Faculty of Electrical Engineering, Mathematics and Computer Science
Microwave-transmission, Radar and Remote Sensing Technology (MTS Radar) Group

Simulation of a Detect and Sense System

G. Berghuis

March 2010

Committee members:
Prof.dr.ir. Erik Theunissen (Supervisor)
Prof.dr. O. Yarovoy
Dr.ir. J.H. Weber

Abstract

Under current regulations, Unmanned Aerial Vehicles (UAVs) are prohibited from operating in non-segregated airspace. The absence of a pilot is considered to be a potential hazard because the pilot is responsible for the separation with other aircraft. In unmanned aviation the pilot, or operator, can never detect possible separation conflicts without the help of Detect, Sense and Avoid (DSA) systems. To make operation in civil airspace possible, there is a need for certified DSA systems. As of yet there are no certified systems available because the requirements for the certification are not yet developed. To help develop these requirements for these systems, research efforts experiment with these systems in both flight-tests and simulations.

One such a research effort is made by the Delft University of Technology (DUT) together with the Netherlands Defence Academy (NLDA). They use a UAV ground station simulator to investigate the 'Operator-in-the-Loop' concept: the DSA system does not autonomously take action, but the operator is alerted when a possible threat is detected and asked to verify the threat and to decide what action to take. The research presented in this thesis is performed in order to increase the fidelity of the DSA models used in this simulator.

DSA systems use sensors to detect potential threats. This project focused on the simulation of the sensors used for detecting non-cooperative traffic. On the basis of a study of the literature the three most commonly used sensors were selected: radar, electro-optical (EO) and infrared (IR). After the subsystems and interfaces were defined the subsystems were designed in a two step iterative approach. The final radar simulation is a link-budget model and uses the Swerling algorithm for the detection calculation. The EO and the IR simulation system use a simplified geometric aircraft model to calculate the static and dynamic signature on the sensor image. The detection calculation is performed by weighing and combining the signatures and comparing them to a threshold value. All designed systems are verified with data obtained from other research projects.

Acknowledgments

After hours of reading literature and writing Matlab scripts I have arrived at the end of my thesis work, and with that my student years in Delft. The time in Delft I spent with great pleasure, this was not possible without a lot of people. I thank everybody I have worked with in Delft, I want to name some people specifically.

First and foremost I thank my supervisor, Prof.dr. Erik Theunissen, for his patience and the excellent advise and guidance he constantly provided. I also owe Ir. Joris Koeners gratitude for his willingness to answer all questions and the social talks we had over coffee.

I wish to thank my parents for their moral support, and in particular my father for the time he put into editing the text making it more readable and understandable. Also for their moral support and the many hours I spent with them I thank my sister, brother in law and of course my little cousin and niece.

For their constant supply of coffee and other social and informative activities I thank everybody at the study association ETV and at our graduation room. Without them the time I spent studying might have been shorter, but a lot less fun.

Geert Berghuis
Delft, March 2010

When the first version of this report was complete and submitted for review by the committee, I noticed that the women of the secretariat were always very willing to help with organizing the final steps necessary for my graduation. For this I thank Stefanie van Gentevoort, Dominique Meijer and Marjon Verkaik-Vonk very much.

Geert Berghuis
Delft, April 2010

Contents

List of Figures

List of Tables

List of Abbreviations

ADS-B	Automatic Dependent Surveillance-Broadcast
AFRL	Air Force Research Lab
CDTI	Cockpit Display of Traffic Information
DS	Detect and Sense
DSA	Detect, Sense and Avoid
DUT	Delft University of Technology
EKF	Extended Kalman Filter
ELOS	Equivalent Level of Safety
EO	Electro-Optical
FAA	Federal Aviation Administration
FOV	Field of View
FPGA	Field Programmable Gate Array
HWIL	Hardware-in-the-Loop
IR	Infrared
LAT	Latitude
LON	Longitude
LWIR	Long Wave Infrared
MIDAS	Multifunction Infrared Distributed Aperture System
NLDA	Netherlands Defence Academy
pdf	Probability Density Function
RCS	Radar Cross Section
SNR	Signal-to-Noise Ratio
SSR	Secondary Surveillance Radar
TCAS	Traffic Collision Avoidance System
UAV	Unmanned Aerial Vehicles

1. Introduction

1.1. Unmanned Aerial Vehicles and rules and regulations

An Unmanned Aerial Vehicle (UAV) is an aircraft that has no pilot on board, it is usually controlled by a pilot or operator on the ground or it can fly autonomously. Today UAVs are primarily used by defense forces around the world as reconnaissance or combat aircraft because they are cheaper and the consequences are less grave when they are shot down with respect to their manned counterparts.

In the future UAVs are expected to be deployed more frequently in civil airspace by both military and civil organizations. Because there were no provisions for unmanned aircraft in the rules and regulations applicable to aircraft, regulatory agencies like the American Federal Aviation Administration (FAA)[1] are trying to develop rules to fit these types of operations into the civil airspace. UAVs can for example not comply with rule 14 CFR 91.113b which deals with the see-and-avoid requirements, it reads: "When weather conditions permit, regardless of whether an operation is conducted under instrument flight rules or visual flight rules, vigilance shall be maintained by each person operating an aircraft so as to see and avoid other aircraft." [1]. The FAA therefore will only allow UAVs in their airspace if they adhere to rules which require measures to assure the UAV have adequate see-and-avoid capabilities [2]. The term Equivalent Level of Safety (ELOS) was introduced meaning that the UAV needs to be able to at least match the see-and-avoid capabilities of manned aircraft. A result of this rule is that when operating outside of class A, restricted or military airspace a chase plane is required to accompany the UAV to perform the see-and-avoid task.

1.2. Detect, Sense and Avoid

The requirement of an assisting chase plane is a consequence of the lack of certified Detect, Sense and Avoid (DSA) systems. These systems are designed to do the sense-and-avoid task described earlier. Implementations can range from fully autonomous systems to simple video links to the operator at the ground station.

Every DSA system needs sensors to sense targets[2] that may form a risk to the UAV. The Detect part of the system identifies one or more targets from the sensor measurements, this can be the human operator or automated detection methods. Finally, if the target is identified measures need to be taken to avoid a loss of separation or even a collision, usually the flightpath of the UAV is adapted. A typical block-diagram of a DSA system is given in Figure 1.1. Some systems will have the sensor fusion performed before the detection algorithm.

[1] The FAA uses the term Unmanned Aircraft System (UAS)

[2] A target or intruder is an object that is in or near the flightpath of the ownship and can or should be detected by the DSA system, both terms are used in literature and both will be used in this document with the same meaning

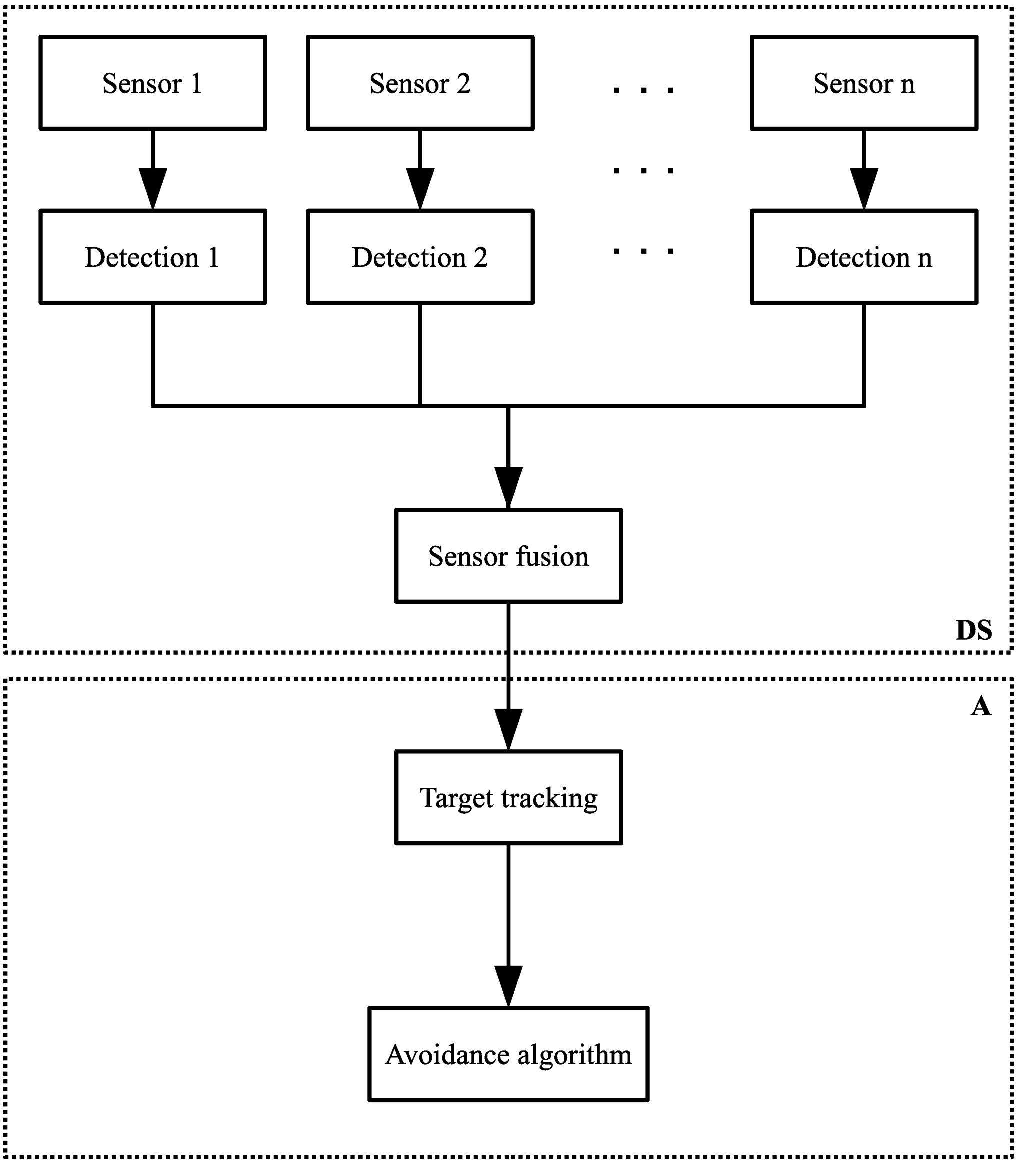


Figure 1.1: Typical DSA system block diagram

These types of systems are already in use in manned aviation. Currently the Traffic Collision Avoidance System (TCAS) is being used in a large number of commercial aircraft. This system detects possible intruders by the responses they send to the secondary surveillance radar (SSR). If an intruder is detected that is or will be to close to the flightpath, the pilot will be warned and a traffic or resolution advisory can be given. When Automatic Dependent Surveillance-Broadcast (ADS-B) becomes the standard, the DSA task can be fulfilled by this system. The drawback is, however, that these type of systems can only detect cooperative aircraft, so this can not form a complete solution: other complementing techniques need to be used. Although the TCAS system is proven technology and certified for use in civil airspace, it is not suitable for use with UAVs. The design of the TCAS system is based on performance characteristics of the (manned) aircraft the system was expected to be used in, UAVs do not necessarily match their performance [3], so the resolution advisories cannot be used for the collision avoidance but the traffic advisories can be used for the detection.

This research project focuses on the Detect and Sense part of a DSA system aimed at detecting cooperative as well as non-cooperative intruders using non-cooperative systems. Non-cooperative intruders are objects that have no working system that assists in the detection of their possible intrusions.

1.3. Simulation programs

For the authorities to certify DSA systems, these systems need to be tested thoroughly. A good and economic way to test and develop these systems is by the use of simulation. Of course, empirical data are necessary to verify the simulation fidelity in the end, but real experiments take a lot of time to organize and cost a lot of money. Above that, some scenarios cannot be performed in experiments because systems are not operational yet, or due to safety reasons, for example when testing a DSA system in a near miss scenario.

There are, in effect, very few simulation systems that can be used to evaluate different DSA system implementations. That is why the Delft University of Technology (DUT) together with the Netherlands Defence Academy (NLDA) created a UAV research simulator [4]. This simulator has a front end which is designed to mimic a ground station where an operator can be presented with all types of scenarios. With this simulator research is conducted on a so called 'Operator-in-the-loop DSA system'. Chapter 2. will describe this and other systems in more detail.

1.4. Goal of the thesis

The goal of this research project is to increase the fidelity of the existing DUT/NLDA simulation effort with regards to the detection function.

This research will be divided into four steps which will help in the achievement the goal:

1. A literature review analyzing the different implementation options of the desired function. Using the information found, the system boundaries and a set of rating criteria will be identified.

2. An analysis of the current simulation effort as used in the DUT/NLDA simulator. A decision will be made whether the used simulation systems will serve as a foundation for the improvements or whether other solutions will be used as the basis for the system design.
3. A decision on what simulation methods will be used in this research and the design and implementation of the chosen methods.
4. An evaluation of and a discussion on the results of the implementation. With the help of the defined rating criteria the results of both the subsystems and the total system a conclusion can be drawn on whether or not the goal is achieved.

1.5. Organization of the document

Chapter 2 describes the existing research programs, including the current approach on which this research project is based, and their technical implementations. Also the requirements and verification of the system that is to be developed will be discussed. In Chapter 3 the design of the system will be given and the implementation will be explained. Chapter 4 evaluates the test results of the implementation and on that basis refinements to the system will be introduced. The conclusion and recommendations follow in Chapter 5.

2. Current systems and requirements

Before designing a simulation system several aspects need to be considered. An important part of the overall system in this study is the is the choice on the type of sensors that are to be simulated. In order to make a well-founded choice on the sensors need to incorporate in the system, a literature study has been done. This chapter describes the types of sensors that are used in DSA systems and will offer an overview of existing simulation and verification projects in this field. On the basis of these projects quality criteria will be identified which have been used to design and verify the final system.

First the relevant sensors will be described and a choice will be made which will be investigated further. Next the attention is focused on the literature on those sensors and finally an overview of the found and used data are presented.

2.1. Sensors

The number of types of sensors used in DS systems is limited. In [5] Zeitlin gives an overview of the sensors commonly used and their strengths and weaknesses. To summarize, there are four non-cooperative sensors that are most commonly used in DS applications. Table 2.1 gives an overview.

Table 2.1: Non-cooperative DS sensor characteristics

Sensor	Strength	Weakness
Radar	Accurate range	Inaccurate azimuth, elevation
Electro-optical (EO)	Accurate azimuth, elevation	No range[3], bad performance under adverse weather conditions and sensitive to intruder color
Infrared (IR)	Accurate azimuth, elevation Exhaust gases improve detection probability	No range[3], bad performance under adverse weather conditions
Laser	Accurate range	Limited FOV

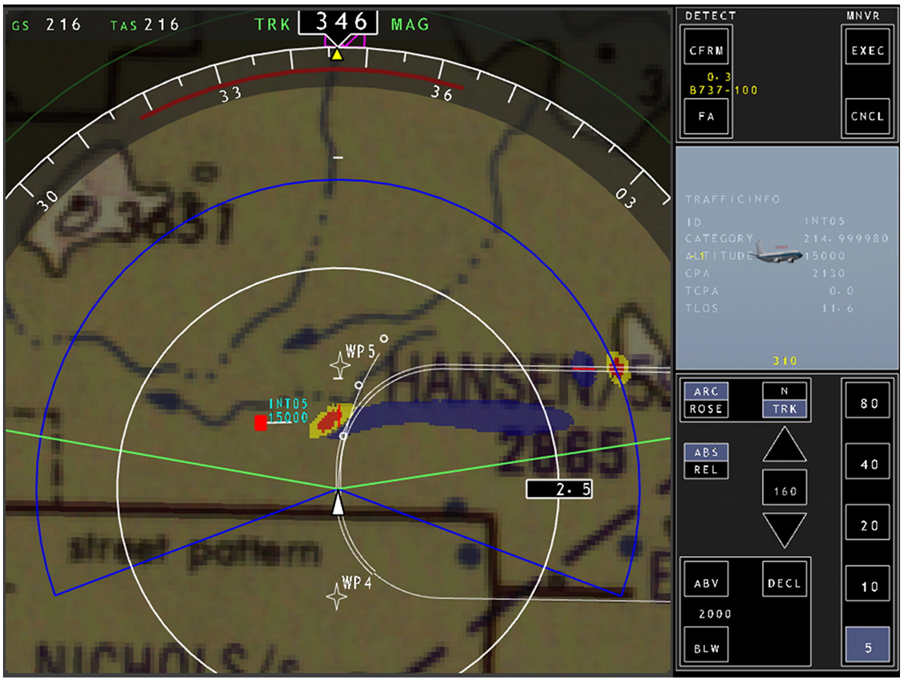
As the table shows, the radar and EO sensors complement each other very well. For this reason most UAVs have at least these two sensors. Infrared systems are considered to be a type of EO system, but the parameters on which detection probabilities of this sensor depend differ so much from the EO type that they have to be researched separately.

[3] There are ways of extracting range information from the images, the PaRCA algorithm will be mentioned in Section 2.4., but also the use of stereo images or knowledge of the geometry of the target can be used

The simulation effort in this research project will focus on the Radar, EO and IR sensor systems as they are widely judged in literature as the most suitable sensors for DSA systems.

2.2. Delft University of Technology and Netherlands Defence Academy

The simulator developed by Delft University of Technology (DUT) and the Netherlands Defence Academy (NLDA) is aimed at researching DSA using an 'Operator-in-the-Loop' concept [4]. DSA systems cannot guarantee perfection, there always has to be a trade-off between false alarm and missed detection probabilities. Therefore this project suggests to include the operator in the loop to help the system identify threats and decide on the resolution. The setup is that an operator is provided with a simulated ground station offering configuration control and information on navigation and flightpath. When the DS system detects an intruder, the operator is alerted and asked to verify the threat. Next the operator or the DSA system can, if deemed necessary, take avoiding action. Figure 2.1 shows the Cockpit Display of Traffic Information (CDTI) the operator is presented with. The blue line indicates the detection range of the EO system and the green line that of the radar system. When an intruder (red square) is detected, the area where the loss of separation occurs is highlighted. The part of the EO image stream where the intruder is detected is extracted and shown to the operator on the right side to make verification of the threat possible.



TRAFFICINFO
ID INT05
CATEGORY 214.999980
ALTITUDE 15000
CPA 2130
TCPA 0.0
TLOS 11.6

Figure 2.1: Screen dump of the CDTI display as shown to the operator

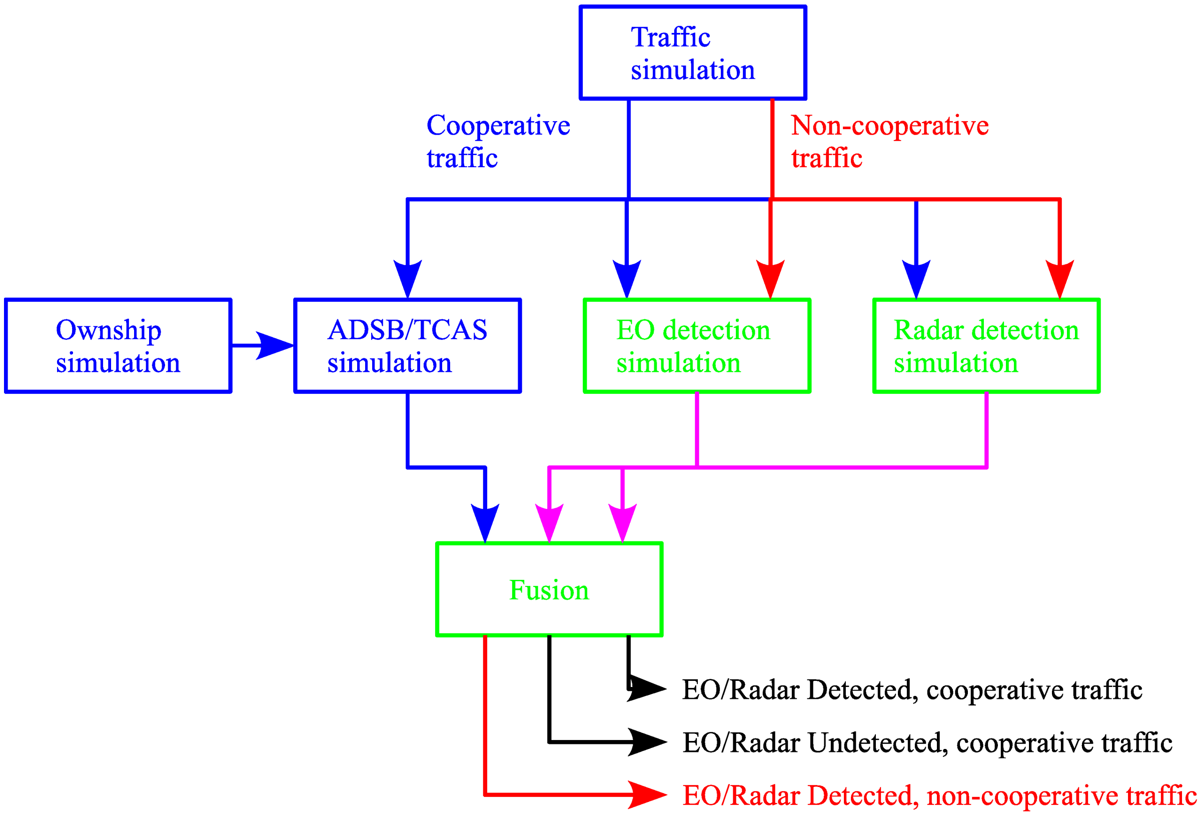

Traffic simulation
Cooperative traffic
Non-cooperative traffic
Ownship simulation
ADSB/TCAS simulation
EO detection simulation
Radar detection simulation
Fusion
EO/Radar Detected, cooperative traffic
EO/Radar Undetected, cooperative traffic
EO/Radar Detected, non-cooperative traffic

Figure 2.2: Block diagram of the sensor simulation in the NLDA simulator; taken from [4]

Figure 2.2 shows the place of the sensors in the existing simulator. The part of the system that simulates the sensors is provided with information from the traffic simulator. Static information, such as sensor parameters and information on the characteristics of the traffic, is stored in a central database. After the sensor fusion a vector pointing to the intruder needs to be fed back to the system which alerts the operator.

2.2.1. Design methodology

In [6] the design methodology which is used in developing the UAV simulator is explained. Originally described by J.A. Hazlett in [7] the method provides a structured way of combining different already existing components into one system. He recommends to:

1. *Start to use modeling linkages to tie together the disparate elements that make up our non-system of systems, to begin develop the non-existent interchanges that take advantage of potential synergies.*
2. *Use models and simulations to develop "wrappers" to encapsulate unruly and uncooperative system elements so that they can interact with other elements in the most opportune manner.*
3. *Use simulations as "fillers" or "placeholders" for not-yet-developed system elements, to take the fullest possible advantage of asynchronous system developments, allowing system elements to come "on-line" when they are ready, rather than waiting for the entire system(s) maturation.*

4. *Develop models and simulations that aid in the actual fusion of battle space awareness inputs, acting as translators, interpreters, facilitators, substitutions or imitators/metaphors, where necessary.*

2.2.2. Sensor simulations

In accordance with Hazletts third recommendation the initial sensor simulations are kept simple. The radar sensor is simply simulated by a boundary decision. When an intruder is within a set range the intruder will be declared detected - this is graphically represented in Figure 2.3.

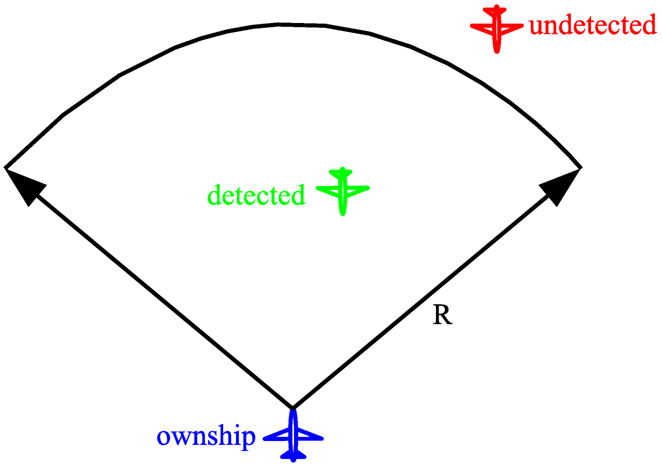


Figure 2.3: Radar simulation as used in the Delft simulator

The EO system retrieves information from the database on size and color of the intruder as well as the sensor parameters. Next a confidence number is calculated by converting the information to a one dimensional space and comparing the contrast between the background and the intruder signature. If the confidence number is large enough, the intruder is declared detected.

2.3. TECVOL

The Italian Aerospace Research Center (CIRA) sponsored a program named TECVOL [8] [9] [10] [11]. This program researches UAV Autonomous Flight Technologies and has the goal to realize a High Altitude Long Endurance unmanned flying laboratory. A component of this program is the development of an autonomous DSA system. This system uses infrared and color cameras as well as a Ka-band pulsed radar.

2.3.1. Sensor simulations

The simulated EO system aims at detecting a target closer than 3 km in distance. The accuracy of this system is set to 0.1° in the angular resolution. In the hardware implementation two infrared as well as two visible spectrum cameras are expected to be used.

Numerical simulations are performed to find the expected performance results. The radar sensor is simulated by a taking an intruder aircraft with a radar cross section of 1 m^2 and calculating the radar detection probability by use of the Swerling algorithm [12]. The navigation data are simulated to be non-ideal by adding a bias and zero mean white Gaussian noise.

2.3.2. Sensor fusion

Based on statistical results of 100 Monte Carlo simulations, the Extended Kalman Filter has been chosen as the best compromise between reliability, computational load and accuracy. Tracking is done only on the basis of the radar sensor - the EO sensor is only used to improve the angular resolution of the detection of the target and therefore only reduces the error in the tracking results.

