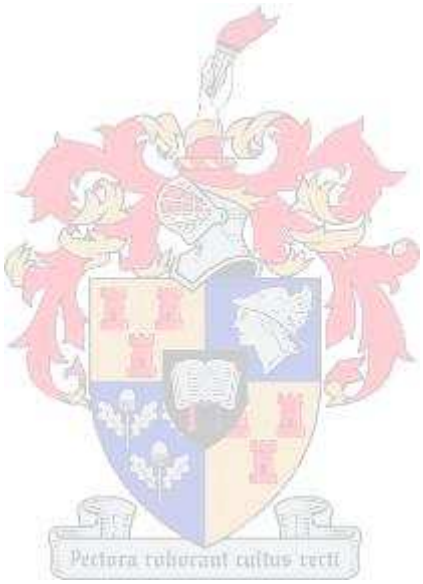
Development and Demonstration of a Performance Evaluation Framework for Threat Evaluation and Weapon Assignment Systems

Martin Louw Truter



Thesis presented in partial fulfilment of the requirements for the degree of
Master of (Industrial) Engineering
in the Faculty of Engineering at Stellenbosch University

Supervisor: Prof JH van Vuuren

March 2016

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: March 1, 2016

Abstract

In a military air-defense environment, ground-based weapon systems are responsible for defending ground assets against aerial threats. These weapon systems are employed in conjunction with an array of sensor systems which are capable of detecting and tracking aerial threats, and providing information for determining the level of danger that the threats pose to the defended system. In this context, the purpose of a *Threat Evaluation and Weapon Assignment* (TEWA) system is to provide decision support to the human operators who are tasked with assigning weapon systems to counter the aerial threats.

The TEWA system typically assigns appropriate system threat values to each of the aerial threats which indicates the level of danger they pose to the defended system. These system threat values are used, in turn, when seeking to optimise the allocation of weapon systems to threats in such a way that the weighted cumulative survival probability of the aerial threats is minimised. These weapon allocations are suggested to a human operator for implementation via a human machine interface.

A large number of TEWA systems are already in use around the world, but due to the confidential nature of this research area, descriptions of the working of these systems are typically not available in the open literature. Despite the critical role these systems play in the current, evolving network-centric warfare environment, there exists no generic framework in the literature for evaluating the performance of TEWA systems.

The work contained in this thesis therefore adds to the South African TEWA knowledge base by determining the extent to which current TEWA-related research in a ground-based air-defense environment forms a coherent foundation from which further system development and performance evaluation can continue. This broad research aim is achieved by developing a performance evaluation simulation framework for TEWA systems and demonstrating the feasibility of locally developed TEWA algorithms. A system-of-systems simulation modelling approach is adopted in the design of this framework.

Using the framework, limitations present in the TEWA algorithms are identified and mitigation strategies are suggested. These strategies include a novel threat value fusion methodology, an alternative weapon system modelling approach and the implementation of a genetic algorithm for solving the weapon allocation problem approximately. Design requirements for an effective human machine interface are also described in some detail and several TEWA system performance metrics are suggested. The working of the framework is finally demonstrated in the context of a comprehensive, near-realistic, but hypothetical, ground-based air-defense scenario.

Uittreksel

In 'n militêre lugafweeromgewing word grond-gebaseerde wapenstelsels gebruik om grondbates teen lugbedreigings te beskerm. Hierdie wapenstelsels word in oorleg met 'n aantal sensorstelsels aangewend wat daartoe in staat is om lugbedreigings op te spoor en te volg, en inligting te verskaf waarvolgens die vlak van gevaar wat hierdie bedreigings vir die verdedigde stelsel inhou, bepaal kan word. In hierdie konteks is die doel van 'n *Bedreigingsafskatting-en-wapentoewysing* (TEWA) stelsel om besluitsteun aan menslike operateurs te bied wat daarvoor verantwoordelik is om wapenstelsels aan lugbedreigings toe te ken.

Die TEWA stelsel heg tipies 'n gepaste stelsel-wye bedreigingswaarde aan elkeen van die lugbedreigings wat die vlak van gevaar aandui wat hul met betrekking tot die verdedigde stelsel inhou. Hierdie stelsel-wye bedreigingswaardes word dan gebruik in die soeke na optimale toewysings van wapenstelsels aan die bedreigings om sodoende die geweegde, geakkumuleerde oorlewingswaarskynlikheid van die lugbedreigings te minimeer. Wapentoewysingsvoorstelle word deur middel van 'n mens-masjien koppelvlak aan 'n menslike operateur vir implementasie voorgelê.

'n Groot getal TEWA stelsels is reeds wêreldwyd in gebruik, maar as gevolg van die vertroulike aard van hierdie navorsingsgebied, is beskrywings van die werking van hierdie stelsels tipies nie in die oop literatuur beskikbaar nie. Ten spyte van die kritiese rol wat hierdie stelsels in die huidige, evoluerende netwerk-sentriese oorlogvoeringsomgewing speel, bestaan daar geen generiese raamwerke in die literatuur waarvolgens die werkverrigting van TEWA stelsels geëvalueer kan word nie.

Die werk wat in hierdie tesis vervat is, dra dus by tot die Suid-Afrikaanse TEWA stelselkennisbasis deur vas te stel tot watter mate die huidige TEWA stelsel-verwante navorsing in die konteks van 'n grond-gebaseerde lugafweeromgewing 'n samehorige grondslag vorm waarop verdere stelselontwikkeling en werkverrigtingsanalise kan voortbou. Hierdie breë navorsingsdoel word bereik deur 'n simulasieraamwerk daar te stel waarvolgens die werkverrigting van TEWA stelsels gemeet kan word en met behulp waarvan die werkbaarheid van plaaslik-ontwikkelde TEWA algoritmes gedemonstreer kan word.

Deur van die raamwerk gebruik te maak, word beperkings in die TEWA algoritmes geïdentifiseer en word strategieë voorgestel waarvolgens hierdie beperkings reggestel kan word. Hierdie strategieë sluit in 'n nuwe metodologie vir die samevoeging van bedreigingswaardes, 'n alternatiewe wapenstelsel-modelleringsbenadering en die implementasie van 'n genetiese algoritme vir die benaderde oplossing van die wapentoewysingsprobleem. Ontwerpsvereistes vir 'n doeltreffende mens-masjien koppelvlak word ook noukeurig beskryf, en 'n aantal TEWA stelsel werkverrigtingsmaatstawwe word voorgestel. Die werking van die raamwerk word uiteindelik aan die hand van 'n omvattende, byna realistiese, maar hipotetiese, grond-gebaseerde lugafweerscenario gedemonstreer.

Acknowledgements

The author wishes to acknowledge the following people and institutions for their various contributions towards the completion of this work:

- First and foremost, I would like to thank my supervisor Prof JH van Vuuren, for being willing to take me on and guide me through the process of researching and writing this thesis. He went through great lengths to ensure that I had everything I needed and that motivation stayed high throughout the two years. His extensive knowledge and dedication helped a tremendous amount as I tried to make sense of the threat evaluation and weapon assignment world, which can become quite complicated at times.
- ARMSCOR for their financial assistance in the form of a LEDGER bursary.
- My fellow students — of the current year and previous year — at the *Stellenbosch Unit for Operations Research and Engineering* (SUnORE) who allowed me to sometimes bounce ideas off them and for the mostly constructive critique that helped me shape this research project.
- Finally, I would like to thank my family and friends for their unending support.

Table of Contents

List of Acronyms

AHP: Analytical Hierarchy Process

AOR: Area of Responsibility

APM: Air Picture Manager

ARMSCOR: Armaments Corporation of South Africa

BMD: Ballistic Missile Defense

BVR: Beyond Visual Range

C2: Command and Control

CIWS: Close-in Weapon System

CoE: Centre of Expertise

CONPOS: Concept of Operation

CPA: Closest Point of Approach

DA: Defended Asset

DM: Deterministic Threat Evaluation Models

DS: Decision Support

DSS: Decision Support System

DTIC: Defence Technical Information Centre

DWTA: Dynamic Weapon Target Assignment

EEM: Exhaustive Enumeration

EO: Engagement Order

ESM: Electronic Surveillance Measures

EWR: Early Warning Radar

FCO: Fire Control Officer

FEC: Formative Element Combination

FO: Fire Order

FPGA: Fully-Programmable Gate Array

FU: Fire Unit

FW: Fire Window

GBAD: Ground Based Air Defence

GUI: Graphic User Interface

HDL: Hardware Description Language

HMI: Human Machine Interface

IFF: Identification Friend-or-Foe

JDL: Joint Directors of Laboratories

KOB: Keep-Out Boundary

LRSAM: Long-Range Surface-to-Air Missile

MAUT: Multi-attribute Utility Theory

MANPRINT: Manpower and Personnel Integration

MRSAM: Medium-Range Surface-to-Air Missile

M&S: Modelling and Simulation

NSGA II: Non-dominated Sorting Genetic Algorithm II

OIL: Operator In Loop

OODA: Observe, Orient, Decide and Act

OPTEMPO: Operational Tempo

OPFOR: Opposing Force

PATRIOT: Phased Array Tracking to Intercept of Target

POPD: Projected Orthogonal Passing Distance

R&D: Research and Development

RAA: Random Assignment Approach

RAM: Rockets, Artillery and Mortars

ROE: Rules of Engagement

SA: Situation Awareness

SAM: Surface-to-Air Missile

SANDF: South African National Defence Force

SHORAD: Short Range Air Defence

SoS: System-of-Systems

SSHP: Single Shot Hit Probability

SWAP: Static Weapon Assignment Problem

TADMUS: Tactical Decision Making Under Stress

TE: Threat Evaluation

TEFM: Threat Evaluation Fusion Model

TEM: Threat Evaluation Model

TEWA: Threat Evaluation and Weapon Assignment

TTWR: Time to Weapon Release

UAV: Unmanned Aerial Vehicle

UCAV: Unmanned Combat Aerial Vehicle

UML: Unified Modelling Language

VGD: Virtual GBAD System Demonstrator

VSHORAD: Very Short Range Air Defence

WA: Weapon Assignment

WAM: Weapon Assignment Model

WASS: Weapon Assignment Solution Selector

WASP: Weapon Assignment Scheduling Problem

WRL: Weapon Release Line

List of Figures

List of Tables

CHAPTER 1

Introduction

To improve is to change; to perfect is to change often.

— Winston Churchill

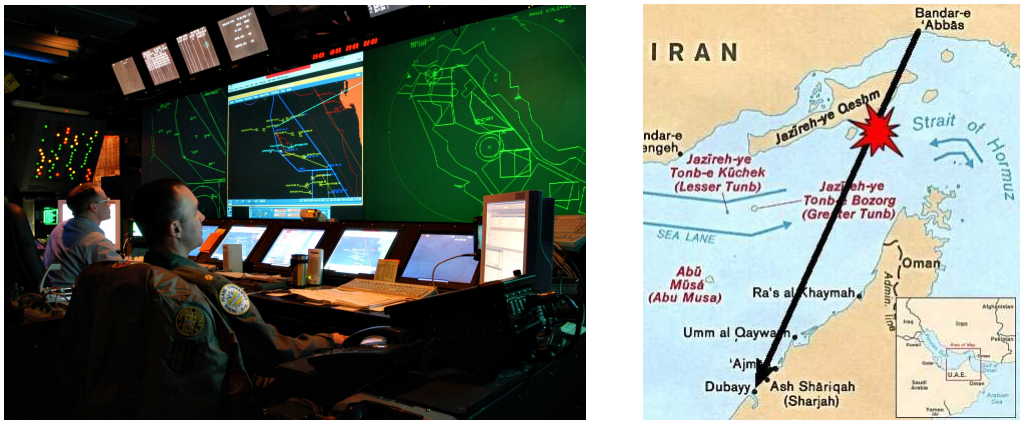
Contents

1.1 Historical Perspective

On 3 July 1988 a United States warship, the USS Vincennes fighting in the Persian Gulf, engaged with Iranian gunboats when an unknown aircraft approached the ship. The USS Vincennes warned the approaching aircraft, but received no reply. Subsequently, the warship wrongfully identified the approaching aircraft, which was a civilian airliner, as an attacking Iranian F14 fighter and launched two SM-2 radar-guided missiles at it in response. A total of 290 people were on board this civilian A300 Airbus. The confrontation that followed resulted in the USS warship destroying the civilian airliner, leaving no survivors [43].

