoidance manoeuvre trajectory profiles in the 3D reference frame. The curve fitting algorithm uses the least squares curve fitting technique which is discussed in Appendix A. The coefficients of the Bezier curve that fits the speed profiles are calculated as:

$$\begin{aligned} C^{\dot{x}} &= B_{ls}\dot{x} \\ C^{\dot{y}} &= B_{ls}\dot{y} \\ C^{\dot{z}} &= B_{ls}\dot{z} = 0 \end{aligned} \tag{5.30}$$

where $C = [c_0, c_1, \ldots, c_6]$ are the Bezier curve coefficients, B_{ls} is the curve-fit matrix which is calculated off-line using the Bezier basis function matrices B (see Appendix A):

$$B_{ls} = (B^T B)^{-1} B^T \tag{5.31}$$

The coefficients of the Beizer curve are used to calculate the trajectory profiles (position, speed, acceleration, and rate of acceleration) in the 3D reference frame. The curve fitted speed profiles in forward (u), lateral (v), and vertical (w) axes are written:

$$\begin{aligned} u(\tau) &= c_0^{\dot{x}}B_0(\tau) + c_1^{\dot{x}}B_1(\tau) + c_2^{\dot{x}}B_2(\tau) + \cdots + c_6^{\dot{x}}B_6(\tau) \\ v(\tau) &= c_0^{\dot{y}}B_0(\tau) + c_1^{\dot{y}}B_1(\tau) + c_2^{\dot{y}}B_2(\tau) + \cdots + c_6^{\dot{y}}B_6(\tau) \\ w(\tau) &= c_0^{\dot{z}}B_0(\tau) + c_1^{\dot{z}}B_1(\tau) + c_2^{\dot{z}}B_2(\tau) + \cdots + c_6^{\dot{z}}B_6(\tau) \end{aligned} \tag{5.32}$$

However, in horizontal manoeuvres the vertical speed ($w(\tau) = 0$), hence:

$$c_i^{\dot{z}} = 0; i = 0, 1, \cdots, 6 \tag{5.33}$$

The relationship between the parameter τ and the avoidance manoeuvre time (T_m) can be used to represent the time $t = T_m.\tau$. Therefore, the acceleration and the jerk profiles are given by:

$$\frac{du}{dt} = \frac{1}{T_m}\left(c_0^{\dot{x}}\frac{dB_0(\tau)}{d\tau} + \cdots + c_6^{\dot{x}}\frac{dB_6(\tau)}{d\tau}\right) \tag{5.34}$$

and

$$\frac{d^2u}{dt^2} = \frac{1}{T_m^2}\left(c_0^{\dot{x}}\frac{d^2B_0(\tau)}{d\tau^2} + \cdots + c_6^{\dot{x}}\frac{d^2B_6(\tau)}{d\tau^2}\right) \tag{5.35}$$

The acceleration and the jerk profiles for the lateral axis can be calculated in a similar way. The acceleration and jerk profiles for the vertical axis are zeros. The position profile is driven by integration of the basis functions with respect to time t.

$$B_i^{int} = \int_0^{T_m} B_i(t)dt; \quad i = 0,1,\ldots,6 \tag{5.36}$$

Hence, the trajectory profiles calculation is reduced to simple matrix multiplication:

$$\begin{aligned} u &= C^{\dot{x}^T}B, & \dot{u} &= \frac{1}{T_m}C^{\dot{x}^T}B', \\ \ddot{u} &= \frac{1}{T_m^2}C^{\dot{x}^T}B'', & x &= x_0 + C^{\dot{x}^T}B^{int} \end{aligned} \tag{5.37}$$

$$\begin{aligned} v &= C^{\dot{y}^T}B, & \dot{v} &= \frac{1}{T_m}C^{\dot{y}^T}B', \\ \ddot{v} &= \frac{1}{T_m^2}C^{\dot{y}^T}B'', & y &= y_0 + C^{\dot{y}^T}B^{int} \end{aligned} \tag{5.38}$$

$$\begin{aligned} w &= C^{\dot{z}^T}B, & \dot{w} &= \frac{1}{T_m}C^{\dot{z}^T}B', \\ \ddot{w} &= \frac{1}{T_m^2}C^{\dot{z}^T}B'', & z &= z_0 + C^{\dot{z}^T}B^{int} \end{aligned} \tag{5.39}$$

where:
B: Is the discretised basis function matrix given by:

$$B = \begin{pmatrix} B_0(\tau_1) & B_0(\tau_2) & \cdots & B_0(\tau_n) \\ B_1(\tau_1) & B_1(\tau_2) & \cdots & B_1(\tau_n) \\ \vdots & \vdots & \cdots & \vdots \\ B_6(\tau_1) & B_6(\tau_2) & \cdots & B_6(\tau_n) \end{pmatrix} \tag{5.40}$$

B', B'': Are the discretised basis function derivatives matrices. More about these matrices can be found in Chapter 3.
C^{u^T}, C^{v^T}, C^{w^T}: Are the vectors of coefficients for forward, lateral, and vertical axis:

$$\begin{aligned} C^{\dot{x}^T} &= \begin{bmatrix} c_0^{\dot{x}} & c_1^{\dot{x}} & \cdots & c_6^{\dot{x}} \end{bmatrix} \\ C^{\dot{y}^T} &= \begin{bmatrix} c_0^{\dot{y}} & c_1^{\dot{y}} & \cdots & c_6^{\dot{y}} \end{bmatrix} \\ C^{\dot{z}^T} &= \begin{bmatrix} c_0^{\dot{z}} & c_1^{\dot{z}} & \cdots & c_6^{\dot{z}} \end{bmatrix} \end{aligned} \tag{5.41}$$

The generated trajectory profiles are then passed directly to the inverse dynamics to generate the control commands for the UAV control system. Although this method has some advantages, such as, simplicity, and reduction the CAS algorithm computational time, it has some drawbacks:

- The curve fitting process accuracy depends on the selected manoeuvre time T_m, and the number of discretising steps n. So using long manoeuvre time reduces the fitting accuracy.

- The trajectory profiles are generated without considering the UAV dynamic constraints, which may lead to controller saturation in the UAV control layer.

- Passing the generated trajectory profiles directly to the inverse dynamics, means that the UAV will not be able to avoid any sudden collision with any pop-up obstacles that may happen during the avoidance manoeuvre execution. However, if the UAV is flying in a well known environment and the possibility of having a pop-up obstacle is canceled this method can be used, but it needs more development to include the UAV dynamics constraints in the curve fitting process.

The other proposed method uses the generated trajectory profiles (position, and speed that generated by the UAV lateral directional model Figure 5.15) as a global trajectory for the local trajectory planning (LTP) which uses the receding horizon technique to generate the trajectory profiles (the LTP method is discussed in Chapter 2). Although the LTP algorithm avoids the curve-fitting method drawbacks because it curve fits a specific portion of the avoidance manoeuvre trajectory profiles at each step of time horizon and also takes the UAV and obstacle constraints into consideration, its computational time is higher. In this thesis this method is used. The two methods are shown in the flowchart in Figure 5.16 which gives the head-on/overtaking avoidance manoeuvre as an example.

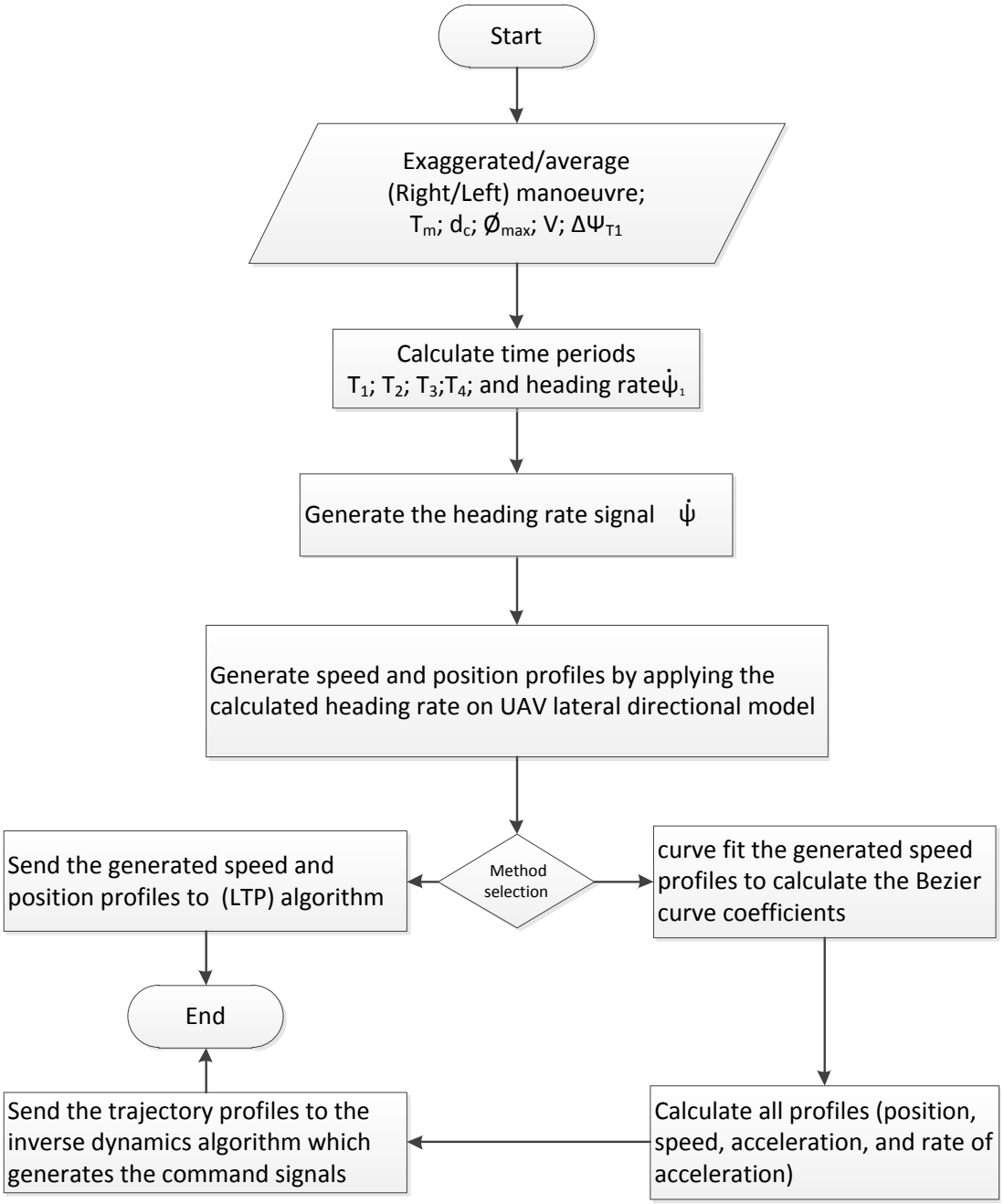

Figure 5.16 CAS Flowchart

5.5 Simulation Results

This section discusses the simulation results for some avoidance manoeuvre scenarios. MATLAB/Simulink are used to test the proposed avoidance manoeuvres for different collision scenarios. Figure 5.17 shows the block diagram that is used to generate the simulation results. The UAV with the controllers are modeled by the point-mass model, and it is assumed that the UAV is tracking the generated command signals exactly. The switch and the dashed link that are shown in the block diagram represent the method for generating the command signals by passing the trajectory profiles, that are generated by the curve fitting algorithm, directly to the inverse dynamic.

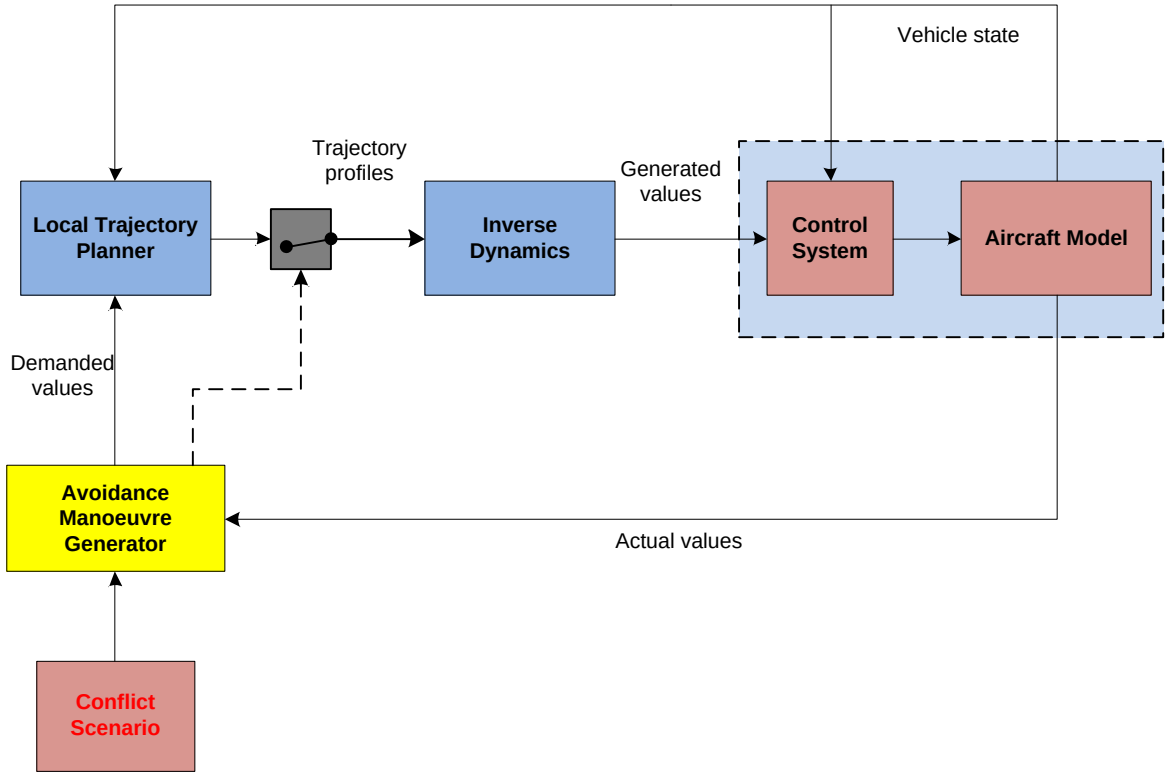

Figure 5.17 Block diagram for CAS simulation

5.5.1 Head-on Scenario Simulation Results

Figure 5.18 shows a head-on conflict scenario where the initial state of the UAV and the intruder are:

1. UAV initial state:

- Speed: $V = 75\ m.s^{-1}$,
- Heading angle: $\psi_0 = 0\ rad$
- The maximum roll angle: $\phi_{max} = \frac{\pi}{3}\ rad$

2. Intruder initial state:

- Speed: $V_b = 50\ m.s^{-1}$
- Heading angle: $\psi_b = -\pi\ rad$
- The range relative to the UAV: $Range = 2500\ m$

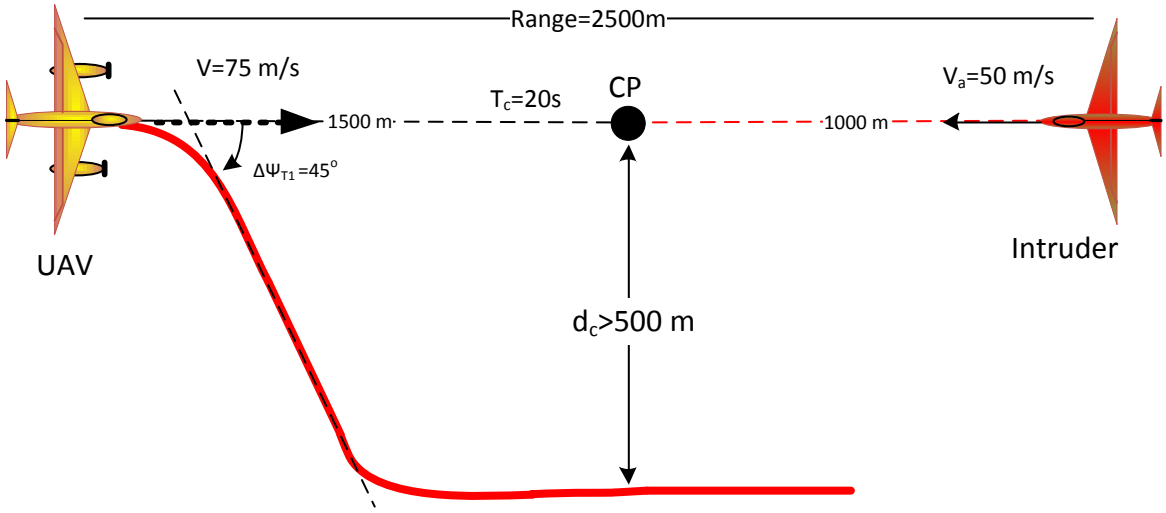

Figure 5.18 Head-on conflict scenario

The UAV and the intruder are at the same level, so the time to collision is $T_c = 20\ s$. The suggested avoidance manoeuvre is average right turn with the following parameters:

- Manoeuvre time $T_m = 40\ s$
- Manoeuvre speed is the same as the initial UAV speed (coordinated turn with constant speed and altitude)
- The clearance distance is $d_c > 500\ m$.
- $\Delta\psi_{T_1} = \frac{\pi}{4}\ rad$

- The maximum heading rate (5.6): $\dot{\psi}_{max} = 12.97\ deg.s^{-1}$

- The heading rate for the average manoeuvre $\dot{\psi}_1 = \frac{\dot{\psi}_{max}}{2} = 6.48\ deg.s^{-1}$

Figure 5.19 shows four subplots for the simulation results of the heading rate, heading angle, roll angle, and flight path angle (from top to bottom):

1. the input (heading rate) to generate the proposed avoidance manoeuvre is shown in the first subplot.

2. The second subplot gives the demanded heading angle ψ_d, which is resulted from the heading rate integration, and the generated heading angle ψ_g which is measured at the UAV (point mass model) output. It can be seen that the generated heading angle is tracking the demanded one with small deviations at the edges due to the curve fitting process and the UAV dynamic delays. However, as can be noticed these differences are negligible.

3. The third subplot shows the demanded roll angle ϕ_d, and the generated one ϕ_g. The relationship between demanded roll angle and the demanded heading rate is given by $\phi_d = \tan^{-1}\frac{V\dot{\psi}_d}{g}$. It can be seen that ϕ_g deviates from ϕ_d, the main reason for this deviations is the UAV dynamics lags. However, the heading angle, UAV speed, and UAV position are the most important quantities that are used in assessing the avoidance manoeuvre performance in this thesis.

4. The fourth subplot in Figure 5.19 shows the demanded and the generated flight path angles, as the manoeuvre is proposed to be in the horizontal plane the demanded flight path angle γ_d is set to be zero. The generated flight path angle γ_g is almost the same as the demanded one with insignificant deviations.

Figure 5.20 gives the simulation result of the speed V, forward speed u, lateral speed v, and vertical speed w. It can be seen clearly that the generated speeds (dashed lines) are tracking the demanded speed (solid lines). The generated UAV speed V_g is approximately 75 $m.s^{-1}$ during the manoeuvre time. Figure 5.22 gives the 3D representation of the manoeuvre which shows that the clearance distance is greater than the specified value that is 500 m.

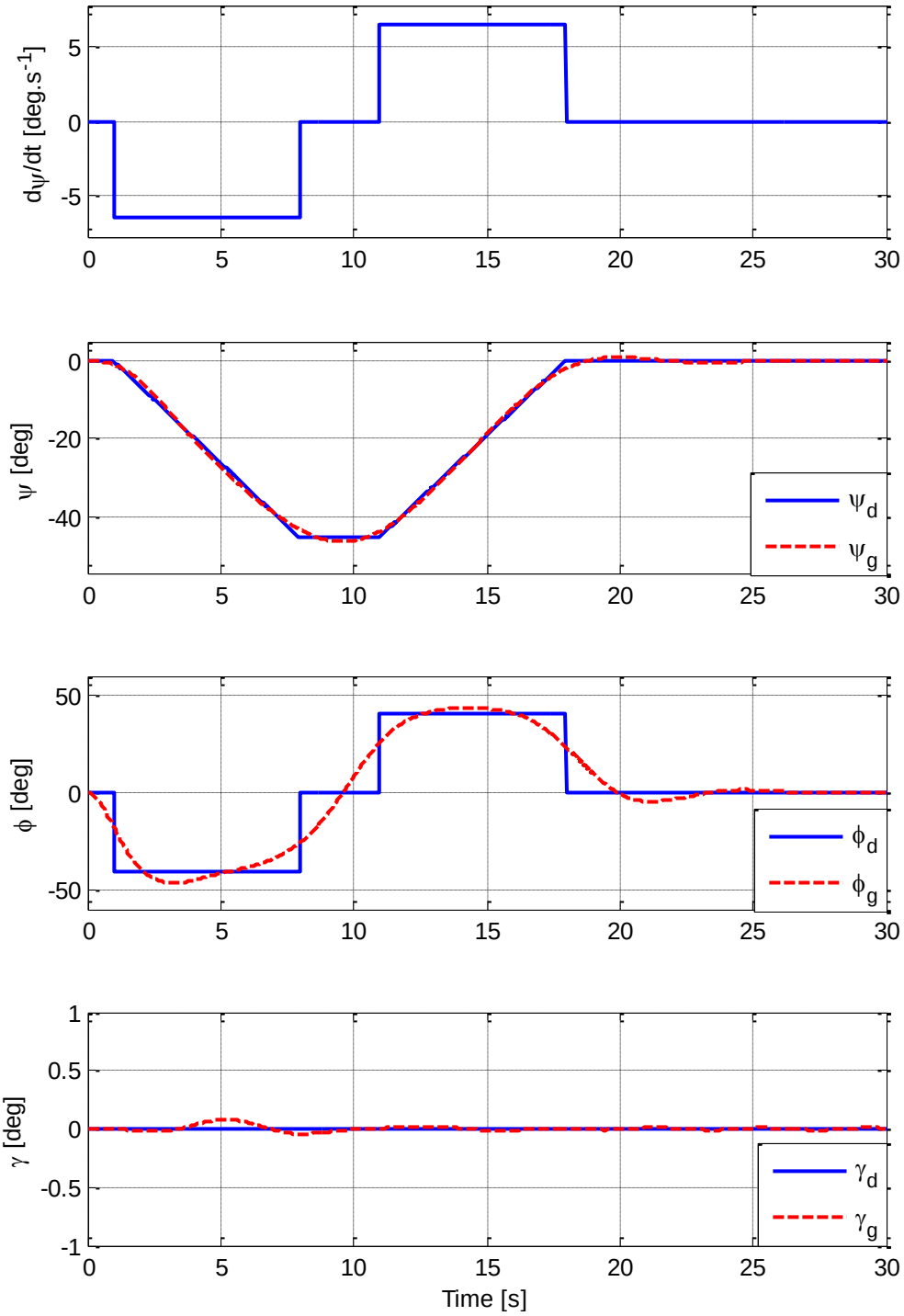

Figure 5.19 Simulation results of attitude (heading rate, heading angle, roll angle, and flight path angle) for the head-on conflict avoidance manoeuvre

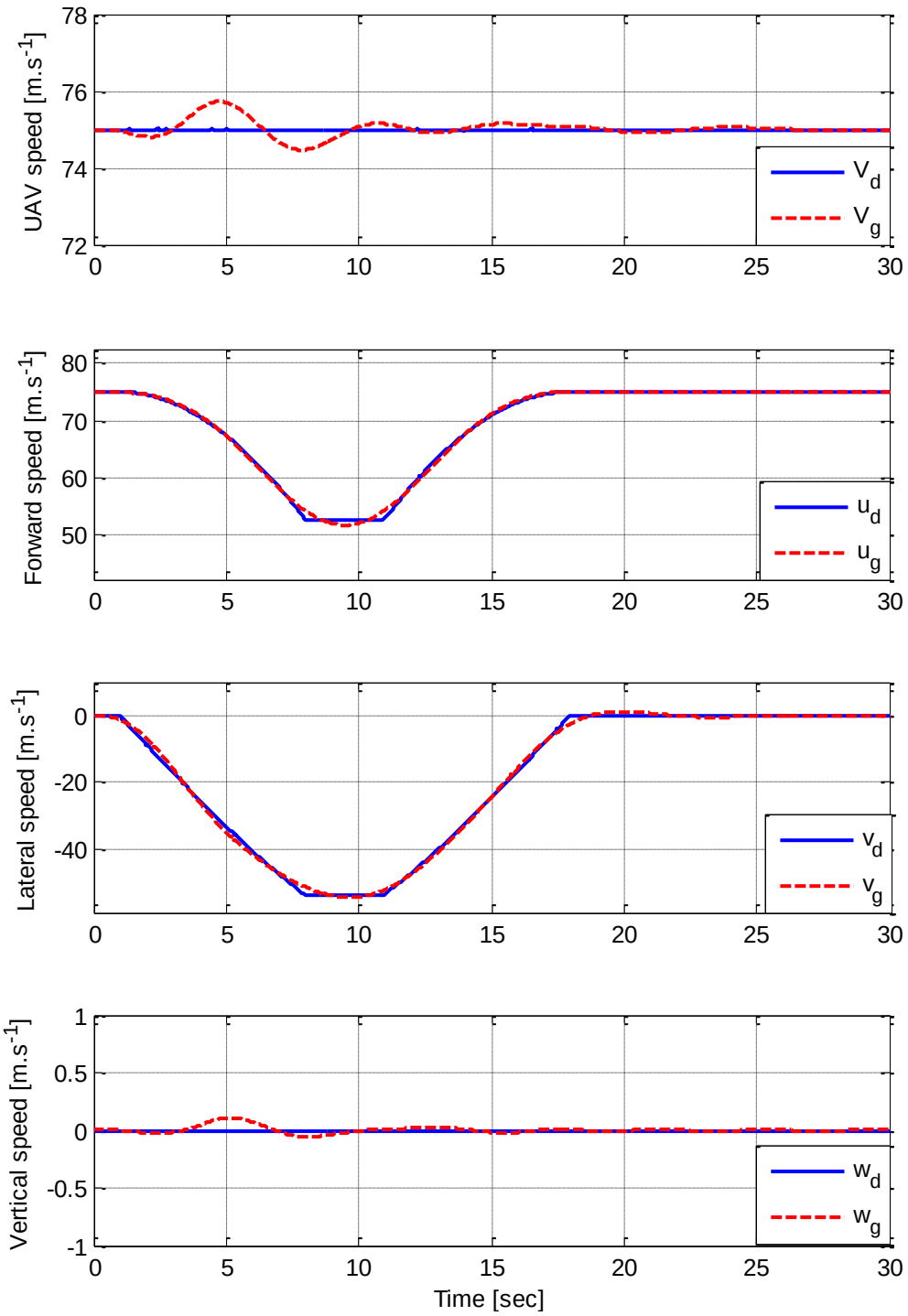

Figure 5.20 Simulation results of speeds (V, u, v , and w) for the head-on conflict avoidance manoeuvre

Figure 5.21 shows the normal load factor and $\frac{1}{\cos(\phi)}$ during the avoidance manoeuvre. It can be seen that the load factor meets the requirements given in CS-23 document [69] and

equals to $\frac{1}{\cos(\phi)}$ hence it satisfies 5.8. This is another factor, in addition to the speed and altitude results, shows that coordinated turn is performed smoothly.