Using the method of Monte Carlo simulations, different conflict scenarios were used to assess the sensors' performance. The errors in the output vectors are shown in [11].

2.4. Air Force Research Lab

The US military is involved in the development and deployment of UAVs, therefore the Air Force Research Lab (AFRL) has programs which investigate DSA systems [13] [14] [15]. They have conducted hardware-in-the-loop (HWIL) simulations and flight tests to test the performance of the components of the designed system. Together with a TCAS system they fitted passive sensors to a Learjet to verify the expected and simulated system performance. Two manned aircraft were used as intruders, a Beechcraft King Air and a Convair 580. The first represented a slow small type air vehicle and the last a commuter transport type.

2.4.1. Sensor simulations

To achieve a 110 degrees field of view, three EO sensors were placed horizontally and coupled to a FPGA. The FPGA was used for image processing and ultimately for target detection. The total resolution was 2048 by 2048 pixels per sensor, which gives a total resolution of 6144 pixels in the azimuth plane, with an output rate of 20 Hertz. The image processing uses information from the inertial measurement unit to correct the image from the sensors for the vibrations of the aircraft. Radar and long wave infrared (LWIR) sensors were also taken on board, but not much information can be found on the implementation in the overall system.

In an earlier study the EO system detection performance on the basis of an optical flow algorithm was researched. A surrogate UAV in the form of a helicopter recorded video of a Beech Bonanza intruder aircraft in several conflict scenarios. Off-line simulations calculated the probabilities of detection and false positives. To investigate the trade-off between high detection probabilities and high false alarm rates, two detection thresholds were used. The detection probabilities were 100% for the low threshold and the high threshold below 3 NM, beyond 3 NM the detection probability would drop to 90%. The value of the false detection probability was nominally 0.045% and 0.01% for the low and high threshold respectively. Detections beyond 4NM were not considered because the intruder would not be visually detectable on the video (a human pilot would detect the intruder at 1.5NM).

During the flight tests with the Learjet, the EO system first detected the small aircraft at an average of 5.5 NM and the large aircraft at 7.4 NM. The system would declare the intruder at 3.9 NM and 4.9 NM average respectively. The range error of the intruders detected by the EO system were initially about 40% and dropped to about 10% after the execution of the PaRCA algorithm [16].

2.4.2. Sensor fusion

An Extended Kalman Filter (EKF) was used to combine the observations of the non-cooperative and those of the cooperative sensors. The filter weighs the tracks from each sensor on the basis of their known noise characteristics.

To improve the range accuracy of the EO sensor the PaRCA algorithm was used. This algorithm can reduce the error in the range information under good circumstances to about 10% by making the UAV execute a series of maneuvers while tracking the relative movement of the intruder.

A review of the EKF performance is only given for the fusion of the TCAS and EO sensors, a fusion between the radar sensor and the EO sensor is not given.

2.4.3. SeFAR

With AFRL sponsorship Northrop Grumman analyzed the DSA requirements [17] using sensor modeling and simulation tools - one of the sensors they chose was a Multifunction Infrared Distributed Aperture System (MIDAS) that is simulated by using known sensor properties and aircraft parameters. The MIDAS sensor is modeled and a limited amount of simulation results is presented.

Later Northrop Grumman followed up on this research efforts by creating the Sensing for UAV Awareness (SeFAR) program. In that program the company started developing a hardware-in-the-loop integration testbed. The goal is to verify DSA system attributes by implementing real-time algorithms and software in a realistic HWIL testbed.

2.5. Amphitech OASys radar

The former Canadian company Amphitech, now merged into ICx Radar systems, published a development report on their DSA radar [18]. In this report capabilities of different radar technologies implemented in their commercially available DSA radar systems are analyzed. Flight tests performed by the NASA ERAST group used this radar system and determined its performance [19]. In 20 scenarios with seven types of aircraft and various conflict geometries the radar detection range was determined. Figure 2.4 shows the radar system installed on the surrogate UAV aircraft.

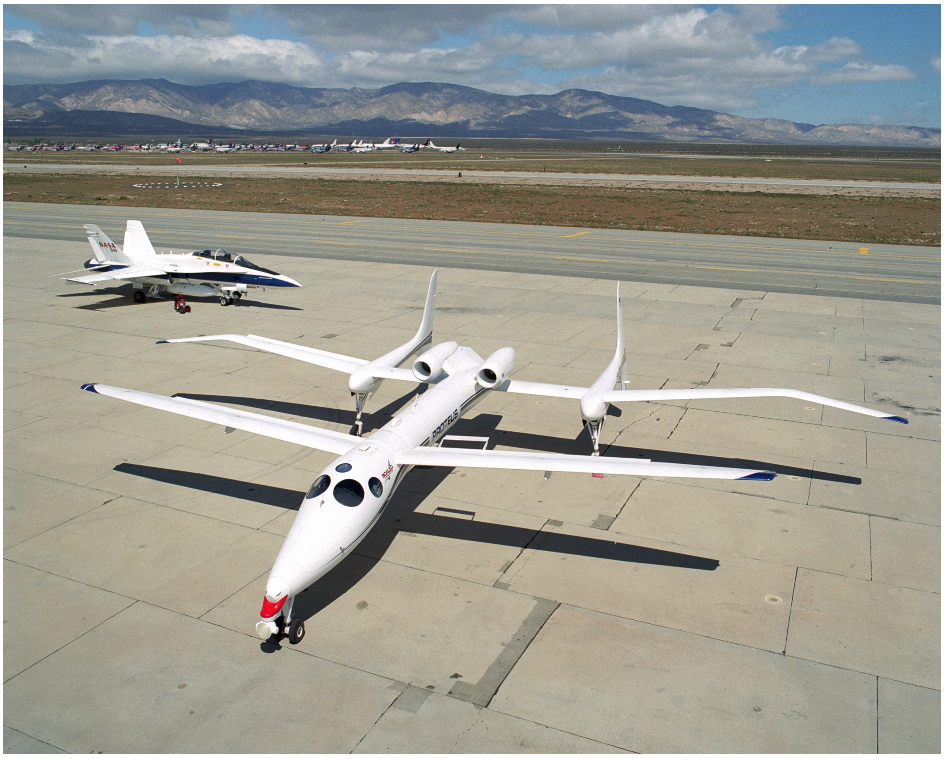


Figure 2.4: The Scaled Composites Proteus Optionally Piloted Aircraft with the OASys radar system under the nose and NASA's F/A-18 used for the flight tests; image courtesy of NASA, www.nasa.gov

The radar parameters are given in Table 2.2. The design of the radar system is based upon the flight tests and was aimed at optimizing the system for UAVs in terms of size, weight, power usage and cost.

Table 2.2: DSA radar sensor parameters

Parameter	**Symbol**	**Value**
Frequency	F	Ka-band, 35GHz
Wavelength	λ	$8.57\cdot 10^{-3}$ m
Maximum detection Range	R_{max}	~15 km
Gain	G	29.5 dB
Noise figure	F	6 dB
Integration gain (coherent)	G_I	7 dB
Power	P	2.5 W
Scan rate		1 Hz
Scan Coverage Azimuth		± 110 degrees
Scan Coverage Elevation		± 20 degrees
5m^2 detection range		5.6 km
Accuracy Az./El.	$\sigma_{x,y}$	1 degree
Accuracy range	σ_R	76.2 m
False alarm rate		min. 0.05, max 0.1
Detection probability	P_D	min. 0.9, max 0.999

2.6. SmartUAV

Under the SmartUAV program [20] [21] [22] Korean researchers focus on radar sensor systems for collision avoidance. A typical radar sensor characteristic was chosen and used in simulations with different collision scenarios. The technology used is a Ka-band pulsed radar and is simulated by use of the Swerling algorithm. For the simulation six types of targets are defined with the radar cross section (RCS) as given in Table 2.3.

Table 2.3: RCS for different obstacles

Type of target	Example target	RCS [dBsm]
Stationary small	Power line	-20
Stationary medium	Building	15
Stationary large	Wooden hill	20
Aircraft small	Trainer	1,76
Aircraft medium	Small fighter	3,01
Aircraft large	Large Fighter	6,99

2.7. System requirements and base of design

From the information found in the presented literature some choices can be made on the starting point of the system design and the requirements for application in the final system. First the system requirements will be discussed and next the design choices will be described.

As described in Section 2.2. the target simulation environment is meant to give the operator an experience that simulates the real world. It is therefore not focused on simulating the different subsystems and sensors with the highest fidelity at all costs. The main goal is to design a system that simulates the behavior of a DS system where a trade-off needs to be made between the (computational) demands and the fidelity. The following three requirements are defined:

1. The system that results from this project needs to be able to run on the hardware used in the target simulation environment.
2. The design of the system should be such that it gives the opportunity of improving the system after this project is finished without having to redesign the whole system.
3. The fidelity of the system needs to be verified - this point will be elaborated upon next.

The predicted level of fidelity is hard to specify, but with the use of the data found in the literature and the choices made on the starting points of the design process the expected performance can be formulated. The system fidelity will be defined by the fidelity of the subsystems, not by the system design. By selecting the source of information for the design of the different subsystems, the corresponding performance numbers can verify the fidelity. Table 2.4 gives an overview of the discussed literature and the type of the relevant data sources found in the literature.

Table 2.4: Data sources to be used for the design and verification of the simulation systems

Sensor	Research project	Data source	Results source
Radar	DUT/NLDA	Range boundary model	Numerical simulation
	TECVOL	Swerling model	Numerical simulation
	Amphitech	**Link budget model (Oasys radar)**	**Link budget analysis**
	Amphitech	**Oasys radar**	**Flight test results**
	SmartUAV	Swerling model	Numerical simulation
Electro-optical	DUT/NLDA	1D projection model	Numerical simulation
	TECVOL	Range boundary model	Numerical simulation
	AFRL	**EO sensors**	**Flight test results**
	AFRL (2)	EO sensors	Flight test results (up to 4NM)
Infrared	TECVOL	Range boundary model	Numerical simulation
	AFRL	IR sensors	Flight test (no data available)
	AFRL (2)	**MIDAS sensor model**	**Numerical simulation**
Sensor fusion	TECVOL	Extended Kalman filter	Numerical simulation
	AFRL	Extended Kalman filter	Flight test (only EO/TCAS)

In Table 2.4 the sources on which the design will be based are highlighted. The radar sensor will be based on the Oasys radar system for which the specifications as listed in Table 2.2, a simple numerical model and flight test results are available. When the system is designed with these specifications the simulation performance is expected to come close to the performance of the real system as found in the NASA flight tests. Most DS research programs that include a radar system simulation use the Swerling algorithm, this algorithm will also be the first choice for this simulation effort. The described implementation focuses on the fidelity of the detection but the false alarms, another important characteristic, are not simulated. False alarms can for example be generated by noise, objects that should not be detected or background clutter. Unfortunately the literature does not describe the false alarms for all three sensors with enough detail to simulate it with an acceptable fidelity. For this reason the simulation of false alarms for all three sensors is left as a recommendation for future research.

The literature does not offer a single favorable solution on the implementation of the EO system simulation. There are however simulation efforts that focus on simulating every part of an EO system to the highest fidelity, the publications of the SPIE (international society for optics and photonics) conference on Optical Modeling and Performance Predictions for example offers intricate simulation methods on both EO and IR systems [23] [24]. The level of detail of these simulation efforts is much too high and the computational demands and development time would exceed the resources available for this project. The TECVOL research uses a very simple model that declares an intruder when it is within a set distance, much like the radar simulation of the current Delft simulator. The current EO simulation in the Delft simulator seems to have a better fidelity than this solution as it takes range, size and color of a target into account. Therefore the current simulation in the UAV simulator is chosen as the basis for the design and improvements will be made to it. With the help of the AFRL research that provides sensor specifications, presented in Table 2.5, and flight test results the design parameters and verification will be based on this research.

Table 2.5: EO sensor characteristics used for design

Resolution azimuth x elevation	6144 x 2048 pixels
Field of view azimuth x elevation	220 x 60 degrees

The literature on IR sensors is less elaborate than the two previously described sensors. Again the TECVOL research has a model that will detect a target within a set distance. The literature on the AFRL research only mentions that IR sensors are used during the flight-tests but does not present their results. That only leaves Northrop Grummans model which takes the static IR signature of a target and relates that to the detection range, it also offers performance data of their MIDAS sensor system. The known specifications of the MIDAS sensor are given in Table 2.6, unfortunately a description of the model is not found in the literature. But when the IR sensor is seen as a type of EO sensor, the simulation system that will be designed for the EO sensor system should also be able to simulate the IR sensor system, albeit with different parameters. Because specific IR simulation methods are lacking, the more developed EO simulation system will be used also for the simulation of the IR sensor system. The specifications and performance data from the MIDAS sensor will be used for the design and verification.

Table 2.6: IR sensor characteristics used for design

Resolution azimuth x elevation	1024 x 1024 pixels
Field of view azimuth x elevation	100 x 100 degrees

3. System overview and implementation

3.1. Introduction

As described in the previous chapter, the system needs to provide the UAV simulator with vectors of the perceived intruders. Keeping in mind the design methodology formulated by Hazlett, the first step is to create a flow diagram in chronological order and specify the linkages. Figure 3.1 gives this diagram for the designed system - from this diagram the behavior and the in- and outputs of the system can be derived. The parts of interest for the final system are the blocks numbered two to six. The traffic simulator is actually part of the existing simulation system, but because of the need for fast and flexible testing a simple traffic simulator is developed for this research.

This chapter will describe the system components as shown in the flow diagram. The numbers in the diagram correspond to the numbering of the subsections of this chapter. Every subsection will state the goal, the in- and outputs of the block and the method of implementation.

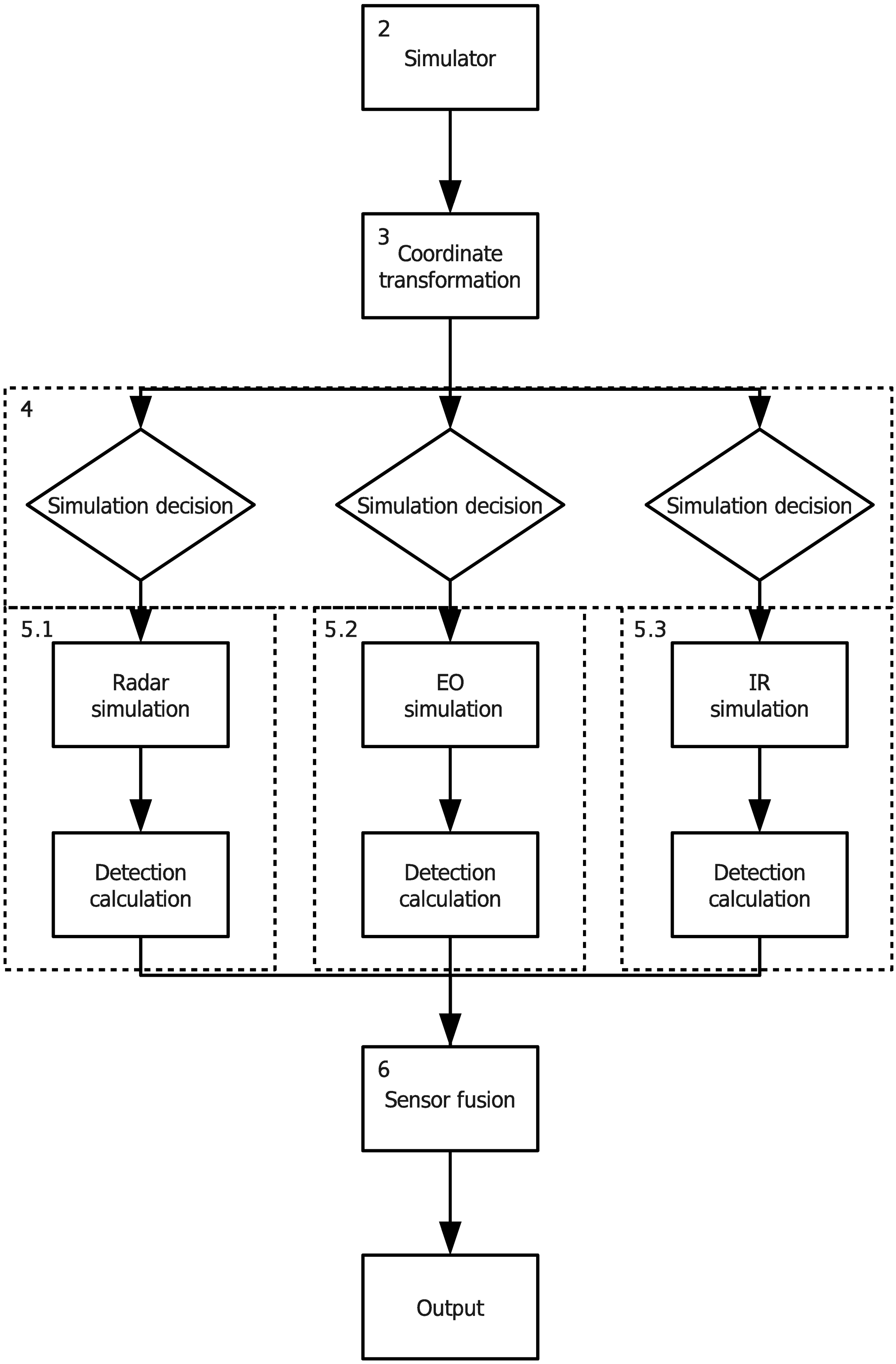
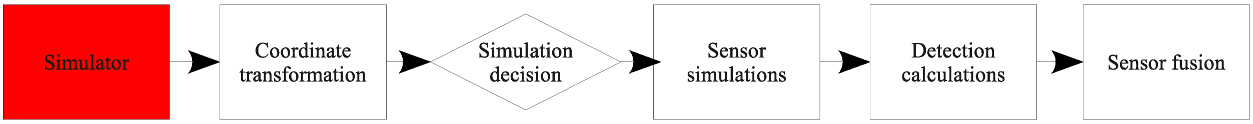


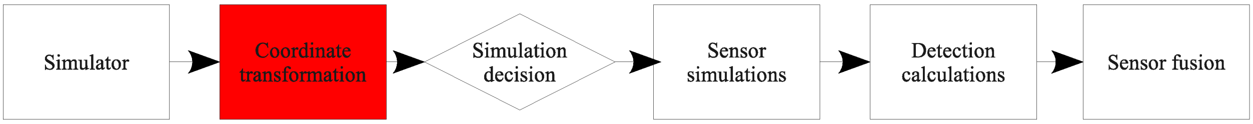
Figure 3.1: Chronological flow diagram

3.2. Simulator



The simulator block provides the system with the position and attitude information of both the ownship and the intruders. It can be configured to generate a certain conflict scenario or load a previously generated one. All information on all aircraft characteristics and vectors is generated and stored in a database before the simulation of the rest of the system starts. When the database is complete the system will run through every simulated time instance and pass the relevant vectors to the next block: the coordinate transform. When the simulation reports its final results, this is stored in the database and the simulator will advance to the next time instance.

3.3. Coordinate transform



The input of the coordinate transform block comes from the simulator in the form of a speed and position vector of the ownship and one intruder. These vectors are given in the form of geodetic coordinates and body angles. As sensors usually give the position of an intruder in the form of bearing and range (and to make the calculations simpler) these vectors are converted to a more convenient form. The output of this subsystem will be two vectors with the origin in the sensor: one containing the polar coordinates of the intruder and the other the attitude of the intruder as seen from the position of the ownship.

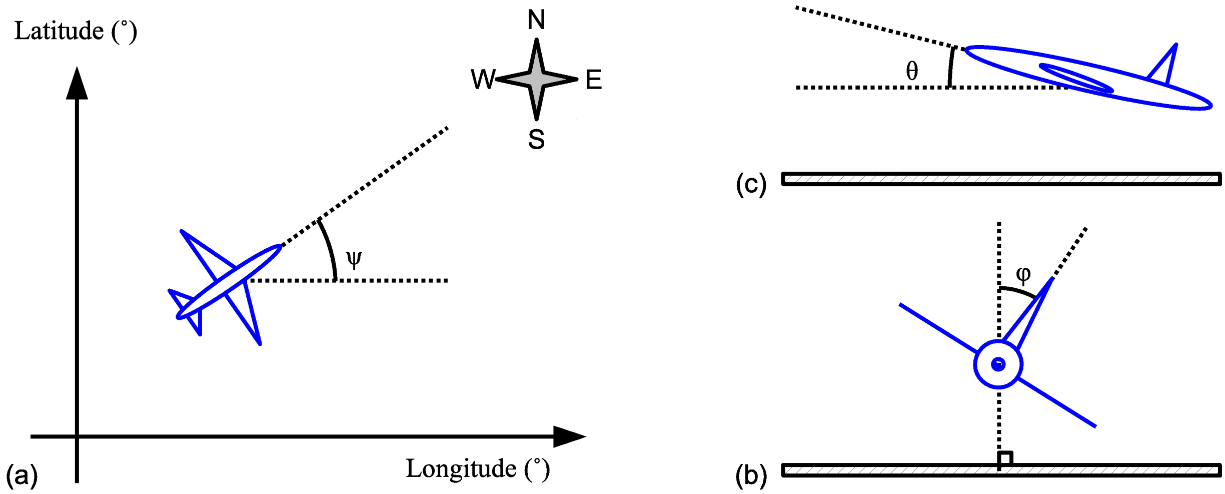


Figure 3.2: Definition of positive angles from (a) top down view, (b) side view and (c) view from back to front

It is essential that the reference angles and coordinates are unambiguous and used consistently - Figure 3.2 gives the definitions of the angles. The location of vehicles is received from the traffic simulator in latitude, longitude and altitude. The attitude is given in the body angles yaw (ψ), pitch (θ) and roll (φ) in the body fixed reference frame with the x-axis pointing North. As the side-slip angle is usually small, this is neglected. When simulating the sensors it is practical and efficient to use relative polar coordinates. The origin of the reference system is placed in the sensor that is simulated and the angles (*p, q*) and range (*R*) to the intruder are calculated. Figure 3.3 gives an overview. For simplicity the sensors are assumed to be in the center of gravity of the aircraft - this will not have a substantial effect because the simulated UAVs are usually not very large.

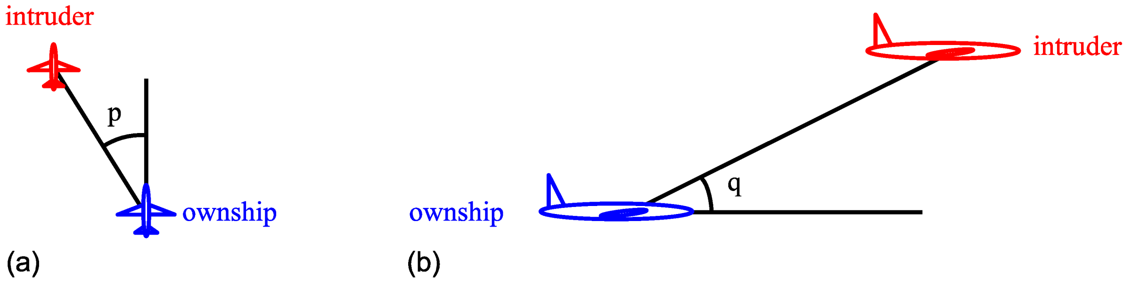


Figure 3.3: Definition of positive angles used for simulation from (a) top down view and (b) side view

To obtain the angles and range the geodetic coordinates must be translated to local North, East, Down coordinates with the ownships' sensors in the origin. First the relative latitude, longitude and altitude of the intruder in degrees is calculated (1).

$$Lat_{relative}=Lat_{inruder}-Lat_{ownship}\ ;\ Lon_{relative}=Lon_{intruder}-Lon_{ownship}\ ;\ h_{relative}=h_{inruder}-h_{ownship} \tag{1}$$

On the basis of the found relative coordinates a first check can be done whether the intruder is far beyond the sensors range.