This tragedy was followed by a number of investigations by the *United States Navy* in conjunction with the *International Civil Aviation Authority* [120]. These investigations concluded that the reason for the aircraft's misidentification can be explained by the small time frame at the disposal of the human operators during which to make a decision, in conjunction with the confusion and added stress resulting from the simultaneous engagement with Iranian gunboats. The *Decision Support System* (DSS) employed by the operators therefore did not succeed in assisting the highly complex decision-making processes associated with engaging multiple threats simultaneously.

As a result of this incident, the United States Office of Naval Research sponsored a research programme with the goal of improving the combat performance of a team operators by means of enhanced training, and providing them with an enhanced DSS [120]. The aim of such a DSS, commonly referred to as a *Threat Evaluation and Weapon Assignment* (TEWA) system,



(a) *AEGIS combat information centre on-board the Vincennes guided missile cruiser*

(b) *Approximate geographic location of the incident*

FIGURE 1.1: *The Iran Air Flight 655 incident [14].*

is to provide the operators with important information related to the level of threat that aerial vehicles pose to allied forces in an air-defense environment. Furthermore, the TEWA system is also responsible for generating high-quality weapon allocation suggestions for engaging aerial threats. The information from the TEWA DSS should then be used in combination with operator judgement to select the most suitable *Weapon Systems* (WSs) to counter the aerial threats.

Approximately fifteen years after the incident described above, on 22 May 2003, a Royal Air Force Tornado jet was returning to base when a US Patriot missile crew wrongly identified the friendly fighter plane as an attacking anti-radiation missile [130]. The blue-on-blue[1] confrontation which followed resulted in the US Patriot missile destroying the Royal Air Force Tornado resulting in two friendly fatalities.

During the build-up to this event, the Patriot battery crew were monitoring for Iraqi ballistic missiles when an unknown aerial entity was identified by their sensor systems and displayed on the *Human Machine Interface* (HMI) of their DSS. The symbol which appeared on the HMI corresponded to that of an anti-radiation missile. To confirm the Patriot system's threat evaluation result, the radar track was interrogated for IFF[2], but no reply was received. After meeting all the criteria laid out by the specific Rules of Engagement, the Patriot crew engaged in self-defence action, thereby destroying the friendly aircraft.

Several possible causes of this accident have subsequently been identified. The investigation board concluded that the Rules of Engagement were not robust enough to prevent blue-on-blue confrontations. The threat evaluation subsystem of the patriot system was also not robust enough — threat classification was based on the flight profiles and characteristics (radar signatures and thermal signatures) of typical radiation missiles and did, as such, not focus on specific Iraqi threats [130]. Similarly, the patriot crews' training were focused on identifying generic threats rather than those specific to the Iraq conflict, and also not on identifying false alarms. The Patriot crews were trained to react quickly, engage early and to trust the Patriot system's results. After detailed investigations, the board concluded that both the operator training and the Patriot firing doctrine were factors contributing to the accident that need to be revised [130].

[1] Synonymous with friendly fire — an inadvertent firing toward one's own or otherwise friendly forces.
[2] An acronym for *Identification: Friend or Foe.*

In this thesis it is advocated that perhaps one of the most effective ways of preventing the re-occurrence of TEWA-related decision support malfunctions such as those described above, is to develop a simulation performance evaluation framework for collectively testing the associated *Threat Evaluation* (TE) and *Weapon Assignment* (WA) algorithms of a TEWA system and to validate the system in conjunction with its intended end-users. Such a testing paradigm must account for different types of scenarios, in an attempt to identify potential algorithmic and/or system errors before commissioning the system. In the above accidents, a scenario generation approach could have identified the flaws in the Patriot system's TE algorithms, making it possible to take corrective action early. Diligent validation of the TEWA system — by incorporating the operators (end-users) during these stages — could also have flagged potential operator-training shortcomings. Since lives are at stake, it is clearly of critical importance that the performance of a TEWA system be thoroughly evaluated before commencing its deployment phase.

1.2 Informal Problem Description

Air-defense system analysis is a multidisciplinary field — command and control, decision theory, engineering and operations research all become intertwined to ensure that available resources (WSs and ammunition) are effectively employed during the process of protecting important assets. Air-defense is broadly defined by NATO[3] as

> *"all measures designed to nullify or reduce the effectiveness of hostile air action."* [141]

More specifically, within the context of a *Ground-based Air Defense* (GBAD) environment, the task of the defenders on the ground is to protect the so-called *Defended Assets* (DAs) from being destroyed by hostile threats such as aircraft. To be able to fulfil this purpose, the defenders are equipped with an array of sensors, *Weapon Systems* (WSs) and a TEWA system.

In this context, the purpose of a TEWA system is to provide decision support to the ground-based defenders in order to ensure that they make optimal use of their available resources (sensors, WSs and ammunition) in respect of protecting their DAs. To this end, the TEWA system is responsible for TE which entails identifying the threats, assessing the level of danger they pose to the defended system, and consequently assigning prioritized threat values to them. After the severity of all the threats have been determined, the WA subsystem makes use of these threat values to optimally assign WSs to threats so as to minimise the cumulative survival probability of the threats [215]. Figure 1.2 contains a graphical representation of the basic physical elements forming part of such a GBAD environment in which a TEWA system is employed.

At the turn of the 21st century, the combination of TE and WA subsystems became fairly common [167]. Even so, there still exist several unresolved complexities related to the implementation of a TEWA system within a GBAD environment. The concept of TEWA has traditionally been considered from the single-platform, static perspective [116]. According to Roux [167], the majority of deployed TEWA systems are used on naval craft, in a point-defense[4] role, as opposed to an area-defence role where a certain geographical area with numerous DAs has to be protected, as is the case with a GBAD environment.

With the current trend towards network-centric warfare[5], it is becoming increasingly important to approach TEWA from a force-level perspective. This means that the TEWA system should be

[3]The *North Atlantic Treaty Organisation* (NATO) is a collection of 28 member states with the essential purpose of safeguarding the freedom and security of all its members through political and military means [142].

[4]Point-defense is defined as the defense of a single entity (stationary or moving).

[5]Network-centric warfare may be seen as a military embodiment of information age concepts which includes

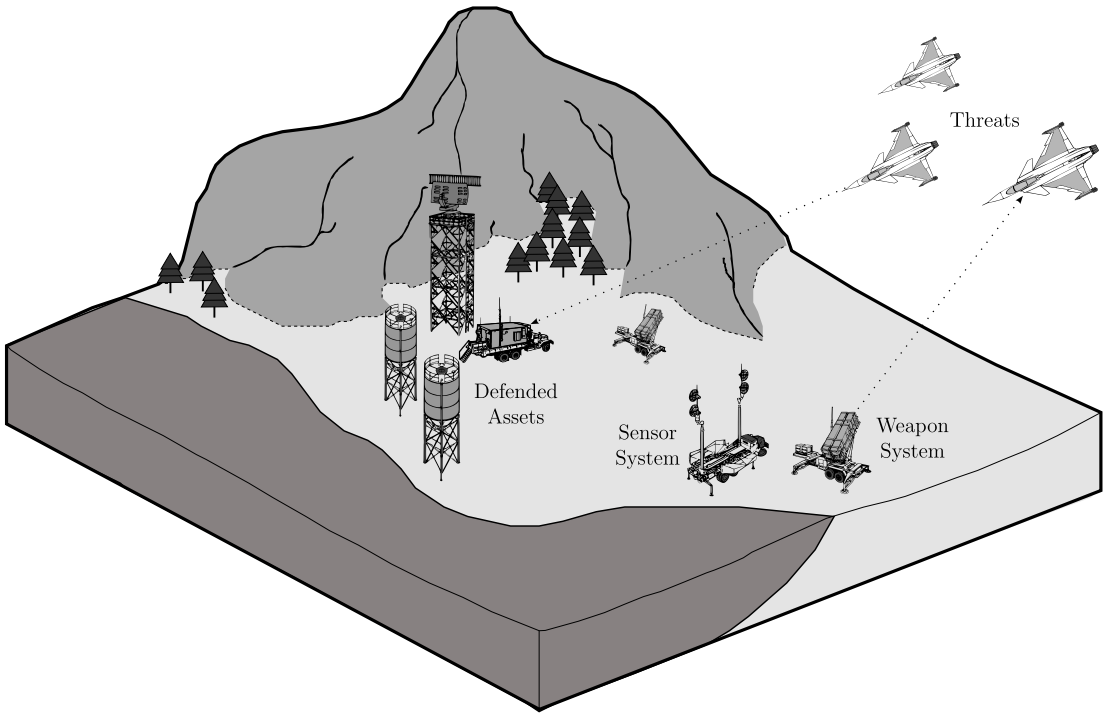


FIGURE 1.2: *The main physical elements within a GBAD scenario.*

considered as a whole, accounting for the human decision makers, DAs, WSs, sensors and threats. The TEWA system may be seen as a core subsystem of a larger GBAD system. Consequently, the TEWA system should be viewed as a dynamic decision making process aimed at the efficient utilisation of restricted tactical resources [72].

Because of the confidential nature of this research area, companies are reluctant to publish details related to the inner working of these systems in the open literature. The majority of TEWA systems are therefore, effectively, proclaimed as black-box systems[6]. Despite the critical role of, as well as the large number of risks involved when deploying these systems (as elucidated in §1.1), there are no standard methods or frameworks available in the open literature for evaluating the performance of TEWA systems.

The aim in this thesis is to integrate and demonstrate the working of previously developed TE and WA algorithms. Throughout such an integration of the algorithms, limitations may be identified and mitigated. Although the TE subsystem based on the above-mentioned algorithms has been tested in isolation, the TE and WA algorithms developed at the Stellenbosch University TEWA Centre of Expertise have never been tested collectively. Such a collective testing approach is crucial in the development of a TEWA DSS where the different subsystems interact to create the characteristic emergent properties of the system. These emergent properties cannot be accounted for by testing the system in a reductionistic manner (*i.e.* testing the TE and WA subsystems separately and projecting the TEWA system functionality from their isolated understanding).

It is acknowledged that this thesis builds upon work completed by previous students of the Stellenbosch TEWA Centre of Expertise which is mainly funded by ARMSCOR[7]. A TEWA

the integration of sensors, engagement systems and decision makers into an effective and responsive whole [223]. A more detailed discussion on the notion of network-centric warfare is provided later in this thesis.

[6] A complex system or device whose internal workings (algorithms) are hidden or not readily understood [55].

[7] The *Armaments Corporation* (ARMSCOR) of South Africa is mainly responsible for managing the acquisition,

system, similar to the one simulated in this thesis, forms part of a larger GBAD system employed by the South African National Defense Force which is used in order to protect critical South African infrastructure from hostile air action.

1.3 Rationale for the Study

The concept of a real-time computerised DSS for helping operators make decisions in respect of TEWA is a relatively new area of research in South Africa [167]. Several considerations related to a TEWA system in a GBAD environment have nevertheless already been researched. It is, therefore, required to integrate all the algorithms and strategies proposed in order to simulate a functioning TEWA system. In addition to serving as a proof-of-concept model, such a simulation is expected to help identify relevant limitations and conflicts between the proposed algorithms as well as opening avenues of investigation for future work. The presence, and extent, of the effect of possible emergent properties resulting from the interaction between the various components of such a system may thus also be evaluated in terms of the system's overall performance.