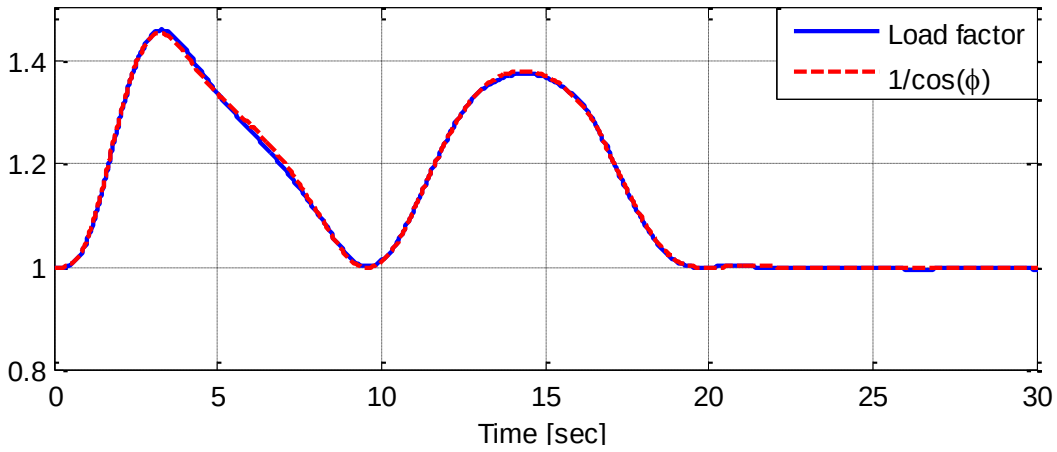

Figure 5.21 Load factor during the avoidance manoeuvre

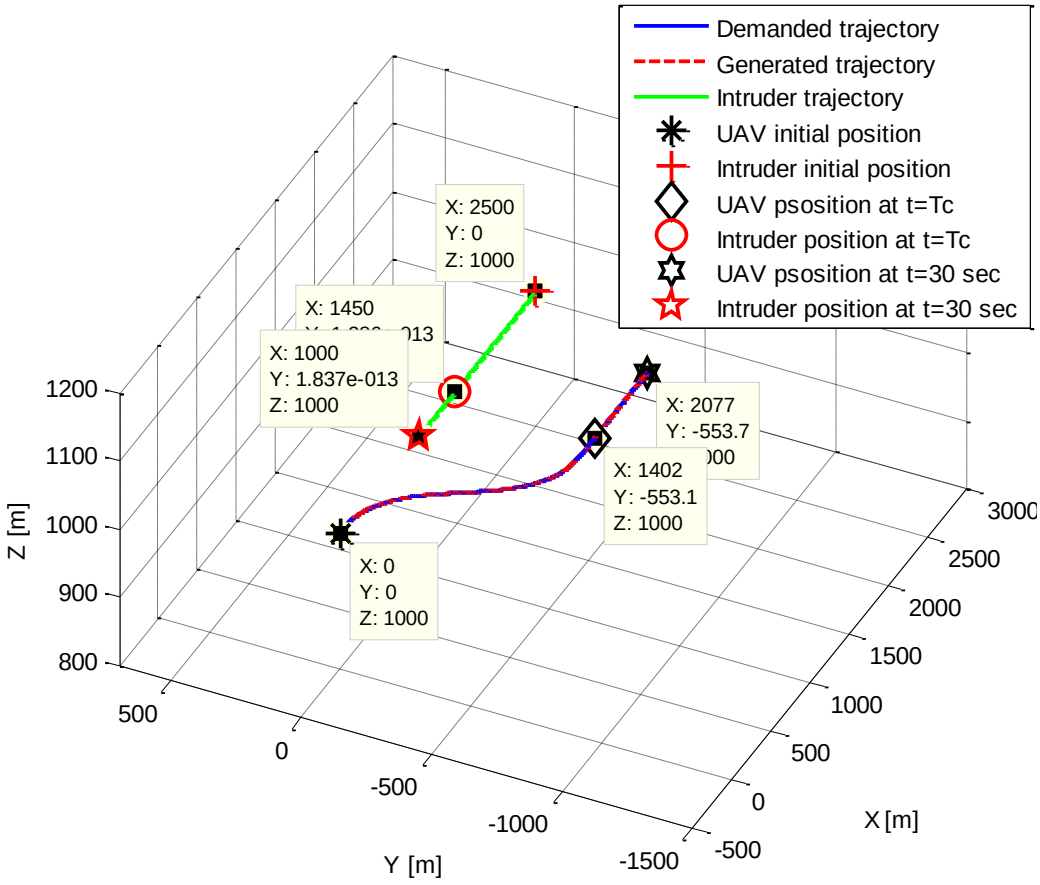

Figure 5.22 3D view of the UAV trajectory for the head-on conflict avoidance manoeuvre

5.5.2 Right Approaching Conflict Scenario Simulation Results (RSL manoeuvre)

Figure 5.23 shows a right approaching scenario used to test the RSL avoidance manoeuvre. The initial state for the UAV and the intruder are:

1. UAV initial state:

   - Speed: $V = 75\ m.s^{-1}$,
   - Heading angle: $\psi_0 = 0\ rad$
   - Maximum roll angle: $\phi_{max} = \frac{\pi}{3}\ rad$

2. Intruder initial state:

   - Speed: $V_b = 75\ m.s^{-1}$
   - Heading angle: $\psi_b = \frac{\pi}{3}\ rad$, that means $\theta_b = \frac{\pi}{6}\ rad$
   - Range relative to the UAV: $R_{y0} = 1500\ m$
   - Time to collision: $T_c = 20\ s$

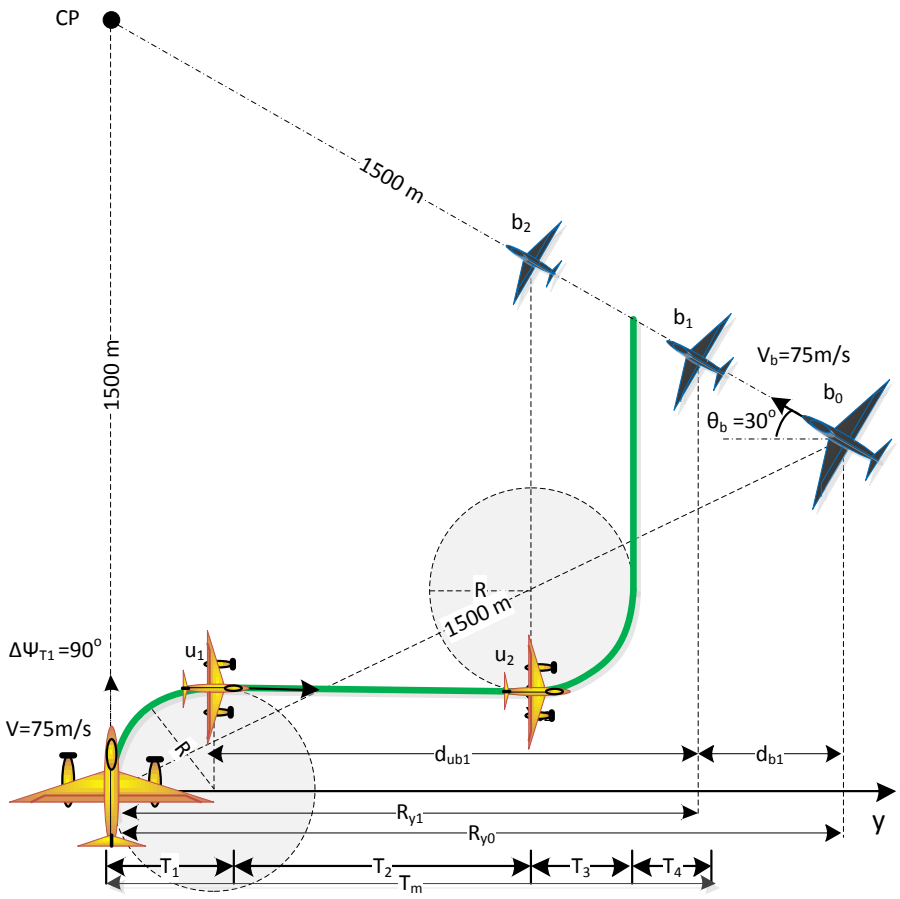

Figure 5.23 Right approaching scenario

The RSL avoidance manoeuvre (exaggerated type) parameters are:

- Selected heading rate: $\dot{\psi}_1 = \dot{\psi}_{max} = 0.2263\ rad.s^{-1}$
- The turn radius: $R = \frac{V}{\dot{\psi}_1} = 331\ m$
- Time period $T_1 = \frac{pi}{2\dot{\psi}_1} = 7\ s$
- $R_{y0} = 1500\cos(\frac{\pi}{6}) = 1300m$, $d_{b1} = V_b\cos(\frac{\pi}{6})T_1 = 455\ m$
- Time period T_2 is calculated using 5.19- 5.21: $T_2 = 4\ s$
- $T_3 = T_1 = 7\ s$, manoeuvre time $T_m = 50\ s$, so $T_4 = T_m - (T_1 + T_2 + T3) = 32\ s$,

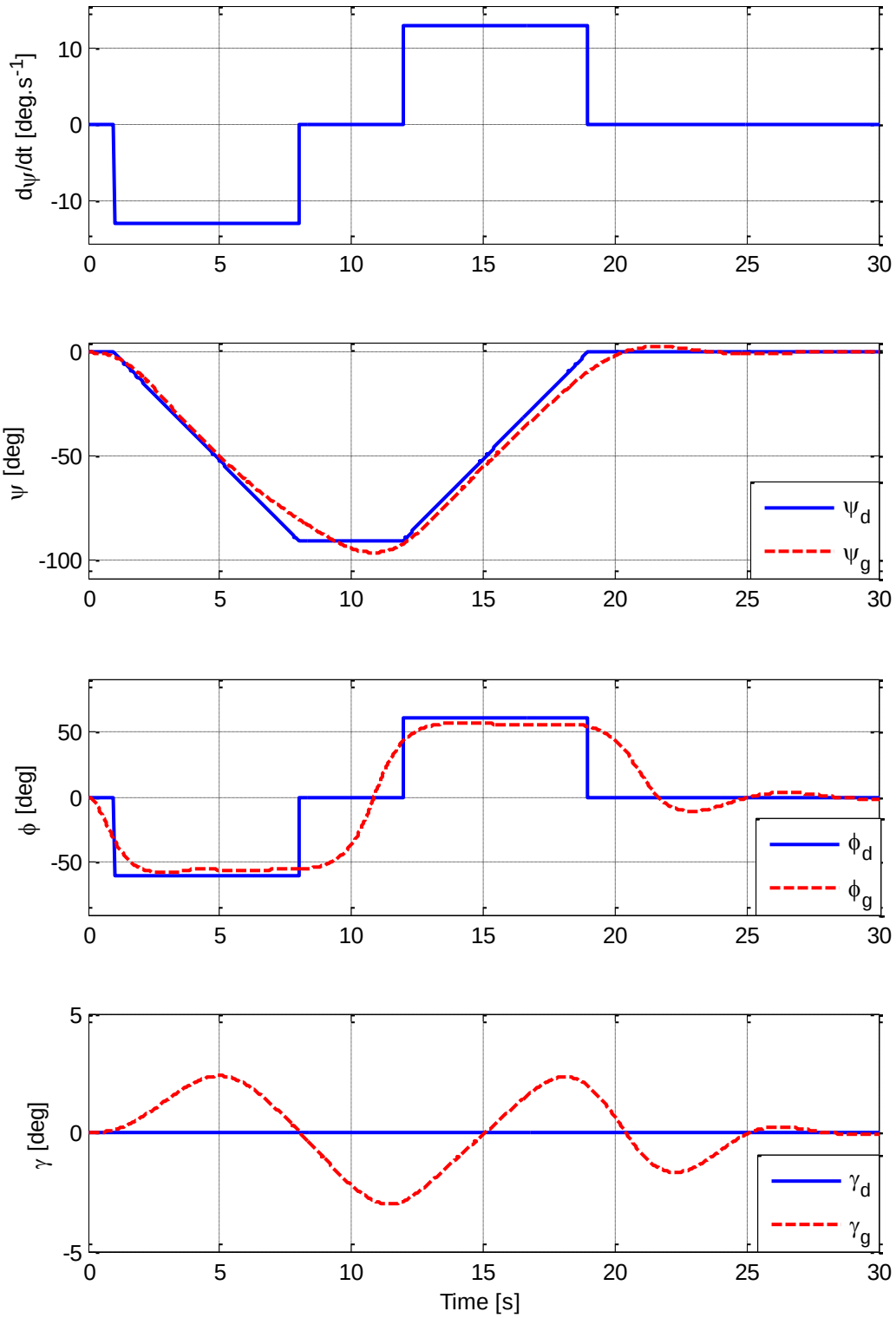

Figure 5.24 Simulation results of attitude (heading rate, heading angle, roll angle, and flight path angle) for the RSL avoidance manoeuvre

Figure 5.24 shows four subplots for simulation results of the heading rate, heading angle, roll angle, and flight path angle (from top to bottom):

1. the input (heading rate) to generate the proposed RSL avoidance manoeuvre is shown in the first subplot. It can be seen that the heading rate is at the maximum value (exaggerated type avoidance manoeuvre).

2. The second subplot shows the demanded heading angle ψ_d, and the generated heading angle ψ_g. It can be seen that the deviations between the generated heading angle and the demanded one are slightly higher than those that have been shown in the head-on collision scenario (previous case) in which the proposed avoidance manoeuvre is average type, where the demanded heading rate is 50% of the maximum heading rate.

3. The third subplot gives the demanded and the generated roll angles ϕ_d, and ϕ_g. The generated roll angle ϕ_g does not exceed the maximum heading angle.

4. The demanded and generated flight path angles are shown in the fourth subplot in Figure 5.24. The avoidance manoeuvre is proposed to be in the horizontal plane so the demanded flight path angle γ_d is set to be zero. The generated flight path angle γ_g fluctuates around zero within the range $\pm 3\ deg$, which is small enough to be negligible.

Figure 5.25 gives the simulation result of the speed V, forward speed u, lateral speed v, and vertical speed w. It can be seen clearly that the generated speeds (dashed lines) are tracking the demanded speed (solid lines). The generated UAV speed V_g is approximately $75 m.s^{-1}$ during the manoeuvre time. Figure 5.27 gives the 3D representation of the manoeuvre which shows that the clearance distance is greater than the specified value that is $500m$.

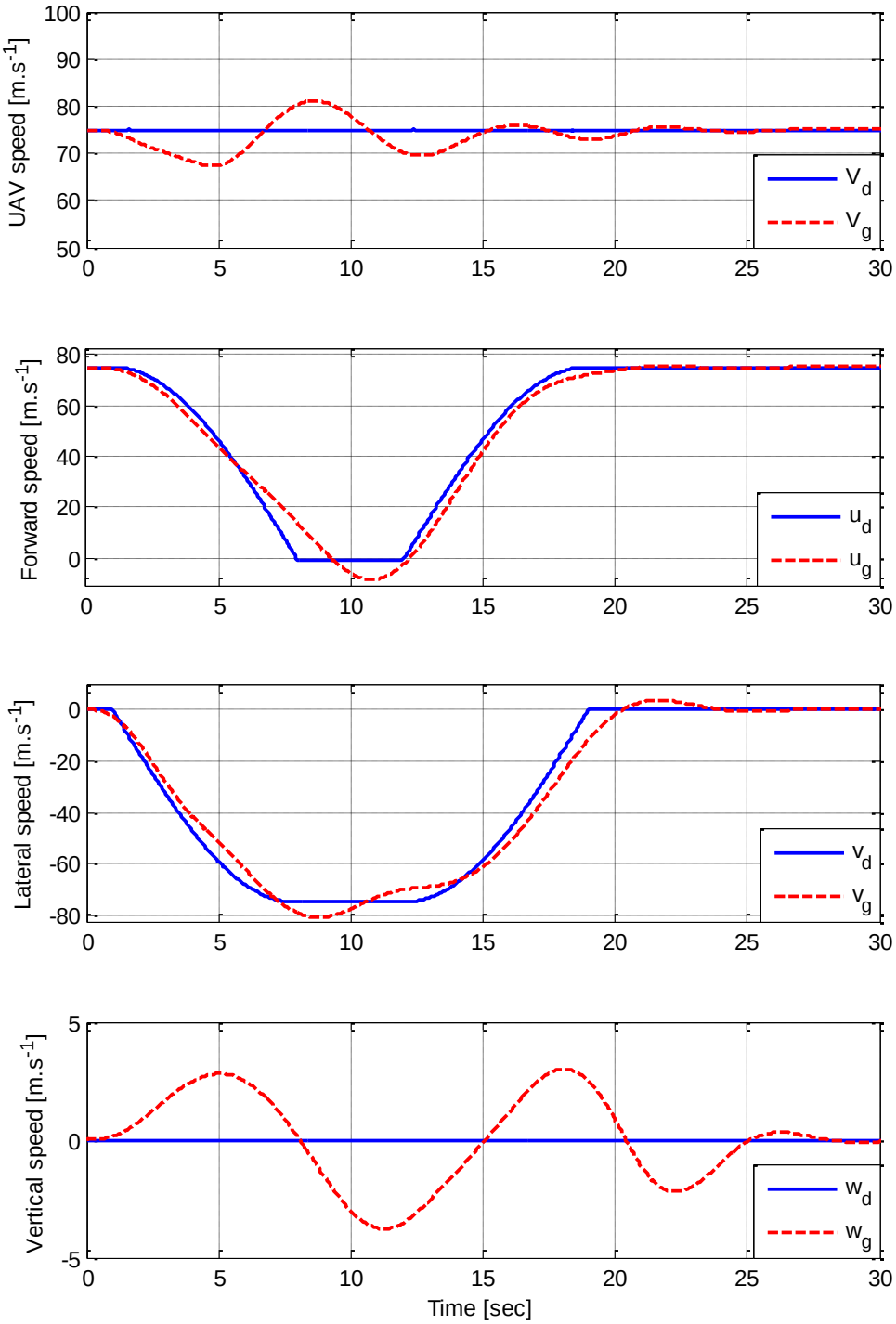

Figure 5.25 Simulation results of speeds (V, u, v , and w) for the RSL avoidance manoeuvre

Figure 5.26 shows the normal load factor and $\frac{1}{\cos(\phi)}$ simulation results. Unlike the head-on conflict scenario, that is discussed previously, there is mismatch between n and $\frac{1}{\cos(\phi)}$ due to the higher value of the heading rate (command) of the exaggerated type avoidance manoeuvre. However, the normal load factor still meet the criteria that given in CS-23 [69].