Next the relative coordinates are converted to distances in meters. Because the simulation is done in a local area, the surface distance per degree is assumed to be constant. The surface distance of one degree on a sphere with a radius of 6,371 kilometers, the mean radius of the earth, is 111,195 meters. This introduces an error of maximum 600 meters per degree if compared to more accurate earth ellipsoid models. To obtain the distances in meters between the ownship and the intruder, a simple multiplication between the relative distances in degrees with 111,195 is performed.

$$X=Lon_{relative}*111195\ ;\ Y=Lat_{relative}*111195*\cos(Lon_{intruder}) \tag{2}$$

Now the parameters *p, q* and the *R* can be found using simple trigonometry functions and correcting for the ownship orientation, this is done in (3) - (5).

$$R=\sqrt{X^2+Y^2+h_{relative}^2} \tag{3}$$

$$p=atan2\left(\frac{Y}{X}\right)-\psi\ ^{4} \tag{4}$$

$$q=asin\left(\frac{h_{relative}}{R}\right)-\theta \tag{5}$$

To aide the sensor simulations in determining what surface is visible to the sensor, relative body angles are defined and calculated. Figure 3.4 shows how to find the relative angles for yaw, pitch and roll. The angles are given in (6). Note that the direction of travel of the ownship is not important because the sensor is assumed to always look right at the target. Also, due to the symmetric nature of the intruder model, the sign of the angle is not relevant: in later calculations this sign will eventually disappear. Again the yaw angle is neglected due to the small angles this parameter takes under normal conditions.

$$\psi_{rel}=\psi_{intruder}-p\,;\theta_{rel}=q-\theta_{intruder}\,;\phi_{rel}=q-\phi_{intruder} \tag{6}$$

[4] The used arctangent variation is called the four-quadrant inverse tangent which calculates results in the interval (-π, π] instead of the standard arctangent function which can only have results in the (-π/2, π/2] interval

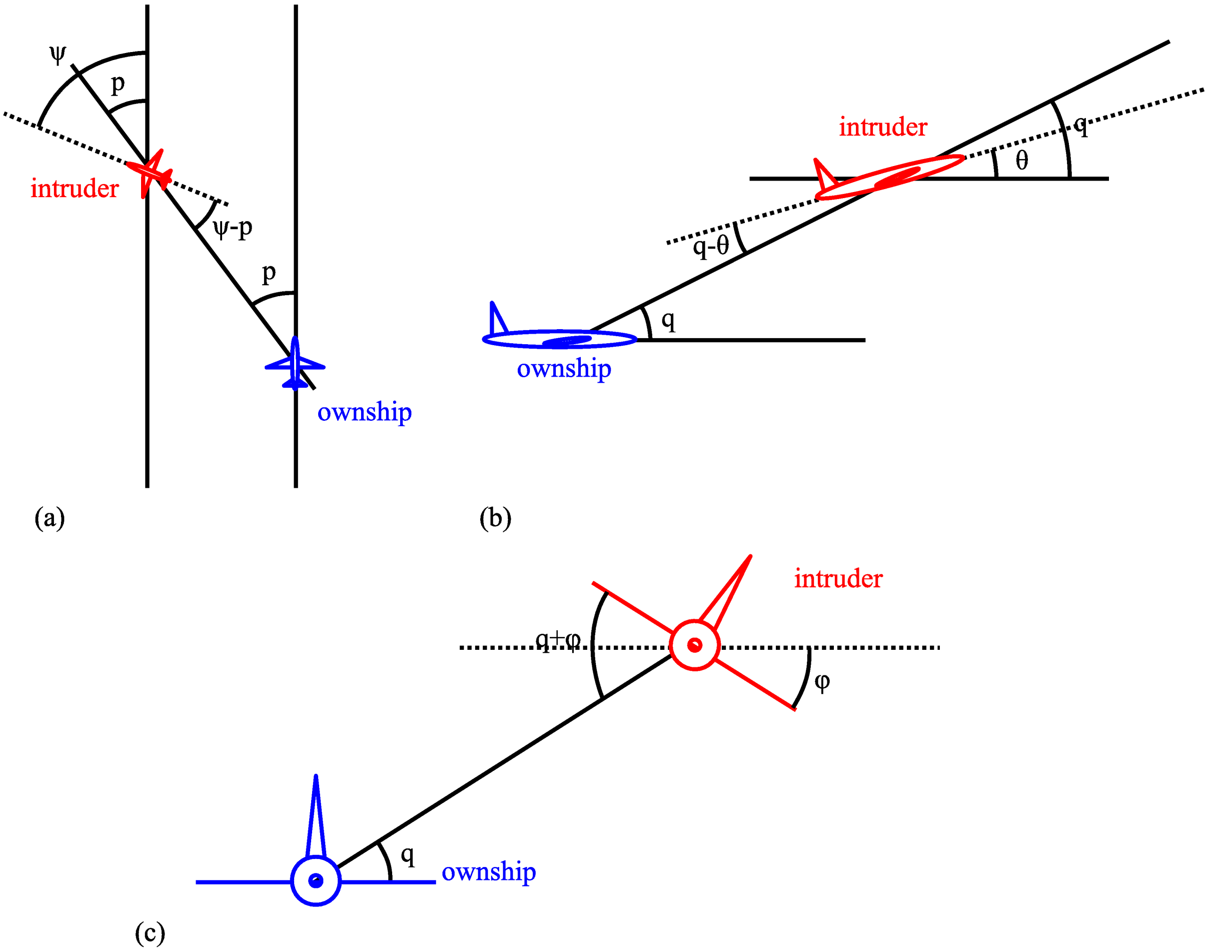
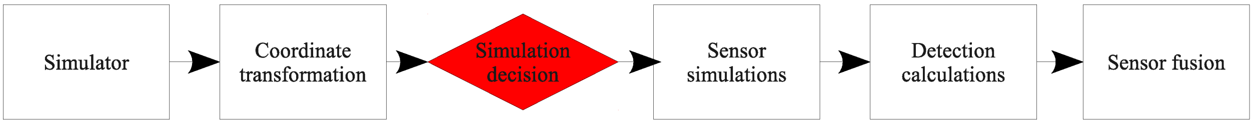


Figure 3.4: Overview of angles from (a) top down, (b) side view and (c) back to front

3.4. Simulation decision



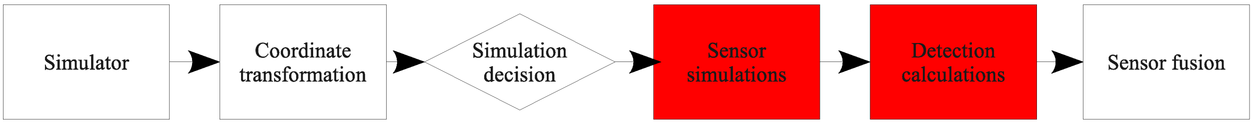
The next step is to determine which sensors need to be simulated. This decision is made for all the sensors, but because the criteria are for all sensors are the same they do not have to be discussed for each separately. There are several reasons why a sensor would not need to be simulated:

1. The user does not want to use the sensor for the simulation.
2. If the interval of scanning of the sensor is lower than the time steps of the simulator, the sensor does not need to be simulated every time step.
3. The intruder is outside the Field of View.

Due to the preliminary choice of the reference frame it is very simple to detect if an intruder is within the FOV. All it takes is a comparison of the intruder angles/range and the FOV angles/range.

The inputs of the decision algorithm are the vectors pertaining to the intruder and the relative attitude vectors. This block outputs the same vectors to the sensor simulations only if the result of this block is that the sensor indeed needs to be simulated.

3.5. Sensor simulation



The sensor simulation calculates whether the simulated sensor will detect an intruder and with what accuracy the sensor would be able to determine the position of the intruder. Every sensor will take the position and attitude vectors generated by the coordinate transform and will output a vector to the target, if it is determined to be detectable, and a confidence number which indicates with what margin the intruder is detected. The sensors also have access to the database which contains parameters on the sensors, simulation and aircraft.

3.5.1. Radar Sensor

Simulating a radar sensor can be done in a variety of ways, ranging from very simple and computationally easy to very accurate but very hard to calculate.

A typical radar system can be divided into three parts:

1. The electromagnetic part; the transmission, reflection and reception of the wave
2. The receiver electronics
3. The detection algorithm

All three parts can be simulated in a very detailed manner, but this is neither practical due to the requirement that the simulation is real-time, nor is it possible to do so in the time available for this research project. For these reasons the decision is made to start with a simple simulation system combining the first two parts in one algorithm and the last part in a separate algorithm.

On the basis of the literature study and the relatively good fidelity and low computational demands a link budget model is chosen for the simulation of the electromagnetic and electronics part of the radar sensor. The output of this calculation is a signal-to-noise ratio, which can be fed to a detection algorithm. Initially the detection will be a simple threshold detection, later more elaborate decision algorithms will be discussed like the Swerling algorithm mentioned in Section 2.7. The block diagram of this system is given in Figure 3.5. The sensor parameters are stored in a database, the simulation vectors are the vectors that are passed on by the previous block.

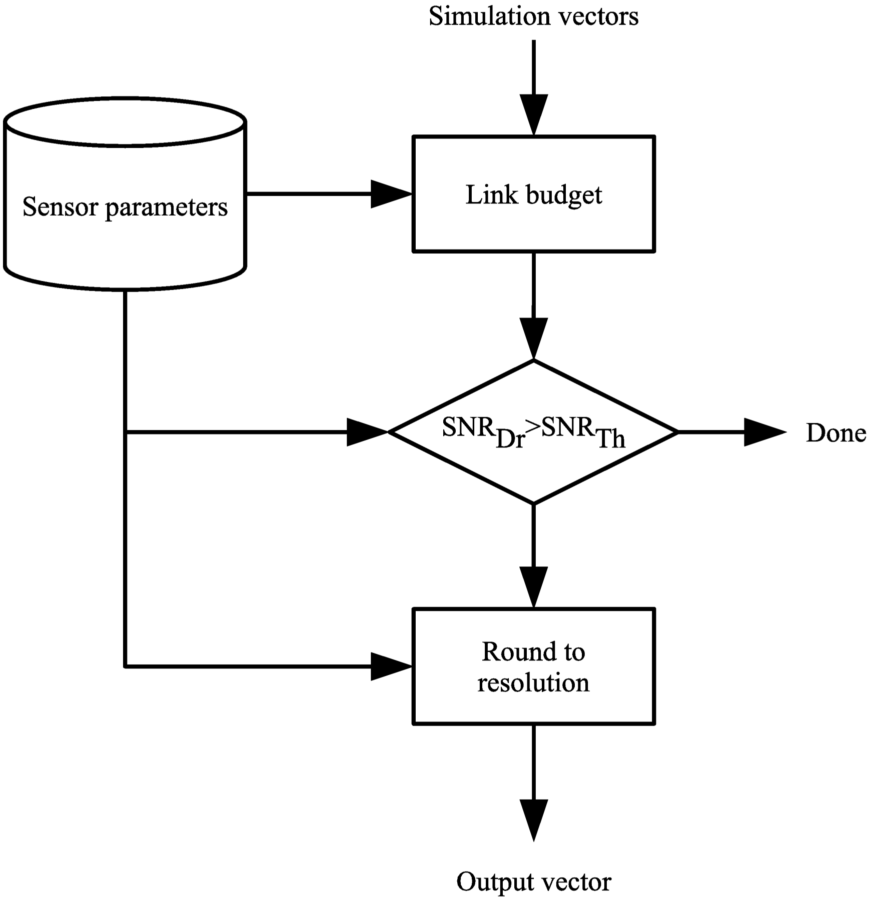


Figure 3.5: Block diagram radar simulation

A comprehensive introduction on radar simulation and the link budget model is given in [12]. First the received signal power needs to be computed. The equation used is given in equation 7, the symbols and parameters are explained in Table 2.2.

$$P_{Dr}=\frac{P_t G^2 \lambda^2 \sigma}{(4\pi)^3 R^4} \tag{7}$$

$$SNR_{Dr}=\frac{P_{Dr}}{k T_e B} \tag{8}$$

When the received power is calculated, the SNR can be determined by subtracting the noise (8). The necessary parameters are retrieved from a central database and the range comes from the simulation.

The detection is simply a threshold detection, so an intruder is detected when the SNR is higher than a set value. The threshold value can be found by using the simulation results as presented in the paper where the radar parameters are also given [22].

When the intruder is detected the resolution of the sensor must be taken into account when returning a vector. This is done by simply rounding the three components (*p,q* and R) of the vector to the nearest point in the resolution-grid of the sensor.

3.5.2. Electro-optical sensor

In the literature described in Chapter 2., the detection of intruders with the help of EO sensors is often based on an optical flow algorithm. Implementing such an algorithm takes a lot of system resources if this simulation has to be done in real time. During the NASA flight tests specialized hardware like FPGA's are used to perform such real time calculations.

To make the system viable for use in a real time simulator with limited computational capabilities the sensor and detection algorithm are simulated. The first step in the design process is the identification of the factors most influential to the detection probability. Three important intruder characteristics are defined which have the most significant effect:

1. Size of the intruder, and consequently the size of the projection on the image sensor created by the intruder
2. The contrast between the intruder and the background
3. The speed of the intruder relative to the background

The first two parameters are simulated in a simple manner in the current implementation, to improve this algorithm the static signature will be calculated in a more accurate way. The static signature, or the number of pixels the image of the target covers, depends on the attitude of the intruder and the position of the intruder relative to the ownship. Computing the static signature can be done in a variety of ways. Two examples are: high fidelity aircraft models and the extraction of the signature from the image stream generated by the simulator. The latter method comes very close to the implementation of an actual detection algorithm and has the same resource requirements. Using high fidelity aircraft models also places a significant load on the computational resources. In order to avoid these requirements a simple aircraft model is created which is expected to produce an accurate enough static signature with a simple set of equations. Again the main characteristics which have the most impact on the signature are identified:

1. The horizontal surfaces, like the wings and horizontal stabilizer
2. Surfaces that only change the signature when the relative heading changes, the main contributor is the fuselage

3. The front and back of an aircraft is usually a rounded surface which is visible more or less independent off the heading and pitch

After some experimenting with different shapes, the model depicted in Figures 3.6 and 3.7 is chosen. The main fuselage is represented by the cylinder (blue), the nose and back by the sphere (red) and the horizontal surfaces by the horizontal plane (green).

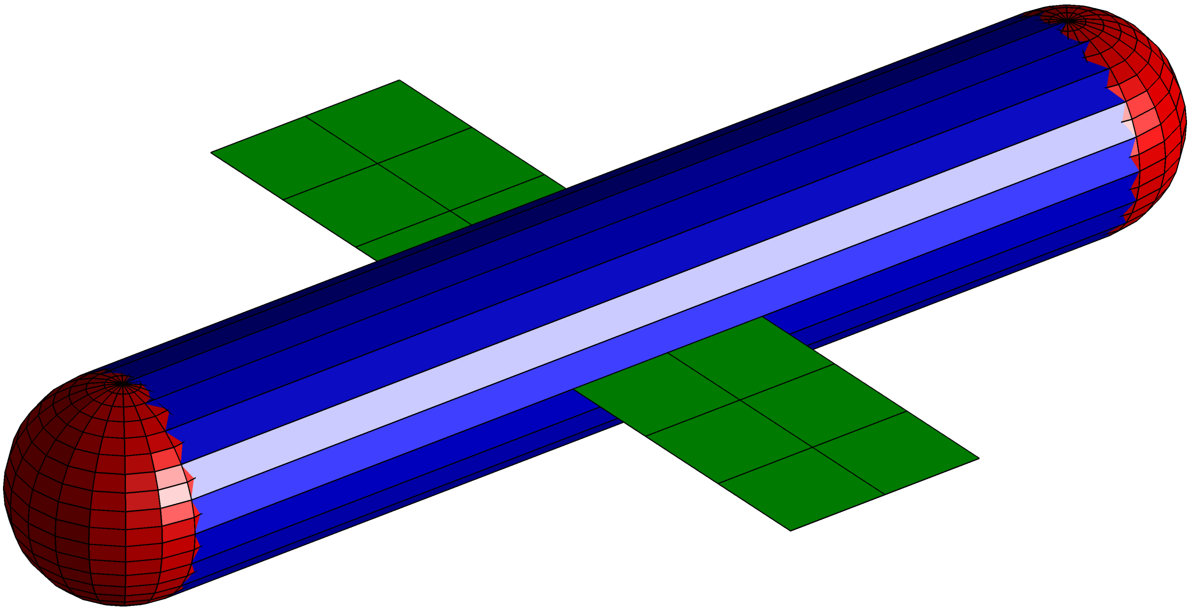


Figure 3.6: Intruder aircraft model

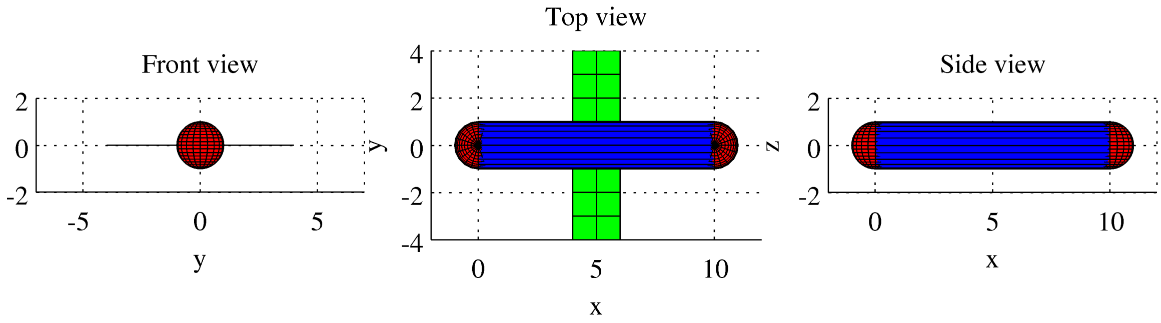


Figure 3.7: Front, top and side view of the intruder aircraft model

Each type of intruder is characterized by three numbers: the visible surface of the sphere, the visible surface of the cylinder when viewing at a right angle and the surface of the horizontal plane. Three aircraft dimensions were chosen to be used in the simulator, their dimensions are given in Table 3.1. The Beech 99 is chosen as the model for general aviation aircraft because it is similar to the Beechcraft King Air used in the AFRL research discussed in the previous chapter. The F-16 represents fighter aircraft and the Boeing 787 represents commercial aircraft. Their dimensions were calculated on the basis of their real-world dimensions [25] [26] [27].

Table 3.1: Standard intruder aircraft dimensions

Aircraft	**Front (red sphere) [m^2]**	**Side (blue cylinder) [m^2]**	**Horizontal planes (green) [m^2]**
Beech 99	12	30	90
F-16	7	20	250
Boeing 787	90	250	400

Now that the model and dimensions of the intruder are determined, the size visible to the sensor can be calculated.

Because it is assumed the sensor always looks right at the target and the relative angles are known, the calculation of the projected surface is done via basic trigonometry functions. Figure 3.8 shows the target, $\boldsymbol{x}$, and its image x'. Equations (9) to (12) calculate the fraction of surface that is projected onto the YZ-plane.

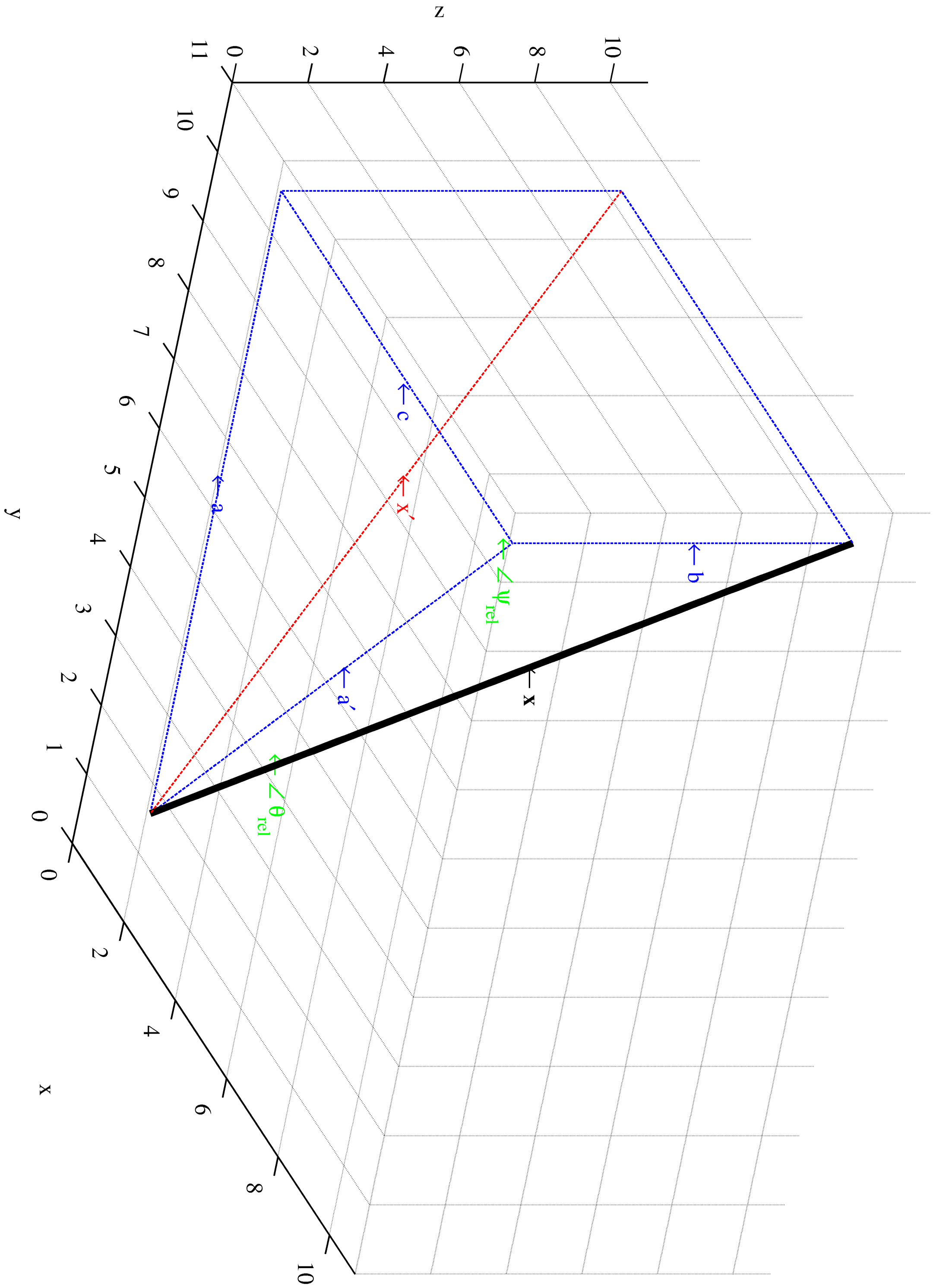


Figure 3.8: Projection of x onto the YZ plane x′

$$a'=x*\cos(\theta_{rel}) \tag{9}$$

$$b=x*\sin(\theta_{rel}) \tag{10}$$

$$a=a'*\sin(\psi_{rel}) \tag{11}$$

$$x'=\sqrt{b^2*a^2}=\sqrt{x^2*\sin^2(\theta_{rel})+x^2*\cos^2(\theta_{rel})*\sin^2(\psi_{rel})} \tag{12}$$

Depending on what is used for *x*, equation (12) can be used to calculate the visible surface of the cylinder or that of the plane. In the first case *x* will be the parameter in the column *side* of Table 3.1, for the second case one more calculation is needed. As can be seen in Figure 3.4 the roll angle is important when determining how large the projection of the plane on the sensor is. Equation (13) corrects for the roll angle, *y* is the area of the plane.

$$x=y*\sin(\phi_{rel}) \tag{13}$$

Now that the surface of the projection is known, the number of pixels covered can be found. As the resolution σ of the sensor is given in radians, it is easy to compute the surface *S* of one pixel at some distance *R*. This is done in equations (14) to (15).

$$S=S_x*S_y=(R*\sigma_{hor})*(R*\sigma_{vert}) \tag{14}$$

$$no.\,of\ pixels\,covered=\frac{surface\ visible}{surface\ per\ pixel}=Q=\frac{x}{S} \tag{15}$$

The speed of the intruder through the sensors field of view is found by calculating the angular velocities of *p* and *q* (16) and combining these in one vector (17) and compensating for the sensor update frequency. Note that the speed in the direction of *R* is not a significant factor because the movement of the intruder in this direction is barely visible due to the poor resolution in the range. The values of p_{t-1} and q_{t-1} are stored in the database for the next time instance.

$$\Delta p=p_t-p_{t-1};\Delta q=q_t-q_{t-1} \tag{16}$$

$$V=\sqrt{(\Delta p)^2+(\Delta q)^2}*F_{sensor} \tag{17}$$

An optical flow algorithm must be able to separate the pixels the target covers from the background, otherwise the target will not have a large enough total signature to be detected. When the contrast is low, the target image needs to be very large to be classified as a target. Therefore the contrast is implemented as a factor *P* which acts as a gain on the static signature.

When the size, speed and contrast are known, they are combined into one factor E_{sign} which represents the total signature. To keep the system flexible and to be able to fine-tune the system, the weighing factors *a* and *b* are introduced (18). When comparing this factor with a threshold value E_{thresh} , a confidence factor *C* can be obtained (19).

$$E_{sign} = a * Q * P + b * V \tag{18}$$

$$C = E_{sign} - E_{thresh} \tag{19}$$

Now the total simulation for the EO system can be summarized in a block diagram shown in Figure 3.9.