From various preceding studies, the need to establish a system development and system integration capability in the GBAD domain was identified. Hence, the research reported in this thesis is aimed at adding to the knowledge-base funded by the South African LEDGER programme, through developing a simulation model for testing the current TE and WA algorithms. The broad aim of the research in this thesis may be formulated as follows:

> *To determine the extent to which the current research related to TEWA-algorithms in a GBAD environment forms a coherent foundation from which further system development and testing can continue.*

1.4 Scope and Objectives

The long-term aim inspiring the research in this thesis is the development of an integrated South-African GBAD system, which includes a TEWA DSS similar to the one demonstrated in this thesis. As explained in §1.3, this research project is aimed at the development of TEWA DSS simulation environment for testing the effectiveness of current, locally developed TEWA algorithms. To achieve this aim, the following seven objectives are pursued:

I To *conduct* a thorough survey of the literature related to:

(a) physical and functional elements required in a successful and effective TEWA DSS in a GBAD environment,

(b) the current state of TEWA-related knowledge,

(c) methods established for accomplishing the TE process,

(d) different formulations of the WA problem,

(e) the suitability of simulation as the preferred paradigm to evaluate the performance of TEWA systems,

(f) requirements for the design of an HMI in a TEWA GBAD context, and

(g) the performance evaluation of complex systems.

maintenance and disposal needs of the South African Department of Defense.

II To *establish* display guidelines for an effective HMI in a TEWA context.

III To *integrate* the TE and WA algorithms developed locally within a generic TEWA performance evaluation simulation framework.

IV To *identify* limitations and conflicts present in the proposed WA and TE algorithms and to *implement* appropriate mitigation strategies.

V To *propose* sensible performance metrics for evaluating the performance of TEWA systems within a simulation performance evaluation framework.

VI To *demonstrate* the functioning of the established simulation performance evaluation framework within the context of a near-realistic, comprehensive GBAD scenario.

VII To *recommend* sensible future research avenues for the continuation of the work presented in this thesis.

It should be noted that during a number of previous LEDGER-funded projects, several assumptions were made in respect of the development of the aforementioned algorithms, which limits the scope of this thesis. As such, these assumptions are still valid and are clarified in the relevant chapters of this thesis. Two of the main assumptions include, among others, only considering fixed-wing aircraft as possible threats and assuming that the kinematic data from sensors systems are accurate and readily available. This thesis does not specifically address ballistic missile defence, although the system concepts derived and methodologies adopted throughout this thesis may be assessed for possible residual capability in such an environment.

1.5 Research Methodology

In order to fulfil the research objectives put forth in §1.4, an appropriate research methodology was established. This methodology is described in this section.

Since there are numerous complexities surrounding a GBAD system, it is important to understand the required level of modelling fidelity in order to achieve the stated objectives, prior to proceeding with the detailed work. The various levels of model fidelity used in an air-defense environment may be represented by the pyramid in Figure 1.3. Recognising the position of a system designer within this hierarchy is critical to understand the level of system fidelity that is required during the evaluation and design of the system [100].

The campaign models at the top of the pyramid typically includes many of end-users to test the system over a period of days, weeks or even months [100]. Consequently, these models are highly aggregated and are therefore limited to identifying the effect of individual systems on the performance of the larger system. The two middle layers — force-level and unit-system analysis — are able to illustrate the functioning of individual components of the larger air-defense system. The base of the pyramid (fundamental or engineering analysis), represents a higher level of detail for individual components of the system and thereby makes it attractive for a system-of-systems modelling approach.

Although the base layer provides the most detail regarding the system components and should therefore allow the system designer to obtain accurate results, this modelling approach requires extensive time and resources to effectively implement [100]. Furthermore, the scarcity of accurate, detailed information required in order to accurately model the different TEWA elements are typically classified and not available in the open literature. The approach followed in this

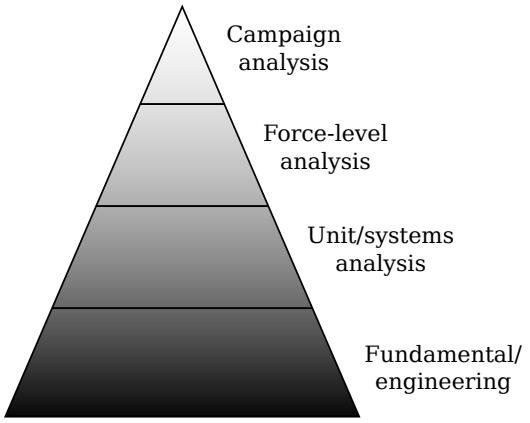


FIGURE 1.3: *Hierarchy of different analysis models [100].*

thesis is a combination of the system and fundamental modelling approaches. Many detailed algorithmic models are implemented to represent the WSs and threats (fundamental), but in order to effectively analyse the performance of the system a more system perspective approach (unit or systems analysis) is employed.

In order to pursue the broad research aim put forth in §1.3, appropriate research methods have to be identified. The methodology has to be executed appropriately in order to achieve the detailed research objectives stated in §1.4. The research methodology adopted to achieve these objectives is described in the remainder of this section.

Objective I and all its sub-objectives is pursued by performing a *literature survey*. This entails identifying and analysing the most important literature related to TEWA systems in general. The majority of the detailed local research in this domain was done at the Stellenbosch TEWA Centre of Expertise. Unfortunately, detailed international research is often not available in the open literature.

The second objective also requires a *literature survey* related to HMIs used in a TEWA context and, consequently, *qualitative methods* are employed in order to derive suggestions for HMI display requirements. There exists a variety of research related to decision support in a military context [41, 113, 225]. As such, this type of research is more easily obtainable in the open literature than the core algorithms of TEWA systems.

Objective III entails developing a simulation model incorporating the majority of the TE and WA related algorithms developed at the Stellenbosch TEWA Centre of Expertise. Prior to starting software development, a suitable simulation environment is first identified. This objective requires *implementation* in the form of computer code of the TE and WA so as to form a functioning TEWA system concept demonstrator which may be used for detailed performance evaluation.

Objective IV requires identifying limitations and conflicts present in the proposed TEWA algorithms. This objective is addressed by employing both *qualitative methods* and *quantitative methods*. The models developed at the Stellenbosch TEWA Centre of Expertise are compared to model formulations present in the open literature in order to qualitatively identify possible areas for improvement. Furthermore, the locally developed algorithms are tested by executing *empirical experiments* so as to qualitatively identify limitations and conflicts present between the algorithms. This objective is therefore, in effect, linked to Objective III. An evolutionary approach is followed to address Objective III — as limitations are identified through execution of Objective IV, mitigation strategies are implemented in the simulation environment, in turn,

addressing Objective III.

After developing a simulation model of a TEWA system and conducting a literature survey related to the evaluation of complex systems, several performance metrics and a general framework is suggested for evaluating the performance of TEWA systems, thereby addressing Objective V. This framework is developed by *qualitatively* evaluating the available literature. Since detailed literature related to the performance evaluation of TEWA systems is scarce to non-existent, similar research is identified and residual capabilities are identified for the application to the performance evaluation of TEWA systems.

In order to demonstrate the functioning of the simulation model developed in pursuit of Objectives III–IV and the performance evaluation framework designed in puruit of Objective V — in effect addressing Objective VI — the various existing TEWA algorithms are *tested* within the context of a comprehensive worked hypothetical example. The algorithms are evaluated *quantitatively* by visualising their results and calculating performance metrics, but also *qualitatively* by interpreting the results from a top-down perspective.

After having achieved all of the above objectives of §1.4, a critical review of the current state of TEWA DSSs is performed in order to determine desirable further work, in fulfilment of the last objective of §1.4.

1.6 Thesis Organisation

As described in §1.5, this research project is executed in three stages. The first stage consists of a thorough *literature review*, the second stage entails the *development and integration* of the constituent TE and WA algorithms and the representation of the simulation model entities and, finally, the third stage entails the demonstration of the suggested *performance evaluation framework*. The chapters within each of these research stages are clarified in this section.

In the second chapter of this thesis, an overview of a TEWA system in a GBAD environment is provided, thereby establishing the context in which the remainder of the thesis should be understood. The concept of network-centric warfare in a GBAD environment is also elucidated. Furthermore, the importance of employing efficient command and control processes so as to integrate the large amounts of information in a network-centric GBAD environment is clarified. After reading this chapter, the reader should have a good understanding of the nature and constituent elements within the battlespace and also be better equipped to understand the explanations contained in the remainder of the thesis.

Chapter 3 contains a concise literature review of the current state of TEWA knowledge from a domestic as well as from an international perspective. The domestic part contains reviews of all the previous work done at the Stellenbosch TEWA Centre of Expertise and provides transparent, general frameworks for the WA and TE subsystems. The current state of international TEWA research, as identified from the literature, is also provided. This chapter closes with an overview of four TEWA systems in use around the world, so as to provide a practical understanding of the operational environment of a TEWA system and thereby promote a system-level (top-down) perspective of a TEWA system.

The next part of the thesis — the development and integration — details the development of the simulation performance evaluation framework which contains the embedded TE and WA subprocesses, and well as the preliminary design of an HMI for a TEWA DSS. The development and integration part opens in Chapter 4 with an overview of simulations in a military environment, as well as the typical structure of simulation paradigms in this context. The methodology followed

to model each of the physical TEWA elements are also described in detail. Chapter 4 closes with the introduction of a hypothetical GBAD scenario, to be used for clarifying the WA-related and TE-related concepts throughout the remainder of the thesis.

The implemented TE process is described in some detail in Chapter 5. A variety of deterministic TE models are described in detail, and the complexity of the threat value fusion process is clarified. Two possible value-based fusion approaches are consequently proposed in order to calculate a single system threat value per threat. A high-level overview of the TE-process architecture within the simulation environment is finally given at the end of Chapter 5.

In Chapter 6, a description of the WA problem is provided, together with a description the various underlying concepts which shape the formulation of this resource allocation problem. The complexities of this non-linear, combinatorial optimisation problem is also described and typical mathematical formulations of the static and dynamic classes of WA problem are given. Five different WA solution approaches are next qualitatively compared. A genetic algorithm — utilised in this thesis for solving the WA problem — is briefly described. The chapter closes with a clarification of the formulation of the dynamic WA problem model adopted in this thesis.

The development and integration part of the thesis closes in Chapter 7 with the presentation of a preliminary design of an HMI to be used in a GBAD environment. This chapter opens with a discussion on the architecture of a typical TEWA DSS, with reference to the different levels of data fusion, and elucidates the importance of providing germane decision support to human operators. In addition, the complexities surrounding the provision of germane decision support in a military context are also provided. In order to assist possible future work in terms of the detailed design of an HMI, the available literature is reviewed in conjunction with existing HMIs in order to provide several design requirements for a TEWA HMI in a GBAD environment.

The final part of the thesis — performance evaluation — opens with the presentation of a performance evaluation framework for TEWA systems in Chapter 8. The underlying concepts of a *System of Systems* (SoS) approach is explained, and this is followed by a review of available performance evaluation approaches for complex systems as identified in the open literature and, finally, four performance evaluation metrics are proposed to be used within the developed performance evaluation paradigm.

The workability of the concepts, strategies and algorithmic models explained throughout this thesis are applied to a comprehensive, near-realistic (but hypothetical) GBAD scenario in Chapter 9. In addition, this experimental demonstration of the constituent TE and WA models within the performance evaluation framework of the previous chapters also serves as an additional verification step (to ascertain the correct functioning of the algorithmic models) and as an additional validation step (establishing the suitability of the developed simulation model for evaluating the performance of TEWA systems).

Finally, the thesis closes in Chapter 10 with a summary of the work in this thesis and an appraisal of the contributions made in this thesis. Suggestions for future work are also provided in order to guide the reader in the selection of future research avenues.

PART I

LITERATURE REVIEW

CHAPTER 2

TEWA System Overview

Information is the oxygen of the modern age. It seeps through the walls topped by barbed wire, it wafts across electrified borders.

— Ronald Reagan

Contents

This chapter provides the reader with an understanding of the physical elements and the predominant theories underlying a TEWA DSSs in the literature, in order to establish a conceptual picture of what a TEWA DSS is. This conceptual understanding will allow the reader to place the simulation and algorithmic development of the following chapters in the proper context. In addition, this chapter also attempts to provide an appreciation of some of the complexities and challenges associated with the practical implementation of a TEWA DSS. All explanations pertaining to TEWA systems in the chapters to follow, are from a systems viewpoint — considering the TEWA DSS as a whole — so as to promote a systems thinking approach throughout the rest of the thesis.

Because a TEWA system typically consists of a multitude of interacting complex systems, it would not be possible to account for all these elements in an entirely generic manner and, as such, the scope of this thesis is also further clarified in this chapter.