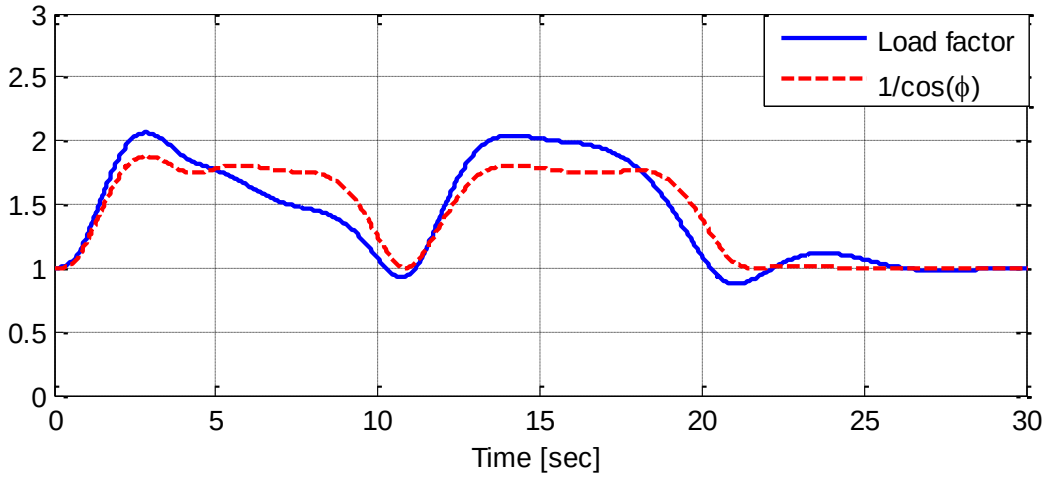

Figure 5.26 Load factor during the avoidance manoeuvre

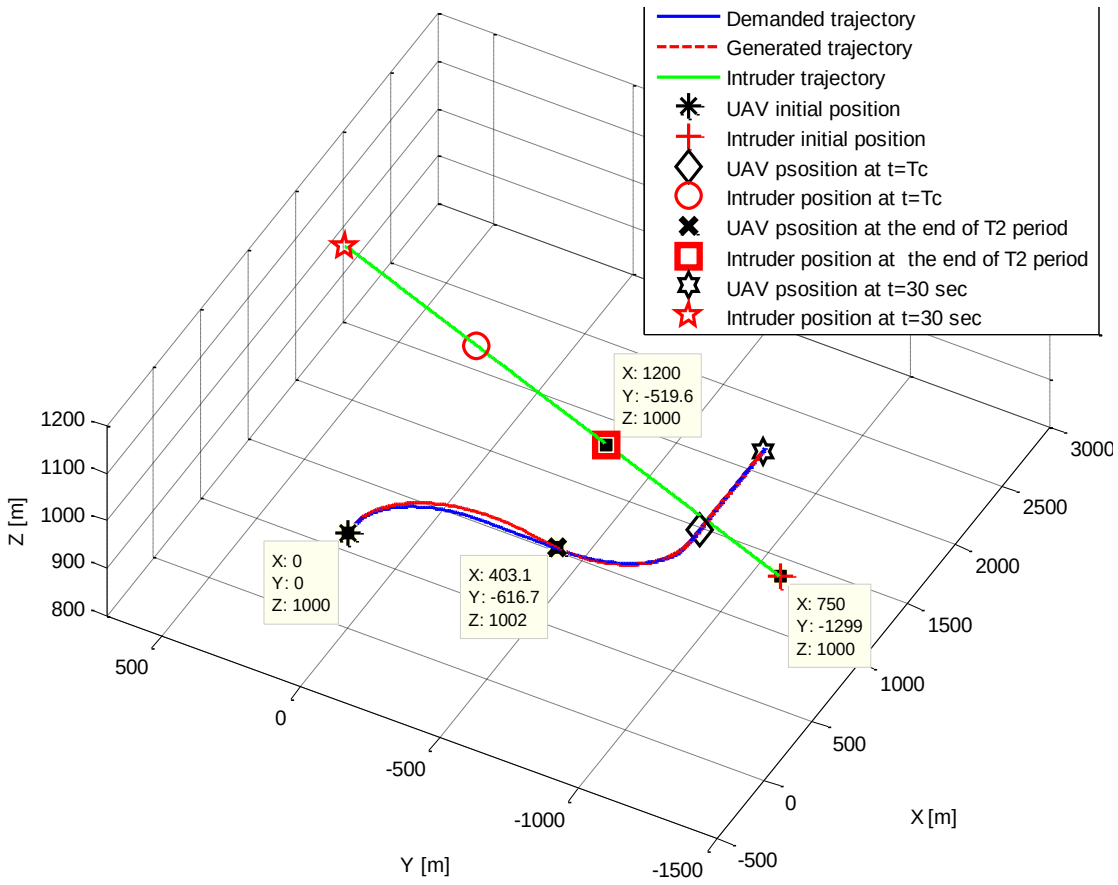

Figure 5.27 3D view of the UAV trajectory for the RSL avoidance manoeuvre

5.5.3 Right Approaching conflict Scenario Simulation Results (Circle Manoeuvre)

Figure 5.28 shows a right approaching scenario used to test the circle avoidance manoeuvre. The initial state for the UAV and the intruder are:

1. UAV initial state:
    - Speed: $V = 60\ m.s^{-1}$,
    - Heading angle: $\psi_0 = 0\ rad$
    - Maximum roll angle: $\phi_{max} = \frac{\pi}{3}\ rad$
2. Intruder initial state:
    - Speed: $V_b = 75\ m.s^{-1}$
    - Heading angle: $\psi_c = 0.89\ rad$, that means $\theta_c = 0.68\ rad$
    - Range relative to the UAV: $Range = 1500\ m$
    - Time to collision: $T_c = 25\ s$

The circle avoidance manoeuvre (exaggerated type) parameters are:

- Selected heading rate: $\dot{\psi}_1 = 0.8\dot{\psi}_{max} = 0.2263\ rad.s^{-1}$
- Turn radius: $R = \frac{V}{\dot{\psi}_1} = 265\ m$
- Time period $T_t = \frac{2\pi}{\dot{\psi}_1} = 27.76\ s$
- $R_{x0} = -312\ m$, $d_{c1} = V_c \cos(\theta_c)\frac{T_t}{4} = 455\ m$
- $T_4 = T_m - (T_t) = 11.24\ s$,

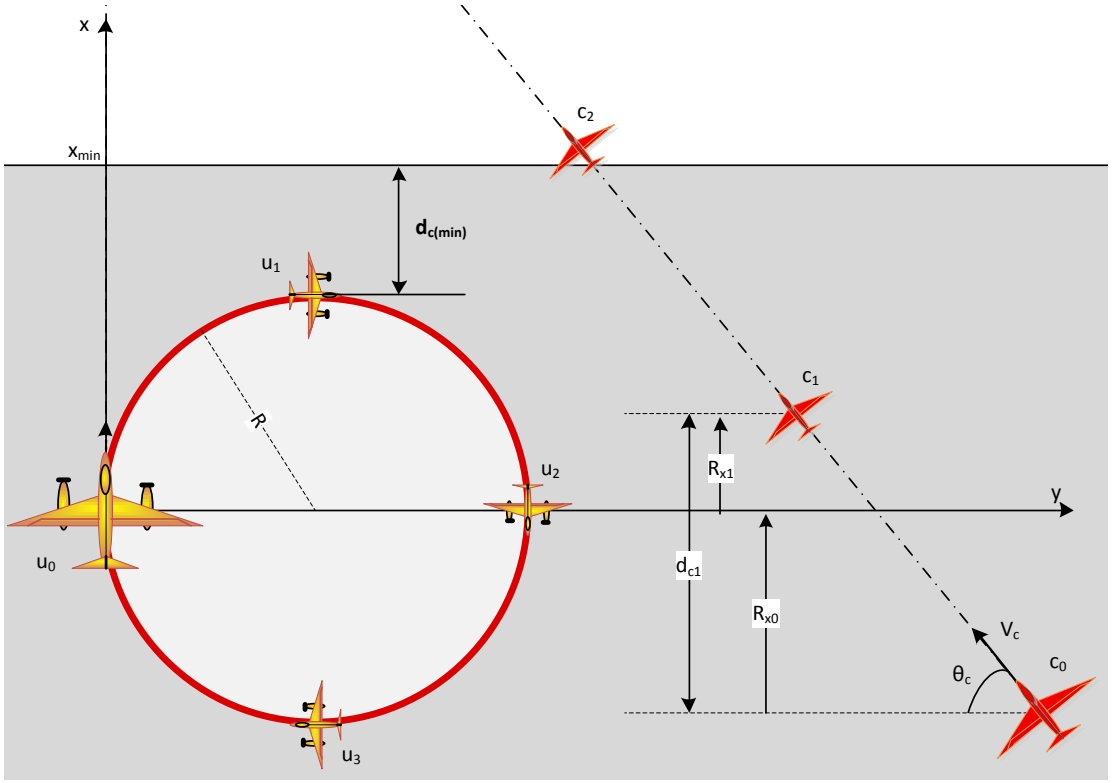

Figure 5.28 The circle avoidance manoeuvre for a right approaching conflict scenario

Figure 5.29 shows the simulation results of the heading rate, heading angle, roll angle, and flight path angle. It can be seen that the generated signals are tracking the demanded signals with acceptable fluctuations. For example, the generated heading angle tracks the demanded one with very small deviations, also the generated flight path angle fluctuates around zero with maximum error less than 2°. However, the UAV dynamics lags can be obviously noticed in the roll angle response in the third subplot in Figure 5.29. Figure 5.30 gives the simulation result of the UAV speed V, forward speed u, lateral speed v, and vertical speed w. It can be seen clearly that the generated speeds (dashed lines) are tracking the demanded speeds (solid lines). The generated UAV speed V_g is approximately $60m.s^{-1}$ during the manoeuvre time. Figure 5.32 gives the 3D representation of the UAV avoidance manoeuvre trajectory, and the intruder trajectory, it also shows the position of the UAV and the intruder at ($t = \frac{T_t}{4}$, $t = \frac{T_t}{2}$, and $t = T_m$). It can be seen that when $t = \frac{T_t}{4}$ the UAV at position is (333, -213,1003), and the intruder position is (154,-1090,1000) so the intruder is at the right of the UAV, this is the reason for using the circle avoidance manoeuvre instead of the RSL/RS-LS avoidance manoeuvre.

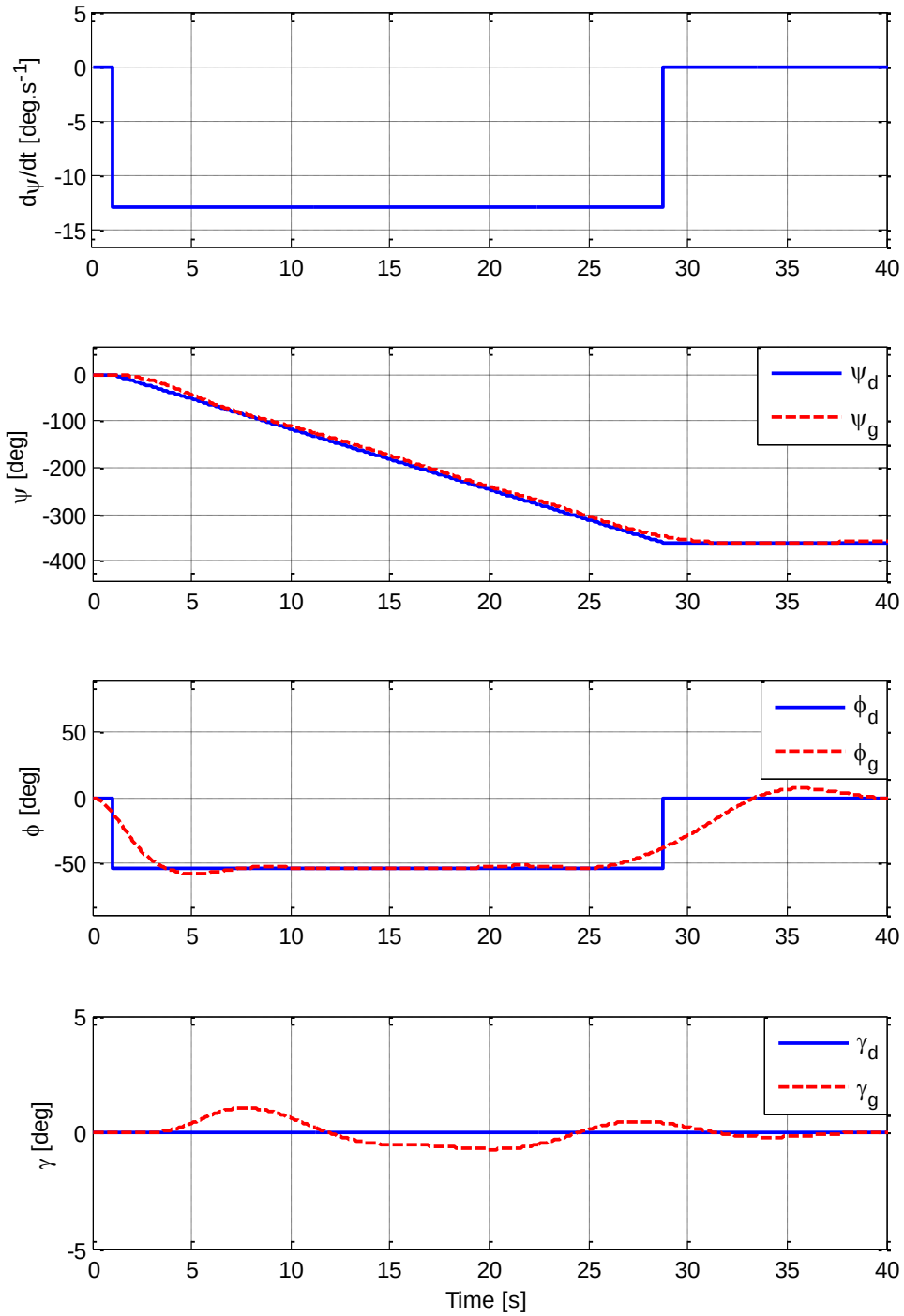

Figure 5.29 Simulation results of attitude (heading rate, heading angle, roll angle, and flight path angle) for the circle avoidance manoeuvre

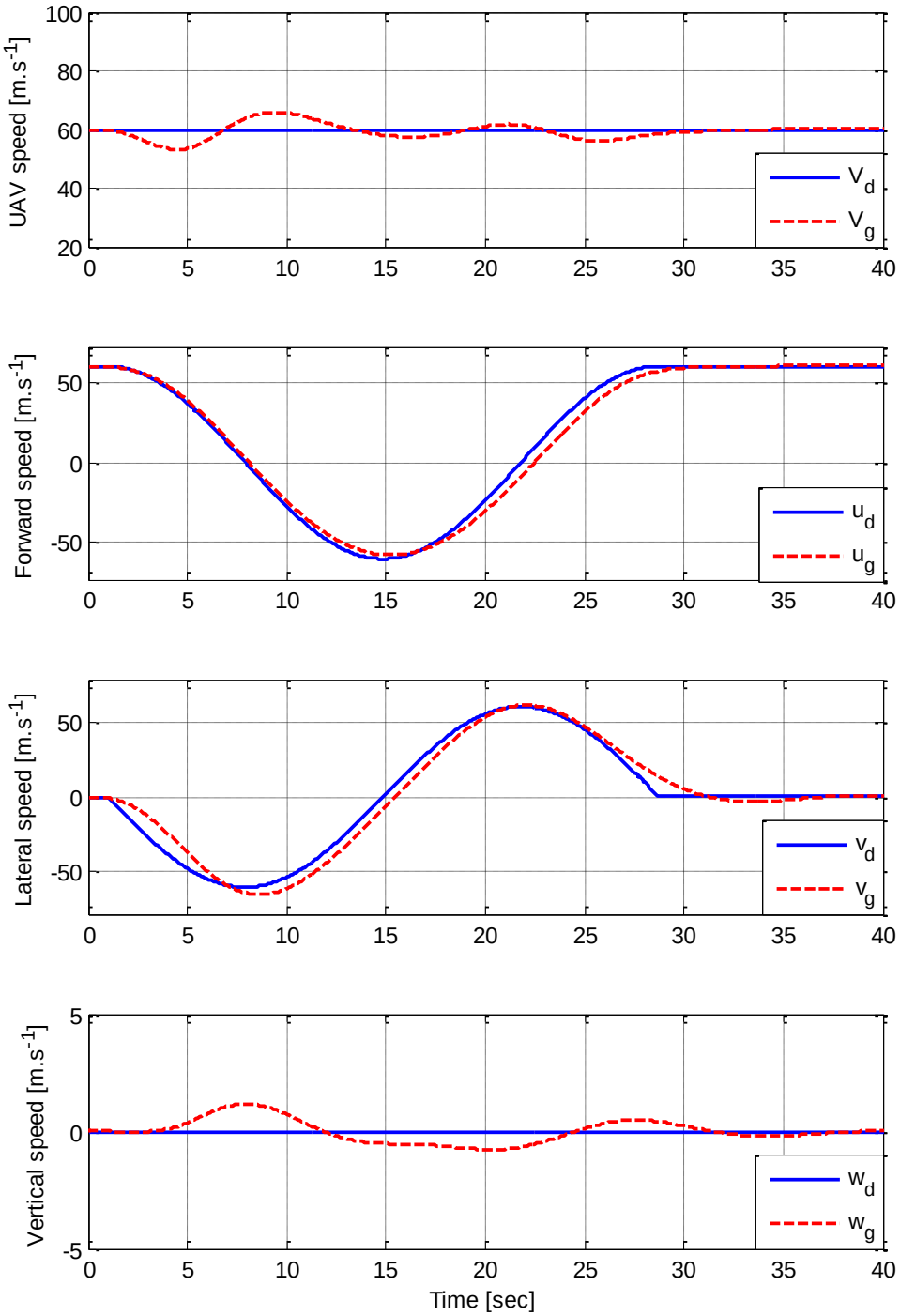

Figure 5.30 Simulation results of speeds (V, u, v , and w) for the circle avoidance manoeuvre

Figure 5.31 shows the normal load factor and $\frac{1}{\cos(\phi)}$ during the circle avoidance manoeuvre. The mismatch between the normal load factor and $\frac{1}{\cos(\phi)}$ is very small. The lower mismatch in this case comparing to the mismatch in the previous scenario (both are exaggerated type avoidance manoeuvre) is due to the lower speed of the UAV ($V = 60\ m.s^{-1}$)

than the speed in the previous case ($V = 75\ m.s^{-1}$).

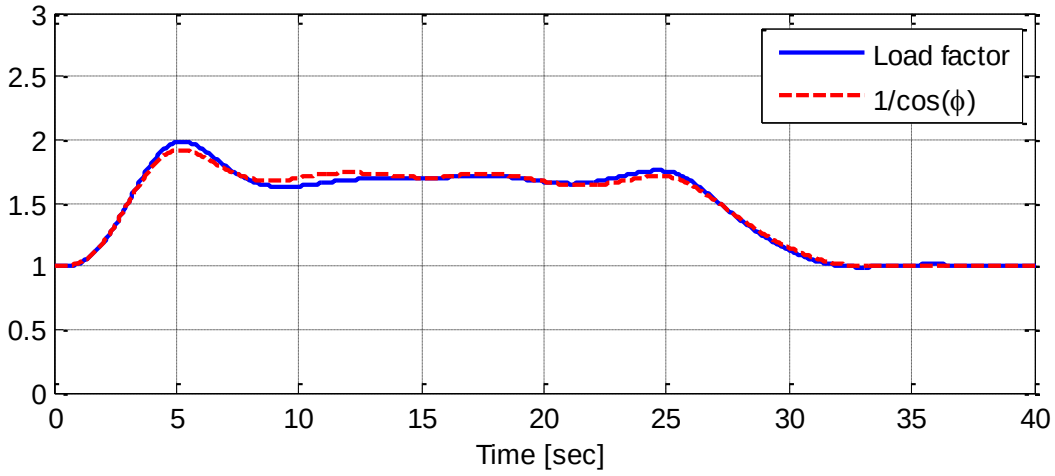

Figure 5.31 Load factor during the avoidance manoeuvre

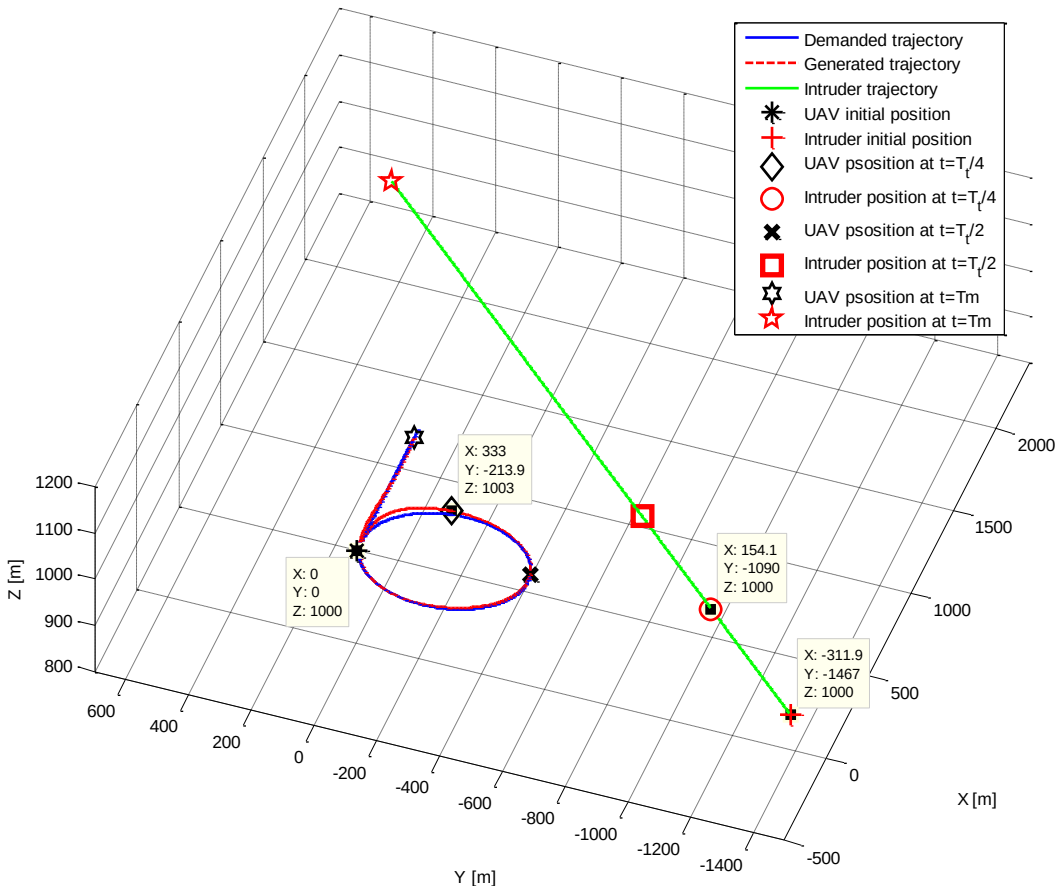

Figure 5.32 3D view of the UAV trajectory for the circle avoidance manoeuvre

5.6 Summary

This chapter discusses the avoidance manoeuvres generating process for different conflict scenarios in which the UAV should change direction (right/left turn) in horizontal plane. Two conflict scenarios are discussed, namely head-on, and right approaching. The rules of the air give general instructions for avoiding different conflict scenarios, but there are no specific procedures for performing the avoidance manoeuvres. Hence, the proposed avoidance manoeuvres that are presented in this chapter are selected based on a pilot's suggestions. A geometric approach is used to parameterize the proposed avoidance manoeuvres.

Chapter 6

Pilot Behaviour-Based Collision Avoidance Manoeuvre Generation

6.1 Introduction

This chapter shows how pilot experience can be used to develop a collision avoidance manoeuvre generation process for a UAV. Although the avoidance manoeuvres proposed in Chapter 5 are intended to be similar to the manoeuvres that are generated by pilots in manned aircraft, they are predetermined manoeuvres. For example, at a specific value of the UAV speed and bank angle the avoidance manoeuvres will be the same for different values of the time to collision. This is not the case in the manned aircraft, where the reaction of the pilot differs from one scenario to another. Consequently the produced avoidance manoeuvres are not predetermined. Hence, a further development is proposed to link the avoidance manoeuvre's parameters with the conflict scenario's parameters. Thus, the generated avoidance manoeuvre changes dynamically based on the conflict scenario in a similar manner that the manned aircraft does. As a result, the UAV behaviour during the conflict becomes more similar to the manned aircraft behaviour.

The proposal in this chapter is to augment the pilot's experiences to the avoidance manoeuvre generating process. In a manned aircraft the pilot uses his/her experience to initiate the suitable manoeuvre to avoid the conflict. Fuzzy logic technique is proposed to model the pilot actions in generating the avoidance manoeuvre for different conflict scenarios.

In this thesis the protocol for achieving the turn for the avoidance manoeuvre is obtained from experience of a single pilot (See Appendix B) . The pilot in collision avoidance case

initiates the turn by considering the time to collision T_c and the aircraft speed (V) to infer the amount of roll rate $\dot{\phi}$ to bank the aircraft at a predetermined bank angle ϕ, and change the aircraft heading angle by a predetermined value $\Delta\psi$. The pilot experience is used to generate the roll rate by using the fuzzy logic technique. Different head-on conflict scenarios are used to test the proposed algorithm by using MATLAB/Simulink. Simulation results are included and discussed.

It is important to mention that the pilot behaviour information during the collision avoidance, which is presented in this chapter, is based on a single pilot experience. Extracting the pilot experience is achieved by extended interviews and a problem discussion with a pilot at the National Flying Laboratory Centre, Cranfield University. Pilot behaviour differs from pilot to another according their experience, type of the aircraft, and the environment that they are flying in. A further development for this method can be achieved by involving more pilots with different experiences.

6.2 Fuzzy Logic Controller Structure

Since it was introduced by Zadeh in 1965, Fuzzy Logic (FL) has become a successful technique for developing complex control systems [78]. Fuzzy logic techniques have been used in a practical control systems successfully since the early 1970s, when the British engineer Ebrahim Mamdani developed an automated control system for a steam engine using the expertise of a human operator [79]. Commenting on his work, Mamdani wrote [80]:

> *The essence of this work is simply that if an experienced operator can provide the protocol for achieving such a control in qualitative linguistic terms, then fuzzy logic can be used to successfully implement this strategy.*

It is appropriate to use fuzzy logic when [81]:

- The system has one or more continuous control variables,
- The mathematical model of the process does not exist, or it is very difficult to encode, or is very complex to be evaluated fast enough for real time operation, or involves too much memory on the designated chip architecture.
- High ambient noise levels must be dealt with or it is important to use inexpensive sensors and/or a low precision microcontroller.