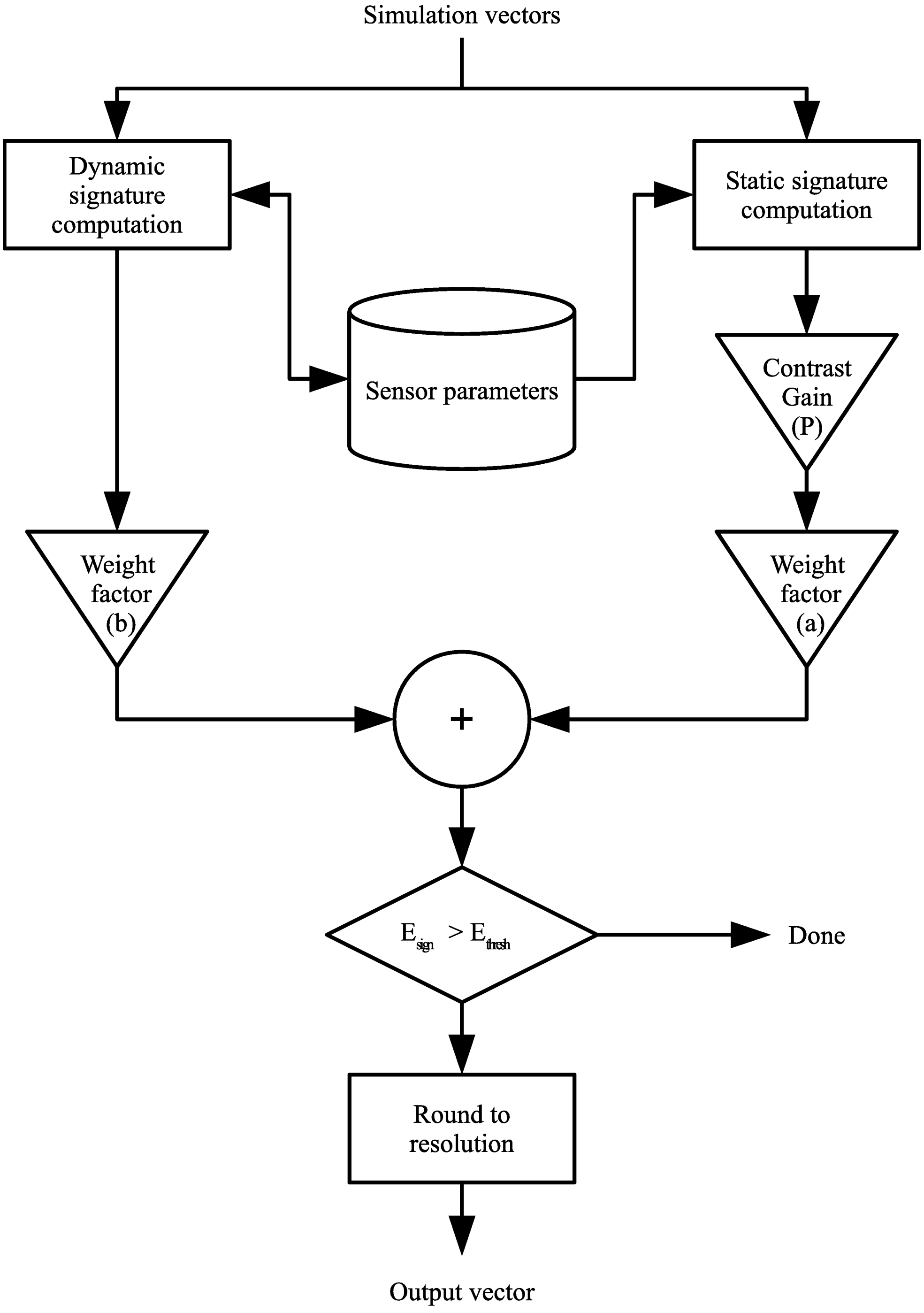


Figure 3.9: Block diagram of the EO simulation system

3.5.3. Infrared sensor

As mentioned before, the infrared sensor is considered to be a special type of EO sensor - therefore it is possible to use the described EO simulation system but with different parameters. Because the IR sensor detects emitted energy caused by heat, the sensor will also be able to detect the exhaust gas from jet engines. As is the case with the static signature computation used in the model for the EO sensor, also extensive models exist for the calculation of the IR signature of aircraft, for example the NATO Infra-Red Air Target Model, which can calculate the IR signature for different aircraft. To keep the simulation simple enough to perform it real-time without too demanding hardware requirements a more simple model is used.

The EO model is modified to incorporate the two most significant parameters that influence the signature: the external jet plume and the internal tail pipe signature [17]. With the EO simulation system it is not possible to simulate the intensity of the different parts of the target aircraft separate from the projected area. This necessitates compensation for the size to be taken into account. Figure 3.10 shows the modified model: the tail section of the aircraft is made bigger to compensate for the effect the internal tail pipe has on the signature and the length of the blue cylinder is increased to take the exhaust plume into account.

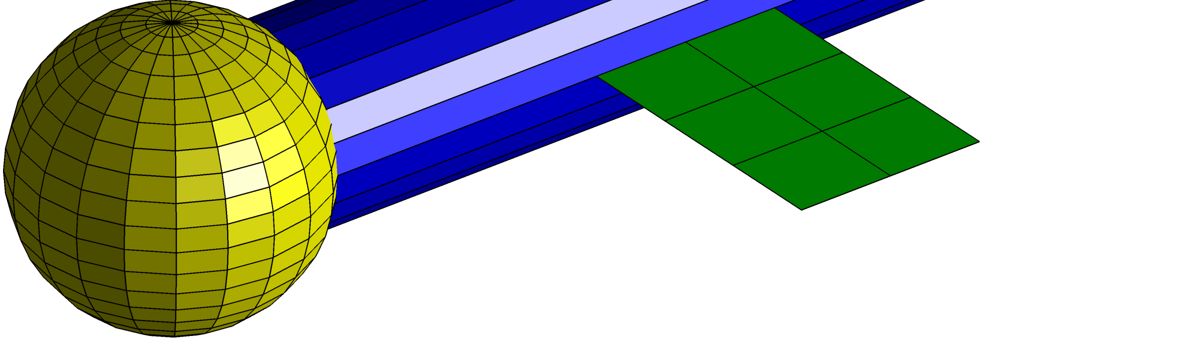


Figure 3.10: Intruder model for the IR sensor, the tail of the aircraft points to the left

The effect of these two changes will be that the aircraft will have a similar signature when looking exactly head on as in the signature with the EO sensor. From every other angle, the signature will be larger. The difference in size of the front and back sphere introduces one more equation to the model. The result of equation (20) gives the surface visible due to the front and back spheres; the equation is simplified in equation (21).

$$x = a(0.5 \times \cos(\psi_{rel}) + 0.5) + b - b(0.5 \times \cos(\psi_{rel}) + 0.5) \quad (20)$$

$$x = (a - b)(0.5 \times \cos(\psi_{rel}) + 0.5) + b \quad (21)$$

The signature of the individual spheres as well as the summation of the two as a function of relative heading is given in Figure 3.11, in which the parameters *a* and *b* are set to 1 and 1.5 for the plot and represent the size of the back and front sphere respectively. In other respects the system is identical to that of the electro-optical sensor, with adjusted parameters derived from [13]. Table 3.2 gives the relevant parameters for the three types of aircraft used in the simulation.

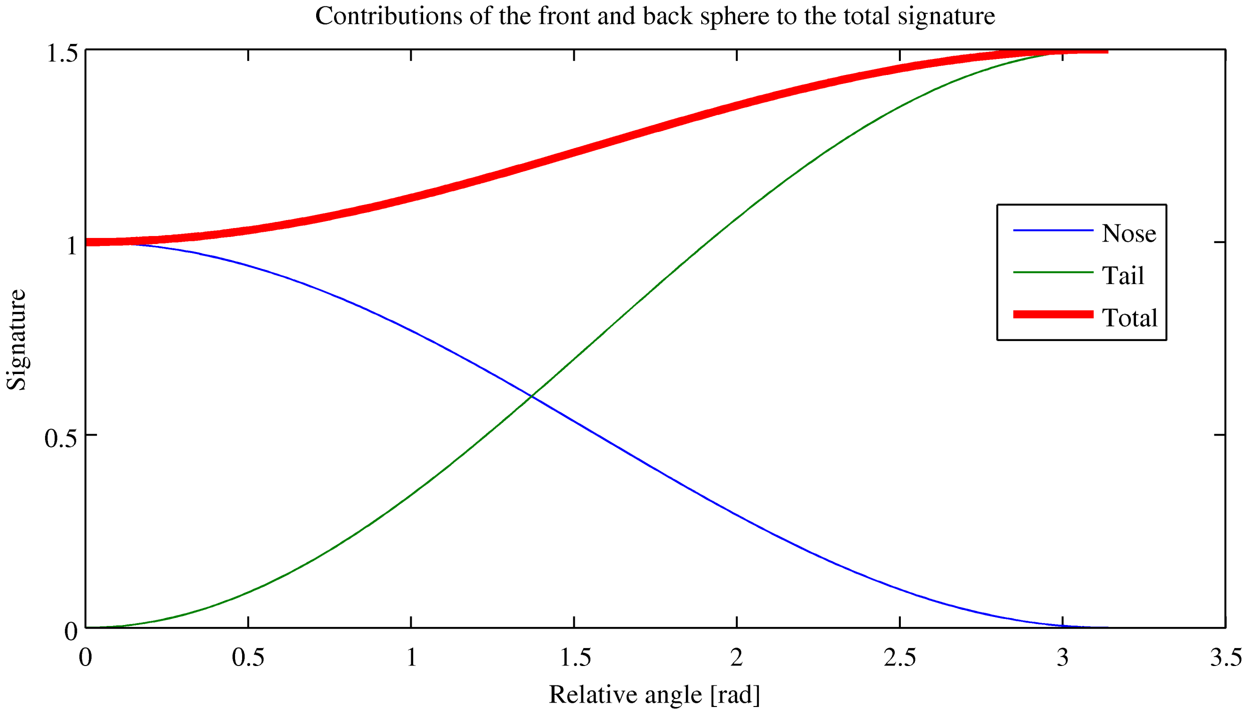
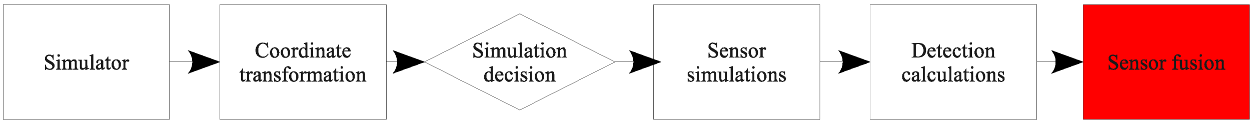


Figure 3.11: Shape of the contribution of the front and back sphere to the static signature

Table 3.2: Aircraft parameters for the IR system simulation

Aircraft type	**Front (red sphere) [m²]**	**Back (yellow sphere) [m²]**	**Side (blue cylinder) [m²]**	**Horizontal planes (green) [m²]**
General aviation	4,6	4,6	11,15	35,52
Small fighter	7	14	14	37,16
Commercial transport	32,5	65	223	353

3.6. Sensor fusion



In order to test the independent sensor simulations as well as the total system and in accordance with the Hazlett design methodology, the first design of the sensor fusion algorithm will be a *filler*. In this implementation the fusion algorithm will pick the most accurate sensor for the detection vectors: that will be the radar sensor for the range and the optical sensors for the azimuth and elevation angle. Of course when the result of a sensor simulation is that a target is not detected, the data from other sensors will be used when available. A block diagram of this algorithm is given in Figure 3.12.

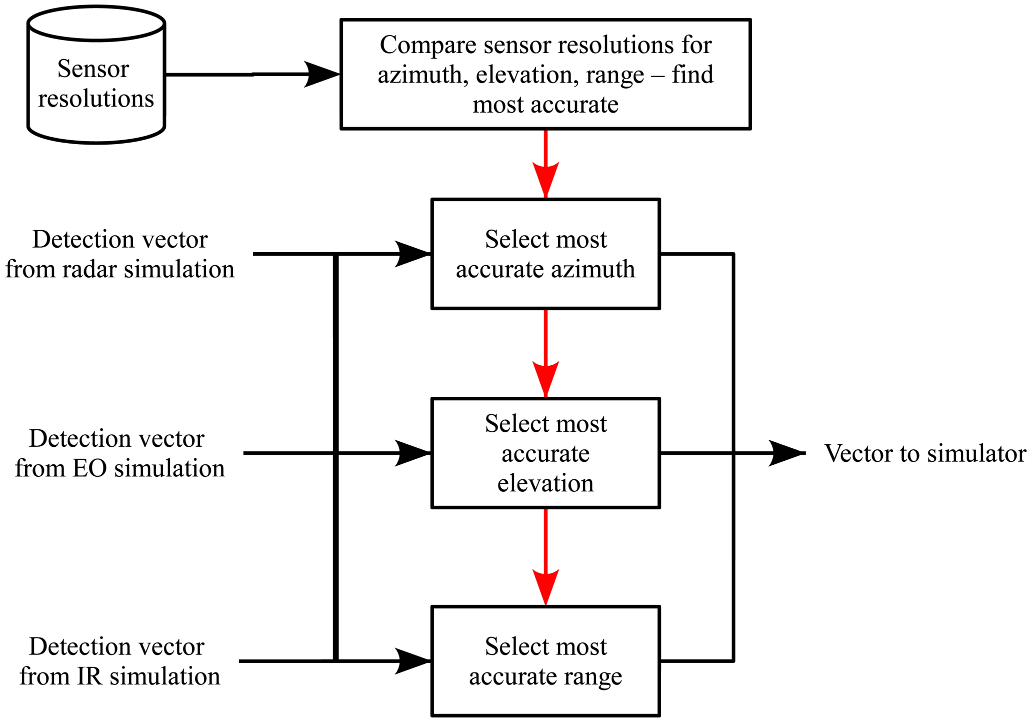

Sensor resolutions
Compare sensor resolutions for azimuth, elevation, range – find most accurate
Detection vector from radar simulation
Select most accurate azimuth
Detection vector from EO simulation
Select most accurate elevation
Vector to simulator
Detection vector from IR simulation
Select most accurate range

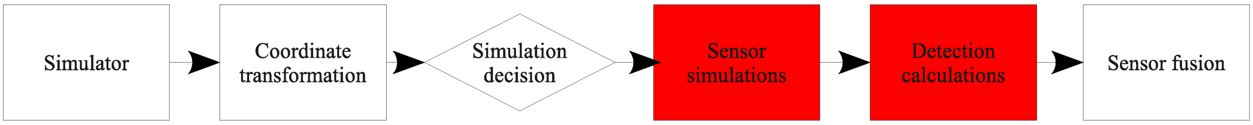
Figure 3.12: Block diagram sensor fusion algorithm

4. Testing and refinement

With the completion of the design and its implementation for the first iteration, the quality of the simulation can be determined. By comparing the results of the individual components as well as the overall results to the data from the studies discussed in Chapter 2, conclusions can be drawn on the system performance.

This chapter will discuss the performance of every relevant component and identifies areas of improvement.

4.1. Radar sensor



4.1.1. Performance

The Oasys radar system specifications define that a target with a 5 m^2 radar cross section is detected at exactly 3.5 sm, that is 6.48 km. The parameters of the simulation system are set accordingly and the resulting system performs as expected.

4.1.2. Improving the RCS modeling and detection algorithm

When the radar system is detecting a target that has a speed relative to the ownship, the radar cross-section of this target will not remain constant if the target is not a perfect sphere. The RCS of a target will depend on the size, material, shape and smoothness of the reflecting surface. Because an aircraft has many sources of reflection the shapes and sizes of a moving target as seen by the ownship will fluctuate constantly. This causes the determination of the RCS of a target from every angle to be difficult and will require a lot of computational power and complicated aircraft models. Figure 4.1 shows the detection distance as a function of viewing angle obtained from the NASA flight tests. There shows no clear correlation between the angle and the detection distance, and thus the RCS. Theoretically this correlation will exist, but because of the fluctuations in RCS due to very small variations in the angle this correlation can only be found when analyzing large high resolution datasets.

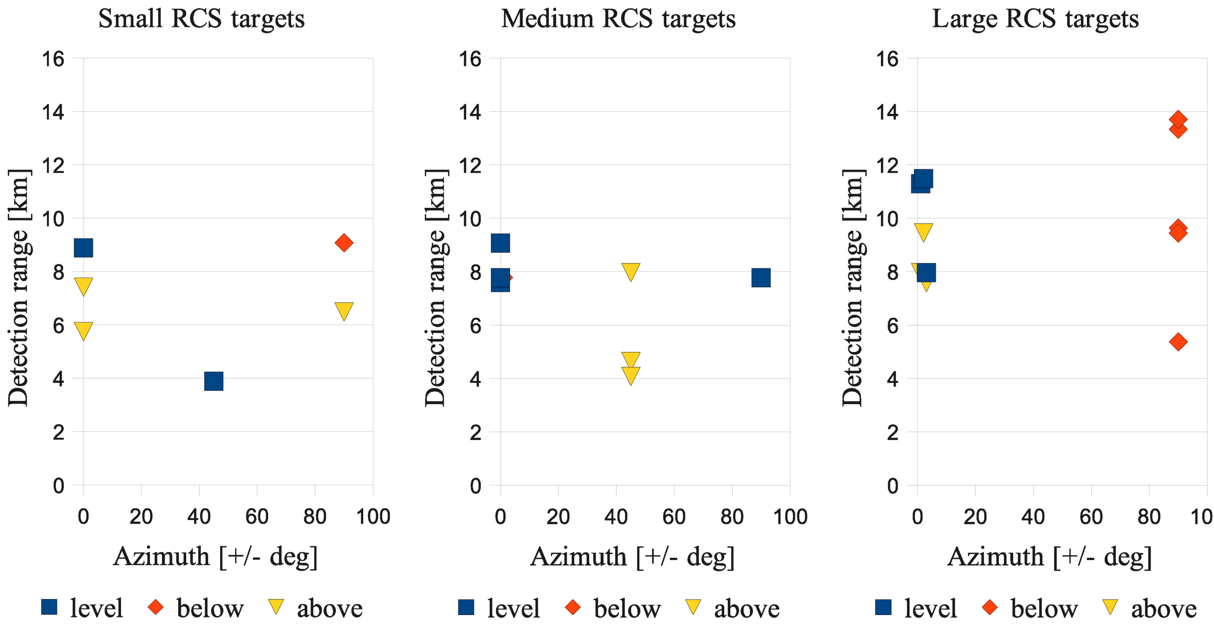


Figure 4.1: Detection ranges for different conflict geometries

In [28] Swerling analyzed how the RCS of different targets fluctuate and he created a stochastic representation in the form of a Chi-square distribution with the number of degrees of freedom depending on the type of target. His models account for fast fluctuations, variations in one scan between pulses, and slow fluctuations, variations between scans, in the amplitude of the received signal.

Swerling defines five types of targets with different characteristics. Type 0 or V is used for targets with a constant RCS, this is equivalent to the previously described system. Types I and II are used for targets consisting of many small scatterers with the difference being that the type II RCS also varies between pulses. Finally types III and IV characterize targets with one dominant and several small scatterers. Aircraft do not consist of one dominant scatterer and can be seen as a collection of more or less equal scatterers. Therefore they can best be represented by type I and II.

If the radar uses coherent pulse integration like the radar that is used as the basis of this simulation effort, fluctuations between pulses in one scan cannot be simulated because these fluctuations cause a decorrelation between the amplitudes from different pulses. This causes that RCS of aircraft detected by the simulated radar system can only be modeled as a Swerling I target. Because most targets a UAV is expected to encounter will probably fall into the category of the Swerling I targets, the simulation will only contain Swerling models I and V. Swerling model V can be used to make the behavior of the radar simulation more stable and can be used for evaluation of other system components.

As was mentioned, the Swerling method uses a stochastic approach to simulate the target RCS fluctuations. The probability density function (pdf) of the Chi-square distribution used in this method is given in equation (22). For Swerling type I targets $N = 1$ resulting in the Rayleigh pdf given in equation (23).

$$f(\sigma)=\frac{N}{(N-1)!\times\bar{\sigma}}\left(\frac{N\sigma}{\bar{\sigma}}\right)^{N-1}\exp\left(-\frac{N\sigma}{\bar{\sigma}}\right) \tag{22}$$

$$f(\sigma)=\frac{1}{\bar{\sigma}}\exp\left(-\frac{\sigma}{\bar{\sigma}}\right);\sigma\geqslant 0 \tag{23}$$

Now that the pdf of the RCS is known, the probability of detection (P_D) of the radar system can be determined. Whether a target is detected by a radar system is determined by the signal-to-noise ratio (SNR) of the received pulse reflected by the target. Previously this was done using equations (7) and (8), but the SNR is dependent on the radar system parameters, the distance to the target, attenuation by for example rain or air, the RCS of the target and non static noise. This noise has several sources and can be described as a zero-mean Gaussian random process for which equation (24) gives the pdf. To find the pdf for the total received signal, i.e. noise plus received pulse, the equation for noise is combined with the equation for the received signal. The outcome is given in equation (25) and is called the Rician pdf. The expression *I_0(.)* is the modified Bessel function of zero order. In the equation ψ^2 expresses the variance of the noise, r is the received signal (signal plus noise) and A is the amplitude of the received pulse.

$$f(n_I,n_Q)=\frac{1}{2\pi\psi^2}\exp\left(-\frac{n_I^2+n_Q^2}{2\psi^2}\right) \tag{24}$$

$$f(r)=\frac{r}{\psi^2}I_0\left(\frac{rA}{\psi^2}\right)\exp\left(-\frac{r^2+A^2}{2\psi^2}\right) \tag{25}$$

When the signal plus noise, r, reaches a certain threshold value, the target is detected by the radar. By integrating equation (25) from the threshold value to infinity the probability of detection is found. If the RCS is also a stochastic function, a conditional pdf *$f(r/\sigma)$* is needed and can be integrated to find the P_D. This derivation will not be presented here, a comprehensive explanation of the total derivation is given in [12].

The integral of the conditional pdf for Swerling models resulting in P_D can be approximated by the Gram-Charlier series. The general expression is given in equation (27).

$$V=\frac{V_T-n_p(1+SNR)}{\omega} \tag{26}$$

$$P_D \simeq 0.5 \times erfc(V/\sqrt{2}) - e^{-0.5V^2}\left(C_3(V^2-1) + C_4 V(3-V^2) - C_6 V(V^4 - 10V^2 + 15)\right) \tag{27}$$

The constants ω, C_3, C_3 and C_6 depend on the type of Swerling model for which P_D is calculated. For the case of Swerling V targets, the values of the constants are given in equations (28) to (31).

$$C_3 = -\frac{SNR + 1/3}{\sqrt{n_p}(2SNR + 1)^{1.5}} \tag{28}$$

$$C_4 = \frac{SNR + 1/4}{n_p(2SNR + 1)^2} \tag{29}$$

$$C_6 = 0.5C_3^2 \tag{30}$$

$$\omega = \sqrt{n_p(2SNR + 1)} \tag{31}$$

For the case of Swerling I targets, the value of P_D can be obtained with equation (32) where Γ_I is the incomplete gamma function and the term SNR is the mean SNR for a target.

$$P_D = 1 - \Gamma_I(V_T, n_p - 1) + \left(1 + \frac{1}{n_p SNR}\right)^{n_p - 1} \Gamma\left(\frac{V_T}{1 + (n_p SNR)^{-1}}, n_p - 1\right) \times \exp\left(\frac{-V_T}{1 + n_p SNR}\right) \tag{32}$$

When the radar system characteristics do not change, the results of equations (27) and (32) will only change as a function of the SNR. For the sake of the computational demand, the P_D can be calculated for different values of the SNR and stored in a lookup table. Figure 4.2 gives the detection probabilities for a range of SNR values. A further rise in efficiency can be achieved by truncating the table so it only contains information on the part of the function where P_D is not zero or one. The values of the truncated table are also shown in Figure 4.2.

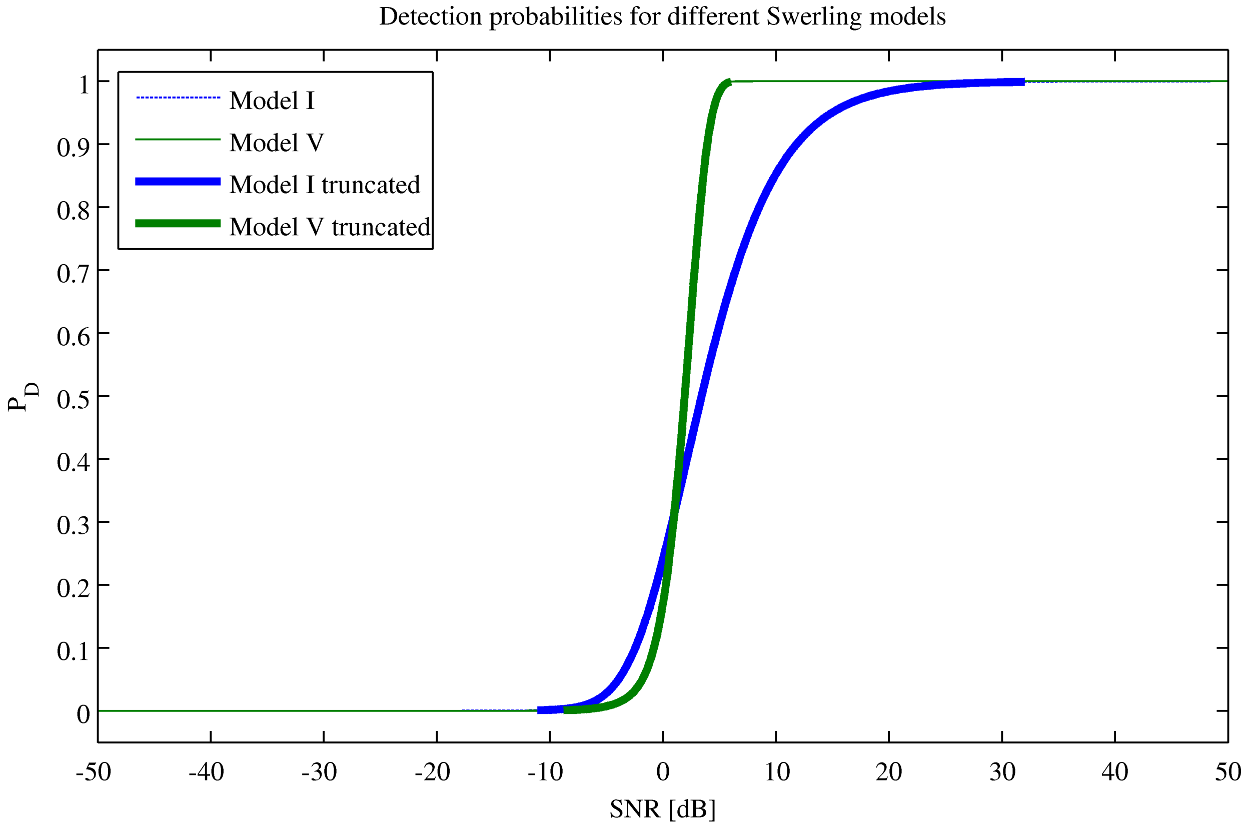


Figure 4.2: Plots of the found detection probabilities for the given radar parameters

4.1.3. Results after refinement

To assess the performance of the radar system simulation, data from the flight tests performed under the NASA ERAST program described in Section 2.5. is used. The seven test aircraft are assigned to one of the three RCS classes according to their size, engine configuration and the aircraft material. The classification is given in Table 4.1.