2.1 Transformation towards Network Centric Warfare

The increase in computational power, level of advance in sensor technologies and the viral spread at which data diffuse, have ushered in what is popularly known as the information age [76]. This new age — a beneficiary of the acclaimed Moore's Law[1] — is marked by an increased ability to generate, organise and distribute information by exploiting sophisticated modern sensor technologies and information architectures. Military forces the world over can achieve greater *Situation Awareness*[2] (SA) by reaping the benefits of this volume of information. This is the general backdrop out of which the concept of *Network Centric Warfare* (NCW) has emerged.

2.1.1 Defining NCW

NCW may be seen as a military embodiment of information age concepts which includes the integration of sensors, engagement systems and decision makers into an effective and responsive whole [223]. According to the United States *Command and Control* (C2) research programme [37], the networking opportunities associated with NCW enable forces to improve the efficiency and effectiveness with which missions can be executed. NCW utilizes computers and communications to link people through the information flows resulting from the inter-operation of different systems. The focus is on improving human behaviour (*i.e.* SA) as opposed to mere information technology. Technologies may, however, be seen as the necessary enabling factor for NCW. As such, one of the main ideas behind NCW is the collaboration and sharing of information to increase the timeliness and effectiveness with which decisions can be made and missions can be executed.

There is no single, universally accepted and all-encompassing definition for the notion of NCW. Nonetheless, Dahl [47] has provided the following working definition for the concept of NCW:

> *"NCW can be broadly defined as deriving power from the rapid and robust networking of well-informed, geographically dispersed military resources. Thereby creating a supreme tempo and a precise, agile form of maneuver warfare. NCW focuses on operational and tactical warfare, but can even affect the strategic domain. It is the dominating theory of war for the information age"* [47].

Central to the effort of incorporating NCW into military doctrine and systems is an understanding of the interaction between the physical, information, social and cognitive domains of warfare. All of these domains form part of the concept of NCW. These domains are briefly described below.

Physical domain: The physical domain is the traditional domain of warfare. This domain spans the land, sea, air and space environments in which operations are normally executed. The physical platforms (*e.g.* weapon systems and sensor systems) and the physical communication infrastructure also form part of this domain. In comparison with the other domains, the physical domain's elements are the easiest to understand and measure. This is also the domain in which combat power has traditionally been measured [183].

[1] The observation that *logic density* (bits per square inch) has corresponded with the curve: $\textit{logic density} = 2^{t-1962}$, where t is time in years. This relation was first observed in 1964 and was found to be valid until the late 1970s. Eventually the trend slowed down, doubling the logic density every 18 months [5].

[2] Situation awareness may be defined as the perception of elements in an environment within a volume of time and space, the comprehension of their meaning and the projection of their status in the near future [66].

Information domain: This domain is where information is collected, processed and managed within the network. It is the domain that facilitates C2 by conveying the crucial mission-related information to human operators. Consequently, and as a result of recent increases in cyber-warfare capabilities, the information domain is becoming increasingly vulnerable to hostile attacks [223]. It is, therefore, of critical importance to protect and defend this domain so as to ensure sustained information superiority during an engagement scenario.

Cognitive domain: The cognitive domain is the mind of the human operators, and encompasses the concepts of leadership, morale, unit cohesion and experience [37]. This domain relates to those factors that influence the ability of a decision maker to establish his SA and, consequently, make well-informed decisions. It is also where an operator's individual judgment resides.

Social domain: This is the domain in which humans interact and share information to form a shared awareness for making collaborative decisions. The elements of this domain include a group's culture, set of values and beliefs. This domain overlaps with the information and cognitive domains, but is distinct from both. To clarify, the social domain is, in essence, shared sense-making — propagating from shared awareness to shared understanding to making collaborative decisions — whereas in the cognitive domain the focus is solely on individual understanding.

From the aforementioned explanations, it is clear that these domains interact to form the unifying concept of NCW. A visual representation of these NCW domains is provided in Figure 2.1 to better clarify the interactions between the various domains.

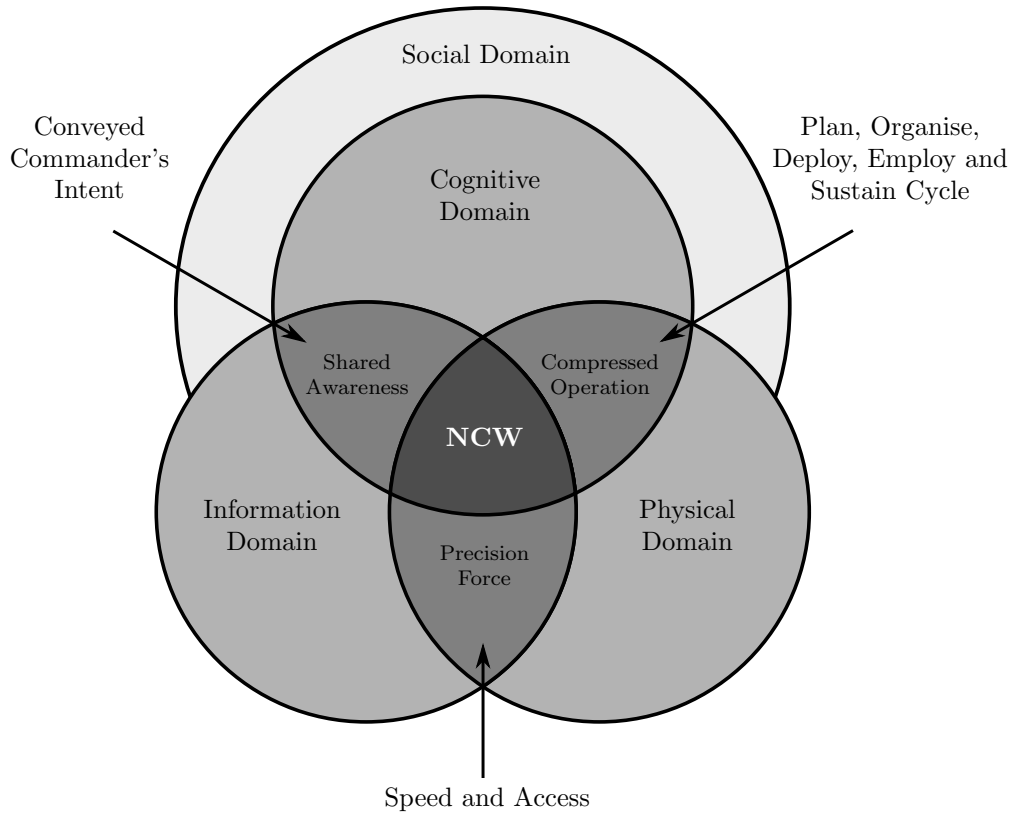


FIGURE 2.1: *The four domains of NCW (adapted from [37]).*

Currently, there are two views on the effects of NCW in modern warfare. The more conservative view is that NCW is simply the evolution that accompanies the digitisation of the conducts of war, while the more radical perspective is that the information age is altering, or revolutionising, the nature of conflict in a fundamental manner [5, 76].

According to the first view, only incremental improvements to modernising current technologies are needed to accommodate the impeding information age. According to Alberts and Hayes [5], this transformation strategy will fall short of the potential benefits that NCW can bring to the battlefield. A more introspective approach is instead needed, where a more "disruptive" change is inflicted. This radical transformation requires reconsideration and redevelopment of the current concepts of C2 that are central to all current military operations. A transformation of C2 is needed to accompany the modern, single-organisational structure that is characteristic of C2 [76]. Basically, this transformation is required in order to translate information superiority into combat power by effectively linking a multitude of knowledgeable entities in the battlespace [6].

There are several enabling technologies that may assist in the transformation towards NCW. According to Edward and Smith [183], three concurrent technologies hold the most potential for NCW: Sensor technology, an increase in computing power and more sophisticated *Weapon Systems* (WSs). These technologies should interact to create potential synergies that may give rise to novel emergent properties so as to address the needs of NCW and, consequently, change the time-tested characteristics of war. This approach towards modernising military forces for the information age is currently the dominating principle in NCW [76, 183, 223].

2.1.2 The Importance of Implementing NCW Principles

The current advances in automation technology, the increasing complexity and diversity of land, aerial and sea scenarios, as well as the abundance of available information, all pose significant challenges to making timely and accurate decisions within a dynamic, high-risk GBAD environment. In light of this, and by analysing modern conflicts, Hutchings and Street [86] have proposed the doctrine development illustrated in Figure 2.2. This procedural doctrinal change for military operations affords an increasing emphasis on information-gathering, sensor-based systems. The figure suggests that sensor needs and information design[3] will dominate as the importance of firepower-based systems decline.

The nature of the information age makes it increasingly difficult to operate in a reductionist manner since technology has compressed the time and scape continuum, thereby introducing the need for more dynamic decision making and also fogging the difference between the strategic, operational and tactical levels of decision making [6]. From Figure 2.2 it is clear that NCW concepts have to be implemented in modern military systems in order for them to remain relevant. The United States has already realised this, as may be seen from how defence projects are increasingly oriented towards the acquisition and integration of sophisticated weapon-sensor platforms and how its research is dedicated to improved information design [19, 213].

One of the potential benefits of NCW is its force multiplying effect resulting in an increased effectiveness and SA from enhanced defence capabilities. NCW enables operators to create and exploit a common SA, and infiltrate the enemy's decision cycle. By successfully utilising these advances in technology, the resulting speed and precision should make it possible to exploit specific battlefield opportunities and operate at a pace that will overwhelm the enemy's ability to

[3] *Information design* is the clear, transparent and unambiguous presentation of information to enhance understanding and decision making [220].

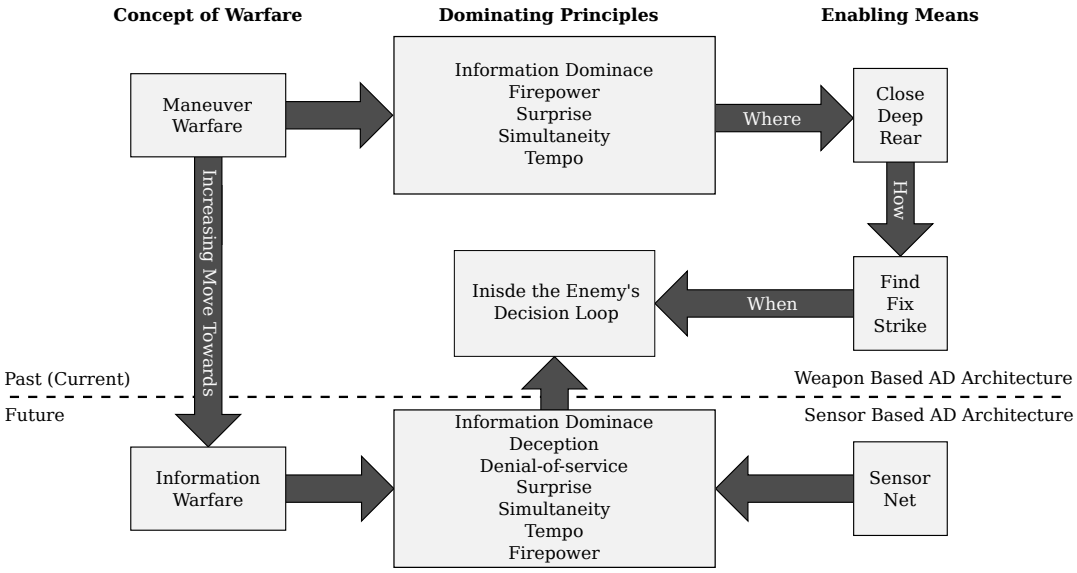


FIGURE 2.2: *Predicted doctrine development (adapted from [86]).*

respond [183]. The main disadvantage currently hindering the implementation of NCW concepts and technologies is that it is highly resource-intensive. Procuring the necessary hardware and software is not only expensive, but involves an ongoing cost (typically with no ceiling) so as to support and maintain the equipment and train the operators [48].

Even so, if military forces wish to increase their effectiveness, increase their survivability, improve their lethality and surprise their enemies, then the concept of NCW should undoubtedly be integrated into their defence systems and concepts of operations. In order to increase the likelihood of success when engaged in a combat scenario, a network-centric approach to the design of military systems is paramount.