- When an expert is available who can specify the rules underlying the system behavior and the fuzzy sets that represent the characteristics of each variable.

Figure 6.1 shows a simple structure of Mamdani type fuzzy logic controller (FLC) which contains four main parts [81]:

1. Fuzzifier: Transforms the numerical values of the measured input signals into fuzzy quantities (linguistic variables) by using membership functions. This transformation is called the fuzzification. A membership function has a value between 0 and 1, and it indicates the degree of belongingness of a quantity to a fuzzy set.

2. Knowledge base: Consists of the linguistic-control rule base and the data base. The data base provides the information which is used to define the linguistic control rules and the fuzzy data manipulation in the fuzzy-logic controller. The rule base (expert rules) specifies the control goal actions by means of a set of linguistic control rules. In other words, the rule base contains rules which are provided by an expert. The FLC looks at the input signals and by using the expert rules determines the appropriate output signals (control actions). The rule base contains a set of (*if-then*) rules. There are many methods for developing the rule base, such as [81, 82]:

    - Using the knowledge of a person who is an expert in the application.
    - Modelling the process.
    - Modelling the control action of the operator.
    - Using an artificial neural network.
    - Using self organized fuzzy controllers.

3. Inference engine: The role of the inference engine starts when the inputs signals are transformed into their linguistic variables. At this point the inference engine evaluates the if-then rules and gives the results as a fuzzy value for a linguistic variable. In other words, the inference engine simulates human decision making by using fuzzy logic rules and fuzzy implication. The output of the inference engine which is a fuzzy value is then transformed into a real output value of the FLC and this is the function of the defuzzifier.

4. Defuzzifier: Transforms the fuzzy values into real quantities and this action is called defuzzyfication. In order to perform its work the defuzzifier uses the membership functions in the reverse way to the fuzzifier.

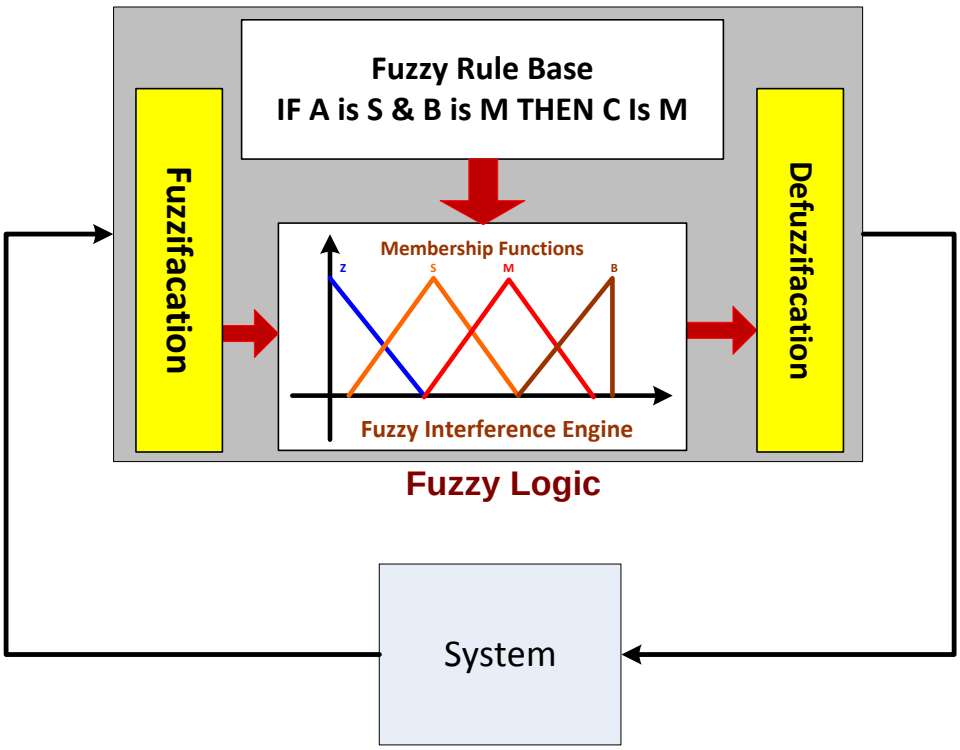

Figure 6.1 Mamdani fuzzy logic system

6.3 Pilot Behaviour During the Collision Avoidance

Understanding how a pilot performs a turn manoeuvre is the first step in defining the fuzzy logic inputs/outputs and producing the fuzzy logic rules. In horizontal avoidance manoeuvres, pilots perform level coordinated turns to avoid conflicts. The level coordinated turn is made by banking the wings in the direction of the desired turn. A specific bank angle is selected by the pilot, so control pressures applied to achieve the desired bank angle. Then appropriate control pressures exerted to maintain the desired bank angle once it is established, the aircraft speed, and altitude. The level coordinated turns are divided into three classes according to the selected bank angle [83]:

1. Shallow turns: In which the bank angle is less than approximately 20°. In a shallow turn the inherent lateral stability of the airplane is acting to level the wings unless some aileron is applied to maintain the bank angle.

2. Medium turns: resulting from a degree of bank (approximately 20° to 45°) at which the airplane remains at a constant bank angle.

3. Steep turns: The turns that result from a degree of bank (45° or more).

The Airplane Flying Handbook [83] (published by FAA) gives more details about the level turns characteristics.

6.3.1 Performing the Coordinated Level Turns

Steep turns are generally flown at 45° bank angles, in both directions, and to a specific heading, and the altitude must be maintained within 100 feet. The following steps give a general summary of the procedure that is used by the pilot to achieve a level steep turn [83]:

1. Roll into the turn: The bank angle is changed from zero towards 45°.

2. When the bank angle passes through 30° of bank, the back elevator pressure should be smoothly increased to maintain the altitude.

3. As the turn continues, increased elevator back pressure will be needed in order to maintain the altitude. Additional back-elevator pressure increases the angle of attack, which results in an increase in drag. Consequently, power must be increased to maintain the entry altitude and airspeed.

4. The roll out from the turn should be timed so that the wings reach level flight when the airplane is exactly on the predetermined heading. While the recovery is being made, back-elevator pressure is gradually released and power reduced, as necessary, to maintain the altitude and airspeed. When beginning to roll out of the turn to the predetermined heading, lead the roll out by one-half the number of degrees of the bank angle. For example, in a 30° bank turn, begin to level the wings 15° degrees before reaching the desired heading.

5. Lower the nose, as the back pressure required to maintain level flight during the turn will cause the airplane to climb once the turn has been completed.

It can be noticed that at a specific speed and altitude values, there are three parameters that control the turn: the bank angle, the roll rate, and the heading angle. [1]. The next sections show how the bank angle and the roll rate are calculated based on the pilot experience, and used in collision avoidance manoeuvre generation process. Hence, the UAV avoidance manoeuvre can mimic a manned aircraft during the conflict.

[1] In this thesis the selected heading angles for the conflict scenarios are discussed in Chapter 5

6.3.2 Avoidance Manoeuvre Bank Angle Selection

In the collision avoidance manoeuvres pilots usually avoid the collision by using medium, or steep coordinated level turns. For example, one of the bank angles 30°, 45°, or 60° is selected to perform the turn part of the collision avoidance manoeuvre. The selected bank angle depends on the conflict scenario:

- 30° bank angle is used for conflict scenarios where:
  - The speed of the aircraft V is very low, near the minimum speed, so a higher bank angle at the low speed may cause stall (see Section 5.2.1), and
  - the time to collision is very large ($T_c > 35s$).
- Banking by 60°, or the maximum bank angle, which is limited by the maximum normal load factor as given by (5.9), is used in critical conflict scenarios where:
  - The speed of the aircraft V is very high, and
  - the time to collision is very short ($T_c < 12s$).
- Generally speaking a bank angle of 45° is the pilot's preferred bank angle in most of conflict scenarios.

The pilot selects a specific value of the bank angle and often uses that value for most of conflict scenarios. It is easier for the pilot if they can achieve the coordinated level turn at a bank angle they are familiar with rather than banking at different angles. However, this differs from pilot to pilot based on their training and experience. In this chapter a bank angle of 45° is considered for all scenarios.

6.3.3 Avoidance Manoeuvre Roll Rate Calculation

To achieve the roll angle, range of a roll rates may be applied. For example, the pilot will try to change the bank angle from zero to 45° rapidly (big roll rate) if the time to collision is within the short range ($T_c < 20s$) and the aircraft speed is in medium/high ranges. Bank angle is changed slowly (small roll rate) if the time to collision belongs to a large range ($T_c > 20s$) and the aircraft speed belongs to slow/medium ranges. The fuzzy logic technique is used to represent the pilot behaviour in generating the roll rate for the turn part of the avoidance manoeuvre. The fuzzy inputs are the time to collision T_c, and the UAV speed

V. The output is the roll rate which is sent to the avoidance manoeuvre generation process to generate the avoidance manoeuvre trajectory profiles. Figures 6.2, and 6.3 show the membership functions of the time to collision T_c and the UAV speed V respectively. The time and the speed ranges are selected based on pilot experience (which may differ from one pilot to another):

- S denotes time to collision membership function Small.
- M denotes time to collision membership function Medium.
- B denotes time to collision membership function Big.

The same notations are used to represent the UAV speed membership functions.

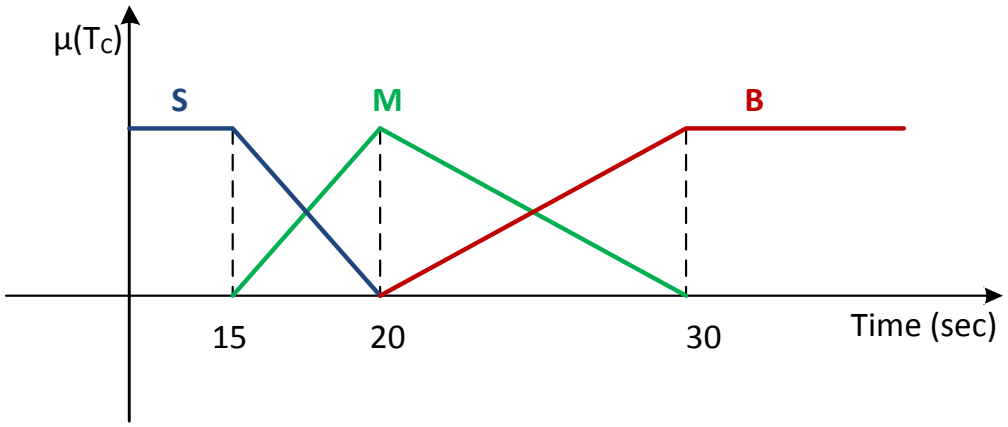

Figure 6.2 Time to collide membership functions

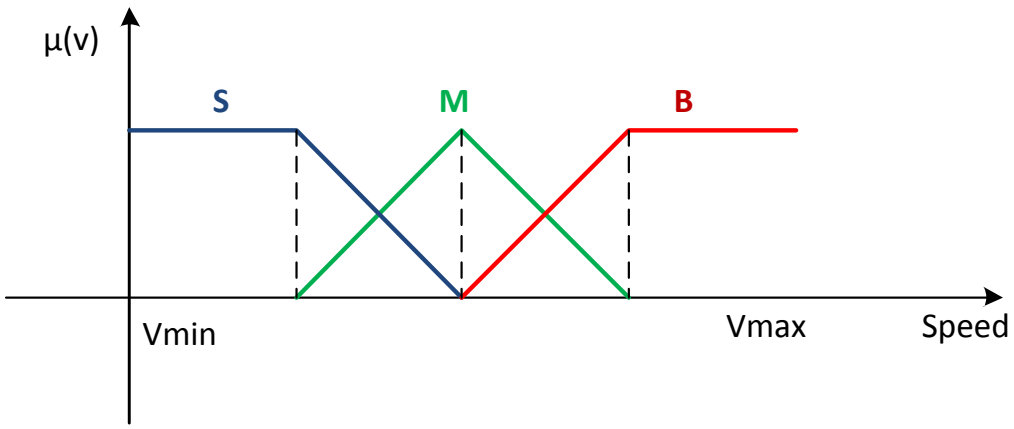

Figure 6.3 Speed membership functions

Figure 6.4 shows the membership functions of the roll rate (the output of the fuzzy system), where:

- S denotes roll rate membership function Small
- M denotes roll rate membership function Medium
- B denotes roll rate membership function Big
- VB denotes roll rate membership function Very Big

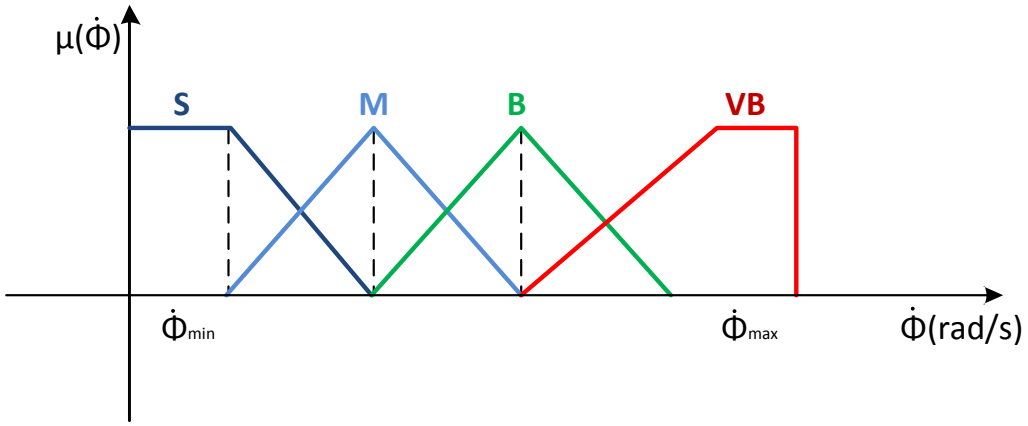

Figure 6.4 Bank Angle (output) membership functions

The maximum allowable value for the roll rate depends on the minimum allowable time to achieve a certain bank angle. The minimum allowable time to achieve a specific roll angle depends on the aircraft type and the speed. However, for small aircraft this time is as short as one second for 90° degree roll [65]. Table 6.1 shows the possible rules for the decision making that are established by using if-then statements based on the pilot's experience. The rules map T_c and V to roll rate $\dot{\phi}$. For example, the first rule in the table is(**IF** T_c is Small **AND** V is Small **THEN** $\dot{\phi}$ is Very Big).

Table 6.1 Fuzzy inference engine (the aircraft starts manoeuvreing first)

V \ T_c	Small	Medium	Big
Small	VB	M	S
Medium	VB	B	M
Big	VB	VB	B

It is worth mentioning here that in some conflict scenarios, where both the aircraft and the intruder are supposed to perform avoidance manoeuvres, the pilot's reaction depends also on the intruder's behaviour. For example, in head-on conflict scenarios, where both the aircraft and the intruder should change their directions (right/left) to avoid the collision, the pilot's reaction, when the intruder starts the avoidance manoeuvre before the pilot does, is different from the pilot reaction when he/she starts the manoeuvre before the intruder. Table 6.2 shows the possible rules for the decision making when the intruder performs the avoidance manoeuvre before the UAV, where 'X' denotes to hold the current states (i.e. without changing the speed, direction, and altitude). However, the behaviour of the intruder is not considered in this thesis.

Table 6.2 Fuzzy inference engine (the intruder starts manoeuvreing first)

V \ T_c	Small	Medium	Big
Small	VB	S	X
Medium	VB	M	X
Big	VB	B	X

6.4 Avoidance Manoeuvre Parameterisation

The pilot's reaction to avoid a collision is to change the aircraft direction according to the rules of the air, by changing the aircraft heading to a specific heading through a level coordinated turn. However, as discussed in the previous section, although the heading and bank angles are predetermined the amount of roll rate that is demanded to achieve the bank angle depends on the time to collision and the UAV speed. The fuzzy logic algorithm generates the roll rate based on the pilot reaction to the conflict scenario. Hence, avoidance manoeuvre trajectory profiles can be produced by using the generated roll rate, and the predefined values of bank angle and heading angle. This section shows how the roll rate is used in the avoidance manoeuvre trajectory profiles calculation. Figure 6.5 shows roll rate that is used to change the UAV bank angle from the level wing state ($\phi_o = 0$) to a predetermined bank angle ϕ_1 at a specific roll rate $\dot{\phi}_1$ during the time period T_{r1}. Then keep banking at the constant bank angle ϕ_1 for a duration of time T_{r2}, finally leveling the wing over the interval T_{r3}.

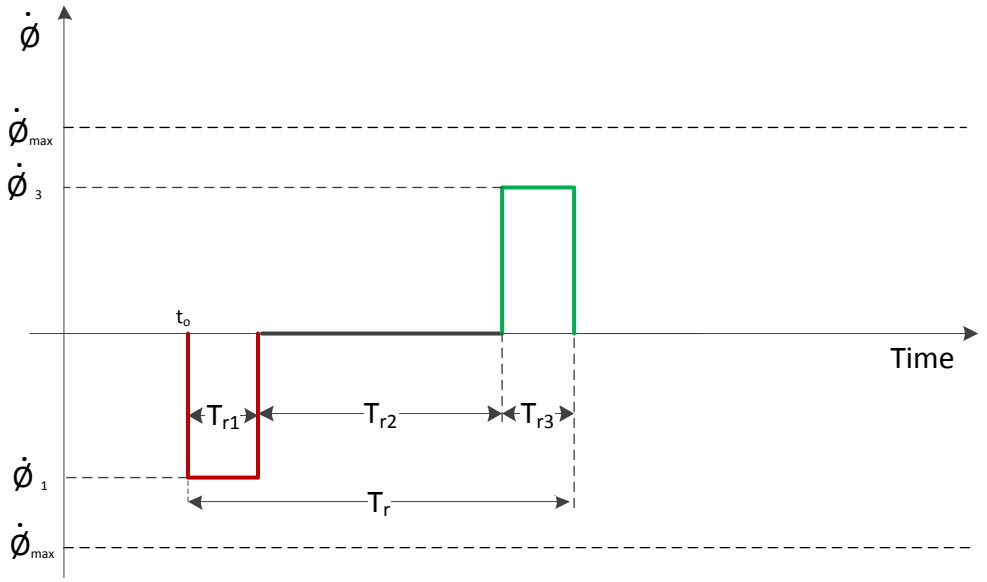

Figure 6.5 Roll rate $\dot{\phi}$

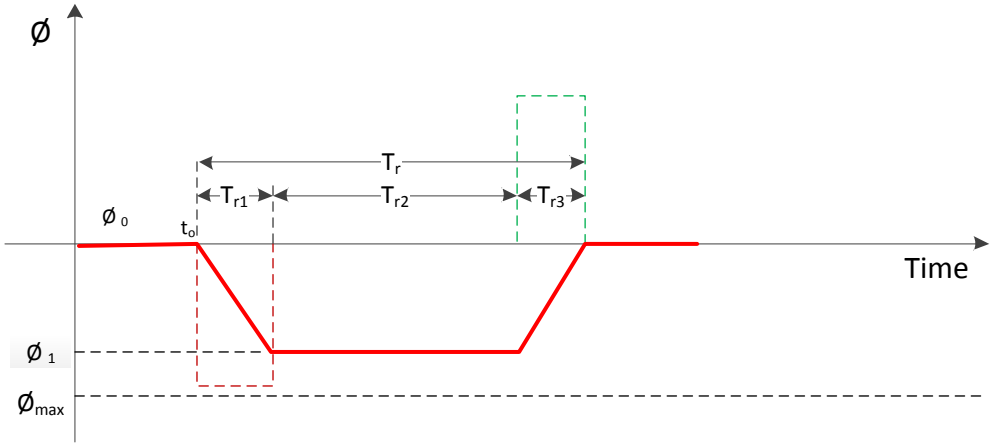

Figure 6.6 Bank angle ϕ

Roll rate integration gives the bank angle:

$$\phi(t) = \int_0^t \dot{\phi}(\tau) d\tau \tag{6.1}$$

By assuming $\dot{\phi}_3 = -\dot{\phi}_1$, so $T_{r1} = T_{r3}$ the bank angle during the time intervals is written as

follows:

$$\phi(t) = \begin{cases} \dot{\phi}_1 t & ; 0 \leq t \leq T_{r1} \\ \phi_1 & ; T_{r1} \leq t \leq T_{r1} + T_{r2} \\ \phi_1 - \dot{\phi}_1 t & ; T_{r1} + T_{r2} \leq t \leq +T_r \end{cases} \tag{6.2}$$

where the initial time is assumed to be zero $t_0 = 0$, and $\dot{\phi}_1 = \frac{\phi_1}{T_{r1}}$. The bank angle is shown in Figure 6.6.

The task now is to find the values of the time intervals (T_{r1}, T_{r2}, T_{r3}) as functions of the generated roll rate $\dot{\phi}_1$, the predefined heading angle $\Delta\psi$, and the predefined bank angle ϕ_1:

1. Time period T_{r1}, and T_{r3} are given by:

$$T_{r1} = T_{r3} = \frac{\phi_1}{\dot{\phi}_1} \tag{6.3}$$

2. Time duration T_{r2}: During this period the UAV bank angle has a constant value equal to ϕ_1. The total heading change during the total rolling time T_r must be equal to a predetermined value $\Delta\psi = \psi(T_r)$ (assuming $\psi_0 = 0$). The relationship between the heading rate and bank angle is given:

$$\dot{\psi}(t) = \frac{g}{V}\tan(\phi(t)) \tag{6.4}$$

Figure 6.7 shows the heading rate during the total rolling time interval T_r that produced by using (6.4).