Table 4.1: Aircraft specifications and RCS designation

Aircraft	Material	Engine configuration	RCS class	RCS [dBsm]
Glider	Composite	None	Small	1,76
Stinson	Cloth	Single prop	Small	1,76
Long EZ	Composite	Single prop	Small	1,76
Extra 300	Metal	Single prop	Medium	3,01
Beech Duchess	Metal	Dual prop	Medium	3,01
King Air	Metal	Dual prop	Large	6,99
FA-18	Metal	Jet	Large	6,99

Because the simulation does not discriminate between the angles on which the intruder approaches the ownship, only range is taken into account. Table 4.2 contains the parameters of interest.

The average detection distances for the small, medium and large RCS targets are respectively 6.91, 7.31 and 9.75 km. Figures 4.3 through 4.5 show the simulated SNR ratio for targets with a different RCS and the calculated probability of detection. At the average detection distances obtained from the NASA flight tests, the simulation shows a probability of detection of 0.9114, 0.9170 and 0.8668. These values approximate the detection probability of 0.9 given as a radar parameter.

Table 4.2: Results of the flight tests under the NASA ERAST program; source [18]

Scenario	Intruder(s)	RCS	Radar detection range [km]
1	Extra 300	medium	7.59
2	Extra 300	medium	4,63
3	Extra 300	medium	7.78
4	Long EZ	small	3.89
5	Stinson	small	5.74
	Long EZ	small	9.07
6	Stinson	small	8.89
9	Glider	small	7.41
10	Glider	small	6.48
11	Duchess	medium	9.07
12	Duchess	medium	7.96
13	Duchess	medium	7.78
14	Duchess	medium	9.07
15	Duchess	medium	7.78
	King Air	large	5.37
16	Duchess	medium	4.07
	King Air	large	9.63
17	FA-18	large	11.3
18	FA-18	large	7.96
19	FA-18	large	13.33
20	King Air	large	9.45
	FA-18	large	11.48
21	King Air	large	7.96
	FA-18	large	9.45
22	King Air	large	7.59
	FA-18	large	13.7

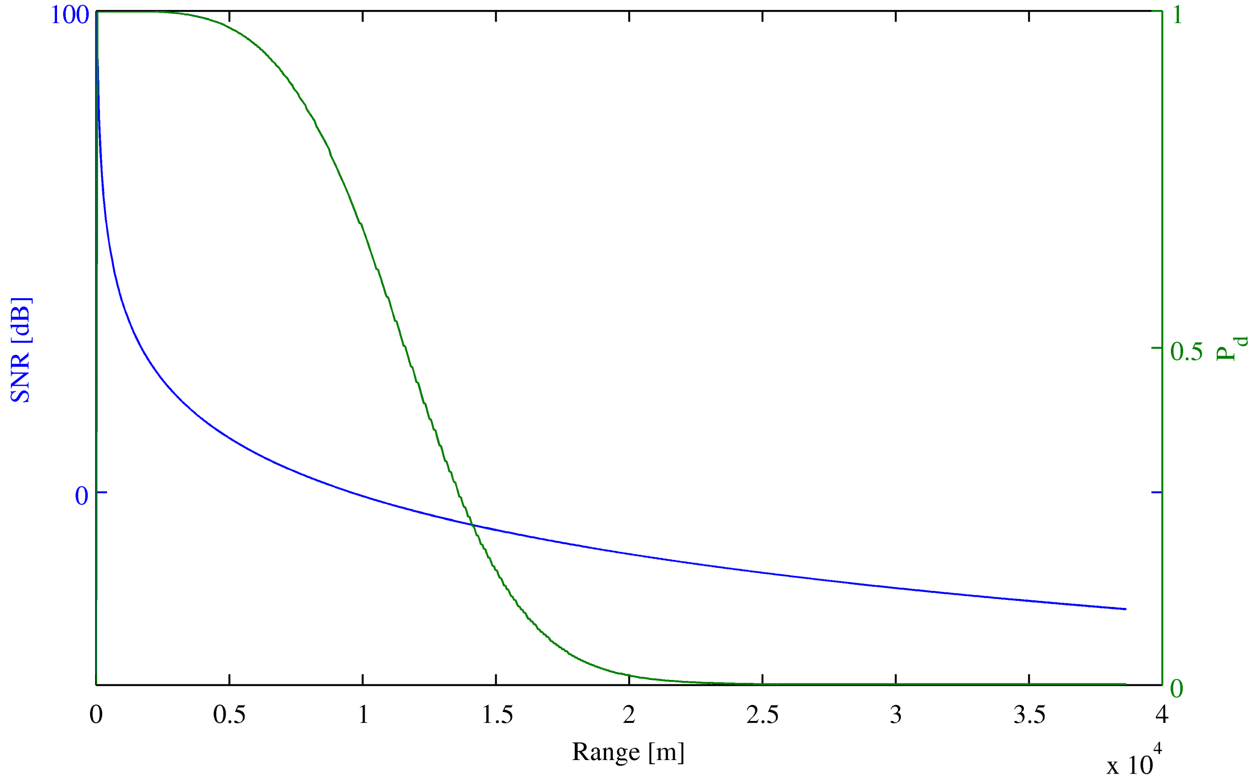


Figure 4.3: SNR and detection probability for a small target (1.76dB)

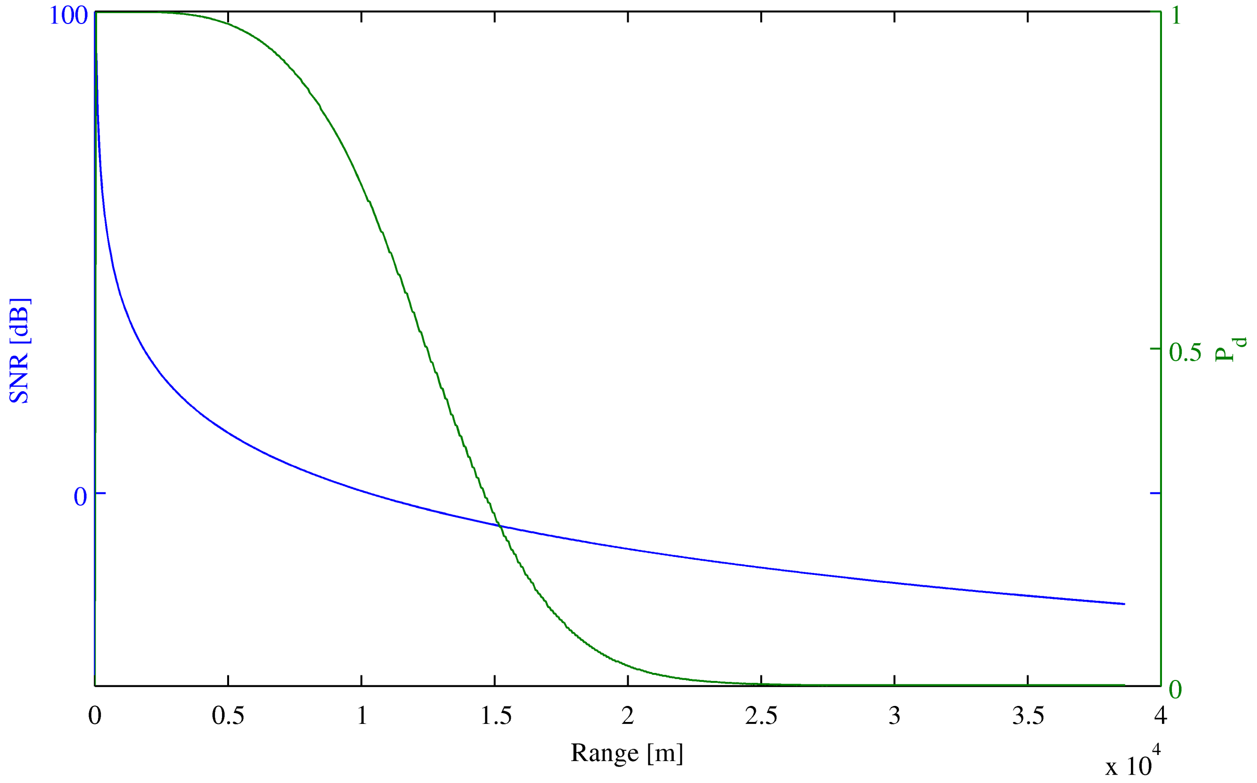


Figure 4.4: SNR and detection probability for a medium target (3.01dB)

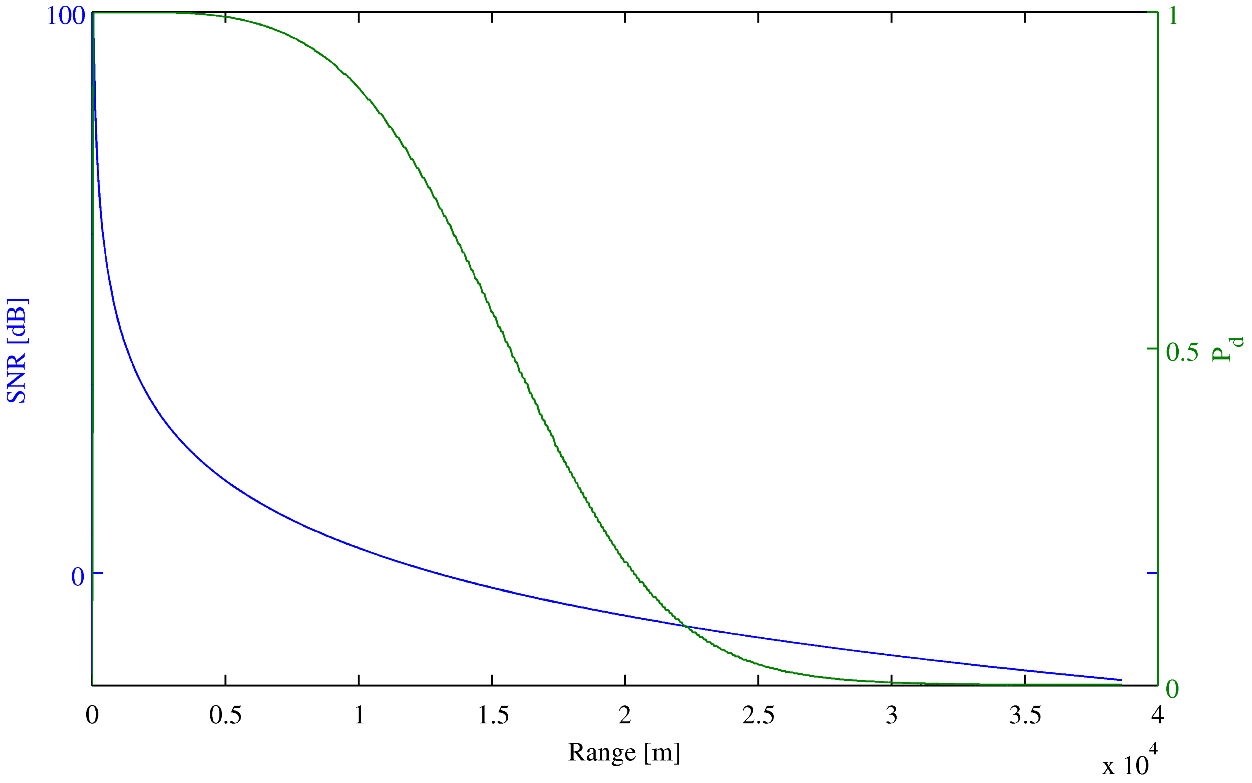
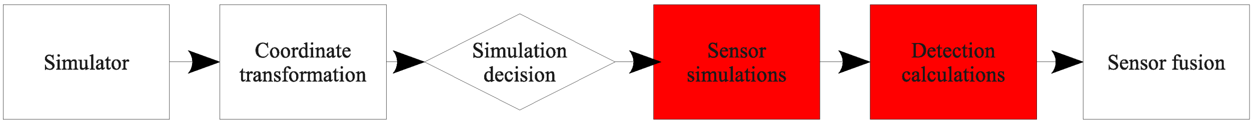


Figure 4.5: SNR and detection probability for a large target (6.99dB)

4.2. Electro-optical sensor



4.2.1. *Simplified geometric aircraft model validation*

The first step in the verification of the EO system simulation is the validation the aircraft model, shown in Figures 3.6 and 3.7, used to calculate the static signature. For this purpose high fidelity images are needed to compare the signature from the model to the real world signature. Because no good real world images are available for verification, flight simulation software is used that renders complex aircraft models with high fidelity. The open-source flight simulator FlightGear was chosen for the level and ease of configuration and the possibility to interface it with Matlab. Figure 4.6 shows the block diagram of the interface effort, the commands to FlightGear were sent via the *http* interface and the screen shots were saved to the file system before being imported into Matlab. Three aircraft models, the Beech 99, the F-16 and the Boeing 787, were modified to have a completely black surface and placed in a simulation environment with a constant background. Next the camera was rotated 180 degrees around the three main axes in the body fixed reference frame. Every 10 degrees of rotation a screen shot was made and imported into Matlab. In Matlab the aircraft image was extracted and converted to a pure black and white representation. Some of the resulting images are shown in Figure 4.7. The quality of the model can be assessed by counting the number of pixels in the image of the aircraft compared to the signature computation of the model. The signature is expressed in the fraction of the maximum number of pixels visible at one angle and are given in Figures 4.8 to 4.10. The mean and maximum errors of the model with respect to the signature images are presented in Table 4.3.

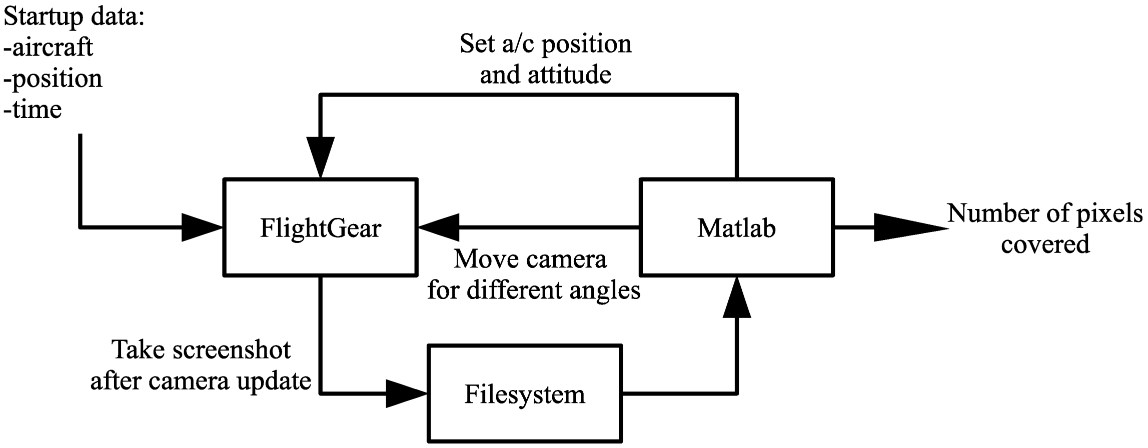


Figure 4.6: Block diagram of the interfacing between Matlab and FlightGear

Table 4.3: Mean and maximum errors aircraft models

Mean errors	**Beech 99**	**F-16**	**Boeing 787**		**Maximum errors**	**Beech 99**	**F-16**	**Boeing 787**
Rotation about x	3.5 %	2.2 %	1.3 %		Rotation about x	10 %	4.9 %	3.4 %
Rotation about y	11.5 %	9.2 %	7.6 %		Rotation about y	18.8 %	22.2 %	14.2 %
Rotation about z	6.2 %	5.9%	2.1 %		Rotation about z	12 %	16.3 %	8.2 %

The figures show that the size of the model signature during rotation approaches that of the signature of the reference images very well. Most mean errors are well below 10% and most maximum errors are below 20%. When considering the trade-off between computational complexity and performance of the model, this is acceptable.

The effect of the errors introduced by the aircraft modeling can be calculated using equations (14), (15) and (18). In these equations the error can be included by adding the term ε and isolating the error term. Equation (33) combines the three equations and shows the relevant part - the term *f* covers the rest of the equation. The influence of the error on the signature can now be expressed as is accomplished in equation (34).

$$E_{sign}=a*Q+f=a\frac{x+\epsilon}{s}+f=a\frac{x+\epsilon}{R^2\sigma_{hor}\sigma_{vert}}+f \tag{33}$$

$$E_{\epsilon}=\epsilon\frac{a}{R^2\sigma_{hor}\sigma_{vert}} \tag{34}$$

When the target is far above or below the detection threshold, the error does not matter as it will not influence the outcome of the simulation. The region of interest is where the signature comes close to the threshold and the error can influence the detection decision. In this region the denominator in equation (34) will take on a value between 1 and 3, which means that, when the value of a is 0.3 and the threshold is set to one, the influence of the error is between 6% and 20% on the detection range. The exact level of error depends on how close the target is detected: the 20% error occurs when the detection range is low. It can be shown that the absolute error for all detection ranges for an error of 20% in surface will result in an error in detection range of approximately 600 meters.

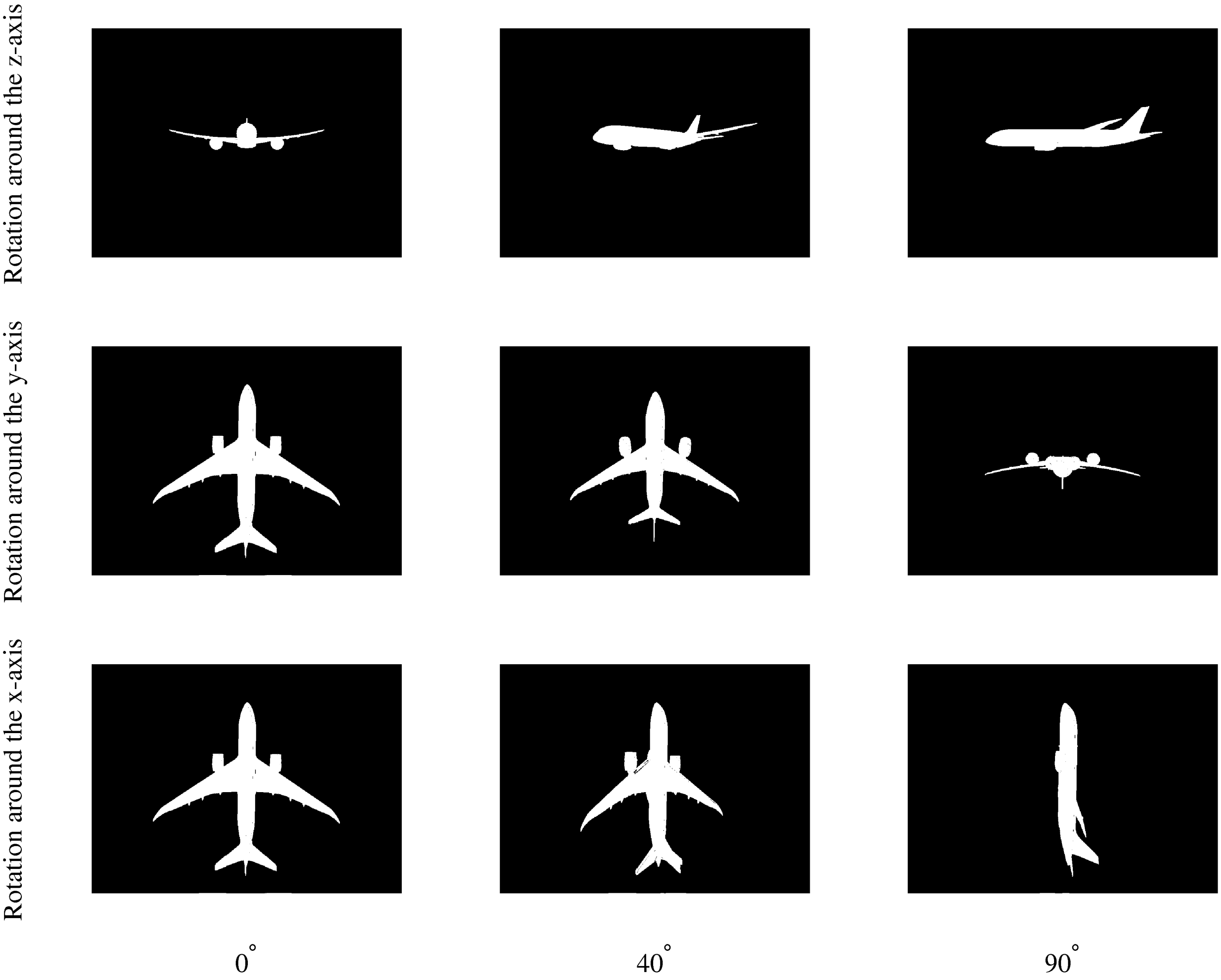

Rotation around the z-axis
Rotation around the y-axis
Rotation around the x-axis
0°
40°
90°