2.2 Command and Control in the Context of TEWA

C2 is a generalised term encompassing a multitude of activities at all levels of an organisation — from motivating individuals, to achieving a common sense of purpose, to assigning responsibilities, to assessing how well the organisation is performing. C2 is therefore inherently an iterative decision making process. Warfare has always been a challenging domain, characterised by the importance of seamlessly implementing C2. Recent designs of military C2 systems have been based on the concept of warfare during the 19th century [167]. After the second world war, the advent of more lethal weaponry and mass mobilisation — as was evident from the large military build-ups during the cold-war era — has resulted in wide-spread implementation of *mission command* [193].

The German army may be seen as the pioneers in this respect — they already implemented mission command principles during both world wars. Mission command was adopted by the United States, as well as by the British Armed Forces, during the late the 1980s, and many other countries also currently adopt similar C2 styles [209, 193]. Although mission command was predominantly implemented after the second world war, its origins can be traced back as far as the Napoleonic Wars (1792–1815) [193].

Mission command is, in essence, a decentralised form of command and control — the commander exercises overall authority and direction so as to enable disciplined initiative by the actors[4] on the ground. As such, mission command relies highly on the trust, responsibility and shared intent of the actors executing the mission on the tactical level, since these actors have freedom of action in the tactical environment. This approach provides the actors, who are executing the mission, with more flexibility in order to adapt to the dynamic nature of military scenarios.

As mentioned previously, however, the increasing complexity and diversity of land, aerial and sea scenarios pose significant challenges to making timely, accurate decisions in a dynamic environment. Therefore, by providing actors with the freedom of choice, without the specification of critical information required to make accurate decisions, an ideal environment is induced for possibly disastrous outcomes; hence the need for effective C2 DSSs.

The information age has created an environment where collective decision making can be used to increase combat power. The knowledge which is gained from this abundance of information may be seen as a key factor to obtain battlefield-dominance. However, the exploitation of knowledge alone is not enough — speed may be seen as the critical enabling factor for exploiting the availability of knowledge. Consequently, the speed with which knowledge can be exploited will collectively determine a military force's C2 effectiveness. The main concern now is how to exploit this abundance of information to the advantage of the operator.

2.2.1 Implementation of C2

The heart of a C2 system is its ability to accommodate large amounts of information. Because of mission command's autonomy, there is typical a lack of information feedback and a general problem of information delay. Some military theorists have suggested that in an increasingly complex environment, it would be advantageous to have a decentralised form of decision making close to the source of complexity (*i.e.* the source of information volumes) [153].

Hence there is a need to move away from the current hierarchical, centralised form of C2 to a more cooperative form of decision making. Although these forms of C2 are being adopted world-wide, the transformation is not fast enough to address the challenges of the information age [4]. If a decentralised C2 paradigm is finally implemented, it is important to consider carefully the quality and quantity of information provided to the different actors, as an overload of information can prove to be counter-productive.

C2 was developed to address the two major challenges of warfare: the *fog* and *friction* of warfare [167]. The fog of warfare refers to the uncertainty associated with any battle scenario, while friction is associated with the difficulty of a commander, or operator, to translate his intent into actions. The information age will not eliminate the fog of war, but it should certainly reduce it. The information age has also had an effect on how problems are solved and how the final solutions are implemented.

By taking a holistic perspective of TEWA, it is acknowledged that the process may be regarded as a dynamic human decision making process aimed at the successful utilisation of available (limited) resources (*e.g.* WSs, ammunition and sensors) during the conduct of command and control activities. The problem is, therefore, how to integrate the massive amounts of information from the ever increasing amount of sensors so as to develop a coherent picture of the current situation, and fashion appropriate responses. This problem forms the basic nature of the C2 challenge of a TEWA DSS.

[4]The notion of an *actor* is used here in order to acknowledge the general context in which the theory has been developed. In the context of TEWA, however, this placeholder may be replaced by *operator* without loss of significance.

2.2.2 The OODA Decision Cycle

Several business process models have been proposed for modelling decision making in a military context or similar real-time, high risk domain [74]. Some of these models include Endsley's model of SA [175], Klein's model of recognition-primed decision making [56], Wohl's stimulus-hypothesis-option-response model [224] and Rasmussen's model of human thinking in supervisory control [225]. Each of these models have certain shortcomings and advantages, but no other model has been as successful as Boyd's *Observe-Orient-Decide-Act* (OODA) cycle, illustrated in Figure 2.3.

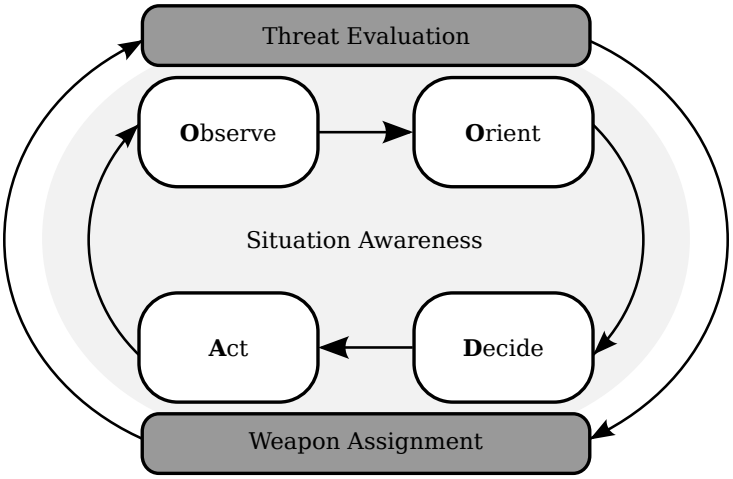


FIGURE 2.3: *Boyd's OODA cycle within a TEWA context.*

The OODA model for decision making was created by Boyd[5] as a result of observing jet fighter pilots in combat [68]. The OODA cycle has become an accepted model for military C2, thereby making it a leading candidate for the operational view of a C2 system architecture. The OODA cycle is currently the dominating process underlying the design of automated and semi-automated decision support systems in the military domain [1].

Referring to Figure 2.3, *Observe* essentially entails the gathering of information from sensor systems. *Orient* is the stage where the main problems with decision making are encountered, since during this stage an operator's interpretation of the current situation is formed. During this stage, the operator reacts to the observed information by processing the information obtained during the *Observe* stage and updates his situational picture. *Decide* is when the operator makes a decision based on the situational picture resulting from the *Orient* stage. Usually, more than one course of action is possible; the operator then determines the appropriate one. *Act* is the realisation of the decision (*i.e.* engaging threats).

An important concept of the OODA cycle is the principle of *Operational Tempo* (OPTEMPO) [169]. OPTEMPO is an acronym that has its origins in the military domain, and refers to the rate of performing military actions or executing military missions [36]. Since C2 is a continuous, cyclic process, the operator who cycles through the OODA cycle faster gains an ever increasing advantage from an increased OPTEMPO. It is, however, not absolute speed that matters, but speed relative to the opposing force. If the defending force's OPTEMPO is higher than that of the enemy, it will allow the defender to infiltrate the enemy's OODA cycle and act before the enemy has a chance to act [180].

[5]Boyd never published any peer-reviewed literature on his OODA cycle. The only available literature from Boyd himself, is the briefings which he presented to the defence establishment in the 1980s [11].

In order to increase the speed with which an operator can cycle through the OODA cycle (*i.e.* increase the OPTEMPO), it is required to decrease the time to observe the enemy, orient to form SA, make a decision and act on that decision. This increased OPTEMPO can be achieved as a result of a variety of practical ways, from increasing the detection rate of sensors to improving decision support for the human operators. To master the OODA cycle, it is therefore essential that technologies are employed that can obtain and process information more rapidly.

The previous TEWA-related research projects on which this thesis builds all suggested the implementation of the OODA cycle for the TEWA process. This model does, however, also have its shortcomings. The OODA cycle has been criticised for its lack of psychological validity, since it does not accommodate any concept of attention or memory. There is also no cognitive representation of real-world states and/or models. Some researchers have also criticised the OODA cycle for not including a deliberate planning process [74].

Although the OODA cycle has some limitations, there does not currently exist a better alternative that has been implemented and which has shown as much success as the OODA cycle [140]. More research will be required in order to develop a superior alternative. Such work is, however, beyond the scope of this thesis. Since the OODA model is superior in its innovative simplicity and since it also typically forms part of an officer's curriculum during officer training, implementation of this model in the C2 architecture is expected to simplify the working of a DSS from the perspective of an operator. It is important to recognise that the goal of the DSS should be to assist the operator during each of the four stages of the OODA cycle. Because no better alternative has proven itself, the OODA cycle will be used in this thesis as a model of the decision cycles embedded within the TEWA process.

2.3 TEWA Processes and Events

The successful neutralisation of threats during an engagement involves the execution of several processes, all off which must be accounted for when analysing the effectiveness of a TEWA DSS. An overview of the events and processes associated with each process within a TEWA scenario are provided in this section. The processes involved in defending assets during an aerial attack may be classified into three groups: Finding and identifying threats, controlling sensors and WSs, and engaging threats [121]. According to Benaskeur[6] [23], these three main air defence processes may be explained by the following five C2 processes:

1. target detection,
2. target tracking,
3. target identification/classification,
4. threat evaluation, and
5. weapons assignment, which consists of
   (a) response planning,
   (b) response execution, and
   (c) damage assessment.

[6]Benaskeur's focus was on naval engagements, but according to him, these processes are applicable to all air defense scenarios.

A functional illustration of the aforementioned air defence C2 processes within the context of a GBAD environment may be found in Figure 2.4.

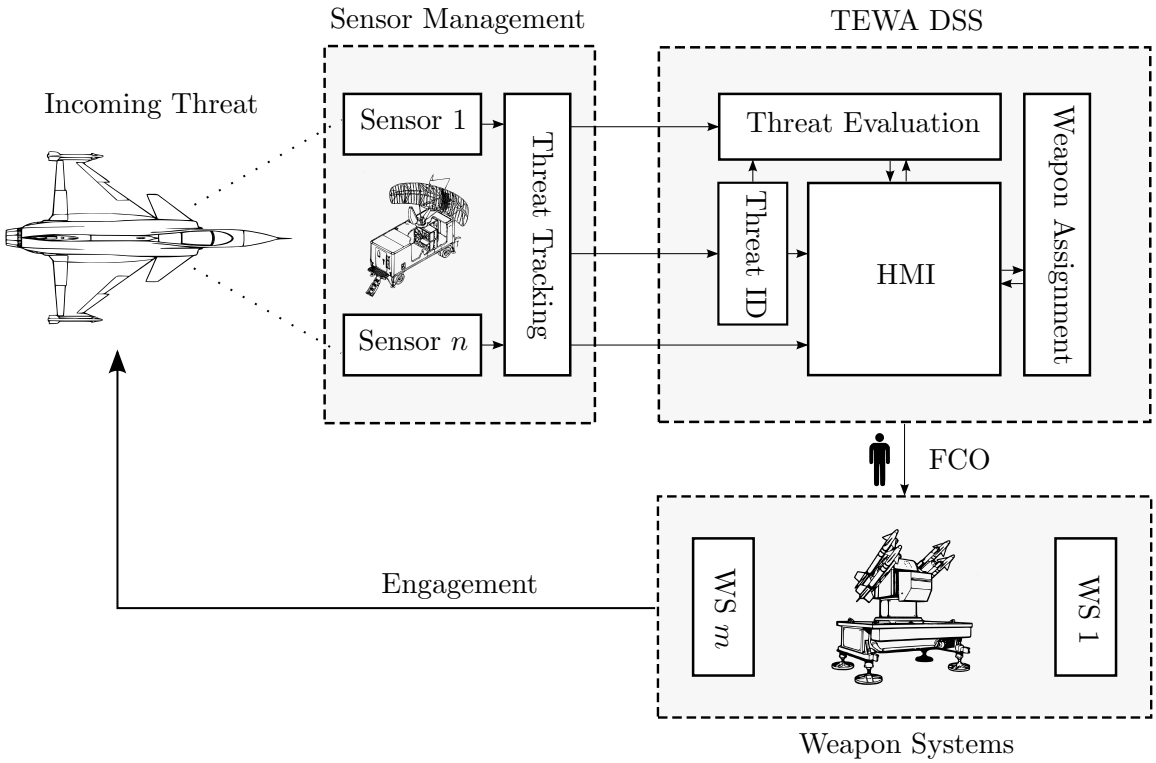


FIGURE 2.4: *Functional layout of a TEWA system.*

During a hostile engagement, the TEWA DSS must initiate a set of consecutive actions. The early detection system will detect the threats and sequentially identify them. This process of early threat identification and hostility classification generally occurs with the aid of intelligence information and operator experience. *Identification Friend-or-Foe* (IFF) and *Electronic Surveillance Measures* (ESM) are also used to verify the identities of aerial entities [121]. Through the identification and classification of the aerial entities, these entities may be categorised as either friendly, neutral or enemy entities. In the case of an enemy or neutral entity, the threat is subjected to further processing within the *Threat Evaluation* (TE) process.