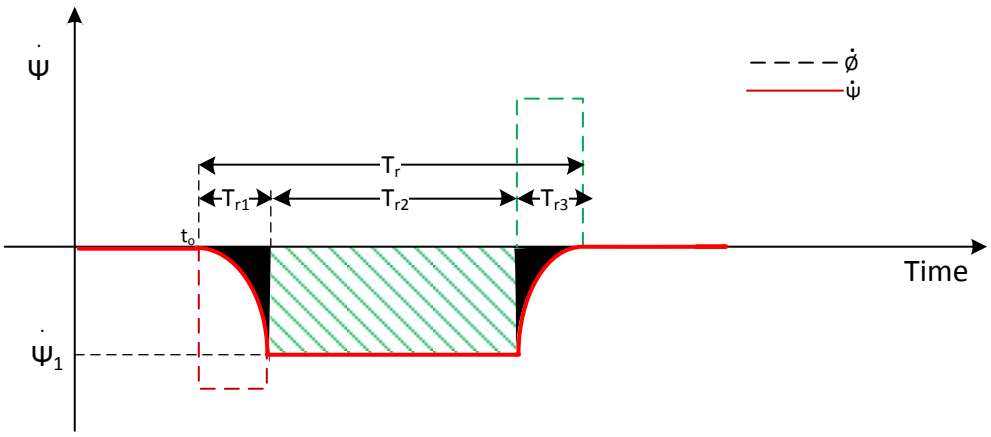

Figure 6.7 The heading rate

The heading rate integration gives the heading angle which is shown in Figure 6.8:

$$\psi(t) = \int_0^t \dot{\psi}(\tau) d\tau \tag{6.5}$$

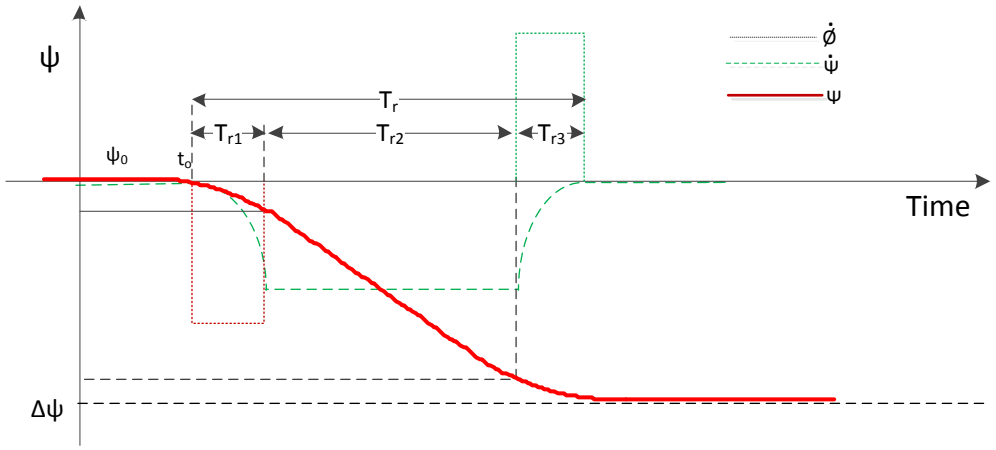

Figure 6.8 The heading angle

Thus, the heading angle at the end of T_r is calculated:

$$\psi(T_r) = \int_0^{T_r} \dot{\psi}(t) dt = \frac{g}{V} \int_0^{T_r} \tan(\phi(t)) dt \tag{6.6}$$

As $T_{r1} = T_{r3}$, and by taking the bank angle values during the different time ranges that given in (6.2), the (6.6) can be rewritten:

$$\Delta\psi = 2\frac{g}{V} \int_0^{T_{r1}} \tan(\dot{\phi}_1 t) dt + \frac{g}{V} \int_{T_{r1}}^{T_{r1}+T_{r2}} \tan(\phi_1) dt \tag{6.7}$$

hence

$$\Delta\psi = -2\frac{g}{V}\frac{1}{\dot{\phi}_1} \ln(\cos(\phi_1)) + \frac{g}{V} \tan(\phi_1) T_{r2} \tag{6.8}$$

where $\dot{\phi}_1 > 0$, and $-\frac{\pi}{2} < \phi_1 < \frac{\pi}{2}$. So the time duration T_{r2} is given:

$$T_{r2} = \frac{V\Delta\psi}{g\tan(\phi_1)} + \frac{2\ln(\cos(\phi_1))}{\dot{\phi}_1 \tan(\phi_1)} \tag{6.9}$$

The various avoidance manoeuvres which were discussed in Chapter 5 are generated based on the assumption of the linearity of the heading angle change, so the radius of the circle is used to find the avoidance manoeuvre parameters. It can be seen in Figure 6.8 that the

heading angle is nonlinear, hence the turn of the avoidance manoeuvre is not a circle. Thus, the parameters of the proposed manoeuvres must be modified:

1. Head-on/Overtaking avoidance manoeuvre (see Section 5.2.2):

   - The time periods T_1 and T_3 are modified to be equal to the total rolling time T_r.
   - The distance D_1, which is used to calculate D_2, hence the time period T_2 is modified to:

     $$D_1 = y(T_r) \tag{6.10}$$

     where $y(T_r)$ is the projection of the UAV position on the y-axis at the end of time period T_r. However, if the summation of traveled distances during time periods T_1, and T_3 is greater or equal to the minimum clearance distance d_c then the time period $T_2 = 0$ (i.e. if $2y(T_r) \geq d_c$ then $T_2 = 0$).

2. The RSL avoidance manoeuvre (see Section 5.3.1):

   - The time periods T_1 and T_3 are modified to be equal to the total rolling time T_r.
   - The projection of the distance between the UAV and the intruder on the y-axis, which is given by (5.20), is modified to:

     $$d_{ub2} = R_{y0} - (y(T_r) + d_{b1}) \tag{6.11}$$

3. The RS-LS avoidance manoeuvre (see Section 5.3.2): The time periods T_1 in the RS part, and T_3 in the LS part are modified to be equal to the total rolling time T_r.

4. The circle avoidance manoeuvre (see Section 5.3.3):

   - The time period T_t is modified to be equal to the total rolling time period T_r.
   - The condition to perform the circle manoeuvre is given by:

     $$R_{x0} \leq x(\frac{T_r}{4}) + d_{c(min)} - \frac{T_r}{4} V_c \sin(\theta_c) \tag{6.12}$$

The UAV positions $y(T_r)$, and $x(\frac{T_r}{4})$ are calculated by using a numerical method as the analytic solution is hard to be achieved due to the nonlinearity of the heading rates and heading

angle change. Figure 6.9 gives the flowchart of calculating $x(T_r)$ and $y(T_t)$ numerically, where $n_1 = \frac{T_r}{T_s}$ is the number of steps and T_s is sampling time.

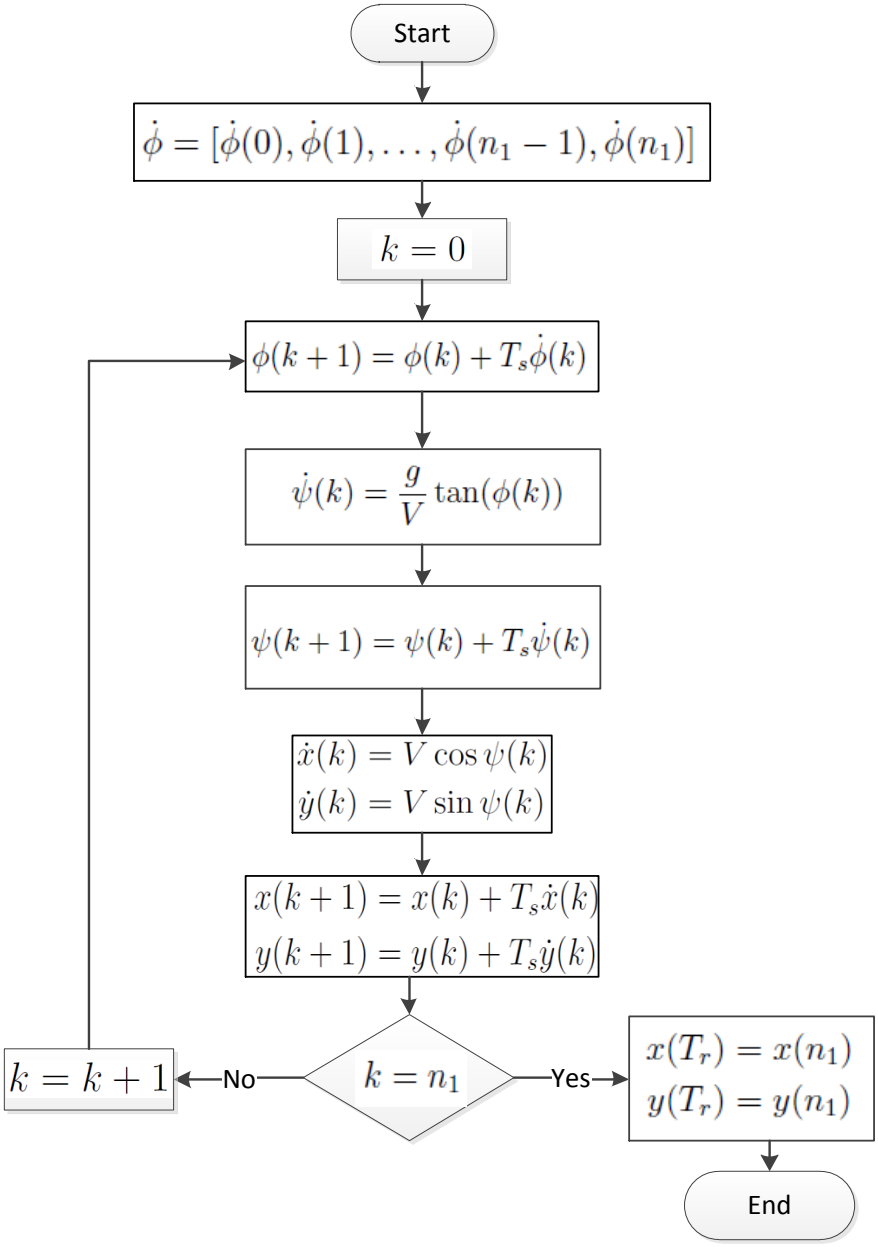

Figure 6.9 Calculation of the UAV position at the end of the time period T_r

The steps after construction of the avoidance manoeuvre heading rate command are the same as the steps described in Chapter 5. The constructed signal of the heading rate of the avoidance manoeuvre is used as input for the UAV lateral directional model to generate the avoidance manoeuvre trajectory profiles (speed, and position). Figure 6.10 shows the

flowchart of the avoidance manoeuvre generation algorithm using fuzzy logic techniques

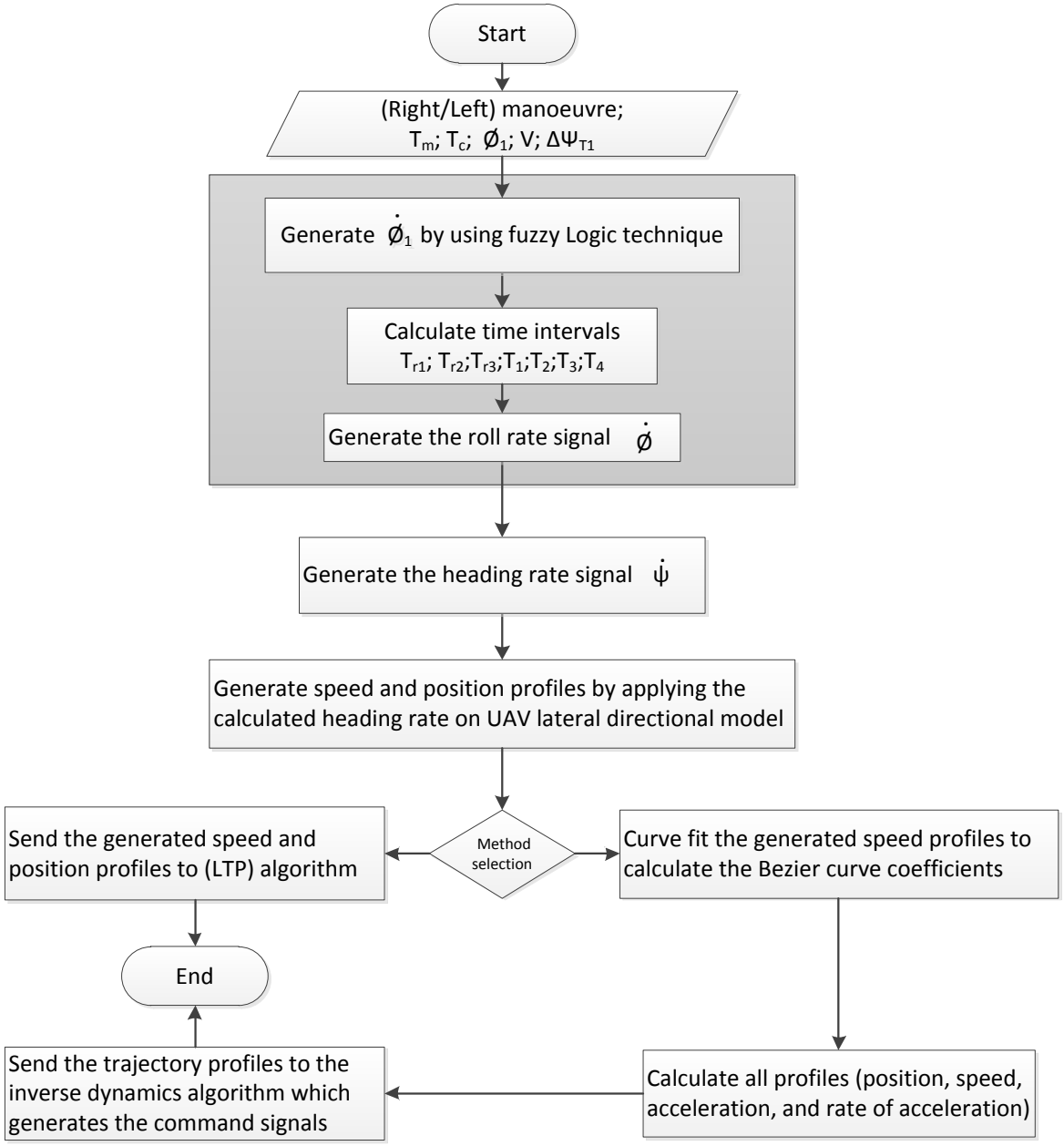

Figure 6.10 Avoidance manoeuvre generation algorithm flowchart

6.5 Simulation Results

The head-on conflict scenario is used to test the proposed algorithm. Different head-on conflict scenarios are established to test the proposed algorithm by using MATLAB/Simulink software. The UAV and the intruder are assumed to be in level flight condition with constant speeds, and they are at the same altitude of 1000 m. Two categories of head-on conflict scenario are used:

- Constant UAV speed with different values of the time to collision.
- Constant time to collision with different values of the UAV speed.

Figure 6.11 shows the block diagram of the Simulink model that is used to test the algorithm where:

- The pilot behaviour-based avoidance manoeuvre generator includes the algorithm that is shown in Figure 6.10.
- As the flight control system design is outside of the scope of this thesis the flight control system and the aircraft model are modeled by the point mass model assuming that the actual values track the generated values perfectly.

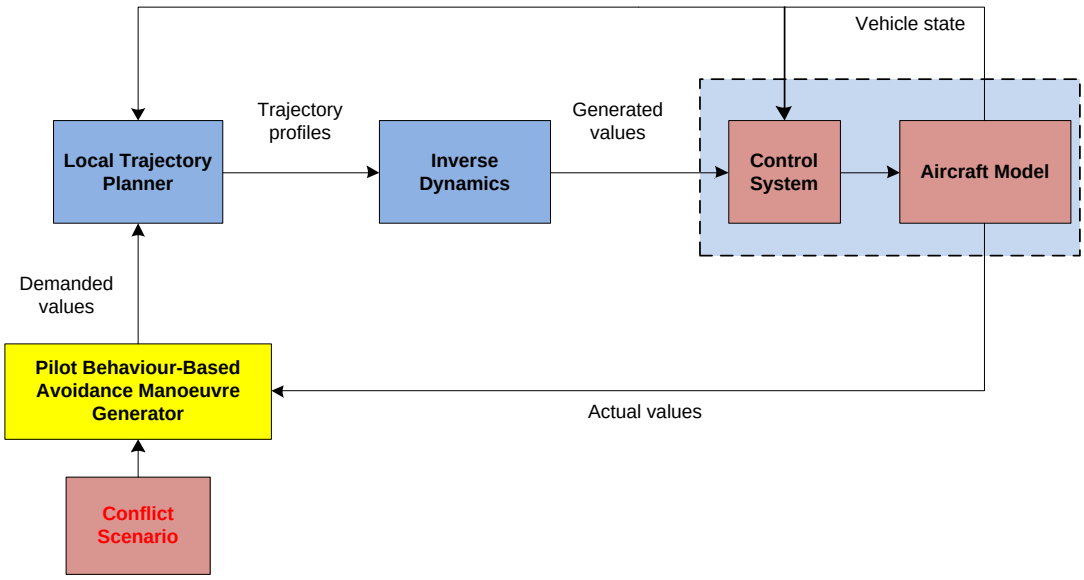

Figure 6.11 System block diagram

6.5.1 Constant V with Different Values of T_c

These scenarios are designed to show the effect of the time to collision T_c on the avoidance manoeuvre while the UAV speed V is constant for all scenarios. Figure 6.12 shows three head-on conflict scenarios where the UAV speed V and the intruder speed V_i are constant in all scenarios, but the intruder initial position is different for each scenario so the time to collision is different for each scenario.

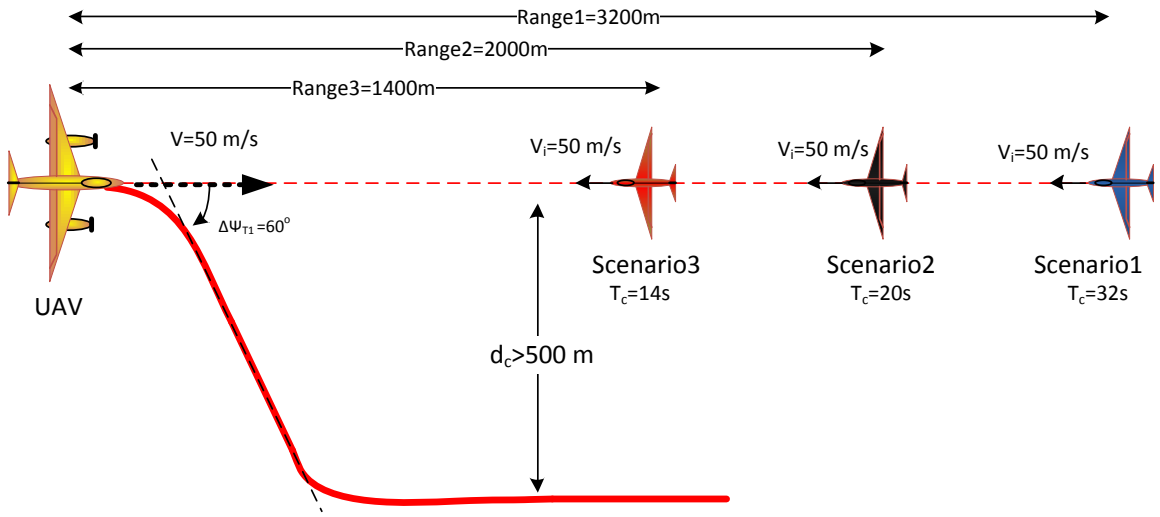

Figure 6.12 Head-on conflict scenarios (1,2, and 3)

Table 6.3 shows the avoidance manoeuvre parameters for each scenario. It can be seen that the heading rate is increased as the time to collision decreased. Hence, the turning part of the avoidance manoeuvre is performed faster.

Table 6.3 Conflict scenarios (1,2, and 3): $V = V_i = 50m.s^{-1}$, $\Delta\psi = 60^o$, $\phi = 45^o$

Parameter	Scenario1	Scenario2	Scenario3
Range [m]	3200	2000	1400
T_c [s]	32	20	14
$\dot{\phi}$[deg/s]	14.89	35	95
T_{r1} [s]	3.02	1.29	0.47
T_{r2} [s]	2.67	4.2	4.92
T_1 [s]	8.71	6.77	5.87
T_2 [s]	2.27	4.28	5.18

Figure 6.13 shows the simulation results of ($\dot{\phi}$, ψ, V, altitude, and the normal load factor and $\frac{1}{\cos(\phi)}$) for the conflict scenarios (1,2, and 3), it can be seen that the level coordinated turn is achieved successfully in all scenarios as the speed and the altitude are held around their demanded values (50m/s for the speed, and 1000m for the altitude) with small deviation. Also the normal load factor is within the acceptable reagin that given in CS-23 document [69], and it is almost identical with $\frac{1}{\cos(\phi)}$, hence the coordinated turn is performed smoothly. The generated heading angle tracks the demanded heading in all scenarios. However, as the roll rate increased, the abruptness of the demanded heading change increased, hence the mismatches between the generated values and the demanded values becomes more noticeable as it can be seen in Figures 6.14, 6.16, and 6.18 that show forward, lateral, and vertical axis speed (on the left column), and heading rate, bank angle, and flight path angle (on the right column) for the conflict scenarios 1, 2, and 3 respectively. These mismatches are small and their effects on the generated avoidance manoeuvres are not noticeable as can be seen in Figures 6.15, 6.17,and 6.19 that show the 3D view of the UAV and the intruder paths during the collision avoidance manoeuvres for the conflict scenarios (1, 2, and 3) respectively.
Figure 6.20 shows the UAV trajectories of the avoidance manoeuvres for the conflict scenarios (1, 2, and 3) in the horizontal plane, where the UAV initial position is $(0,0)$. It can be noticed that the curvature of the avoidance manoeuvre for Scenario 3 is higher than the curvature of the avoidance manoeuvre for Scenario 1, which has a larger time to collision. This is similar to the expected performance of piloted aircraft, where the pilot tries to avoid the conflicts with a shorter time to collision, in a faster manner than the conflicts with longer time to collision by applying a higher roll rate.

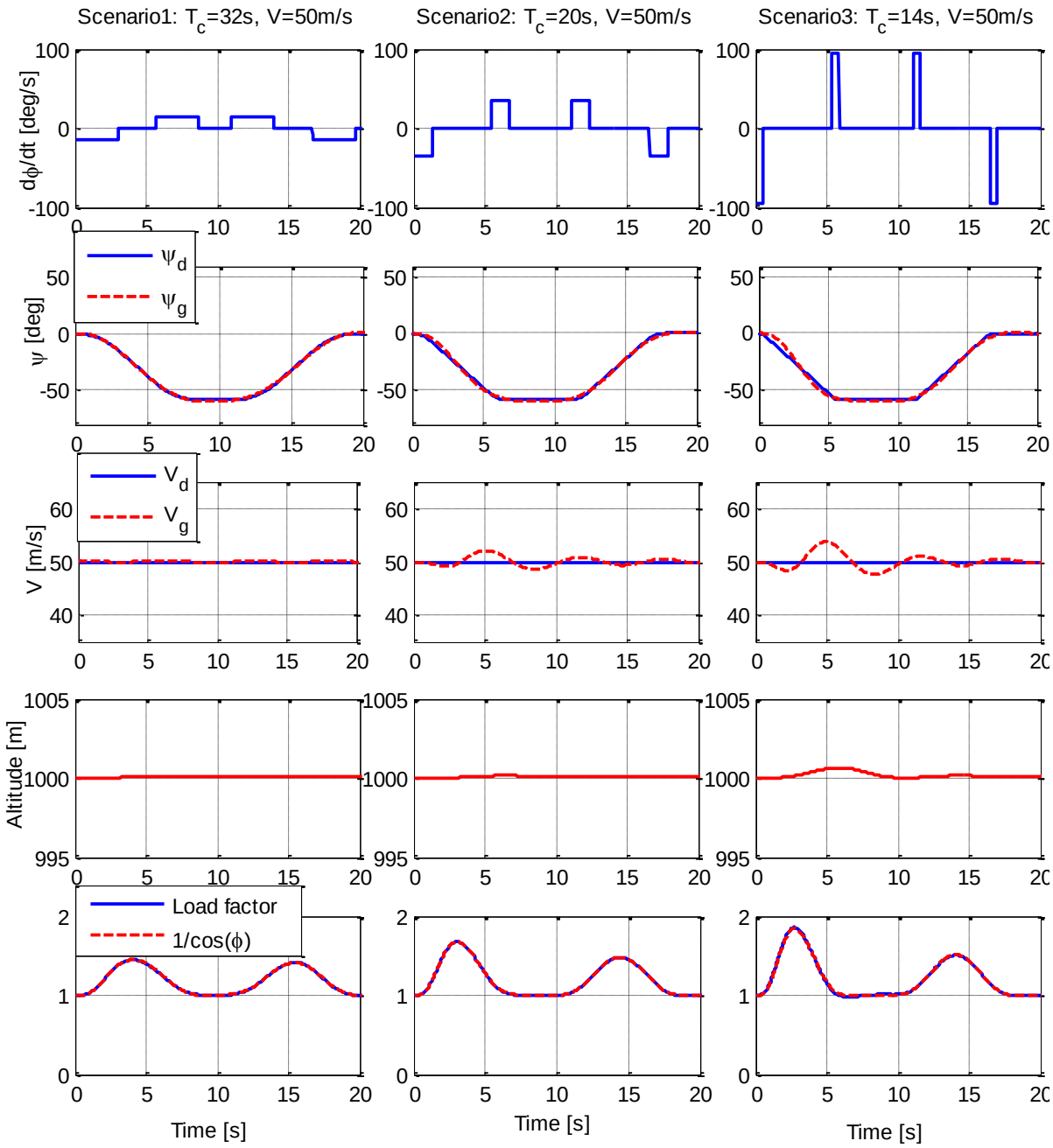

Figure 6.13 The simulation results of ($\dot{\phi}$, ψ, V, and altitude) for scenarios (1, 2, and 3)