Figure 4.7: Signature of a Boeing 787 rotated around the body axes

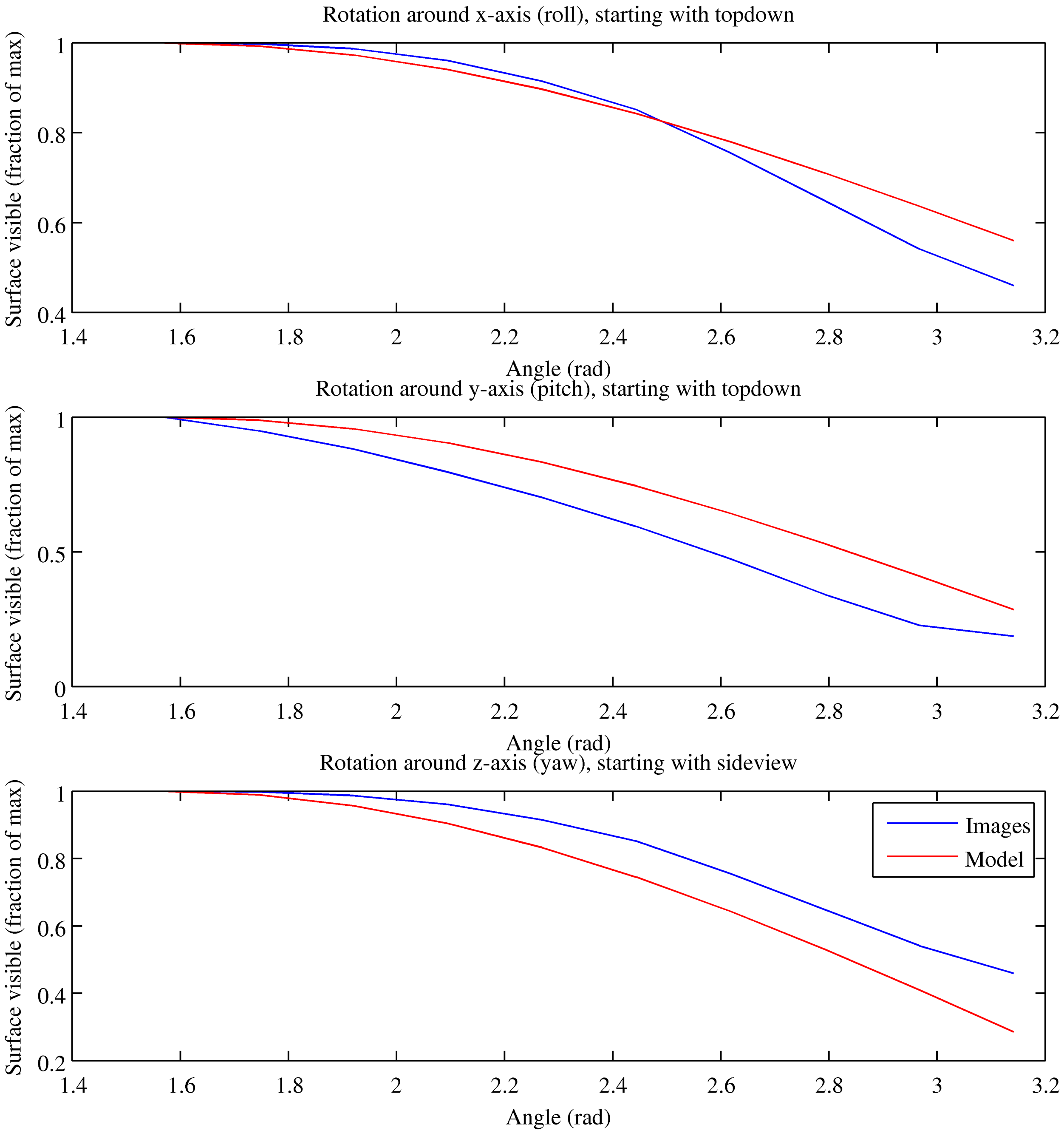


Figure 4.8: Signature comparison model and images of the Beech 99

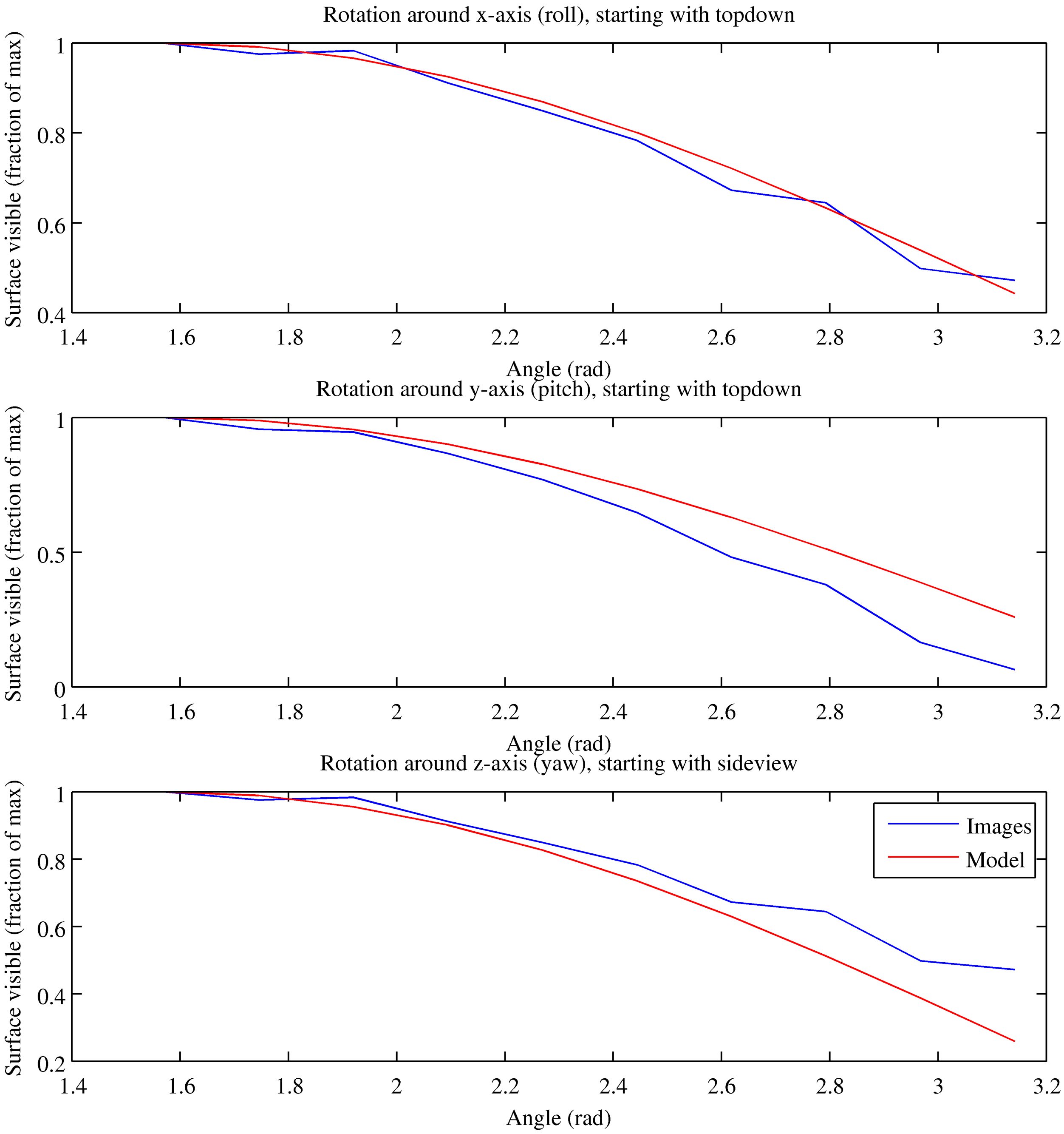


Figure 4.9: Signature comparison model and images of the F-16

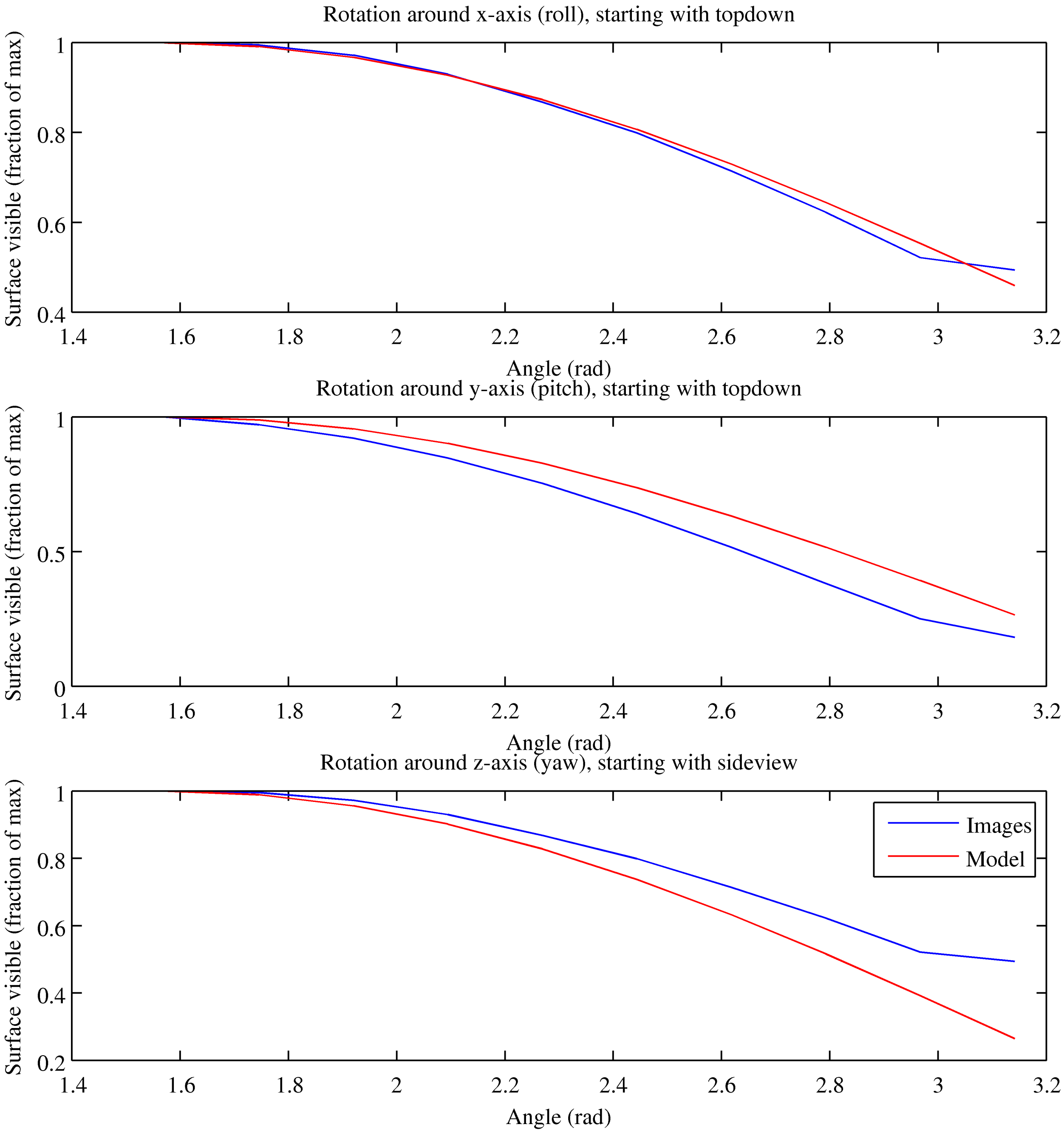


Figure 4.10: Signature comparison model and images of the Boeing 787

4.2.2. Simulation performance

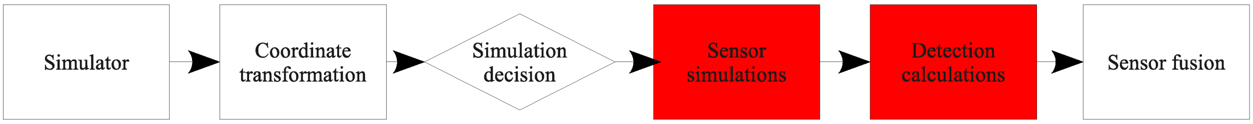
To assess the performance of the simulation the results are compared to results from literature, Table 4.4 shows the flight test performance as found in the AFRL studies and the simulation performance.

Table 4.4: Average intruder detection of the EO system

Geometry	Aircraft type	Detection literature [km]	Detection simulation [km]
Ascending Head-On	King Air	4,8	4
Ascending Head-On	Convair	5,4	9,5
Level abeam	King Air	6,7	6,2
Level Head-On	King Air	6,7	3,3
Level Head-On	Convair	7,7	9,1

As the table shows, the simulated performance does not compare well to the data from the literature. Because the information on the performance is rather limited, it is hard to find the causes for the mismatch in performance. When an image detection algorithm is used based on an optic flow algorithm, as the literature suggests, aircraft with a large velocity across the sensor image (like in the level abeam scenario) are expected to be detected sooner than aircraft with a very small velocity (like in the head on scenarios), the reasons for which cannot be deduced from the available data.

4.3. Infrared sensor



4.3.1. Simulation performance

Data for the validation of the simulation system are not obtained easily. One earlier described paper by Northrop Grumman [17] gives the performance of a simulated IR sensor. In Figure 4.11 the detection distance as a function of relative angle as found during the simulation is compared to the data found in the paper. The data obtained from the literature is not very accurate as the resolution of the dataset is not very high.

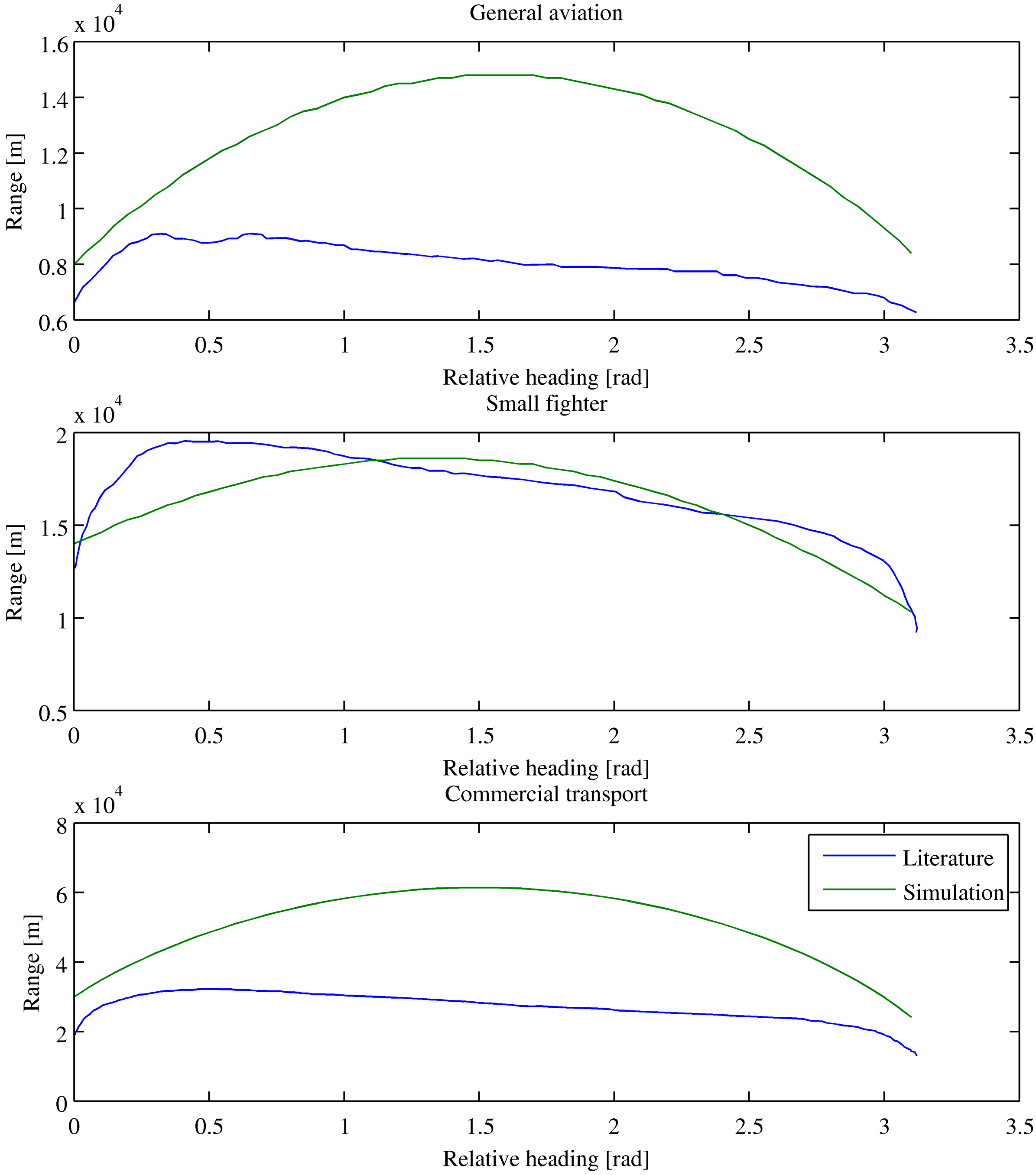


Figure 4.11: Comparison of performance between IR system simulation and literature

The maximum errors in detection range are approximately 90% for the general aviation, 25% for the fighter and 100% for the commercial transport. This performance is evidently poor - the next section focuses on improvements of the fidelity of the simulation.

4.3.2. Refinement

Figure 4.11 allows the conclusion that the signature calculation for the *side* of the aircraft is the biggest problem. This can be caused by the low contrast between the aircraft fuselage skin and the background; the modeled fuselage area corrected for its contrast appears to be too large. As was mentioned before, the main contributors to the IR signature are the exhaust pipe and gases. Also due to the friction of air flowing around the leading edges of the aircraft heat is generated at the front. Figure 4.12 shows the most common heat sources on a fighter aircraft. Of course the friction and therefore the heat signature for the leading edges is larger when the aircraft moves faster.

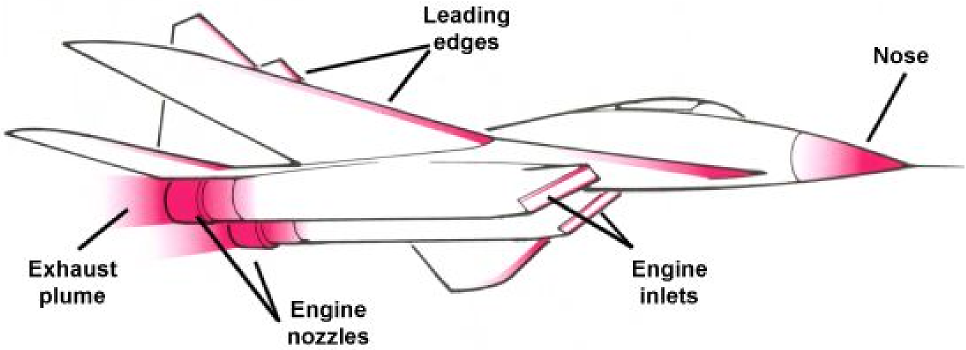


Figure 4.12: Heat sources on a fighter aircraft, the red color indicates heat; image courtesy of aerospaceweb.org

The extra sources of heat on the front of the aircraft diminishes the influence on the signature; consequently the other surfaces will have a larger influence. One aspect that has not been included in the simulation is the size of the exhaust, plume and pipe, relative to the fuselage. The three main types of engines, the propeller, the turbofan and the turbojet, all have a different exhaust characteristic - this was not taken into account in the simulation and could probably have given rise to errors. Table 4.5 sums up the main differences in the engine characteristics.

Table 4.5: Characteristics of different propulsion types used in aviation

Propulsion type	A/c speed	Temperature of airflow	Amount of airflow
Propeller	Low subsonic	Cold	Large
Turbofan	Subsonic	Mix cold/hot	Large cold/small hot
Turbojet	Subsonic/Supersonic	Hot	Small

From the table the conclusion can be drawn that the influence the exhaust system has on the signature will be largest for the turbojet and smallest for the propeller.

To correct for the identified sources of errors in the simulation, the aircraft parameters are changed as follows:

- The contribution of the side to the signature is reduced by a factor two to account for the low contrast
- The contribution of the back area is doubled for the turbofan and turbojet engined aircraft to account for the exhaust pipe and gas

Table 4.6 shows the new parameters.

Table 4.6: Modified aircraft parameters for the system simulation

Aircraft type	Front (red sphere) [m^2]	Back (yellow sphere) [m^2]	Side (blue cylinder) [m^2]	Horizontal planes (green) [m^2]
General aviation	4,6	4,6	6	35,52
Small fighter	7	14	7	37,16
Commercial transport	32,5	65	110	353

4.3.3. Results after refinement

After the modification of the parameters, the simulation result of the infrared systems is again compared to the performance found in literature. Figure 4.13 shows the new detection range as a function of relative heading. This time the maximum errors are approximately 25% for the general aviation, 50% for the small fighter and 30% for the commercial transport type of aircraft. For the general aviation and commercial transport types this is a dramatic improvement, for the small fighter type of aircraft these new parameters mean a degradation in performance.

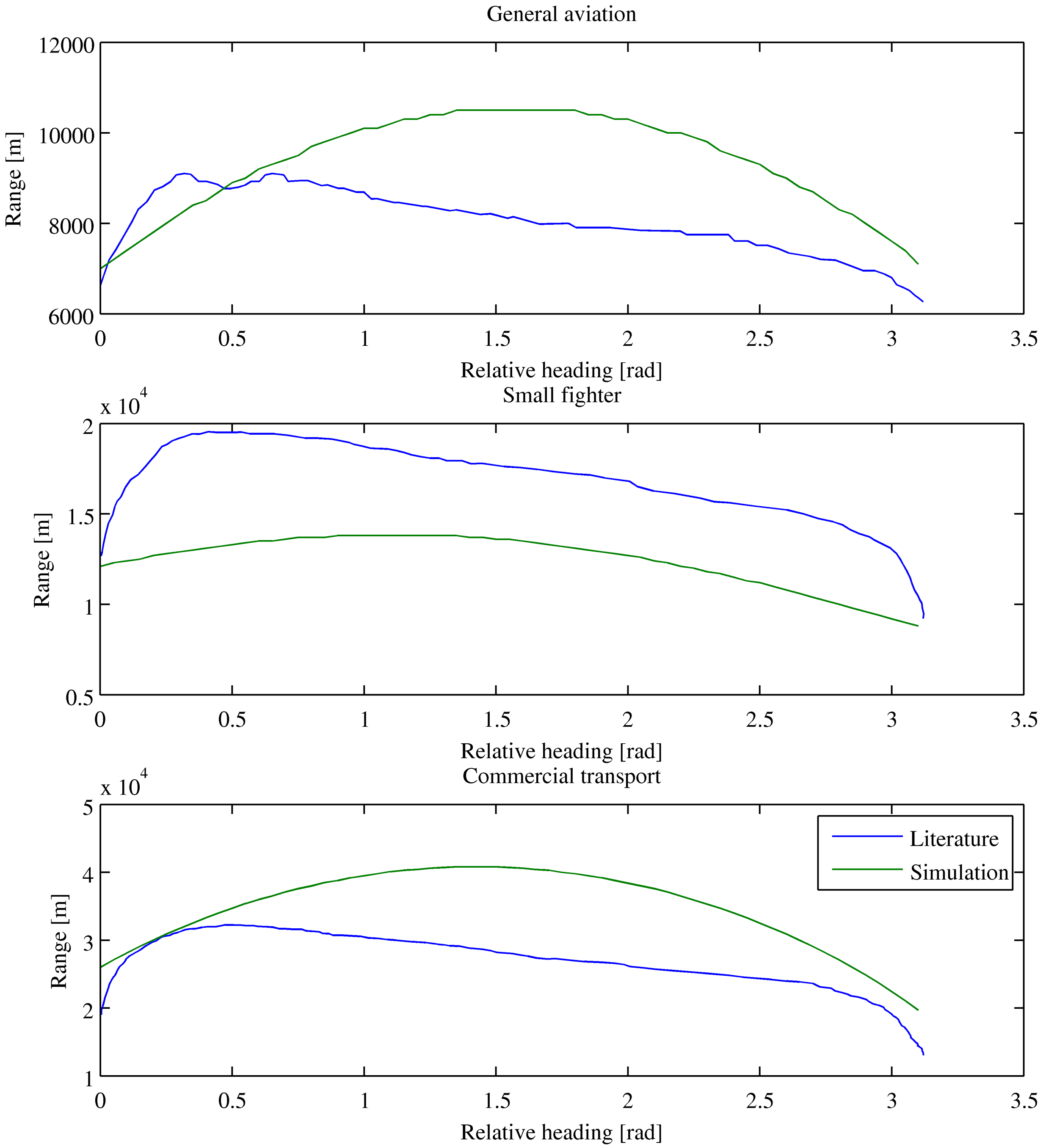
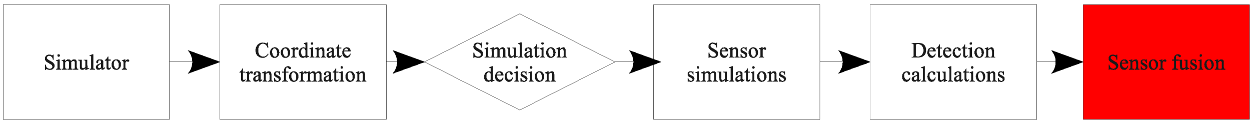


Figure 4.13: Comparison of performance between IR system simulation and literature after refinement

Due to a lack of more extensive data which could be used to assess the performance of the simulator in more detail, the errors presented are the best that can be achieved at this time. Of course the parameters of the simulator could quite easily be changed to output values close to the performance found in literature. Such an arbitrary modification of parameters would not be based on rational and explainable considerations.

4.4. Sensor fusion



Because the sensor fusion algorithm is simply a combination of vectors with the best resolution, the operation can be verified easily. Figure 4.14 shows the output of the sensor fusion block when an intruder aircraft overtakes the ownship at a flight path that lies to the left and is higher than the ownships' flight path. In this scenario the EO system detects the intruder within a range of 5100 m, beyond that the intruder is only detected by the radar system. The range of the intruder is constantly determined by the radar system so this behavior does not change, but the figure clearly shows a decrease in error in the determination of the azimuth and elevation angles if the EO system detects the target.

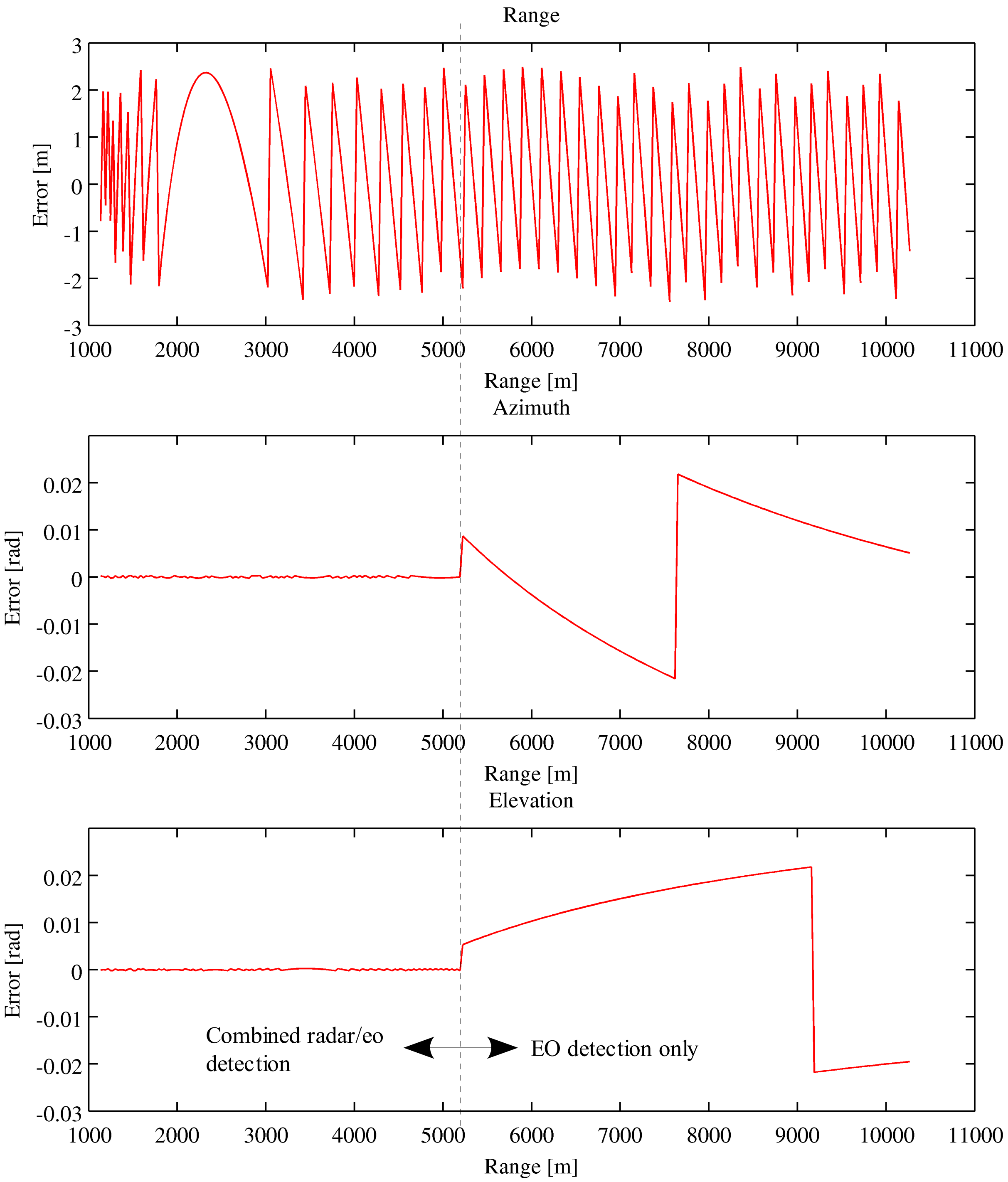


Figure 4.14: Errors in the sensor fusion block output illustrating the fusion of sensor detections with different resolutions

Initially the presented implementation of the sensor fusion algorithm was meant to be a filler to assess the overall system performance without the error introducing effects of this block. When the place of this system in the NLDA simulator is considered, this fusion algorithm might actually be the best for the target system. In the UAV simulator a target is shown on the CDTI and the range information is shown dependent on what sensor resolution. Figures 4.15 and 4.16 show the CDTI in two scenarios. The first where a target is detected by the EO system and range information is not present - the green line emanating from the origin indicates the direction in which the target is detected. The second scenario shows a target when both the radar and EO system detect the target and range information is also available - now the yellow circle indicates the target. Both Figures also show a target that is identified using the TCAS system called “TRF3”, for this target also the altitude and vertical speed are known and displayed.