If a threat cannot be identified positively as friendly, and is therefore still a candidate for engagement, undesirable incidents are avoided by clearly established *Rules of Engagement* (ROE). Prior to an armed conflict, a national authority typically establishes ROE that describe the circumstances under which aerial threats may be engaged [42]. These rules are established in an attempt to prevent fratricide[7] incidents, among others. It is compulsory to consult the current ROE so as to ensure that there are no conflicts with policy and/or laws adopted before threats are engaged.

As the threats approach more advanced sensor systems, TE is conducted in respect of those targets that are not positively identified as friendly. Different TE models, which run concurrently, are typically used to calculate system threat values for all targets which, in turn, are used to prioritise the threats according to the levels of danger they present to *Defended Assets* (DA).

After determination of all the threat values, the control stage consists of sending the threat values to the *Weapon Assignment* (WA) subsystem in order to determine which (if any) ground-based

[7] *Fratricide* represents a safety concern, generally related to WSs, and may be defined as an accidental attack on friendly forces by other friendly forces [12].

resources should be assigned to counter the threat. The WA subsystem determines the best allocations of WSs and ammunition, within the available time frame, to counter these threats. In advanced WA models, optimisation techniques are utilised to generate a schedule of events (engagements), within specific time frames, according to which threats are to be countered, thereby protecting the DAs. From the suggestions provided by the WA subsystem, the operator interacts with a human-machine interface to select a suitable action for execution.

The engagement stage realises when the operator selects an outcome and an *Engagement Order* (EO) is sent — as soon as the WSs are positioned and locked-on, the threats are engaged. After this stage, a damage assessment sub-process is executed in order to determine whether engaged threats were, in fact, successfully neutralised. Hereafter, the C2 cycle is repeated in order to ensure the long-term survival of the DAs.

Such dynamic C2 and battle management require fast and effective decision aids providing high-quality allocations of resources (sensor pairing and WA) so as to aid in successful combat engagement and real-time damage assessment [30]. The main purpose of the TEWA DSS is, therefore, to assist the operator in making decisions during the various decision making processes explained above.

2.4 Physical Elements of a GBAD System

In order to be able to succeed in the C2 processes described in the previous section, the GBAD system has to be equipped with various surveillance sensors, fire control radars, WSs and human machine interfaces. Since the goal when utilising a TEWA DSS is to neutralise hostile threats, these threats are essentially also part of the system, together with the DAs which they attempt to destroy. An overview of these physical elements is provided in this section.

2.4.1 Physical Environment

Since a GBAD scenario takes place in the real world, ideal theoretical performance is rarely achieved. In view of this, the *physical environment* refers to all the real-world conditions which influence the TEWA system's performance and can be further divided into the *natural environment* and the *induced environment* [121]. Natural environmental conditions include, but are not limited to, the state of the atmosphere, meteorological conditions and properties of the terrain. The induced environment, on the other hand, includes all man made conditions that can potentially have an effect on the functioning of the TEWA system, such as smoke, debris and electrical interference. The effects that all these dynamic environmental conditions may have on the functioning of the TEWA system have to be taken into account when the performance of the system is evaluated.

2.4.2 Detection and Tracking Sensor Systems

Sensor systems are used to gather information about the tactical environment in real time (or close to real time). The measured information provided by the sensors, forms part of the crucial input information required by the TEWA system. The threats are detected and monitored by the GBAD system sensor grid, which typically consists of a network of local and/or remote sensors. Generally, the sensor grid consists of a combination of radars, thermal sensors and electro-optical sensors as well as other electronic support measures. Two typical sensor systems used in a GBAD system are shown in Figure 2.5.

(a) *Reutech's DBRXL (RSR 320) 3D ground-based dual band radar [160]*

(b) *Mantis Air defence protection system [137]*

FIGURE 2.5: *Two examples of typical ground-based sensor systems.*

During an engagement, long-range search radars, with slow update rates, are used to detect aerial entities. Table 2.1 provides typical specifications of different types of radars. As may be seen from the table, these long-range radars can have exceptional ranges (200+ km) which enables them to identify possible threats very early on. During these early stages of detection, not much may be known about the aircraft and so only very rudimentary TE is typically possible (*e.g.* by means of flagging models at these stages). The detected aircraft are flagged for further measurement by tracking radars, which are used to track single targets or clusters of smaller targets. Hence, as the threats come into range of the more accurate medium- to short-range radars, more information can be gathered and more sophisticated TE algorithms can be activated in respect of the incoming aircraft. The information obtained by the sensor system usually include threat-specific kinematic and positional data, such as heading, altitude, distance and speed.

TABLE 2.1: *Example specifications of typical 2D surveillance radars [168].*

	Update Rate	Range	Range Accuracy	Azimuth Accuracy
Long-range	±10 sec	200+ km	100–500 m	0.5–2 deg
Medium-range	2–4 sec	50–200 km	20–100 m	0.1–0.5 deg
Short-range	≤1 sec	0–50 km	5–20 m	±0.1 deg

Radar cross-sections, and other signature-related measurements of aerial entities, may also be used to determine certain additional information about the threats, such as threat type [121]. In addition to detecting threats, sensors also have to be used as an intermediate step to guide WSs, before the built-in sensors of the projectiles (*e.g.* missiles) take over for the final guidance in order to achieve a positive hit on a threat [200].

A major threat to air defence radars is the continued development of anti-radiation missiles and anti-radiation drones. Sensors are becoming more susceptible to attack from high-speed anti-radiation missiles, which home in on the radiation emission sources of radars. These threats to the sensor grid encourage the use of passive[8] search and tracking systems [28].

[8]In contrast to active sensors, which require external sources of power (*e.g.* excitation voltages) to provide output signals, the output power of passive sensors is mostly provided by the power of the measured signal [73].

Another threat to the effective functioning of sensor systems is jamming by the opposing force. *Jamming* is the process of deliberately interfering with sensor systems by transmitting frequency-matched power signals in the direction of the sensor [121]. The jamming signal therefore competes with the sensor's signal and makes it increasingly difficult for the system to accurately distinguish a threat from the interference. The two main forms of jamming are *noise jamming* and *deceptive jamming*. Noise jamming has a similar effect to superimposed thermal noise or clutter, thereby making it increasingly difficult to maintain sensor-threat signal integrity. Deceptive jamming, on the other hand, induces errors into targeted measurement channels (such as the elevation, azimuth or range channels) of a tracking radar, thereby degrading the ability of the tracking sensor to aid in the guidance of WSs. As such, deceptive jamming is mainly used against tracking radars whereas noise jamming is typically employed against search radars.

In conclusion, it is clear that many difficulties are associated with the effective management of the sensor net. Sensor management is a relatively new area of research which is described by Xiong and Svensson [226] as *"a system or process that seeks to manage or coordinate the usage of a suite of sensors or measurement devices in a dynamic uncertain environment, to improve the performance of data fusion and ultimately that of perception."* The management of sensor systems is beyond the scope of this project, but relevant problems and approaches towards effective sensor management may be found in [138, 226].

2.4.3 Threat Characterisation

All aerial vehicles within the area of responsibility that have the potential to cause harm to a DA should be classified as threats. This includes *Rockets, Artillery and Mortars* (RAM), electronic warfare platforms, *Unmanned Aerial Vehicles* (UAVs) and manned aerial vehicles. For the purposes of this thesis, only fixed wing aircraft (a sub-class of manned aerial vehicles) are considered as possible threats. As may be seen in Figure 2.6, however, the currently available ground-to-air WSs are not able to counter all these unmanned threats before they are launched from launch vehicles. As such, certain threats (such as ballistic missiles and guided missiles) have to be individually categorized as threats that are available to be engaged by ground-based WSs.

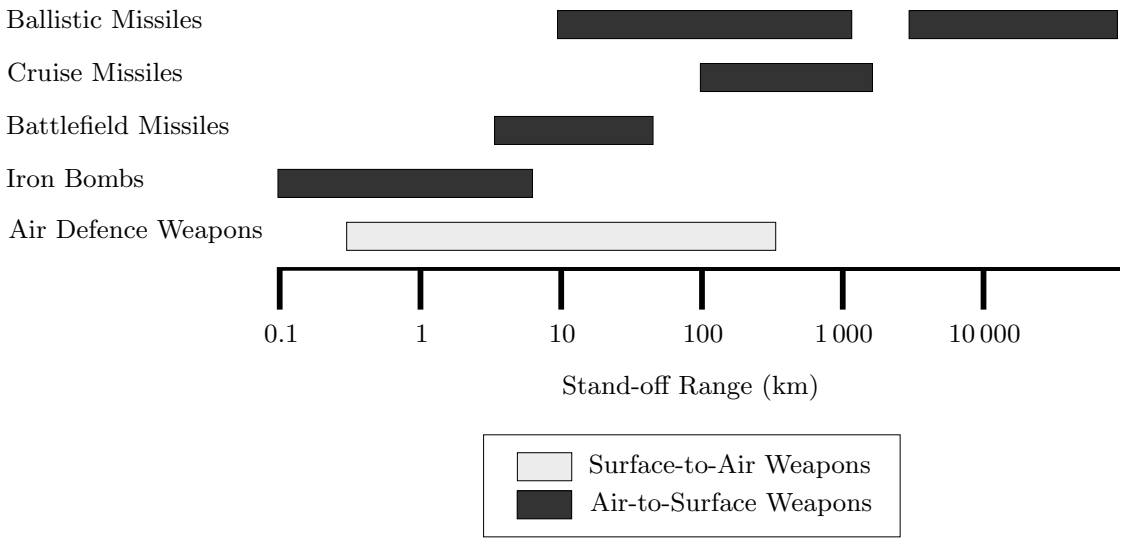


FIGURE 2.6: *Stand-off distance comparison of land-attack weapons and range spans of air defence WSs [121].*

Threats are evaluated during the TE process by different algorithmic models — more detail on this process will be provided in the next chapter. Typical models include those utilizing certain measured variables (*e.g.* heading and speed) to estimate the intent of different threats, whereas more sophisticated models attempt to predict the most probable aircraft attack maneuvre. Different attributes are also used in conjunction with the models in order to increase their accuracy. These attributes may include the threat type, weapon envelope and origin [169]. During this process, each threat is allocated a threat value which is a measure used to compare the levels of danger that different threats present to the defended system.

Although the scope of this thesis is limited to the evaluation of fixed wing aircraft as possible threats, numerous sources have argued that the use of manned fixed-wing aircraft for attacking is declining. According to Hutchings and Street [86], reliance on fixed-wing aircraft for attacks on DAs are predicted to decrease drastically— tactical air-to-surface missiles, *Unmanned Combat Aerial Vehicles* (UCAVs) and supersonic cruise missiles will be the main threats to future (2020+) GBAD systems. Even so, the algorithms developed for fixed-wing aircraft may perhaps be tailored for other threats if this functionality is required in the future.

2.4.4 Ground-based Weapon Systems

WSs are strategically deployed in order to protect DAs by countering threats. A WS can be defined as any equipment, under the defenders' control, which can be used to eliminate/deter a threat. Most WSs are semi-automated[9] systems which cooperatively interact with sensor systems in order to destroy or deter threats. To destroy threats, a WS can either hit a threat with projectiles and warhead fragments or subject the threat to a large pressure gradient [121]. Two examples of typical ground-to-air WSs are depicted in Figure 2.7.