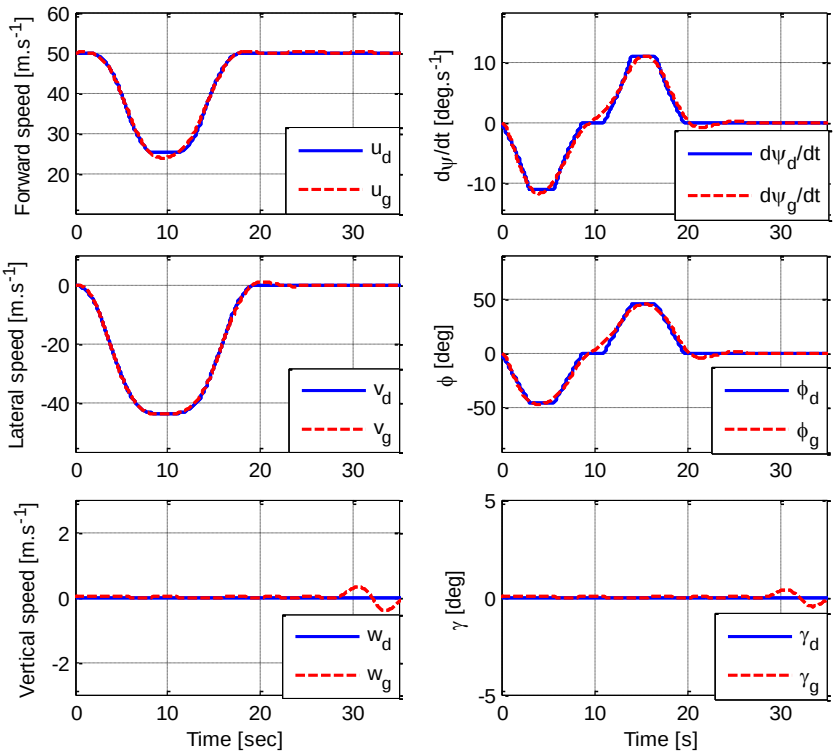

Figure 6.14 Scenario1 simulation results (Left): u, v, and w; (Right): ψ, ϕ, and γ

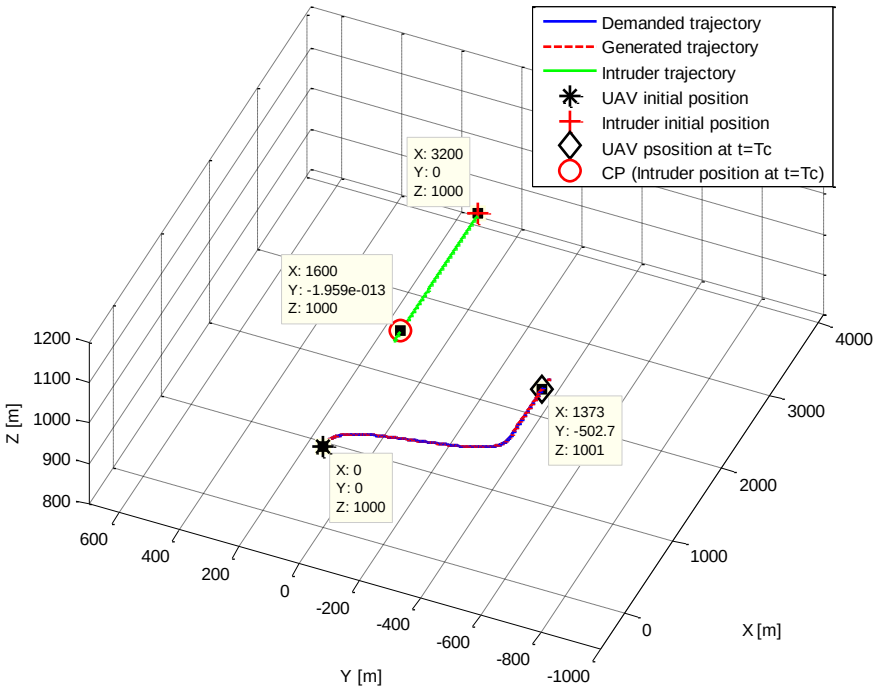

Figure 6.15 Scenario 1: 3D view of the UAV, and the intruder trajectories

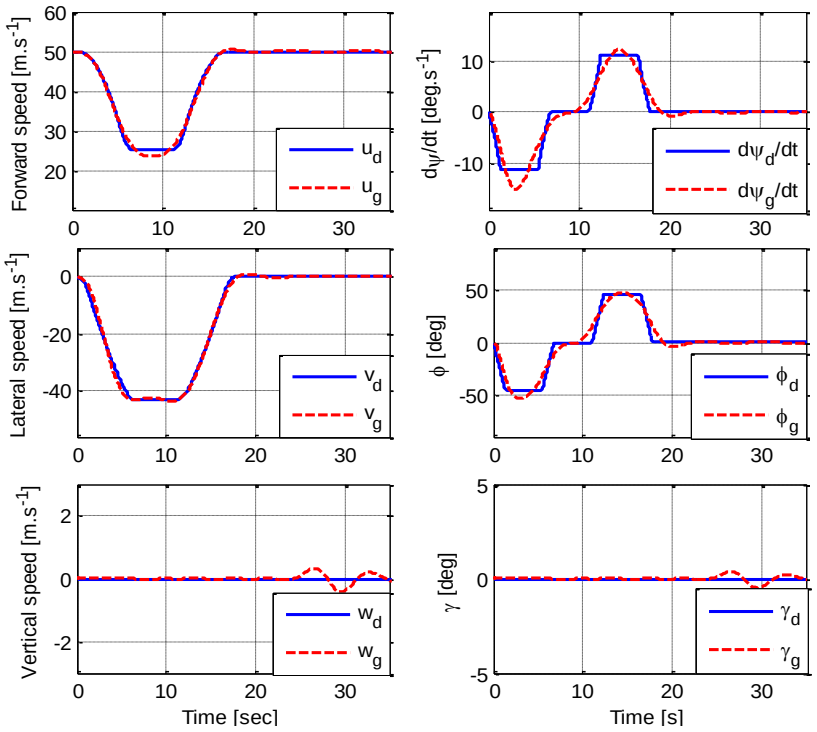

Figure 6.16 Scenario2 simulation results (Left): u, v, and w; (Right): $\dot{\psi}$, ϕ, and γ

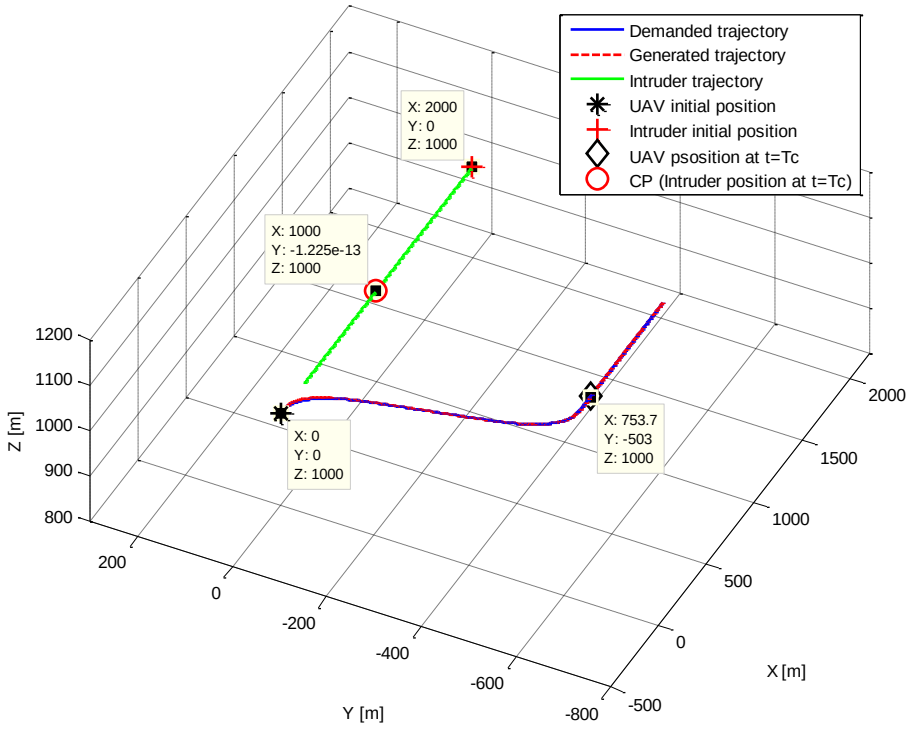

Figure 6.17 Scenario 2: 3D view of the UAV, and the intruder trajectories

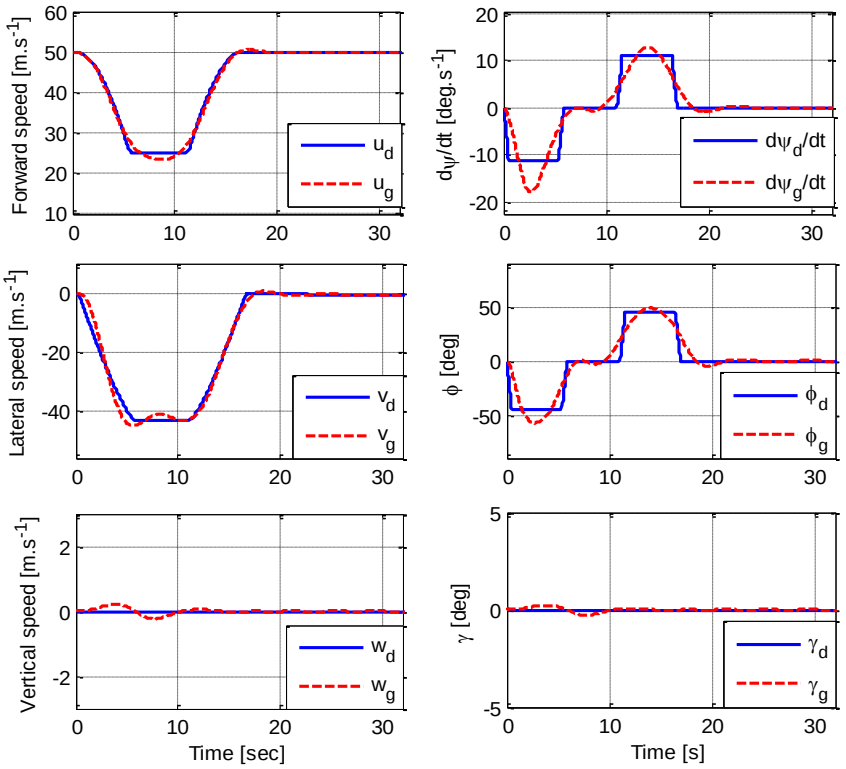

Figure 6.18 Scenario3 simulation results (left): u, v, and w; (Right): $\dot{\psi}$, ϕ, and γ

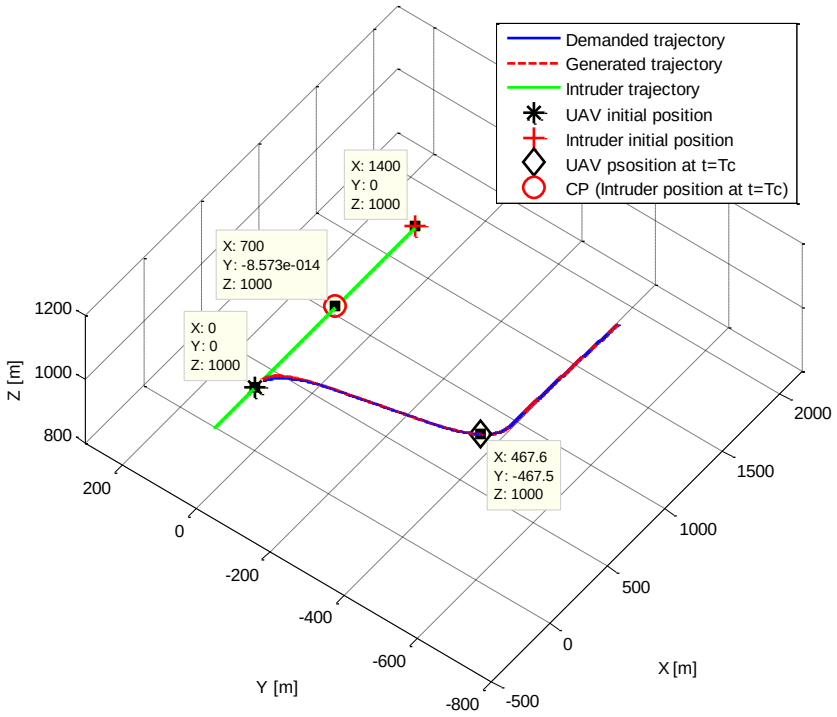

Figure 6.19 Scenario 3: 3D view of the UAV, and the intruder trajectories

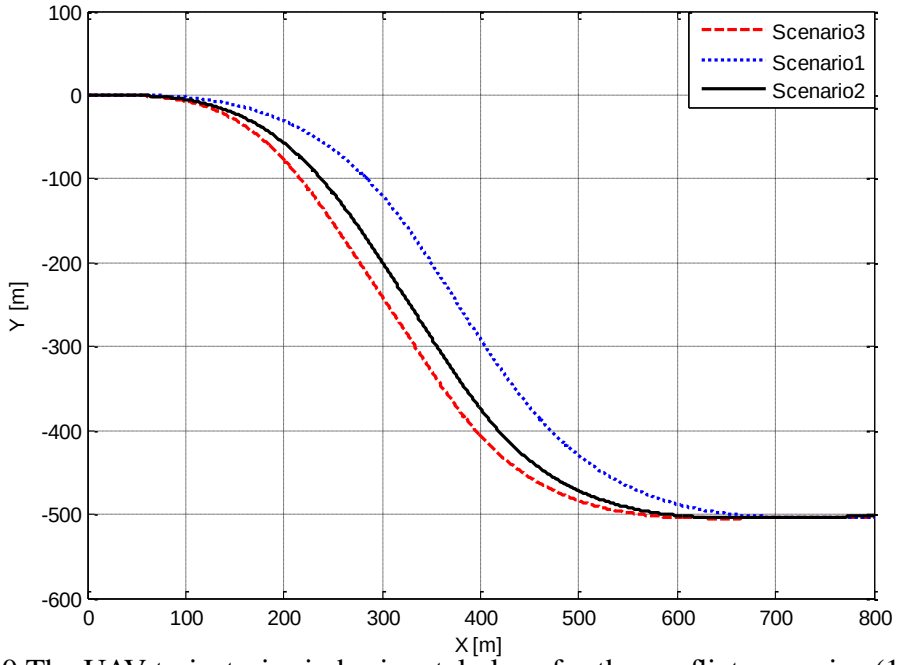

Figure 6.20 The UAV trajectories in horizontal plane for the conflict scenarios (1, 2, and 3)

6.5.2 Constant T_c with Different Values of V

These scenarios are designed to show the effect of the UAV speed V on the avoidance manoeuvre when the time to collision T_c is constant for all scenarios. Figure 6.21 shows three head-on conflict scenarios where the summation of the UAV speed V and the intruder speed V_i values are constant $V + V_i = constant$.

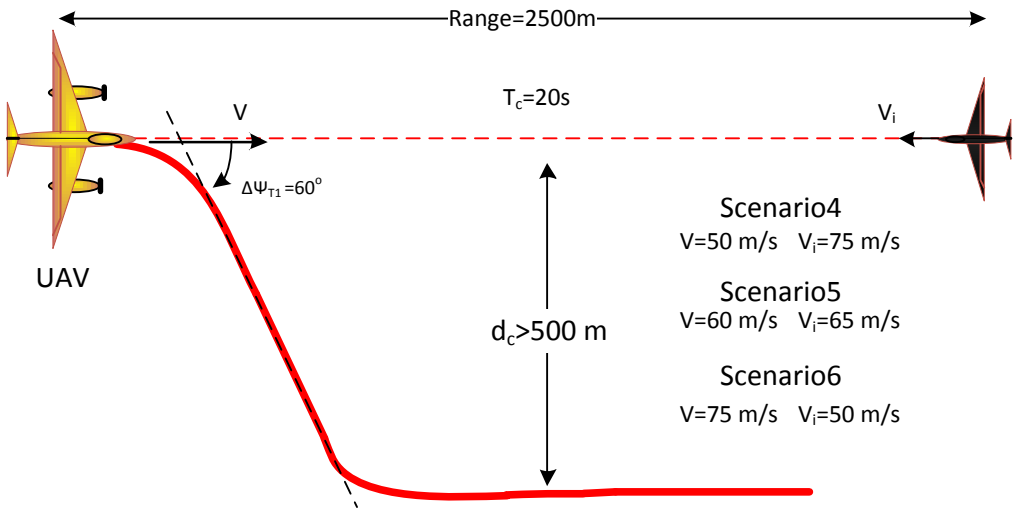

Figure 6.21 Head-on conflict scenarios (4,5, and 6)

The intruder is at the same range in all scenarios. Thus, the time to collision T_c is the same in all scenarios.

Table 6.4 shows the avoidance manoeuvres parameters for each scenario. It can be seen that the heading rate is increased as the UAV speed increased. The time period T_2 in Scenario 6 is zero because the traveled distance during the time period T_1, and T_3 is greater than the minimum clearance distance $d_c = 500m$ (i.e. $2y(T_r) > d_c$).

Table 6.4 Simulation scenarios: $Ra = 2500m$, $\Delta\psi = 60^o$, $\phi = 45^o$

Parameter	Scenario4	Scenario5	Scenario6
Vi $[m/s]$	75	65	50
V $[m/s]$	50	60	75
T_c [s]	20	20	20
$\dot{\phi}$[deg/s]	35	48.92	85.74
T_{r1} [s]	1.29	0.92	0.52
T_{r2} [s]	4.2	5.59	7.54
T_1 [s]	6.77	7.43	8.59
T_2 [s]	4.28	1.59	0
$y(Tr)$ [m]	157	209	304

Figure 6.22 shows the simulation results of ($\dot{\phi}$, ψ, V, altitude, and the normal load factor and $\frac{1}{\cos(\phi)}$)) for the conflict scenarios (4, 5, and 6). The generated speed, altitude and heading angle track the demanded values in all scenarios (with insignificant deviations). Thus, the level coordinated turn is achieved successfully. In Scenario 6 the UAV start rolling out immediately after the heading angle reaches its predetermined value because $T_2 = 0$.
Figures 6.23, 6.25, and 6.27 show forward, lateral, and vertical axis speed (on the left column), and heading rate, bank angle, and flight path angle (on the right column) for the conflict scenarios 4, 5, and 6 respectively. It can be seen that heading rate and the bank angle in scenario 6 are changed from their minimum values to their maximum values without settling at zero value for a period of time because of the zero value of time period T_2. This is not the case for scenarios 4 and 5 where T_2 has non-zero value. Figure 6.24, Figure 6.26, and Figure 6.28 show the 3D view of the UAV and the intruder paths during the collision avoidance manoeuvres for the conflict scenarios (4, 5, and 6) respectively. It can be seen that the UAV in Scenario 6 travels a longer distance in the y-axis to achieve the avoidance manoeuvre. The longer distance is needed to achieve the predetermined heading angle

($\psi = 60°$), because the higher speed turn has a larger radius of turn. Figure 6.29 shows the position profiles of the avoidance manoeuvres for the conflict scenarios (4, 5, and 6) in the horizontal plane, where the UAV initial position is $(0, 0)$.

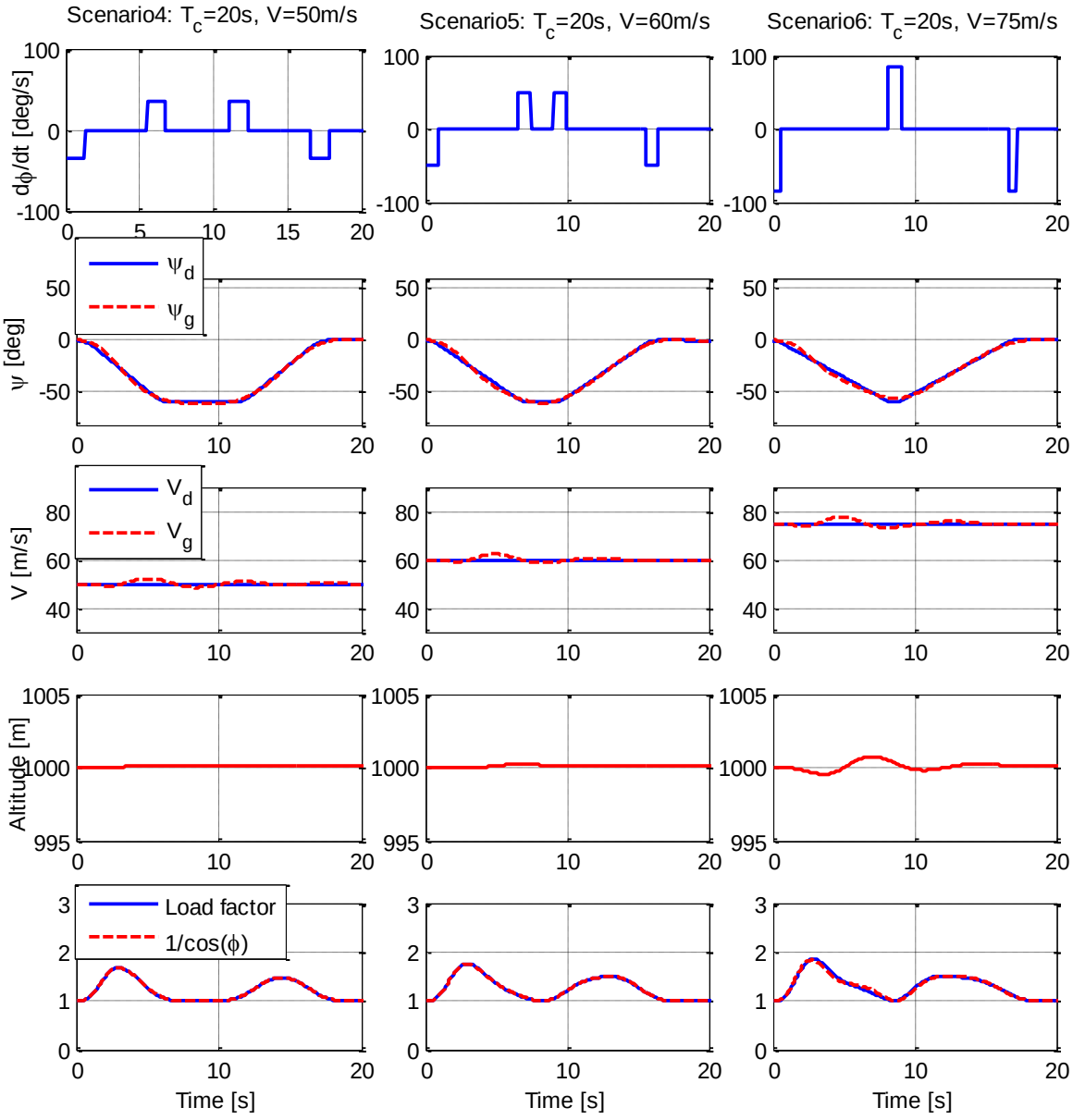

Figure 6.22 The simulation results of ($\dot{\phi}$, ψ, V, and altitude) for scenarios (4, 5, and 6)

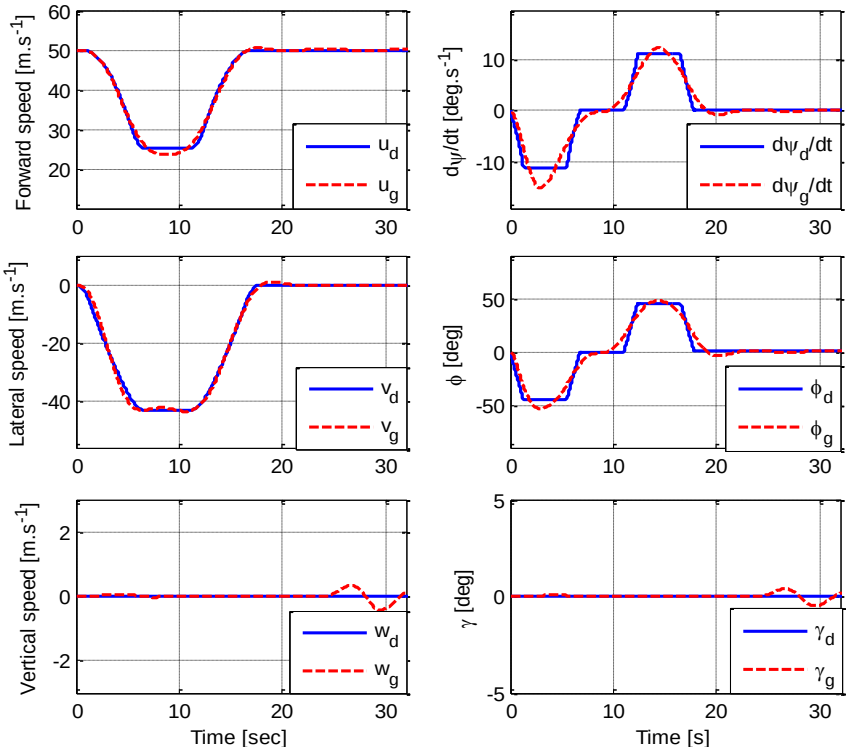

Figure 6.23 Scenario4 simulation results (left): u, v, and w; (Right): $\dot{\psi}$, ϕ, and γ