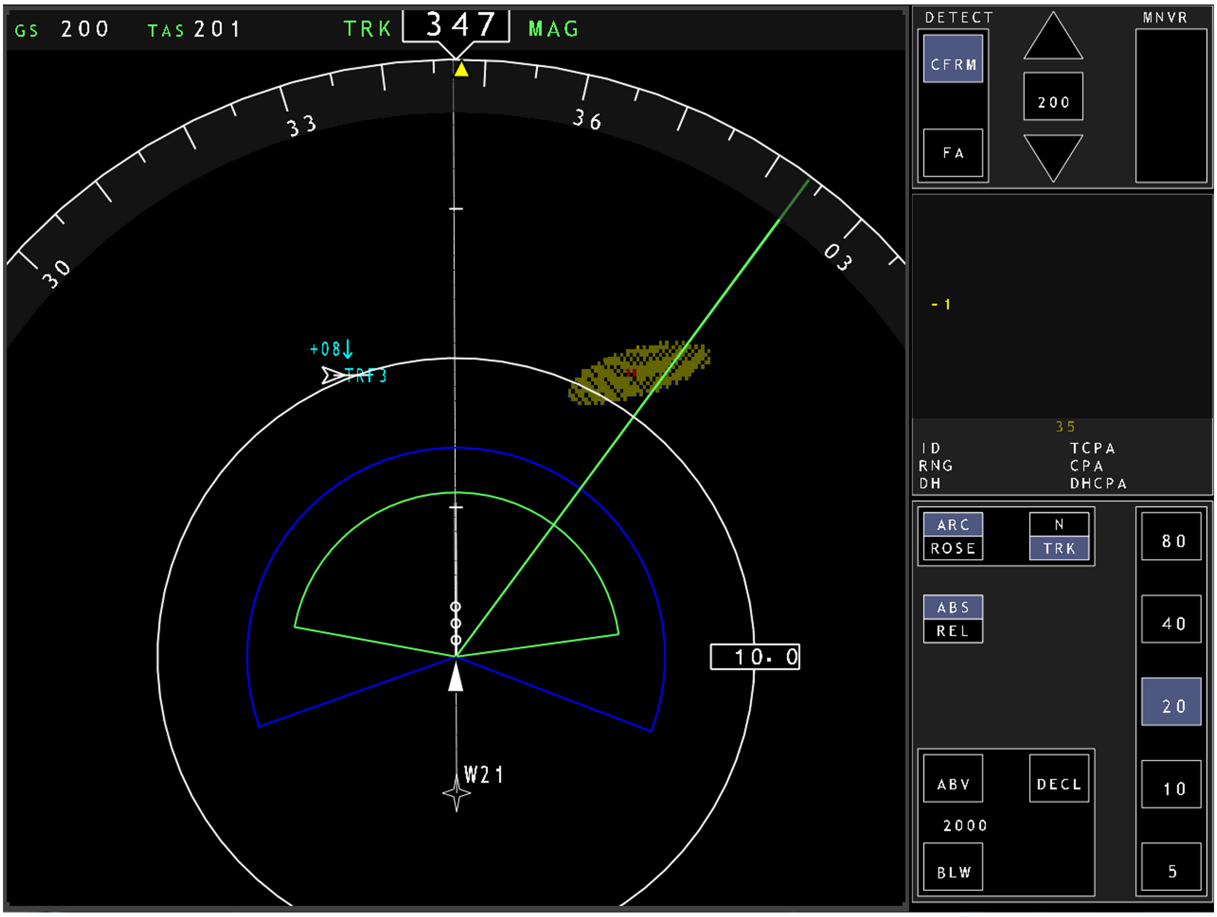
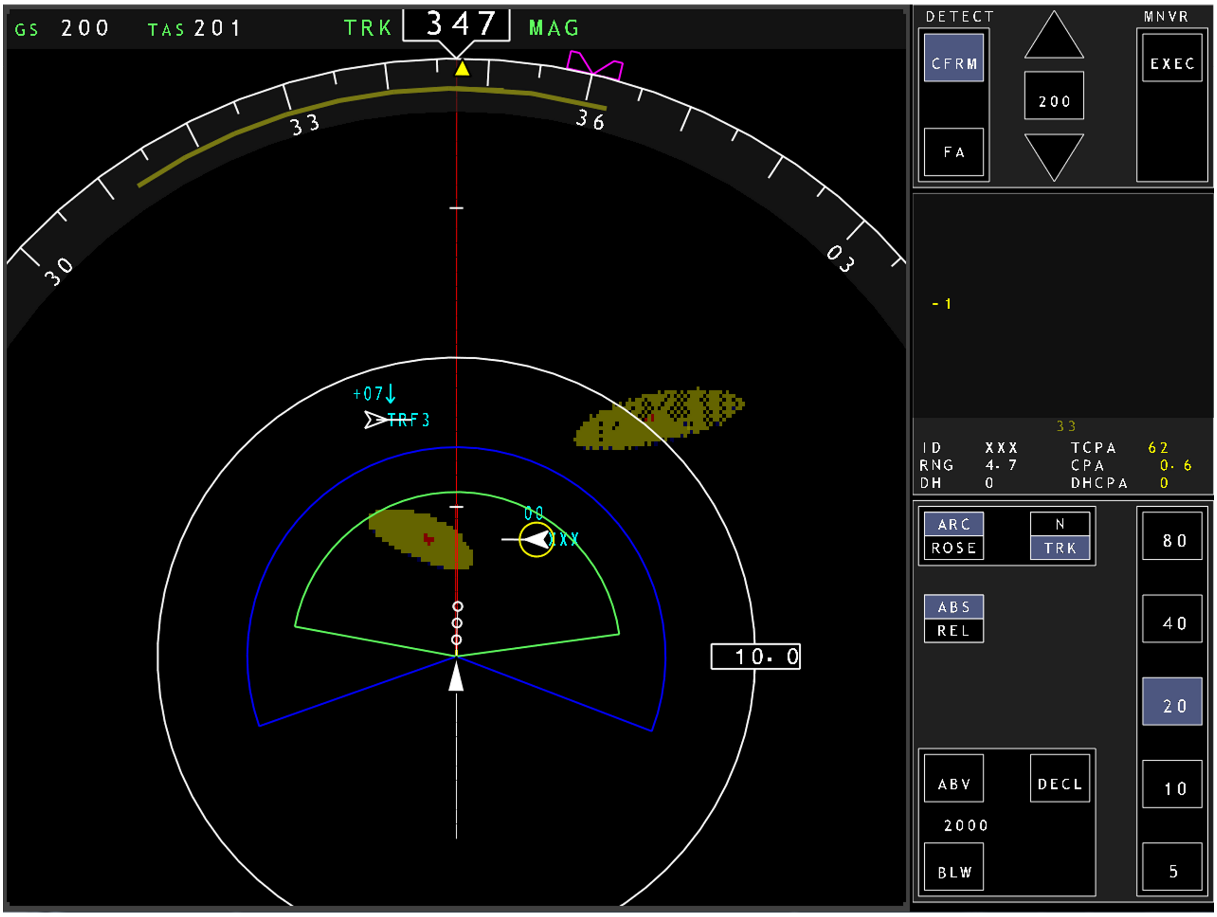


Figure 4.15: CDTI showing EO detection only


GS 200 TAS 201 TRK 347 MAG

ID XXX TCPA 62
RNG 4.7 CPA 0.6
DH 0 DHCPA 0

Figure 4.16: CDTI showing EO/Radar detection combined

To enable the use of the resolution information, the simulator must know which sensor is detecting the target. When a sensor fusion algorithm is used to combine tracks using a statistical method, as used most often in real systems, this information is lost. With the designed system this information is included in the output of the sensor fusion block.

4.5. Total system results

To test the total system numerous scenarios were simulated. This section will display the outcome of some of these simulations to show how parameters that can be changed in the algorithm influence the simulation result. To assess the performance with respect to the computational demand the scenario depicted in Figure 4.21 was timed. The total simulation for these four intruders with 1800 time steps took less than 3 seconds in Matlab running on a 2.1 GHz dual core processor.

A short explanation on the scenarios in Figures 4.17 to 4.22 follows:

- Figure 4.17 shows how the FOV of the three sensors are situated, the ownship is situated in the blue star and is stationary. The intruder is indicated by the line and the simulated detections of the sensors are indicated with red circles. The FOV range only indicates the maximum range set, a target can be missed by the sensor in a closer range.
- Figure 4.18 makes the way the probability of detection of the radar sensor varies with range.
- Figure 4.19 shows what happens when the angular resolution of a sensor is changed.
- Figure 4.20 shows the workings of the fusion algorithm by a scenario containing one intruder traversing two FOVs. All combinations are shown: no detection, detection by the radar sensor, detection by the radar and the EO sensor and detection by the EO sensor alone.
- Figure 4.21 shows what happens when the scan rate is changed.
- When the scan rates for two sensors differ and are not synchronized the detections are not fused. Figure 4.22 shows an intruder trajectory which is detected by both the radar and IR sensor, but only half of the IR detections are fused with the radar detections. This problem should be solved by the target tracker which is not part of this research, but is included in the UAV simulator.

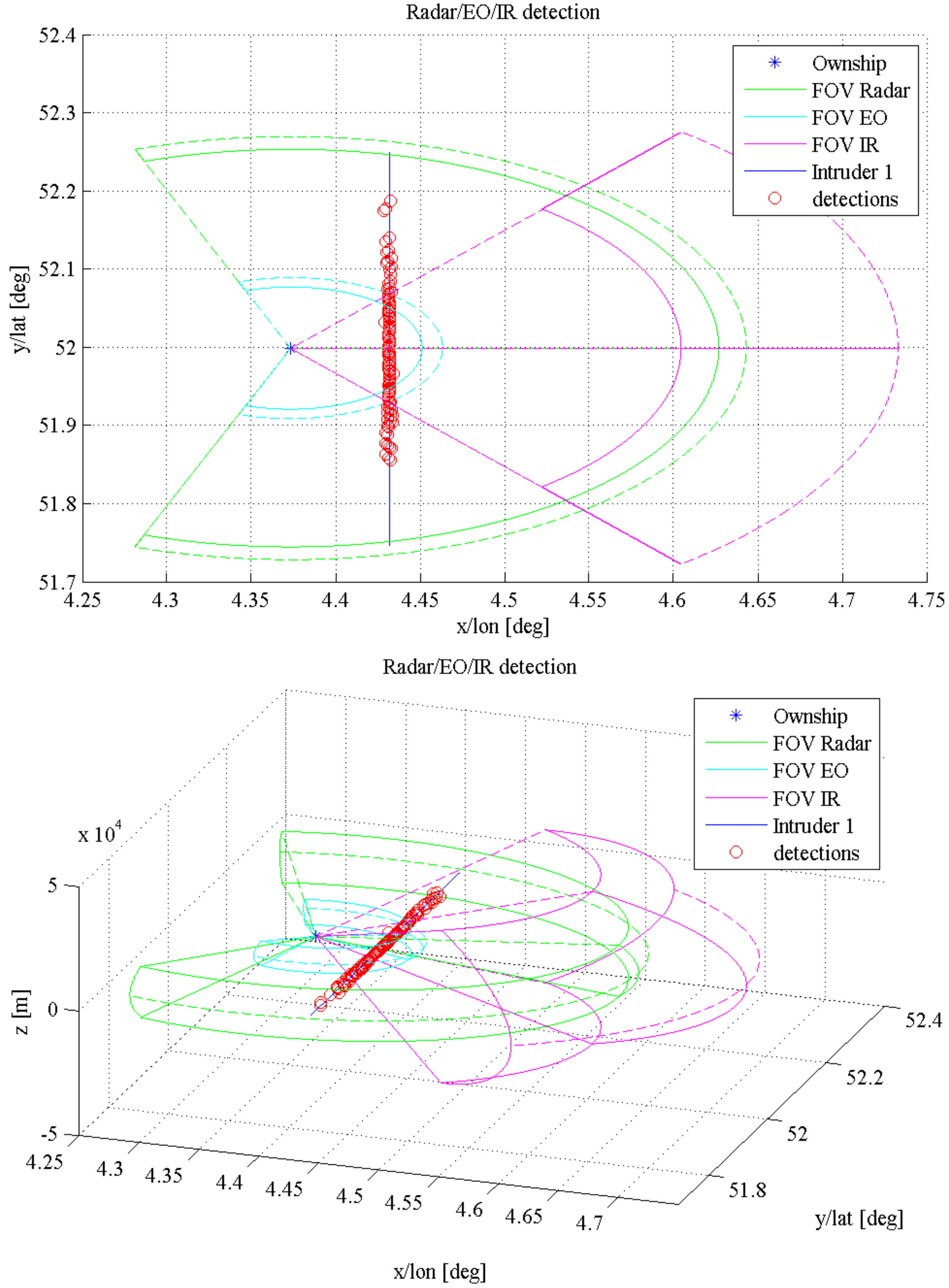


Figure 4.17: Plots showing all three sensor FOVs

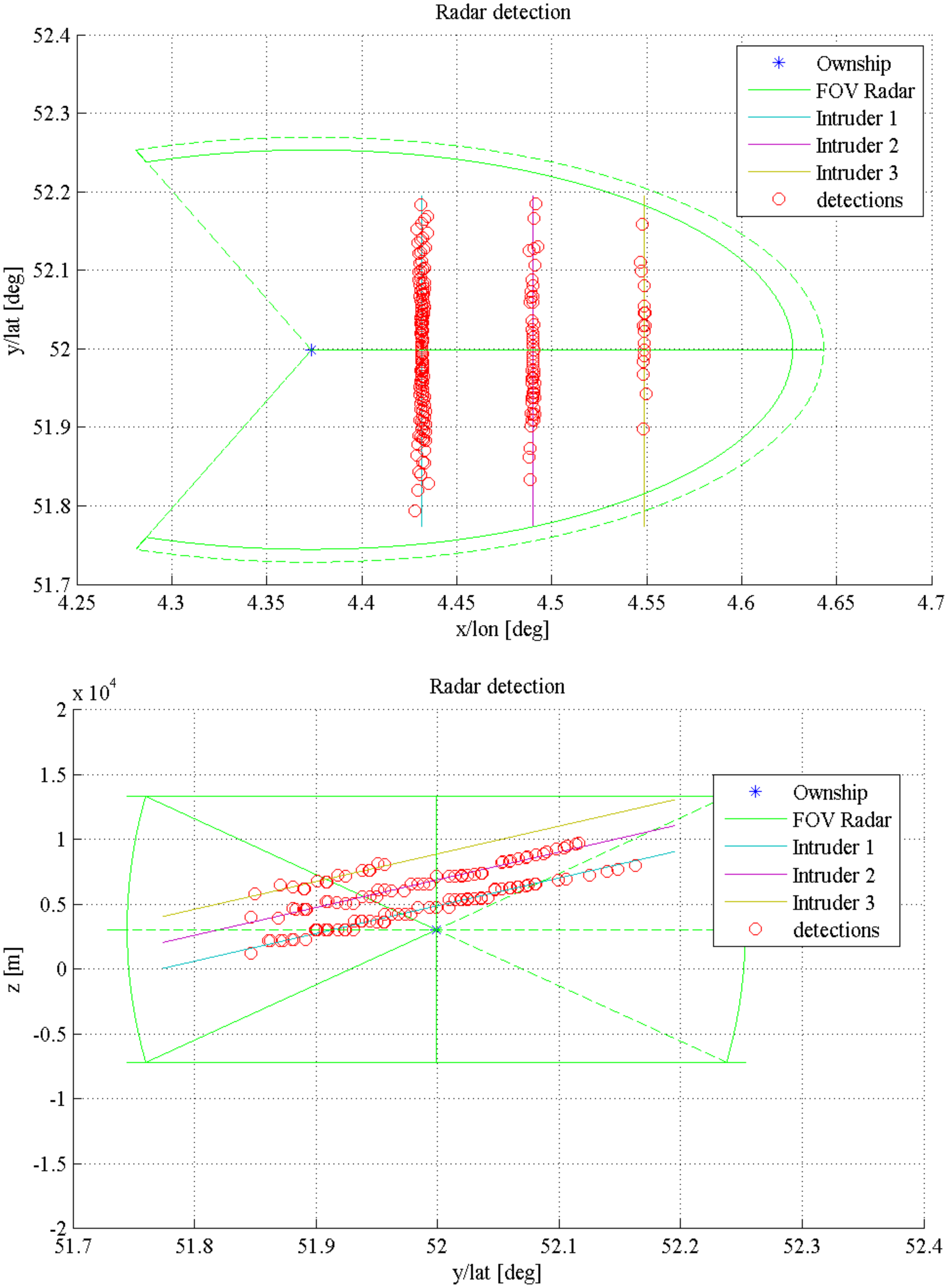
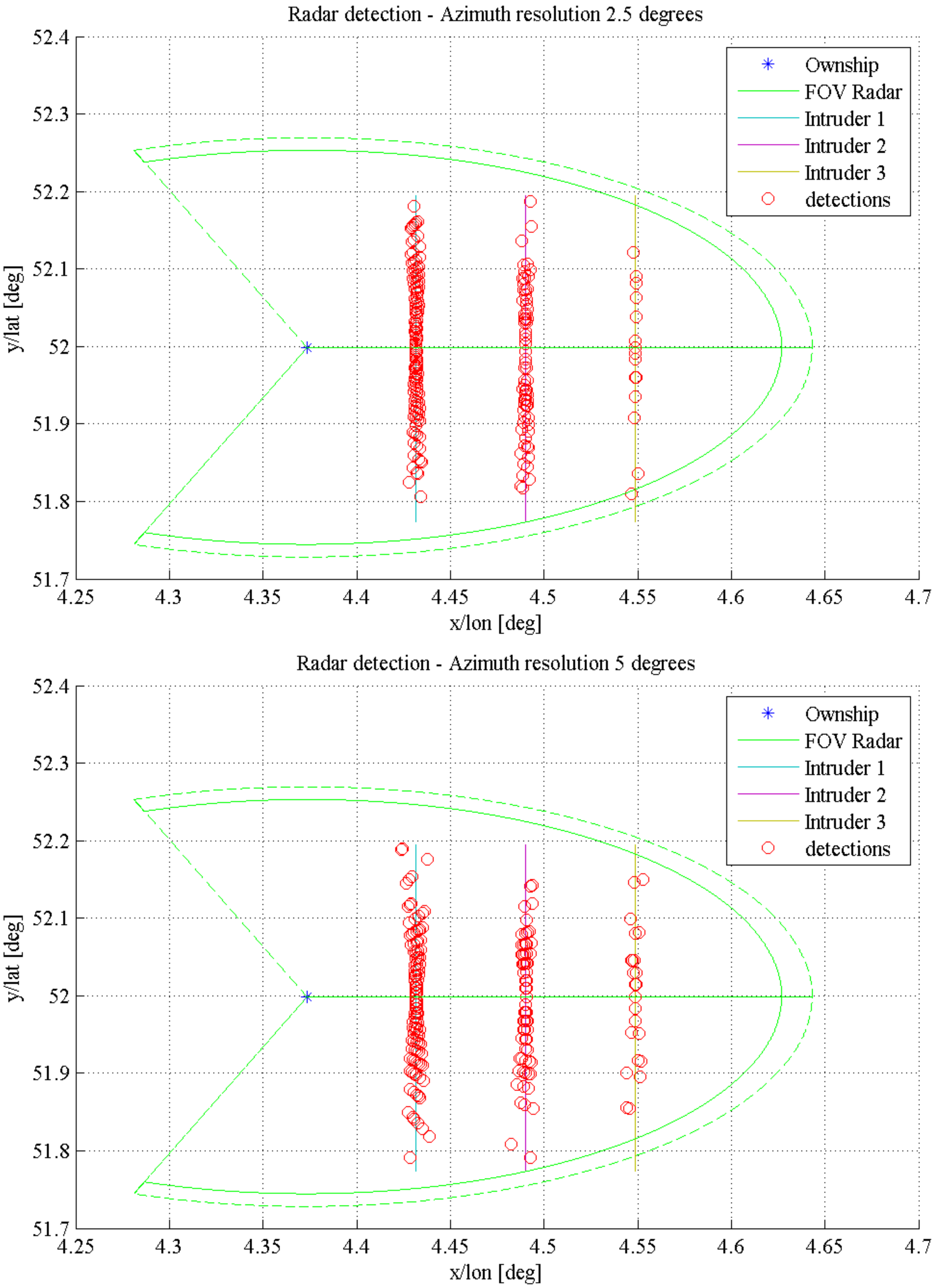


Figure 4.18: Scenario showing the radar probability of detection decreasing with range