(a) *Rheinmetall Denel Munition Oerlikon Skyshield [162]*

(b) *Starstreak missile installation [200]*

FIGURE 2.7: *Two examples of typical ground-based WSs.*

There exist a variety of different types of WSs, but the three main classes of WSs include kinetic kill vehicles[10], directed energy weapons[11] and electronic counter measures. Kinetic kill vehicles can further be broken down into the two subclasses of gun WSs and missile WSs. The focus in this thesis will be on kinetic kill vehicles. The stand-off ranges of the WSs employed by

[9] A human operator still makes the final decision.

[10] A collective term referring to all projectile-fired weapons.

[11] Directed energy weapons emit highly focused energy in order to damage targets and can include high-energy microwave devices, lasers, radio frequency transponders or sonic weapons [223].

combat aircraft as well as the possible ranges for air defence weaponry are shown in Figure 2.6. The placement of WSs is of major concern for the defending force and is a separate optimisation problem in its own right, which is beyond the scope of this thesis. Some information on this topic is, however, provided to build an understanding of the typical formation of a GBAD system.

The different types of WSs are generally deployed in a layered arrangement, which encompasses various different types of WSs into an integrated near-impenetrable air defence. The WS in a specific layer are of similar nature and are mainly responsible for defending the airspace within the associated layer. There are typically four layers, namely an inner, middle, outer and in-depth layer [155]. The effective ranges and altitudes of the different layers are depicted in Figure 2.8.

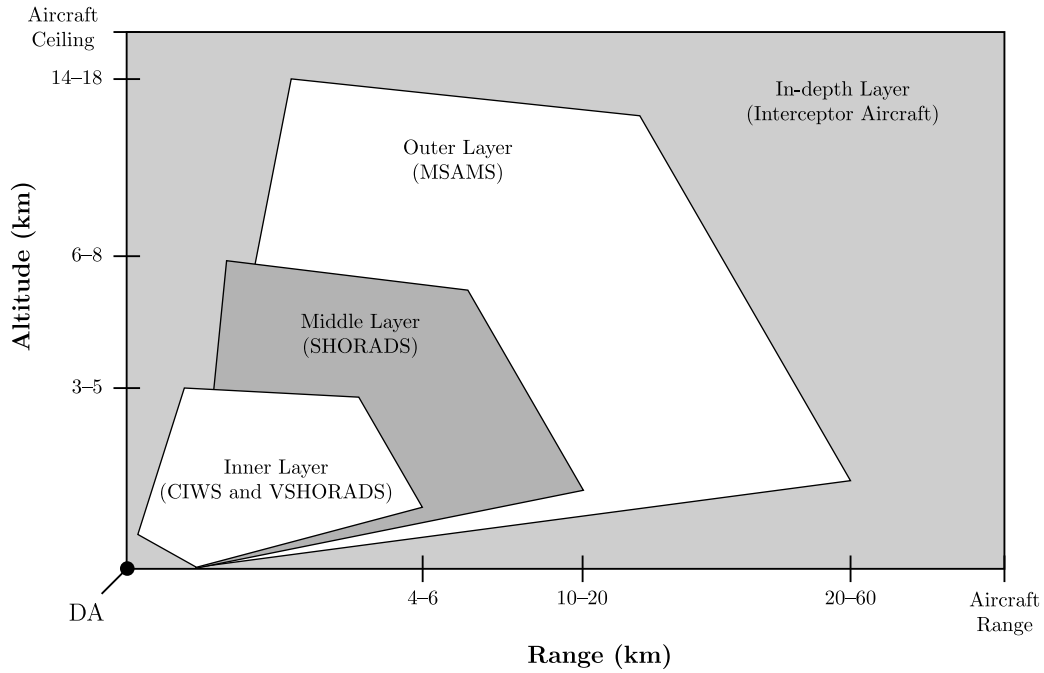


FIGURE 2.8: *Layered air-defense in context [167].*

The inner layer consists of *Close-in WSs* (CIWSs), such as the Mantis 35 mm guns, and *Very Short-Range Air-Defence* (VSHORAD) systems. The middle layer consists of *Short-Range Air Defence* (SHORAD) systems, such as Thale's Starstreak missile shown in Figure 2.7. The outer layer employs *Medium-Range Surface-to-Air Missiles* (MRSAMs), such as Denel Dynamic's Umkhonto Missile, whereas the in-depth layer corresponds to *Long-Range Surface-to-Air Missiles* (LRSAMs) and employs intercept aircraft to counter threats [155]. The WA subsystem in this thesis is mainly concerned with the inner, middle and outer layers, since these are the layers during which ground-based WSs are generally used. South Africa's GBAD system employs three types of WSs with different maximum ranges: 35 mm guns (4 km), VSHORAD missiles (6 km) and various short–medium range missiles (15–30 km) [53].

A WS can destroy a threat with a so-called *Single-Shot Hit Probability* (SSHP) which is a function of the range at which the target is engaged, the threat type and characteristics as well as the conditions of the surrounding environment [21]. The SSHP is the probability that a WS will hit the intended target with a single shot[12]. A successfull hit may refer to one (or more) of three damaging mechanisms whereby it can destroy or damage a threat:

[12] *Shot* may here refer to a single launched missile or a burst of rounds, depending on the type of WS employed.

- Projectiles or surface-to-air missiles directly hit the target,
- fragments ejected by the warheads carried by projectiles or surface-to-air missiles hit the target, and
- pressure from the warhead blast is applied to the skin of the target.

There are two different objectives when employing WSs — *attrition* and *deterrence*. Attrition refers to assigning the most effective WS in order to destroy a threat. In contrast, with deterrence the goal is to scare away threats in an attempt to force them to abort the attack. This goal is generally achieved by allocating multiple WSs to the threat (similar to the notion of suppression during ground operations). In this thesis, the focus is only on attrition, since no freely available information could be obtained to prevail in the mode of deterrence attack. Model building in support of engagements in deterrence mode is a highly complex task because of the large number of irreducible uncertainties involved in determining the success of a deterrence attack.

2.4.5 Defended Asset Characterisation

Prior to an engagement, a prioritised list of critical assets is typically compiled from all those under the commander's control. From this list, certain assets are classified as DAs [167]. A DA may be described as any entity (or a cluster of entities) of strategic importance to a commander and can range from a fixed installation, such as a command centre, bridge or a barracks, to moving objects, such as convoys or ships. In this thesis, only fixed installations are considered as possible DAs.

Each asset has an importance value assigned to it by the defending force. This importance value quantifies the relative importance to the defending force of protecting the asset in question. Several variables may influence the importance of a specific asset, such as its repairability, vulnerability and vital importance [59].

Repairability of a DA refers to the ability to recover from inflicted damage, and is determined based on the manpower, equipment and time required to repair the asset to a functioning state. *Vulnerability*, on the other hand, refers to the extent to which an asset is susceptible to damage and surveillance during an attack. Armour, position, countermeasures and camouflage are several measures which may be utilised to decrease an asset's vulnerability. Finally, *vital importance* is the degree to which the mission's success relies on a specific DA. Assessing the impact that the destruction of an asset will have on the mission's success is one way of determining its vital importance. A command centre, for example, is more important in respect of ensuring mission success (for maintaining C2 superiority) than a redundant sensor system.

The resulting importance value — which is usually a collective representation of the variables explained above — is used to prioritise DAs so that more important DAs are afforded preference when defending assets and allocating resources (WSs, ammunition and sensors). When a DA is destroyed during a mission, its importance value will is reset to zero, since it is not worth protecting anymore. It is therefore possible that the importance value of DAs can change as a battle unfolds.

2.4.6 The Human Machine Interface

Decision making in a GBAD environment is clearly a highly complex task. It is, however, ultimately a human operator who decides whether and how each aerial threat should be engaged

— not a fully automated system. It is therefore of the utmost importance to ensure that the decision support information communicated to the human operator is as clear and uncluttered as possible. Otherwise, ineffective decision making in a GBAD environment can result in severe (possibly catastrophic) consequences if inappropriate decisions are made. The human operator should be afforded the opportunity to make effective use of the available data for the purposes of analysis, interpretation and decision-making.

The *Human Machine Interface* (HMI) is a console which displays certain information collected by the sensor systems and results of the TE and WA processes to the operator. Besides the information provided to assist an operator in decision making, the operator can also interact with the HMI to issue commands, change settings and perform other operations [217]. Operators interacting with a typical HMI employed within a military environment may be seen in Figure 2.9.

The form of decision making explained above requires the integration of various data sources [113]. In order to succeed in this highly stressful and dynamic decision making environment, it is required that the FCO should possess a high level of tactical expertise and knowledge of the type of threats, prevailing legal frameworks and assessment heuristics from experience [41]. Training and experience are, however, not enough to ensure tacit decision making. According to Morrison *et al.* [133], the importance of ensuring that information is meaningful, timely and easily accessible cannot be underestimated. Another important consideration of the proposed solutions it that the solutions should correspond and reaffirm the FCO's intuition. When making decision in a life threatening situation, it is important that the FCO understands the reasoning behind the decision making process and not follow suggestions blindly. The preliminary design of a HMI is described in a later chapter of this thesis, taking into account the various human factors and technological limitations, and aiming to document design guidelines for the detailed design of an effective HMI.

FIGURE 2.9: *Operators interacting with a DSS through a HMI [160].*

2.5 Chapter Summary

The objective of this chapter was to provide an holistic view of the constituent theories and physical components of a TEWA DSS within the context of a GBAD environment. The first part (§2.1) contained an introduction to the concept of NCW and a description of the modern concept of NCW (§2.1.1). Special emphasis was also placed on the importance of implementing NCW principles into a TEWA DSS (§2.1.2). After forming a foundation for the concept of NCW, the notion of C2 was introduced in §2.2. The well-known OODA cycle was described in §2.2.2, which will be used later in this thesis to model the C2 processes associated with the TEWA decision cycle.

The second part of this chapter contains a discussion on the physical elements forming a TEWA system. Special emphasis was placed on the physical environment (§2.4.1), sensor systems (§2.4.2), aerial threats (§2.4.3), WSs (§2.4.4), DAs (§2.4.5) and the HMI (§2.4.6) of such a system.

CHAPTER 3

The Current State of TEWA Knowledge

If I have seen further than others, it is by standing upon the shoulders of giants.

— Isaac Newton

Contents

This chapter opens with an introduction to the origins of this research project, after which previous research available to the author is elucidated from both a *Domestic* and an *International* viewpoint. The domestic research refers to all the TEWA-related work done in South Africa, mainly by members of the TEWA Centre of Expertise at Stellenbosch University, while the international research includes work by members of external international institutions which is available in the open literature. This chapter concludes with a brief overview of existing TEWA DSSs, in order to provide some context on the typical nature of these systems.

Several international studies have been conducted on TEWA DSSs in a GBAD environment. Because of the sensitive nature of this work, however, most of the detailed research done by these studies has been classified. As such, the majority of the studies available to the author were conducted within the TEWA Centre of Expertise at Stellenbosch University.

3.1 South African GBAD Programme

This project has its roots in the South African GBAD programme, which started in the late 1990s and mainly entailed modernising the South African air-defense capability. This project was broken down into several phases. Although the first phase involved the establishment of

an early local warning system responsible for early threat detection and threat evaluation, the contract mainly entailed integrating existing, proven equipment [167].

After the project's concept exploration stage, it was decided to subcontract the development of a TEWA system to an external company. This external company had already developed a combat proven point-based TEWA DSS and it was assumed that they would be able to adapt the current system so that it is applicable in a South African area-based environment. The company in question delivered the product as a black-box system, thereby making it extremely difficult for South African companies to modify and improve upon the system.

The TEWA system was commissioned, but the end-product required modifications to the algorithms before the initialisation of the full-scale utilisation stage. This is the main reason for formation of the TEWA Centre of Expertise at Stellenbosch University — to build a South African knowledge base of TEWA-related expertise. Modifications were required to the procured system, because some advanced test scenarios generated counter-intuitive solutions. Because of this contradiction, the system would not be able to provide the essential decision support required by the operators. These problems may have been the result of a conversion from the point-based to area-based system, since some algorithms/assumptions are not directly transferable. Without more information, however, it is not possible to determine the exact root-cause of this problem.