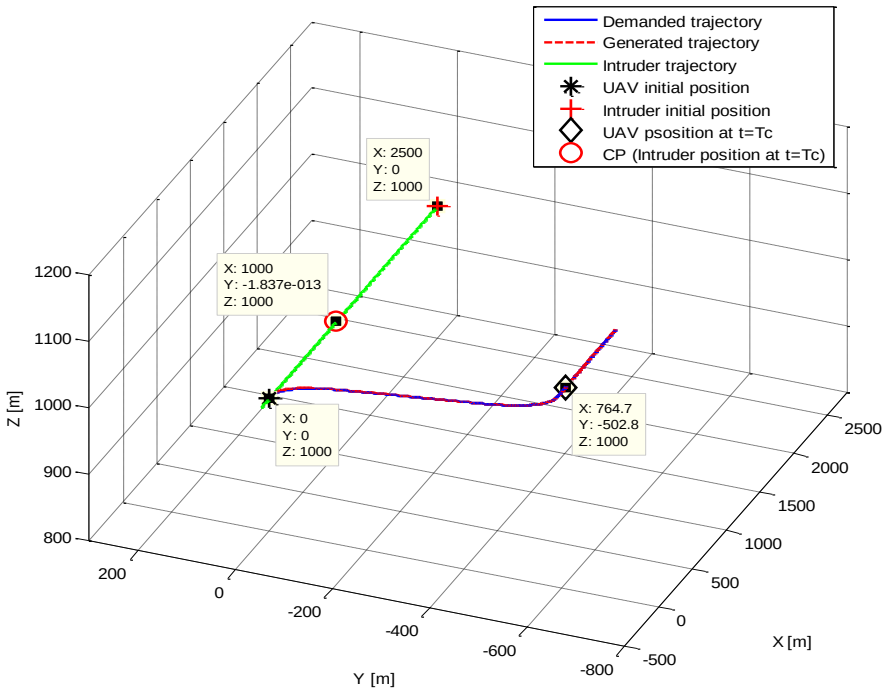

Figure 6.24 Scenario 4: 3D view of the UAV, and the intruder trajectories

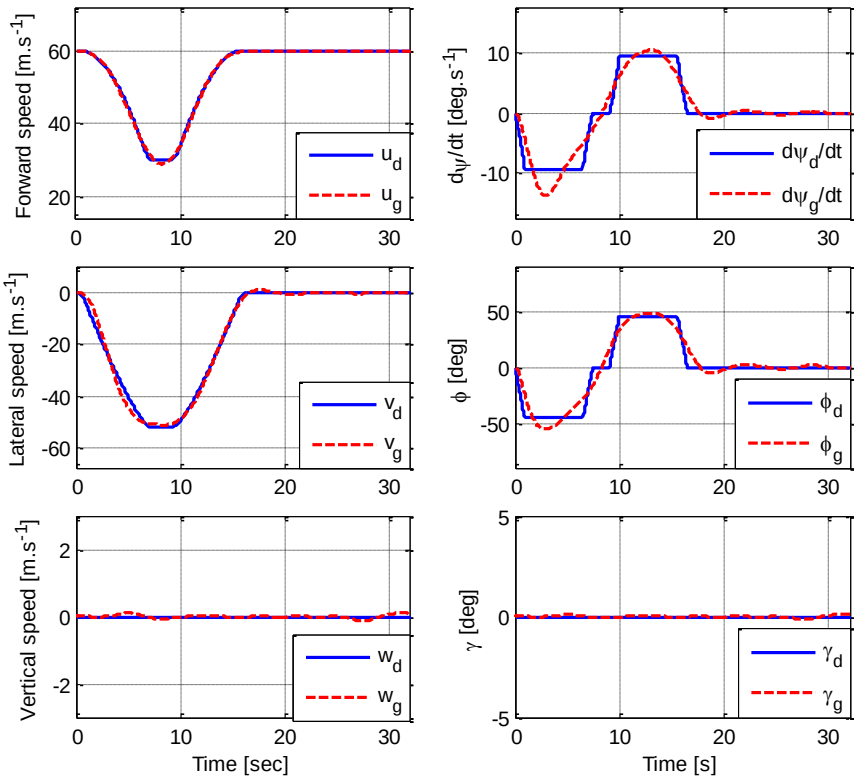

Figure 6.25 Scenario5 simulation results (left): u, v, and w; (Right): $\dot{\psi}$, ϕ, and γ

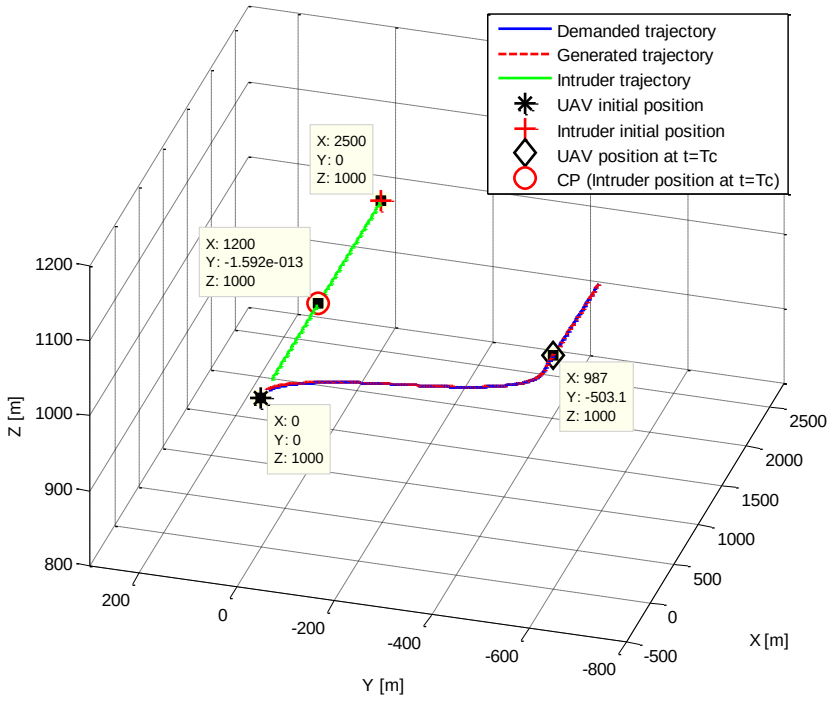

Figure 6.26 Scenario 5: 3D view of the UAV, and the intruder trajectories

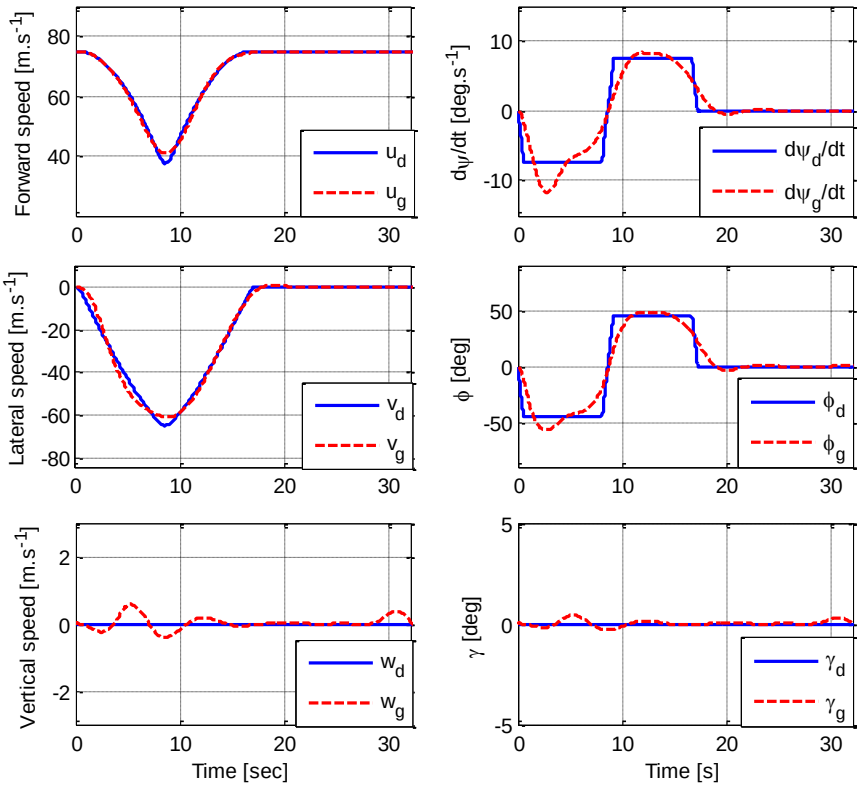

Figure 6.27 Scenario6 simulation results (left): u, v, and w; (Right): $\dot{\psi}$, ϕ, and γ

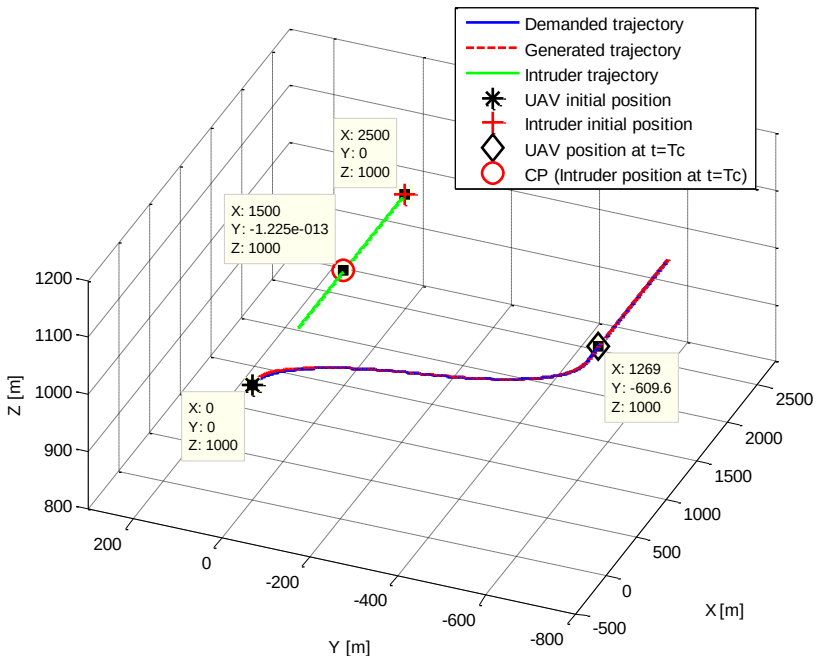

Figure 6.28 Scenario 6: 3D view of the UAV, and the intruder trajectories

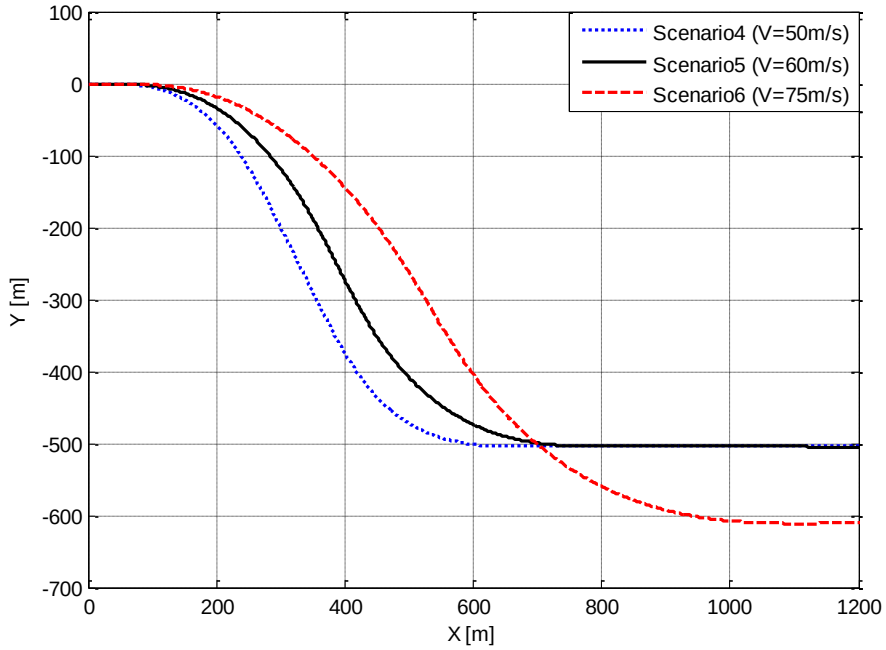

Figure 6.29 The UAV trajectories in the horizontal plane for the conflict scenarios (4, 5, and 6)

6.6 Summary

This chapter discusses how pilot experience can be used in the collision avoidance manoeuvre generation process for the UAV. This chapter presents a proposal for linking the avoidance manoeuvre's parameters with the conflict scenario's parameters using the pilot behaviour during the conflict. This is to augment the pilot experiences in the avoidance manoeuvre generation process. In manned aircraft, pilots use their experience to initiate the suitable manoeuvre to avoid the conflict. The fuzzy logic technique is proposed to model the pilot actions in generating the avoidance manoeuvre for different conflict scenarios.

Chapter 7

Conclusions and Recommendations for Future Work

Operation of UAV's in civil airspace is restricted by the policies of aviation authorities which require full compliance with rules and obligations that apply for manned aircraft. Trajectory tracking and collision avoidance are issues that a UAV must deal with in a way that gives the UAV the ability to avoid conflict situations. Thus, any UAV that will be operated in civil/non-segregated airspace must be equipped with a collision avoidance system that has the ability to avoid conflict scenarios in full compliance with the rules of the air. The main aim of this thesis is to develop a control algorithm for collision avoidance system for aircraft that are flying under VFR conditions. The developed algorithm could be used at different levels of the aircraft autonomy. For example, the developed algorithm can be used as an advisory system for manned aircraft to help the pilot for see and avoid, or it can be a guidance system for a UAV in order to enable it to fly in civil airspace. The following sections give conclusions for the proposed algorithms and techniques that have been developed in this thesis and discusses their limitations, as well as giving some recommendations for future work.

7.1 Local Trajectory Planning Algorithm

An optimal local trajectory generation that uses B-splines is proposed for a real-time collision avoidance algorithm. Online avoidance manoeuvre generation, optimisation, and global trajectory tracking for different conflict scenarios are tested successfully in a simulation environment. The predicted trajectory is generated by using MPC techniques. Essentially, a finite-horizon optimal control problem is periodically solved in real-time hence

updating the aircraft trajectory to avoid obstacles and track a predefined global trajectory. The aircraft and obstacle constraints are augmented in the cost function using a penalty function method. The computational time for the real-time collision avoidance algorithm is reduced significantly by using output space to formulate the optimal control problem, and augmenting the vehicle/obstacle constraints in the cost function. A coarse grid approach is proposed to help the optimal control problem solver to escape the local minima and ensure sufficient coverage of the overall design space. Differential flatness of the system for a fixed wing aircraft is used to produce an inverse dynamic model for the UAV. Hence, the generated local trajectory profiles passed to the inverse dynamic model to generate the coresponding control signals. The simulation results show that the proposed approach allows the UAV to track a predefined global trajectory, as well as avoiding collisions with different types of conflict scenarios in real-time. In the simulation and for the tested scenarios the proposed algorithm successfully allows the UAV to achieve the following tasks:

1. Satisfy the UAV constraints.
2. Track a predefined global trajectory.
3. Avoid static obstacles.
4. Avoid moving obstacles.

However, the developed algorithm suffers from limitations. Some of these limitations can be summarized:

- The intruder trajectory has been estimated using a straight projection method in which the intruder states are projected into the future along a single trajectory. Although this method is simple, there is no direct consideration of the uncertainties.
- The proposed method for avoiding local minima reduces the possibility of getting trapped in local minima but does not overcome it totally.
- The UAV has been modelled by a three Degree of Freedom (3-DoF) point-mass model based on some assumptions (e.g. still wind, zero sideslip, flight path angle γ equals pitch angle θ).
- No effort has been directed towards investigating the effect of length of the horizon time (t_h), and the sampling resolution of the global trajectory on the algorithm efficiency.

Some recommendations for future work are:

- A method for intruder future trajectory estimation should be developed taking into consideration the intruder manoeuvrability. One proposal is to develop a model that predicts the intruder manoeuvre based on limitations of the intruder, and uncertainties of measured data.
- Monte Carlo simulation should be performed for a more through evaluation of the proposed proposed algorithm.
- Test the algorithm with external disturbances (i.e. wind), and internal failures (e.g. UAV actuator failures)
- Test the algorithm with a six Degrees of Freedom (6-DoF) model of a UAV.
- As the inverse dynamics generates the command inputs for the receding horizon time, a MPC can be proposed to design the UAV controllers (in the control layer).
- The algorithm should be experimentally tested using a flight simulator, and flight test.
- Develop a more effective algorithm for the optimal control problem solver to avoid local minima.

7.2 Decision Making System

A multi-layer DMS is developed for a sense and avoid system based on the rules of the air in VFR conditions. The proposed DMS architecture is engineered to be implemented for different functionalities for manned aircraft, and at different level of autonomy of UAS, or at different classes of the flight control modes that are specified by CAA (see Section 4.7 for more details about flight control modes). For example, in manned aircraft (Class-0 flight control mode) the proposed DMS can be used to reduce the overall pilot workload by performing the flowing tasks:

1. Collisions detection and risk prioritizing (Layer-1), so the pilot will know where the traffics are (Section 4.8.1, and Section 4.8.2).
2. Determine the collision scenario type (i.e. head-on, left/right approaching, and overtaking/overtaken) by using Layer-1, and Layer-2. This will help the pilot to decide what actions are needed to avoid the conflict (Section 4.8.4).

3. Evaluate the collision type, and generates conflict resolution advisories (Layer-3) to help the pilot to initiate a safe avoidance manoeuvre (Section 4.8.5).

Currently available commercial collision avoidance systems that are used for see and avoid (e.g. the Potable Collision Avoidance System PCAS) can just perform the first task (see Section 4.8.1). Earlier discussion shows that most of the required time for a pilot to recognize an approaching aircraft and initiate an avoidance manoeuvre is spent on the decision making process. Hence, a decision making system that could be used as an advisory system will effectively save time and help the pilot to avoid the conflicts safely.
For a remotely piloted UAV (Class-1 flight control mode) the proposed DMS architecture could enhance the remote pilot awareness of the conflict, as it is can perform a decision making process similar to a pilot in manned aircraft. So the remote pilot could act in the same manner as the pilot on board in case of manned aircraft. Moreover, the remote pilot can switch to the autonomous mode by engaging Layer-4 of the DMS which initiate a safe avoidance manoeuvre (Section 4.8.6). A GUI for the DMS is proposed for test and simulation purposes with a brief comparison with currently used commercial PCAS.

Some limitations of the developed DMS are:

- The type of intruder has not been considered in the decision making process. The rules of the air suggest different actions for different intruders. For example, when two aircraft are converging at approximately the same level, the aircraft that has the other on its right shall give way, except as follows [16]:

    1. Power-driven heavier-than-air aircraft shall give way to airships, gliders and balloons

    2. Airships shall give way to gliders and balloons

    3. Gliders shall give way to balloons

    4. Power-driven aircraft shall give way to aircraft which are seen to be towing other aircraft or objects

- The case when the intruder performs aggressive/unpridectable manoeuvres (e.g. turn left where it supposed to turn right according to the rules of the air) are not considered in the decision making process.

However, the proposed DMS architecture gives flexibility for any further development layer by layer, or the whole system for research or certification purposes. Some recommendations for future work are:

- Develop an intruder recognition system to be added to the DMS, so the decision making process can take into consideration the type of the intruder.
- Decisions in response to aggressive/unpridectable manoeuvres by the intruder should be developed.
- The system can be developed to include a cooperative operating mode. In case the UAV and the intruder are equipped with DMS, the system should work in a coordinated manner. For example, when an RA, or avoidance manoeuvre is initiated for conflicting aircraft, the other aircraft will be initiated with a coordinated RA/avoidance manoeuvre. Hence, the overall system safety is enhanced.

7.3 Collision Avoidance Manoeuvre Generation

Collision avoidance manoeuvres are proposed for a subset of the possible conflict scenarios. The proposed manoeuvres are generated to be similar to the manoeuvres that are performed by manned aircraft. A level coordinated turn is proposed for the turning part of the avoidance manoeuvres (Section 5.2.1). This type of turn is selected according to pilot suggestions as it is the preferred manoeuvre by pilots to perform a horizontal avoidance manoeuvre in see and avoid conditions. The avoidance manoeuvres generation process takes the rules of the air and the safety requirements into consideration. A geometric approach is used to parameterize the proposed avoidance manoeuvre. Then the parameterized manoeuvres are discretised and used as a global trajectory to be tracked using the developed local trajectory planning algorithm. The collision avoidance manoeuvres are proposed for conflict scenarios where the UAV and the intruder are in a cruising state (constant speed level flight). The following conflict scenarios have been considered:

1. Head-on/Overtaking conflict scenarios: One type of the avoidance manoeuvre is used for all head-on conflict scenarios (Section 5.2.2).
2. Right approached conflict scenarios: Different cases are discussed, then, three main types of the avoidance manoeuvres for these cases are proposed (Section 5.2.3).

The weakness with this method is ensuring that a manoeuvre has been defined for every possible situation. Another drawback is that a limited set of scenarios has been defined, which is a subset of all possible scenarios. Hence, the effectiveness of this method cannot be proven by just simulation and testing it for these limited number of conflict scenarios. To start ensuring every scenario is covered one would have to define the set of all scenarios and make sure the set was covered. Proving that the methods work for every scenario is hard to be achieved. However, Monte Carlo simulation can be established in order to show the method's effectiveness. Other recommendations for future work are:

- Propose and generate avoidance manoeuvres for conflict scenarios that need a change in aircraft vertical trends (i.e. increase/decrease the decent/climb rate, or level off). Constant climb/decent rate avoidance manoeuvres can be used to avoid these type of conflict scenarios.

- Optimal control problem can be formulated in order to optimise the generated avoidance manoeuvres. This can be achieved by calculating the optimal heading angle and bank angle that minimise a defined cost function. The cost function could be the distance of the avoidance manoeuvre, the time of the avoidance manoeuvre, or the fuel consumption. The resulting avoidance manoeuvre may no longer resemble a pilot generated avoidance manoeuvre.

7.4 Pilot Behaviour-Based Collision Avoidance Manoeuvre Generation

A further development in avoidance manoeuvre generation algorithm is proposed by augmenting pilot experience in the avoidance manoeuvre generation process. Hence, the UAV mimics manned aircraft behaviour in avoiding the conflict scenarios. A fuzzy logic technique is proposed to represent the pilot model during the avoidance manoeuvre based on the pilot behaviour during different conflict scenarios (Section 6.3). This includes:

1. Define the fuzzy logic system inputs/outputs: The fuzzy logic system inputs/outputs are defined by determining the main factors the affect the pilot reaction and use them as inputs for the fuzzy system. This research shows that the demanded roll rate that is generated by the pilot depends mainly on the time to collision and the aircraft speed. Thus, the time to collision, and the UAV speed are used as inputs for the fuzzy logic system, while the roll rate is chosen to be the output (Section 6.3.3).

2. The knowledge-based rules for the fuzzy logic system are established using pilot experience.

3. The outcome of the fuzzy logic algorithm is linked to the avoidance manoeuvre generation algorithm. Then the generated avoidance manoeuvre is parameterized and discretised (Section 6.4).