Radar detection - Azimuth resolution 2.5 degrees

Radar detection - Azimuth resolution 5 degrees

Figure 4.19: Plots showing the influence of a change in angular resolution

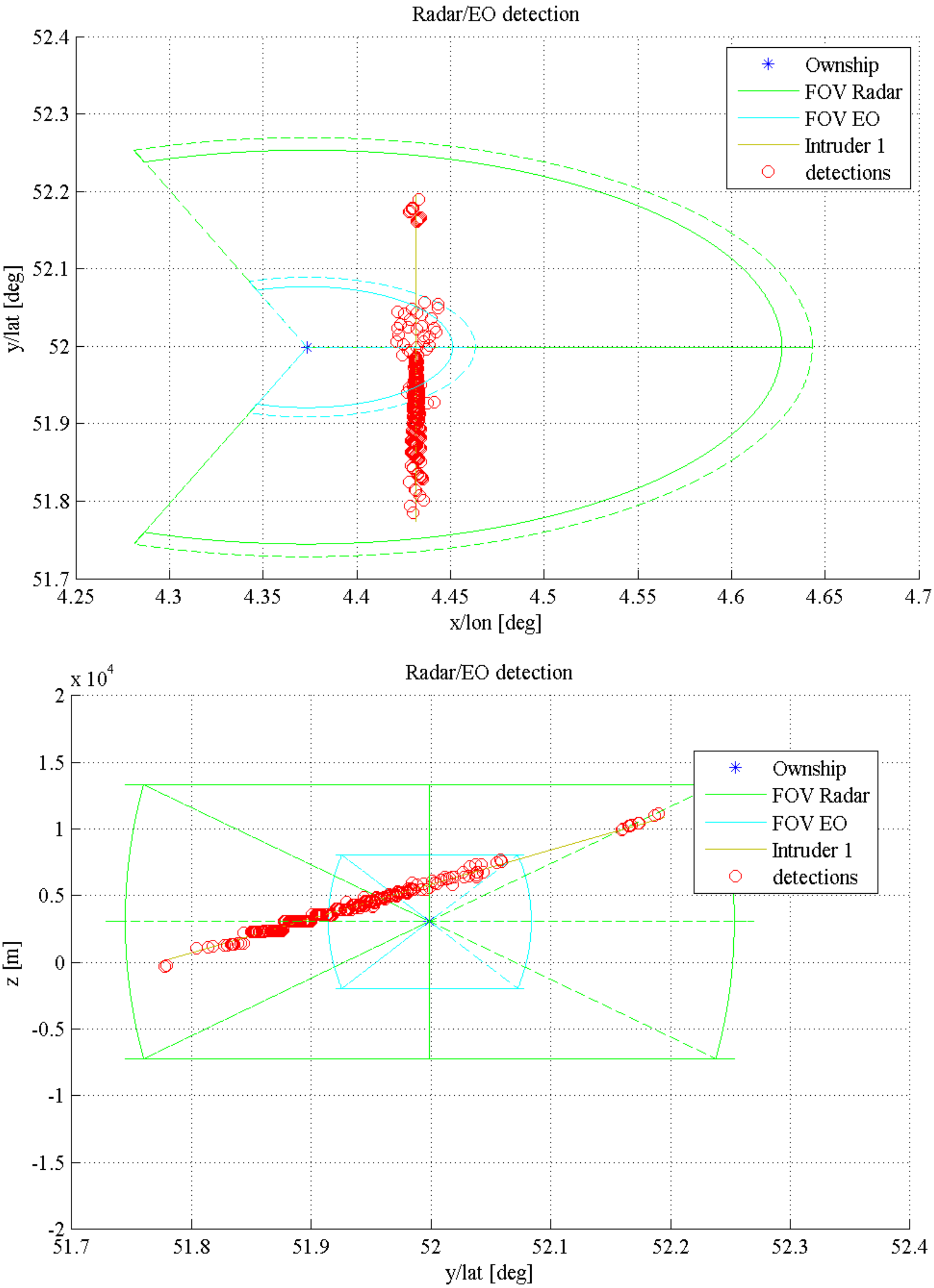


Figure 4.20: Scenario showing the different sensor resolutions

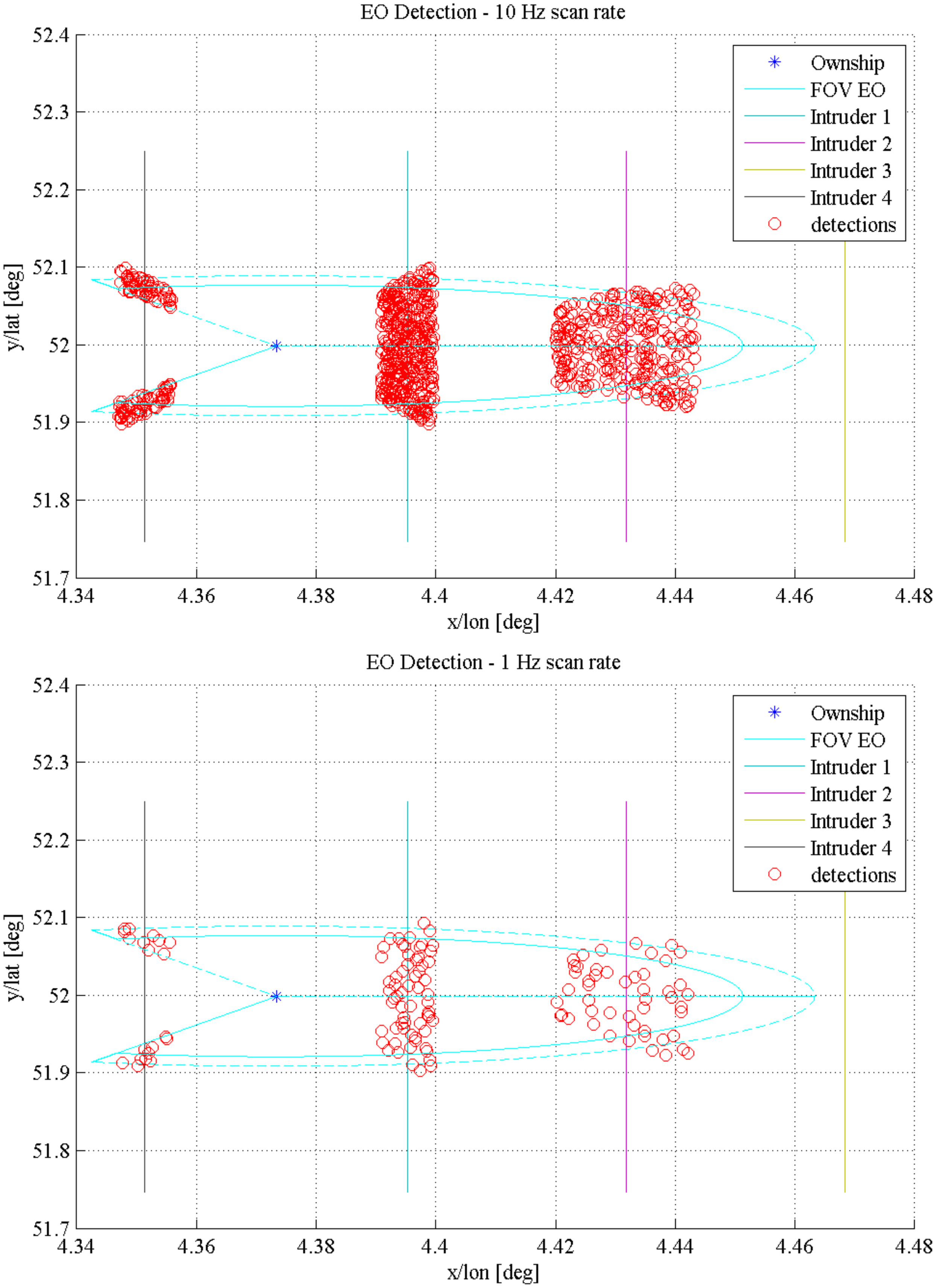


Figure 4.21: Plots showing the influence of scan rate

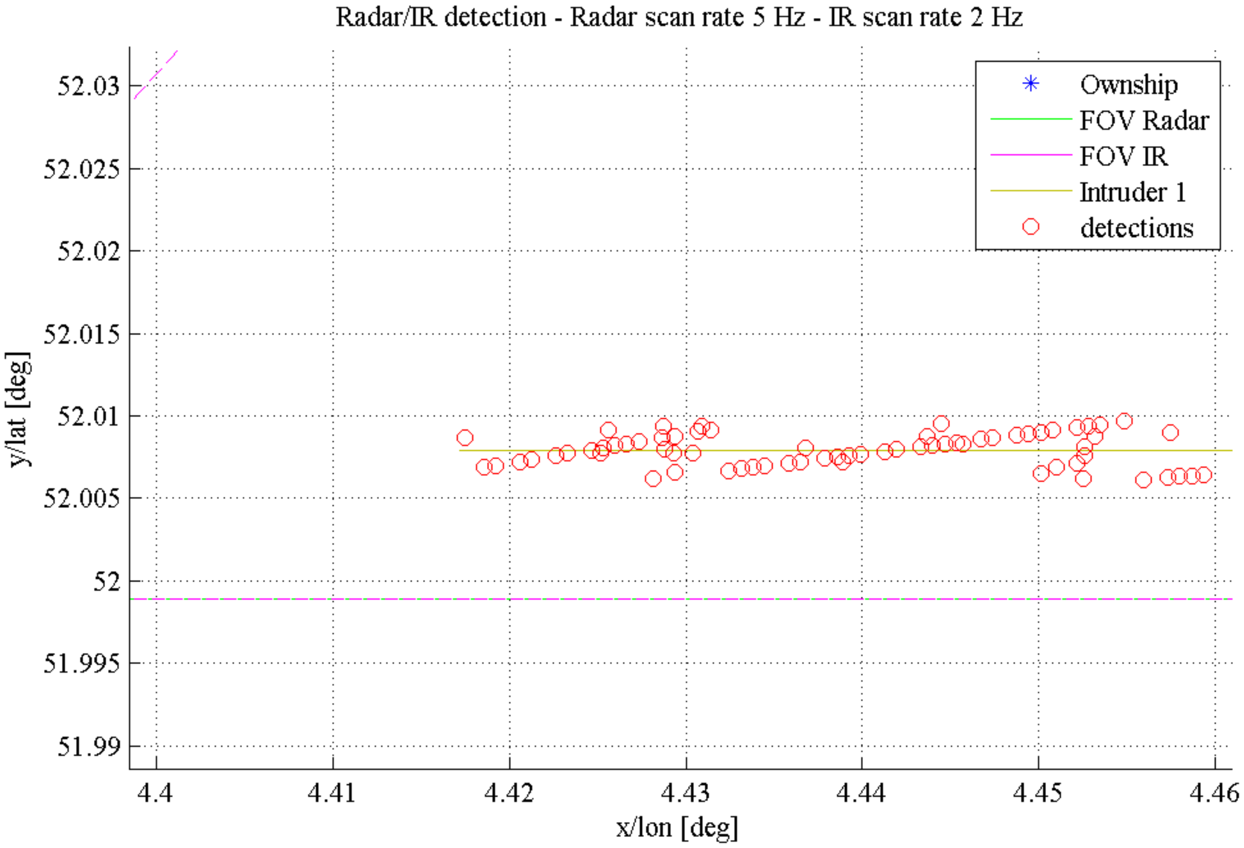


Figure 4.22: Scenario showing with two sensors with different scan rates

5. Conclusion and recommendations

An Unmanned Aerial Vehicle (UAV) needs a system to detect obstacles in its path, for instance other aircraft. To assist in the design of these UAVs, a research simulator exists to simulate its performance. Part of this simulator is the Detect and Sense (DS) system. This thesis aims to improve this component of the existing simulator, as a contribution to the iterative process. So the goal of this research project is to develop a system which is able to provide an existing UAV simulator with an improved detect and sense simulation system.

On the basis of the distinctive requirements for such an endeavor, the results of this research will be discussed. Next the results for the future use of the system will be analyzed and possible integration solutions will be presented. Finally some recommendations will be given for future research based on improvement areas identified in this research project.

5.1. System requirements and results

5.1.1. System complexity

The first requirement was that the level of complexity of the system should be such that it could run on the target simulator hardware, mostly standard computer hardware. With this constraint the design choices of the simulation had to reflect as well as possible the behavior of the actual systems. This resulted in the choice for a stochastic model of the radar system and a simplified aircraft geometry model for the simulations of the optical systems. It is shown (in Section 4.4.) that the computational demands of the resulting system was low. Therefore these choices do not pose problems for complying with the hardware requirements on the target system. The process of integration of the results of this project with that system will be discussed later in this chapter.

5.1.2. System level design and design methodology

The simulator was to be designed using Hazlett's design methodology, which basically states that the design has to have a strong and simple basis. When the basic framework is complete, the different subsystems and their interfaces can be refined in an iterative fashion.

This research project follows this method, by first defining the block diagram of the simulation system (Figure 3.1) and their interfaces (Chapter 3.). These interfaces are defined so that they are compatible with the target system, in order to make the transition from the development environment to the simulator possible.

The next step was to design the first (simple) simulation algorithms (Section 3.5.). After testing the subsystems as well as the overall system for its performance and fidelity, the results were reviewed and improvements were introduced to increase the quality of the simulation (in Chapter 4.).

This approached resulted in an asynchronous development of the different subsystems, in which each of them could be tested in a flexible and thorough manner. Future development of the subsystems can be done quite easily because their place in and interfaces with the other subsystems are well defined. Some suggestions on further development are given in Section 5.3.

5.1.3. Fidelity of the simulation

Because the traffic simulator is not part of the system which will be part of the final implementation, the results of this block are not discussed. Also the workings of the coordinate transform and simulation decision blocks are trivial and need no review, their results are found to be correct during the simulations.

The core of this research project are the simulations of the sensors which have to detect an object in the path of a UAV. Three types of sensors are used: radar sensors, optical sensors and infrared sensors.

The results of the radar simulation were compared to the data obtained by the flight tests performed under the NASA ERAST program in Section 4.1.3.. All seven test aircraft were classified on the basis of their size, engine configuration and construction material in one of three classes. Each class was given an average radar cross-section and the average detection range as found with the flight tests was compared to the simulated detection range with a given detection probability. The result is that the detection probabilities were off by 1,1%, 1,7% and 3,3% for the three classes.

Electro-optical sensors are simulated by calculating three parameters: the static signature, the dynamic signature and the contrast. For the computation of static signatures a simple geometric aircraft model was developed (Section 3.5.2.). Again three classes were defined to represent aircraft targets: general aviation, fighter aircraft and commercial transport aircraft. The model was verified (in Section 4.2.1.) by the use of a high fidelity flight simulator and the results show that the mean errors were between zero and ten percent and the maximum errors between zero and 20 percent. The maximum error of 20 percent would translate in a 600 meters error in detection distance. Unfortunately the data from the literature which was necessary for the verification of the EO system were not adequate. But the simulation system is designed in such a way that parameters can be adjusted easily to adept the system results to new available test data.

The electro-optical and infrared system share many algorithms because they are essentially the same from a system engineering perspective an differ only in operating in a different part of the spectrum. To account for differences in detectable features the aircraft geometry model was altered slightly (Section 4.3.2.). Even more than with the electro-optical system, the verification data for the infrared system was lacking. Therefore on a very coarse scale the performance in detection range for the three aircraft classes can be said to have a maximum error of 25%, 50% and 30%. Again: this system can be tuned when more test data are available.

The final subsystem that has been developed is the sensor fusion algorithm. This was designed as a simple algorithm: it chooses from and combines the inputs with the best resolution to one output vector. It could be shown that this solution performed as desired (Section 4.4.), and thus can play a part in the final implementation in the UAV simulator.

All subsystems combined constitute a working simulation that works utilizing all the discussed subsystems. The results of this total system are evaluated with the conclusion that the simulations were computationally very efficient: a scenario containing four intruders with the duration of 1800 time steps took a little more than 3.5 seconds of time spent on the simulation of the whole system.

5.2. Integration in production environment

Most systems and algorithms discussed in this thesis were designed and tested using the software package Matlab. The target UAV simulator is implemented in the C programming language, so conversion would be necessary. Two methods can be used to convert the developed code from the Matlab scripting language to C: the code can be manually rewritten and integrated, or the code can be compiled by the Matlab compiler which is available under the Delft university license.

Both methods have advantages and disadvantages. Manually rewriting the code ensures that the resulting code is easier to understand and adaptable when problems arise during integration or when further improvements are needed. Rewriting the code of course takes effort and time and requires knowledge about both programming languages as well as about the developed code. In order to assist this eventual translation of the developed Matlab scripts, Appendix A lists the relevant functions and their in- and outputs. Furthermore the developed algorithms do not require additional libraries: all functions are available in the C libraries already used in the production environment.

5.3. Recommendations for future research

As was stated, the result of this research is seen as one iteration in the design process for the UAV simulator. Naturally more steps will follow by instigating more research to improve the simulator fidelity. To help future research the areas of improvement which became apparent during the development and testing will be listed next.

Within the frame of requirements for the simulation system, the primarily focus for improvement seems to be increasing the fidelity of the simulation without a corresponding increasing computational demand. As those demands are shown to be very low in the designed system, there seems to be room in future research for increasing the fidelity substantially without reaching computational limits. In this perspective some recommendations can be suggested.

5.3.1. Optical flow algorithm

The choice to simulate the optical flow algorithms instead of just implementing one fully is made on the basis of the computational demands constraint. Even with custom hardware like FPGA's it is difficult to achieve a frame rate higher than one Hertz on a resolution that is high enough for the application. It might however be possible to implement these algorithms by using video hardware used in most computers - the hardware manufacturer Nvidia for example provides the CUDA architecture [29]. This provides developers with the possibility to use the graphics processing unit (GPU) for calculations. The GPU is optimized for processing large amounts of data in parallel which is necessary for efficient optical flow calculations.

5.3.2. Contrast

A fundamental aspect in the simulation of all three sensors is the contrast between the target and its background. In the developed simulation this aspect is not addressed in full nuance. But the results indicate that more precise determination of the levels of contrast might yield a substantial improvement of the fidelity of the simulation. In the electro-optical simulation this would imply that the viewing aspect, the paint scheme and the background need to be factored in the simulation in order to calculate the contrast of the target with respect to the background. In the same way the background can be included in the infrared sensor simulation.

In the radar simulation the background scattering could be included to improve the accuracy of the calculated contrast. Several very complicated methods to achieve this are available, each of them demand a lot of money, time and computational power. Maybe a first less demanding approach could be to expand the number Swerling of models as suggested by Schnidman in [30]. He proposes to model the clutter not as a zero mean Gaussian (as done in this research project), but as either a constant or a fluctuating mean Gaussian process. This necessitates at least two more models to be defined and the corresponding equations need to be implemented.

Another area of possible improvement in the radar simulator can be found in the RCS calculation. At the moment the viewing aspect is neglected – introducing this calculation should improve the fidelity of the radar sensor simulation. Again, this can be done in a very complicated way with excellent fidelity but with the same downsides in terms of demands. In [31] for example Shirman gives some methods of determining the RCS of aerial targets with different computational requirements.

5.3.3. Atmospheric attenuation

Clear air attenuation, rain and fog for example can have a large influence on the received SNR and therefore the performance of the radar sensor. The used simulation method (the link budget model) allow for the addition of the effects of atmospheric attenuation without much rewriting of the code. Because rain and fog also substantially reduce the performance of the optical sensors, it might be worthwhile to incorporate these factors in the simulation of all sensors.

5.3.4. False alarms

False alarms can for example be generated by the optical sensors because of clouds or the terrain background. The establishment of thresholds in the detection algorithms of sensor systems define the trade-off between false alarm rate and the probability of detection. Currently the sensor simulations only take the latter into account while the former is neglected. For practical use of UAVs it is evidently useful to also simulate the false alarms, especially if they could have a noticeable effect on the tracking filters used in the simulator.

5.3.5. Influence of the tracking filter on the detection algorithm

If a tracking filter predicts where a target will probably be in the near future and this information is shared with the sensor detection algorithm, the detection probability for that target can be increased. The radar simulation already provides such a feature in the form of an increase in the pulse repetition frequency. This feature is not included in the simulation of the optical sensors. There is not much information available which suggests that this method is used in optical sensor systems and what performance increase can be achieved. It should however not be to difficult to implement this feature in the simulator by for example making the detection threshold variable.

Bibliography

[1] FAA, "FAR Part 91 Sec. 91.113 effective as of 09/01/2004," *Code of Federal Regulations*.

[2] K. Davis, *Interim Operational Approval Guidance 08-01*, Aviation Safety Unmanned Aircraft Program Office AIR-160, FAA, 2008.

[3] J. Tadema and E. Theunissen, "Integrated Conflict Prediction and Multi-Dimensional Prevention/Resolution," *Infotech@Aerospace*, Seattle, Washington: AIAA, 2009.

[4] E. Theunissen and J. Tadema, "A Simulation Approach for Evaluating Operator-in-the-Loop DSA concepts," Honolulu, Hawaii: AIAA, 2008.

[5] A.D. Zeitlin, "Technology Milestones–Detect, Sense & Avoid for Unmanned Aircraft Systems." *Infotech@Aerospace*, Rohnert Park, California: AIAA, 2007.

[6] G. Koeners, M. De Vries, A. Goossens, J. Tadema, and E. Theunissen, "Exploring Network Enabled Airspace Integration Functions for a UAV Mission Management Station," *25th Digital Avionics Systems Conference, 2006 IEEE/AIAA*, 2006, pp. 1-11.

[7] J.A. Hazlett, "MODELING AND SIMULATION AND C4I: THE QUEST FOR DOMINANT BATTLESPACE AWARENESS," 1996, pp. 31–36. *Flight Simulation Technologies Conference,* San Diego, CA, USA: AIAA, 1996

[8] G. Fasano, D. Accardo, A. Moccia, and L. Paparone, "Airborne Multisensor Tracking for Autonomous Collision Avoidance," 2006, pp. 1-7.

[9] D. Accardo, A. Moccia, G. Cimmino, and L. Paparone, "Performance Analysis and Design of an Obstacle Detection and Identification System," *Infotech@ Aerospoace*, 2005, pp. 1-18.

[10] A. Joulia and C.L. Tallec, "UAVs What and How to Sense To Avoid What," *Infotech@Aerospace*, Rohnert Park, California: AIAA, 2007.

[11] G. Fasano, D. Accardo, C. Carbone, U. Ciniglio, F. Corraro, and S. Luongo, "Multisensor based Fully Autonomous Non-Cooperative Collision Avoidance System for UAVs," *Infotech@Aerospace*, Rohnert Park, California: AIAA, 2007.

[12] B. Mahafza, *MATLAB simulations for radar systems design*, Boca Raton FL: CRC Press/Chapman & Hall, 2004.

[13] O. Shakernia, W. Chen, S. Graham, J. Zvanya, A. White, N. Weingarten, and Vincent Raska, "Sense And Avoid (SAA) Flight Test and Lessons Learned," *Infotech@Aerospace*, Rohnert Park, California: AIAA, 2007.

[14] J. McCalmont, J. Utt, and M. Deschenes, "Detect and avoid technology demonstration," *1st UAV Conference,* Portsmouth, Virginia: AIAA, 2002.

[15] J. Utt, J. McCalmont, M. Deschenes, and M. Taylor, "Development of a Sense and Avoid System," *Infotech@Aerospace*, Arlington, Virginia: AIAA, 2005.

[16] O. Shakernia, W.Z. Chen, and V.M. Raska, "Passive Ranging for UAV Sense and Avoid Applications," *Infotech@ Aerospoace*, 2005, pp. 1-10.

[17] W. O'Neil and W. Chen, "See and Avoid Sensor System Design Part I - Coverage Modeling," San Diego, California: AIAA, 2003.

[18] R. Bernier, M. Bissonnette, and P. Poitevin, *DSA Radar - Developement Report*, Baltimore, USA: Amphitech, 2005.

[19] R. Wolfe, "Nasa erast non-cooperative dsa flight test," *Proceedings of AUVSI Unmanned Systems 2003*.

[20] Y.K. Kwag, M.S. Choi, C.H. Jung, and K.Y. Hwang, "Collision Avoidance Radar for UAV," *International Conference on Radar*, Shanghai, China: IEEE, 2006, pp. 1-4.

[21] Kwag, Y.K.; Kang, J.W.; , "Obstacle awareness and collision avoidance radar sensor system for low-altitude flying smart UAV," *Digital Avionics Systems Conference, 2004. DASC 04. The 23rd* , vol.2, no., pp. 12.D.2- 121-10 Vol.2, 24-28 Oct. 2004

[22] Y.K. Kwang and Y.H. Kwang, “Performance Simulation of Radar Sensor Based Obstacle Detection and CA for Smart UAV,” 2005.

[23] T.T. Kolehmainen, “Simulation of imaging system's performance,” *Proceedings of SPIE*, San Diego, CA, USA: 2004, pp. 204-212.

[24] B. Andresen and Society of Photo-optical Instrumentation Engineers., *Infrared technology and applications XXII : 8-12 April, 1996, Orlando, Florida*, Bellingham Wash. USA: SPIE, 1996.

[25] “http://www.airliners.net/aircraft-data/stats.main?id=66,” *Beech 99 Airliner | Airliners.net*, Oct. 2009.

[26] J. Casanova, http://www.boeing.com/commercial/airports/acaps/787sec2.pdf, “*787 Airplane Characteristics for Airport Planning,*” Oct. 2009

[27] “http://www.lockheedmartin.com/products/f16/index.html,” *F-16 Fighting Falcon | Lockheed Martin*, Oct. 2009.

[28] P. Swerling, “Probability of detection for fluctuating targets,” *Information Theory, IRE Transactions on*, vol. 6, 1960, pp. 269-308.

[29] “http://www.nvidia.com/object/cuda_home.html,” *CUDA Zone -- The resource for CUDA developers*.

[30] D. Shnidman, “Radar detection probabilities and their calculation,” *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 31, 1995, pp. 928-950.

[31] Y. Shirman, *Computer simulation of aerial target radar scattering, recognition, detection, and tracking*, Boston: Artech House, 2002.

Appendix A. Function reference

The tables in this appendix give the names and in- and outputs of the functions used for the simulations. When the value in the *unit* column consist of multiple units, this means that the given in- or output is an array.

Table A.1: Overview of functions and in- and outputs

Function	Type	Name	Units	Description
runscript.m	none		[ ]	This script calls the initialization routines, runs the simulator and for every plane at every time calls the simulate routine and saves the result
initradar.m	none		[ ]	Saves the set radar sensor properties to a globally accessible variable
initeo.m	none		[ ]	Saves the set EO sensor properties to a globally accessible variable
initir.m	none		[ ]	Saves the set IR sensor properties to a globally accessible variable
simulate	input	lla_ownship	[rad,rad,m]	The geodetic coordinates of the ownship.
	input	att_ownship	[rad,rad,rad]	The angles that define the attitude of the ownship
	input	lla_intruder	[rad,rad,m]	The geodetic coordinates of the intruder.
	input	att_intruder	[rad,rad,rad]	The angles that define the attitude of the intruder
	output	detected	[rad,rad,m]	Polar coordinates indicating where a target is detected by the sensors with respect to the ownships' position.
simulateRadar	input	p	[rad]	Heading from the ownship to the intruder (see Fig. 3.3)
	input	q	[rad]	Pitch from the ownship to the intruder
	input	R	[m]	Range from the ownship to the intruder
	output	p	[rad]	Perceived heading from the ownship to the intruder
	output	q	[rad]	Perceived pitch from the ownship to the intruder
	output	R	[m]	Perceived range from the ownship to the intruder
	output	conf	[ ]	Confidence number to indicate how strong the detection is

Table A.2: Overview of functions and in- and outputs (continued)

Function	Type	Name	Units	Description
simulateEO	input	p	[rad]	Heading from the ownship to the intruder (see Fig. 3.3)
	input	q	[rad]	Pitch from the ownship to the intruder
	input	R	[m]	Range from the ownship to the intruder
	input	phi_rel	[rad]	Relative roll of the intruder
	input	theta_rel	[rad]	Relative pitch of the intruder
	input	psi_rel	[m]	Relative yaw of the intruder
	output	p	[rad]	Perceived heading from the ownship to the intruder
	output	q	[rad]	Perceived pitch from the ownship to the intruder
	output	R	[m]	Perceived range from the ownship to the intruder
	output	conf	[ ]	Confidence number to indicate how strong the detection is
simulateIR	input	p	[rad]	Heading from the ownship to the intruder (see Fig. 3.3)
	input	q	[rad]	Pitch from the ownship to the intruder
	input	R	[m]	Range from the ownship to the intruder
	input	phi_rel	[rad]	Relative roll of the intruder
	input	theta_rel	[rad]	Relative pitch of the intruder
	input	psi_rel	[m]	Relative yaw of the intruder
	output	p	[rad]	Perceived heading from the ownship to the intruder
	output	q	[rad]	Perceived pitch from the ownship to the intruder
	output	R	[m]	Perceived range from the ownship to the intruder
	output	conf	[ ]	Confidence number to indicate how strong the detection is
inFOV	input	p	[rad]	Heading from the ownship to the intruder (see Fig. 3.3)
	input	q	[rad]	Pitch from the ownship to the intruder
	input	R	[m]	Range from the ownship to the intruder
	input	FOV	[rad,rad,rad, rad,m]	The field of view for the sensor in azimuth minimum, maximum, elevation minimum, maximum and range respectively
	output	inrange	[ ]	Boolean indicating whether target is in the FOV

1. *Table A.3: Overview of functions and in- and outputs (continued)*

Function	**Type**	**Name**	**Units**	**Description**
sensorfuse	input	detection	[rad,rad,m]	Array of detections containing the polar coordinates resulting from all sensor simulations
	output	obspoints	[rad,rad,m]	Vector pointing to the target
addplane	input	sim	[ ]	Database containing the simulation parameters to which the information needs to be added
	input	name	[ ]	Optional text identifier for a simulated aircraft
	input	type	[ ]	Optional text identifier for the type of aircraft
	input	position	[deg,deg,m]	Position vectors where the aircraft is at every time instance in the simulation
	input	orientation	[rad,rad,rad]	Orientation vectors describing the orientation of the aircraft at every time instance
	input	rcs	[dBsm]	The radar cross section of the target
	input	swerlingtype	[ ]	The Swerling model type to be used for the target in the radar simulation (can be 0, 1 or 5)
	input	size	[ ]	The size of the target used for the EO and IR simulation to determine which geometric model parameters need to be used
	input	contrast	[ ]	Contrast of the target for the EO simulation system
	input	contrastIR	[ ]	Contrast of the target for the IR simulation system
	output	sim	[ ]	Same database as passed as input updated to include the new aircraft