Some limitations of this algorithm are:

- The pilot behaviour information during the collision avoidance is extracted from a single pilot experience. Pilot behaviour differs from one pilot to another according their experience and the type of the aircraft.

- Two factors are considered in pilot reaction modelling, these are time to collision T_c, and aircraft speed V. However, there are other factors which have effects on the pilot decision that have not been considered such as the intruder type, and multiple conflict scenarios.

- It is difficult to validate the generated manoeuvres. CAA gives general instruction to avoid collisions based on the rules of the air, but there are no specific procedure or detailed requirements to perform the avoidance manoeuvres.

Recommendations to develop this method in future work are:

- Propose and develop a method for validating the avoidance manoeuvre: A Turing-type test [84] can be proposed to develop a framework that can be used to validate the generated avoidance manoeuvres. The literature includes a small amount of research that explores the feasibility of a Turing-type test for UAVs. For example, Young [85, 86] has proposed a Turing-style test to define and evaluate key metrics of autonomous vehicle performance. Duke et al. [87, 88] have proposed Turing-type test for the performance of airborne systems within the National Airspace System (NAS) under Instrument Flight Rules (IFR). However, the developed Turing-type test that is presented in [87] has proposed some modifications in the Federal Aviation Regulations (FARs) to be carried out. Also a Turing-type test has been proposed by Kalik and Prokhorov [89] for intelligent automotive vehicles. However, using Turing-type tests for UAVs has not yet been explored in depth.

- Develop a method for involving a number of pilots to extract the pilot behaviour during the conflict. The proposal is to extract a number of pilots experiences and

combine them to design a single fuzzy logic algorithm that models the pilot behaviour during the conflict. Some suggestions to achieve this proposal are:

1. Extended interviews with the pilots: This method has been used in this thesis as the experience has been extracted from a single pilot. This technique can be used if the number of pilots is small, but would be inefficient for a larger number of pilots.

2. Questionnaire: This method is effective for a large number of pilots. One challenge to this method is that each pilot will try to describe their experience in different ways, not all of which will fit neatly into a fuzzy rule. Hence, a method for designing a questionnaire form should be developed for extracting the experience from responses to questionnaires and putting these into fuzzy rules. One suggestion for designing the questionnaire form is to design it after making some interviews with few pilots or with an expert pilot then design the questionnaire.

3. Experimental approach: Putting a number of pilots into a simulator and running them through a number of scenarios and recording their responses. The experimental approach may be easier, because the responses can be quantified, and weighted according to the success of the avoidance. An independent observers of the experiment can be involved to judge the success of the pilot's avoidance manoeuvre. However, the cost of this method is high comparing to the questionnaire, or the extended interviews method, as it needs more facilities to be established (e.g. simulator, recording devices, and an independent observer)

4. Combination of the previous methods can be considered.

Bibliography

[1] C. K. Lai, M. Lone, P. Thomas, J. F. Whidborne, and A. K. Cooke, "On-board trajectory generation for collision avoidance in unmanned aerial vehicles," in *2011 IEEE Aerospace Conference*, pp. 1–14, March 2011.

[2] U.S. Department of Transportation, "Integration of Civil Unmanned Aircraft Systems (UAS) in The National Airspace System (NAS) Roadmap," tech. rep., U.S. Department of Transportation, FAA, Washington, DC, 2013.

[3] T. Hutchings, S. Jeffryes, and S. Farmer, "Architecting UAV sense and avoid systems," in *2007 IET Conference on Autonomous Systems*, (London, UK), pp. 1–8, November 2007.

[4] J. Pellebergs, "The MIDCAS project," in *27th International Congress of the Aeronautical Sciences*, (Nice, France), 2010.

[5] L. Wong, W. Chen, J. Kay, and V. Raska, "Autonomous Sense and Avoid (SAA) for Unmanned Air System (UAS)," in *RTO-MP-SCI-202 Symposium on Intelligent Uninhabited Vehicle Guidance Systems*, (Munich, Germany), 2009.

[6] F. P. Osinga, *Science, Strategy and War: The Strategic Theory of John Boyd*. London: Routledge, 2006.

[7] U.S. Department of Transportation , "Advisory Circular: Pilots' Role in Collision Avoidance," Tech. Rep. 90-48C, Federal Aviation Administration, 1983.

[8] Australian Transport Safty Bureau, Australian Transport Safety Bureau (ATSB), Adelaide, *Limitations of the See-and-Avoid Principle*, November 2004.

[9] Safety Regulation Group, " Manual of Air Traffic Services Part 1," Tech. Rep. CAP-493, Civil Aviation Authority, West Sussex, UK, March 2014. Edition 5.

[10] A. Berry, J. Howitt, D. Gu, and I. Postlethwaite, "Continuous local motion planning & control for micro-air-vehicles in complex environments," in *AIAA Guidance, Navigation, and Control Conference*, (Toronto, Ontario), 2010.

[11] J. Kuchar and L. Yang, "A review of conflict detection and resolution modeling methods," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, pp. 179–189, December 2000.

[12] A. Zeitlin, "Sense and avoid capability development challenges," *Aerospace and Electronic Systems Magazine*, vol. 25, pp. 27–32, October 2010.

[13] B. Albaker and N. Rahim, "A survey of collision avoidance approaches for unmanned aerial vehicles," in *International Conference for Technical Postgraduates (TECHPOS)*, (Kuala Lumpur, Malaysia), pp. 1–7, December 2009.

[14] R. L. Ennis. and Y. J. Zhao, "Characterization of aircraft protected zones," in *AIAA's 3rd Annual Aviation Technology, Integration, & Operations (ATIO) Forum*, (Denver, CO), November 2003.

[15] T. Williamson and N. Spencer, "Development and operation of the traffic alert and collision avoidance system (TCAS)," *Proceedings of the IEEE*, vol. 77, pp. 1735–1744, November 1989.

[16] T. Thom, *Air Pilot's Manual - Aviation Law & Meteorology'*, vol. 2. Shrewsbury: Airlife Publishing, 2004.

[17] RTCA Special Committee 186, "Minimum aviation system performance standards for automatic dependent surveillance broadcast (ADS-B)," in *RTCA*, RTCA Special Committee, 1997.

[18] R. Y. Gazit, *Aircraft Surveillance and Collision Avoidance using GPS*. PhD thesis, Stanford University, August 1996.

[19] R. Holdsworth, *Autonomous In-Flight Path Planning to Replace Pure Collision Avoidance for Free Flight Aircraft Using Automatic Dependent Surveillance Broadcast*. PhD thesis, Swinburne University of Technology, 2003.

[20] P. Y. Oh, "Flying insect inspired vision for micro-air-vehicle navigation," in *Proceedings AUVSI's Unmanned Systems, North America 2004*, pp. 2201–2208, 2004.

[21] L. Muratet, S. Doncieux, Y. Briere, and J. Meyer, "A contribution to vision-based autonomous helicopter flight in urban environments," *Robotics and Autonomous Systems*, vol. 50, no. 4, pp. 195–209, 2005.

[22] R. Kephart and M. Braasch, "See-and-avoid comparison of performance in manned and remotely piloted aircraft," *IEEE Aerospace and Electronic Systems Magazine*, vol. 25, pp. 36–42, May 2010.

[23] B. Breen, "Controlled flight into terrain and the enhanced ground proximity warning system," *Aerospace and Electronic Systems Magazine, IEEE*, vol. 14, pp. 19–24, Jan 1999.

[24] R. Asep, A. K. Achaibou, and F. Mora-Camino, "Automatic collision avoidance based on supervised predictive controllers," *Control Engineering Practice*, vol. 4, no. 8, pp. 1169–1175, 1996.

[25] R. Carlson and J. Lee, "Detecting near collisions for satellites," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, no. 3, pp. 921–929, 1997.

[26] A. W. Merz, "Maximum-miss aircraft collision avoidance," *Dynamics and Control*, vol. 1, no. 1, pp. 25–34, 1991.

[27] K. L. Bowers, "Determining the feasibility of a flight profile in a free flight environment," in *Proceedings AIAA/IEEE Digital Avionics Systems Conference*, pp. 81–86, 1996.

[28] C. Tomlin, G. Pappas, and S. Sastry, "Conflict resolution for air traffic management: a study in multiagent hybrid systems," *IEEE Transactions on Automatic Control*, vol. 43, pp. 509 –521, April 1998.

[29] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *IEEE International Conference on Robotics and Automation*, vol. 2, pp. 500–505, March 1985.

[30] A. Chakravarthy and D. Ghose, "Obstacle avoidance in a dynamic environment: a collision cone approach," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 28, pp. 562–574, September 1998.

[31] M. Y. Cho, A. J. Lichtenberg, and M. A. Lieberman, "Minimum stopping distance for linear control of an automatic car-following system," *IEEE Transactions on Vehicular Technology*, vol. 45, no. 2, pp. 383–390, 1996.

[32] C. Tomlin, J. Lygeros, and S. Sastry, "A game theoretic approach to controller design for hybrid systems," *Proceedings of the IEEE*, vol. 88, pp. 949–970, July 2000.

[33] U.S. Department of Transportation, *Intrduction to TECAS II Version 7.1*. FAA, 2011.

[34] S. Wollkind, J. Valasek, and T. R. Ioerger, "Automated conflict resolution for air traffic management using cooperative multiagent negotiation," in *Proceedings AIAA Guidance, Navigation, and Control Conference*, vol. 2, (Providence, RI), pp. 1078–1088, August 2004.

[35] D. Sislak, M. Rehak, M. Pechoucek, D. Pavlicek, and M. Uller, "Negotiation-based approach to unmanned aerial vehicles," in *IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications (DIS 2006)*, pp. 279–284, June 2006.

[36] J. C. Hill, F. R. Johnson, J. K. Archibald, R. L. Frost, and W. C. Stirling, "A cooperative multi-agent approach to free flight," in *Proceedings of the International Conference on Autonomous Agents*, (NY, USA), pp. 1205–1212, 2005.

[37] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *Robotics & Automation Magazine*, vol. 4, pp. 23–33, March 1997.

[38] D. Shim and S. Sastry, "A situation-aware flight control system design using real-time model predictive control for unmanned autonomous helicopters," in *AIAA Guidance, Navigation, and Control Conference 2006*, vol. 2, (Keystone, Colorado), pp. 855–862, 2006.

[39] D. Shim, H. Chung, and S. Sastry, "Conflict-free navigation in unknown urban environments," *Robotics & Automation Magazine*, vol. 13, pp. 27–33, September 2006.

[40] D. Shim, H. Kim, and S. Sastry, "Decentralized nonlinear model predictive control of multiple flying robots," in *Proceedings. 42nd IEEE Conference on Decision and Control*, vol. 4, pp. 3621–3626, December 2003.

[41] T. Dong, X. H. Liao, R. Zhang, Z. Sun, and Y. Song, "Path tracking and obstacles avoidance of UAVs - fuzzy logic approach," in *The 14th IEEE International Conference on Fuzzy Systems, 2005. FUZZ '05.*, pp. 43–48, May 2005.

[42] K. Sigurd and J. How, "UAV trajectory design using total field collision avoidance," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, (Austin, TX), August 2003.

[43] A. L. Smith and F. G. Harmon, "UAS collision avoidance algorithm minimizing impact on route surveillance," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, (Chicago, Illinois), 2009.

[44] J. Goss, R. Rajvanshi, and K. Subbarao, "Aircraft conflict detection and resolution using mixed geometric and collision cone approaches," in *Collection of Technical Papers - AIAA Guidance, Navigation, and Control Conference*, vol. 1, (Providence, Rhode Island), pp. 670–689, 2004.

[45] A. Alsaab and R. Bicker, "Behavioral strategy for indoor mobile robot navigation in dynamic environments navigation in dynamic environments," *International Journal of Engineering Science and Innovative Technology (IJESIT)*, vol. 3, pp. 533–542, January 2014.

[46] V. Dobrokhodov, I. Kaminer, K. Jones, and R. Ghabcheloo, "Vision-based tracking and motion estimation for moving targets using small UAVs," in *American Control Conference, 2006*, June 2006.

[47] J. R. Derek Ebdon, *White Paper: Sense-and-Avoid Requirement for Remotely Operated Aircraft (ROA)*. HQ ACC/DR-UAV SOM, June 2005.

[48] E. A. Euteneuer and G. Papageorgiou, "UAS insertion into commercial airspace: Europe and US standards perspective," in *IEEE/AIAA 30th Digital Avionics Systems Conference (DASC)*, pp. 5C5–1–5C5–12, October 2011.

[49] Safety Regulation Group, "Unmanned Aircraft System Operations in UK Airspace - Guidance," Tech. Rep. CAP-722, Civil Aviation Authority, West Sussex, UK, August 2012.

[50] T. Dubot, "Integrating civil unmanned aircraft operating autonomously in non-segregated airspace: towards a dronoethics?," in *Rights and Duties of Autonomous Agents*, vol. 885, (Montpellier, France), 2012.

[51] O. Shakernia, W.-Z. Chen, S. Graham, J. Zvanya, A. White, N. Weingarten, and V. M. Raska, "Sense and avoid (SAA) flight test and lessons learned," in *AIAA Infotech@Aerospace 2007 Conference and Exhibit*, (Rohnert Park, CA), pp. 1–15, AIAA, May 2007.

[52] I. Cowling, *Towards Autonomy of a Quadrotor UAV*. PhD thesis, Cranfield University, School of Engineering, 2008.

[53] O. A. Yakimenko, "Direct method for rapid prototyping of near-optimal aircraft trajectories," *Journal of Guidance, Control, and Dynamics*, vol. 23, pp. 865–875, Sept. 2000.

[54] G. Basset, Y. Xu, and O. Yakimenko, "Computing short-time aircraft maneuvers using direct methods," *Journal of Computer and Systems Sciences International*, vol. 49, no. 3, pp. 481–513, 2010.

[55] C. K. Lai and J. Whidborne, "Real-time trajectory generation for collision avoidance with obstacle uncertainty," in *Guidance, Navigation, and Control and Co-located Conferences*, American Institute of Aeronautics and Astronautics, Aug. 2011.

[56] A. Berry, *Continuous Local Motion Planning and Control for Unmanned Vehicle Operation within Complex Obstacle-Rich Environments*. PhD thesis, University of Leicester, 2010.

[57] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *Systems Science and Cybernetics, IEEE Transactions on*, vol. 4, pp. 100–107, July 1968.

[58] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of Mathematics*, vol. 79, pp. 497–516, July 1957.

[59] G. Lafferriere and H. J. Sussmann, "A differential geometric approach to motion planning," in *Nonholonomic Motion Planning* (Z. Li and J. Canny, eds.), vol. 192 of *The Springer International Series in Engineering and Computer Science*, pp. 235–270, Springer US, 1993.

[60] O. Pettersson and P. Doherty, "Probabilistic roadmap based path planning for an autonomous unmanned helicopter," *Journal of Intelligent and Fuzzy Systems*, vol. 17, no. 4/2006, pp. 395–405, 2006.

[61] L. Piegl and W. Tiller, *The NURBS Book*. Springer Verlag, 1997.

[62] L. Piegl, "On NURBS: a survey," *Computer Graphics and Applications*, vol. 11, pp. 55–71, January 1991.

[63] I. D. Cowling, O. A. Yakimenko, J. F. Whidborne, and A. K. Cooke, "Direct method based control system for an autonomous quadrotor," *Journal of Intelligent & Robotic Systems*, vol. 60, pp. 285–316, 2010.

[64] H. Sira-Ramírez and S. K. Agrawal, *Differentially Flat Systems*. Marcel Dekker, 2004.

[65] B. L. Stevens and F. L. Lewis, *Aircraft Control and Simulation*. John Wiley, second edition ed., 2003.

[66] M. V. Cook, *Flight Dynamics Principles*. Elsevier, 2nd ed., 2007.

[67] V. Becerra, "Solving optimal control problems with state constraints using nonlinear programming and simulation tools," *Education, IEEE Transactions on*, vol. 47, pp. 377–384, Aug 2004.

[68] G. J. Holland, T. McGeer, and H. Youngren, "Autonomous Aerosondes for economical atmospheric soundings anywhere on the globe," *Bull. Amer. Meteor. Soc.*, vol. 73, pp. 1987–1998, 1992.

[69] EASA, "Certification specifications for normal, utility, aerobatic, and commuter category aeroplanes CS-23," tech. rep., EASA, 2012.

[70] I. Postlethwaite, J. F. Whidborne, G. Murad, and D.-W. Gu, "Robust control of the benchmark problem using H_∞ methods and numerical optimization techniques," *Automatica*, vol. 30, no. 4, pp. 615 – 619, 1994.

[71] N. V. Dakev, J. F. Whidborne, A. J. Chipperfield, and P. J. Fleming, "Evolutionary H_∞ design of an electromagnetic suspension control system for a maglev vehicle," *Proc. IMechE J. Syst. Contr. Eng.*, vol. 311, no. 4, pp. 345–355, 1997.

[72] L. Huo and L. Baron, "The self-adaptation of weights for joint-limits and singularity avoidances of functionally redundant robotic-task," *Robotics and Computer-Integrated Manufacturing*, vol. 27, no. 2, pp. 367 – 376, 2011.

[73] R. Toro, C. Ocampo-Martinez, F. Logist, J. F. Van Impe, and V. Puig, "Tuning of predictive controllers for drinking water networked systems," in *the 18th IFAC World Congress*, vol. 18, (Universita Cattolica del Sacro Cuore, Milano, Italy), pp. 14507–14512, 2011.

[74] C. F.Spence, *AIM/FAR 2008: Aeronautical Information Manual/Federal Aviation Regulations*. McGraw-Hill Professional, 2007.

[75] J. Clemens, *Portable Collision Avoidance System (PCAS) Model XRX Owners Manual*. Zaon Flight Systems, Inc., Frisco, Texas, USA.

[76] K. Yang, Y. Kang, and S. Sukkarieh, "Adaptive nonlinear model predictive path-following control for a fixed-wing unmanned aerial vehicle," *International Journal of Control, Automation and Systems*, vol. 11, no. 1, pp. 65–74, 2013.

[77] H. R. Martens, "A comparative study of digital integration methods," *SIMULATION*, vol. 12, no. 2, pp. 87–94, 1969.

[78] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, pp. 338–353, 1965.

[79] F. M. McNeill and E. Thro, *Fuzzy Logic: A Practical Approach*. AP Professinal, 1994.

[80] E. H. Mamdani, "Application of fuzzy logic to approximate reasoning using linguistic synthesis," *IEEE Transactions on Computers*, vol. C-26, pp. 1182–1191, December 1977.

[81] E. Cox, "Fuzzy fundamentals," *Spectrum, IEEE*, vol. 29, pp. 58–61, October 1992.

[82] J. Mendel, "Fuzzy logic systems for engineering: a tutorial," *Proceedings of the IEEE*, vol. 83, pp. 345–377, March 1995.

[83] U.S. Department of Transportation, *Airplane Flying Handbook*. FAA, 2004. FAA-H-8083-3A.

[84] A. M. Turing, "Computing machinery and intelligence," *Mind*, vol. LIX, no. 236, pp. 433–460, 1950.

[85] L. A. Young, J. A. Yetter, and M. D. Guynn, "System analysis applied to autonomy: Application to high-altitude long-endurance remotely operated aircraft," in *In AIAA Infotech@Aerospace Conference*, (Arlington, VA), 2005.

[86] L. A. Young, "Feasibility of Turing-style tests for autonomous aerial vehicle intelligence," in *AHS International Specialists Meeting Unmanned Rotorcraft: Design, Control and Testing*, (Chandler, AZ), 2007.

[87] E. L. Duke, C. C. Vanderpool, and W. C. Duke, "A Turing test for UA operations: Defining operational requirements," in *U.S. Air Force T&E Days*, (Destin, Florida), 2007.

[88] E. L. Duke, C. C. Vanderpool, and W. C. Duke, "Turning PINOCCHIO into a real boy: Satisfying a turing test for ua operating in the nas," in *Collection of Technical Papers-2007 AIAA InfoTech at Aerospace Conference*, vol. 2, pp. 1675–1690, 2007.

[89] S. F. Kalik and D. V. Prokhorov, "Automotive Turing test," in *Workshop on Performance Metrics for Intelligent Systems (PerMIS 07)*, 2007.

Appendix A

Least Squares Bezier Curve-fit

A least squares curve-fit minimises the square of the difference between the data to be fitted (y) and a model of that data (p) as shown below:

$$S = \sum_{i=1}^{m} (y_i - P_i(C))^2 \tag{A.1}$$

where
S: sum of the residual error,
m: number of data point to be fitted,
y_i: data to perform curve-fit to at position i,
$P_i(C)$ estimate of data at position i using a polynomial data model with design C, given by:

$$P_i(C) = \sum_{j=1}^{n} C_j B_{j,i} \tag{A.2}$$

where
$B_{j,i}$: value of the j^{th} Bezier basis function at position i in the curve,
C: design coefficients.

If the model being used to describe the data is linear, as with a Bezier curve, then a closed form solution is available. This solution is calculated as follows:
S is a function of the design vector C and can be minimised by setting it's partial derivatives with respect to C_i to zero:

$$\frac{\delta S}{\delta C_j} = 0 \quad for \quad j = 1, 2, \cdot, n \tag{A.3}$$

now,

$$\frac{\delta S}{\delta C_j} = \frac{\delta}{\delta C_j}(\sum_{i=1}^{m}(y_i^{actual} - \sum_{j=1}^{n} C_j B_{j,i})^2) = 2\sum_{i=1}^{m}(y_i^{actual} - \sum_{j=1}^{n} C_j B_{j,i})(-B_{i,j}) \tag{A.4}$$

Therefore the conditions for S being minimised can be written as:

$$\frac{\delta S}{\delta C_j} = -2B_{i,j}\sum_{i=1}^{m}(y_i^{actual} - \sum_{j=1}^{n} C_j B_{j,i}) = 0 \tag{A.5}$$

This can be re-arranged to give:

$$(B^T B)C = B^T Y \tag{A.6}$$

Therefore the design vector that minimises the least squares difference between a Bezier curve and a given data set (y) can be calculated from:

$$C = (B^T B)^{-1} B^T Y \tag{A.7}$$

Note that the term least squared

$$(B^T B)B^T = B_{ls} \tag{A.8}$$

is fixed by the Bezier basis functions and the chosen resolution of the curve, therefore this term only needs to be calculated once, at the start of a simulation. The Bezier curve that best fits a given data set Y in a least squares sense can therefore be calculated on-line by a single matrix-vector multiplication:

$$C = B_{ls} Y \tag{A.9}$$

Appendix B

Summery of Extended Interviews and Discussions with a Pilot

Extended interviews and discussions about the problem have been carried out with Mrs Susan Szasz who is a pilot at National Flying Lab, Cranfield University. She was helpful in giving her experience and advice that were useful in achieving this thesis. The following table summarises the topics that discussed in the sessions of the interviews.

Session	Date	Discussed Topics
Session 1	18/2/2014	• General discussion about the rules of the air. • The rules of the air in different airspace classes. • Conflict scenarios.
Session 2	1/3/2014	• Conflicts detection. • Conflicts scenarios recognition. • Introduction to conflict avoidance manoeuvres for different conflict scenarios.
Session 3	25/3/2014	• Introduction to head-on conflict scenarios. • Avoidance manoeuvres for the head-on conflict scenarios (right/left manoeuvres). • The factors that affect the pilot's decision when initiating avoidance manoeuvres.
Session 4	15/4/2014	• Introduction to the approaching conflict scenarios. • Avoidance manoeuvres for approaching conflict scenarios where the pilot performs right/left avoidance manoeuvre. • Avoidance manoeuvres for approaching conflict scenarios where the pilot performs full circle avoidance manoeuvre.
Session5	29/4/2014	• Discuss in more detail the factors that affect the pilot behaviour when initiating the conflict avoidance manoeuvre (Time to collision T_c and aircraft speed V). • Discuss the effect of the time to collision T_c on the roll rate when initiating a conflict avoidance manoeuvre. • Discuss the effect of the aircraft speed V on the roll rate when initiating a conflict avoidance manoeuvre.

Mrs Susan Szasz Confirmation

Sign: Susan Szasz

Date: 13/11/14.