During 2003, the contract to further the development of the GBAD system was placed with *Denel Integrated Systems*. In 2012, the system was delivered to its end-user, the South African Army, which led to the start of the utilisation phase. According to Denel's newsletter [157], the system is currently in the support phase and undergoing upgrades, while the next stages of the GBAD program are initiated. An example of a typical upgrade that forms part of this project, includes integration of Denel Dynamics's *Beyond Visual Range* (BVR) Umkhonto surface-to-air missiles.

Furthermore, in March 2014 Rheinmetall AG[1] signed a contract with the South African Army to modernise the ageing South African air-defense systems. The contract made provision for implementation of the *Skyshield* fire control units as well as refurbishment of the existing 35 mm guns with the purpose of using *Rheinmetall's* latest generation *Ahead Airburst*[2] ammunition, giving them MK VII status. These improvements are therefore focused on modernising South Africa's CIWSs. The contract is scheduled for completion by 2017 [162].

From the preceding discussion, it is clear that the South African GBAD system is undergoing continuous development and there are, as such, still numerous areas for improvement. The objectives in this thesis are structured in such as way so as to add to the South African knowledge base in respect of TEWA DSSs, with the long-term goal of developing a proudly South African TEWA DSS.

3.2 Domestic TEWA Knowledge

Domestic research related to the South African GBAD program may be categorised into two main research areas: *Threat Evaluation* (TE) and *Weapon Assignment* (WA). This section is

[1] *Rheinmetall AG* is an international supplier of automotive components and defense equipment. Their defense sector is advertised as setting global standards in the discipline of network-enabled warfare capabilities, which includes electro-optics and simulation technology [161].

[2] Each Ahead Airburst round contains a large amount of sub-projectiles and a programmable fuse. Depending on the range and speed of the threat, the fire control computer determines the optimum detonation time and programs the projectiles as it traverse the barrel [161].

a chronological review of the various contributions to this research, emphasizing their main outcomes and focus points.

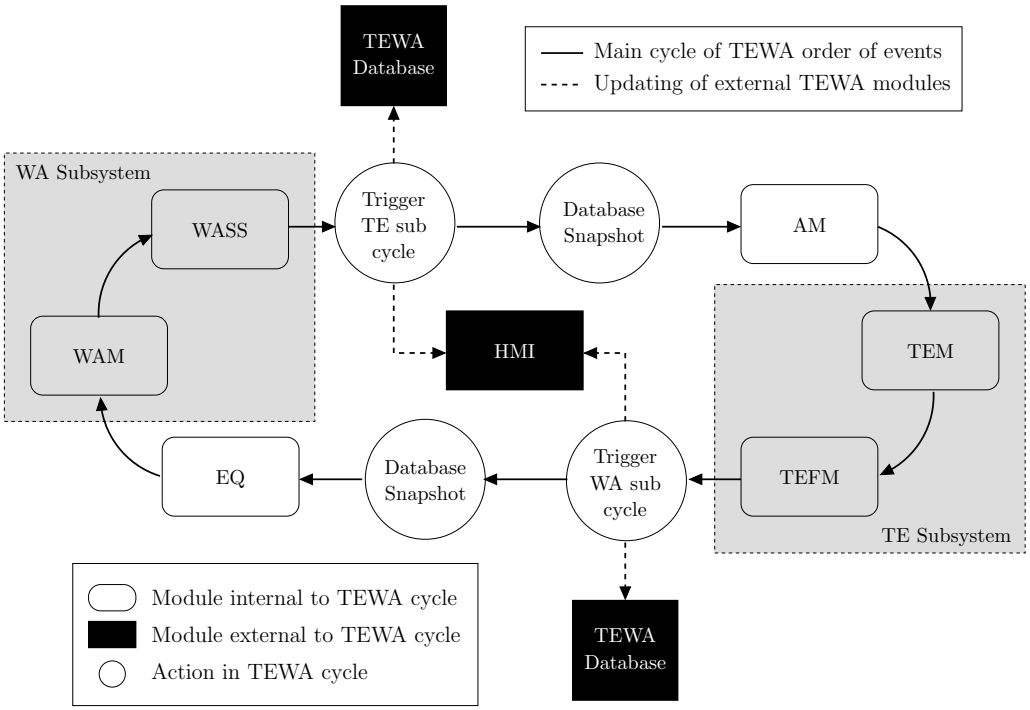


FIGURE 3.1: *Order of events within the TEWA cycle [117].*

3.2.1 Threat Evaluation

In 2010, Roux and Van Vuuren [167] designed a generic TE subsystem architecture for a ground-based air-defense environment. Special consideration was placed on determining the required sequence of events in a TEWA cycle and the details surrounding the TE models. The proposed sequence of events for a TEWA system is depicted in Figure 3.1. In the proposed TE subsystem architecture, an *Attribute Management* (AM) component analyses the measured threat attributes (*e.g.* speed and altitude) which are obtained from the sensor systems. This information is then used to calculate a number of derived attributes, such as acceleration, which are in turn sent to the *Threat Evaluation Model* (TEM) component. The TEM component is the core component of the TE subsystem, and contains a number of TE models each of which attempts to quantify the individual level of threat of each detected aircraft with respect to the *Defended Assets* (DAs).

The TEM component consists of three hierarchies of mathematical TE models. These three levels of models, ranked in order of increasing sophistication, include various *flagging models*, *deterministic models* and *stochastic models*.

At the lowest level of sophistication Roux and Van Vuuren [167] proposed a suite of binary flagging models each of which flags aircraft for attention if an abrupt change is detected in their kinematic behaviour, possibly suggesting hostile behaviour. Typical flagging criteria may include drastic changes in altitude, deployment of counter measures, dropping of paratroopers or any other hostile actions [167]. Although these flagging models do not rank the aircraft according to the level of threat posed to the defended system, they should aid the the operator in analysing a air-defense scenario, especially when a large number of unknown aircraft are detected. These

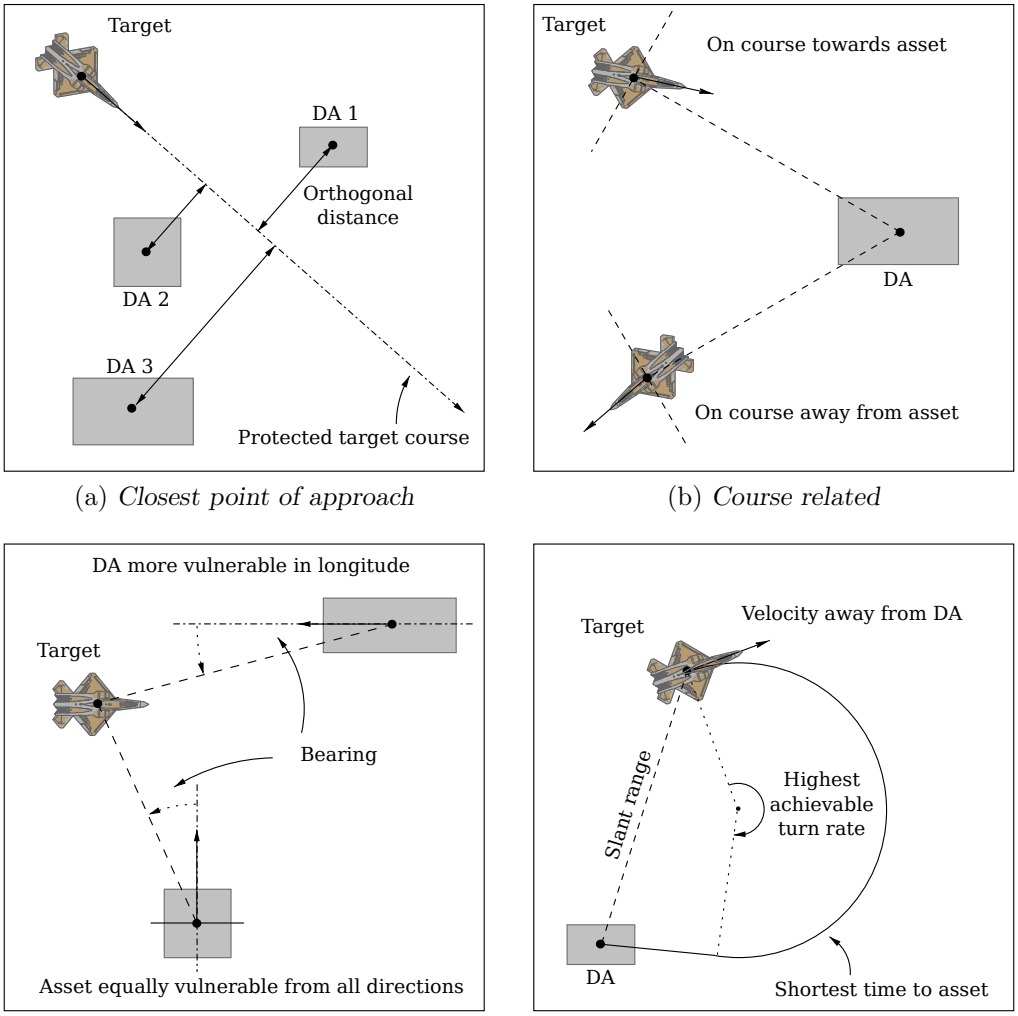
models are easily implementable since only aircraft kinematic data are used to flag a specific aircraft — no derived attributes are utilised, nor is any pre-deployment information on aircraft type, weapon envelope or doctrine required.

At the next level of sophistication, deterministic models, utilize the measured kinematic and positional data of aircraft, together with the attributes derived from these data to estimate the threat level associated with an aircraft. These models calculate certain characteristics of an aircraft to determine the level of threat that it poses to the DAs, and are typically based on the time that an aircraft is estimated to travel to a DA or any course, heading or distance related measure. More information on a number of some specific deterministic models are provided later in this thesis. For the implementation of deterministic models, basic pre-deployment information, such as the position of a DA, the importance value associated with a specific DA and, in some cases, maximum aircraft turn radii, is required [59]. The exact amount of information required, however, depends on the specific implemented deterministic model. For example, no aircraft-specific information is required to determine the distance between a threat and a DA. When the time to weapon release of an aircraft is estimated, however, it is required to know the threat's most probable weapon envelope, as well as the attacking aircraft's weapon specifications, is required.

Heyns and Van Vuuren [78] developed four deterministic models with the help of domain experts for populating the TEM component of their proposed TEWA system. The principles on which these deterministic models rely are illustrated in Figure 3.2. These models are applicable to fixed-wing aircraft and were designed only to evaluate a threat based on kinematic data. As mentioned previously, these models attempt to quantify the level of threat posed by an aircraft to the defended system, accounting for its course, heading and distance related measurements. Roux and Van Vuuren [168] expanded and improved upon the deterministic models proposed by Heyns and Van Vuuren [78].

The final level of TEMs, proposed by Roux and Van Vuuren [168], are a suite of probability-based stochastic models . These models take into account aircraft kinematic data, detailed DA deployment data, pre-deployment data about the enemy arsenal as well as doctrine information. All this information is collectively used, in real time, to estimate the probability that an aircraft will achieve a kill with respect to a particular DA. Although Roux and Van Vuuren [167] proposed that this information may be determined solely from pre-deployment data and expert opinions, Van Staden and Van Vuuren [216] developed a hidden Markov modelling paradigm for classifying the *Formative Element Combination* (FEC) of an aircraft. The FEC of a specific threat includes information such as the most probable aircraft type, origin, weapon envelope and attack technique associated with a detected threat. This modelling approach facilitates FEC determination based on objectively measured data rather than on subjective, error-prone expert opinion and pre-deployment information. If this FEC information is known, then the advanced models of Roux and Van Vuuren [167] are expected to yield more realistic results than the simpler flagging and deterministic models [216].

The output of each the TEM described above, is a threat value in the real interval interval $[0, 1]$, with 1 denoting the most threatening behaviour and 0 indicating no threat at all. The three tiers of TEMs are meant to run concurrently in the TEWA system. A combination of TEMs may be selected depending on the amount of available from the sensor systems and the requirements of the FCO. For example, in the absence of certain sensor and/or preprogrammed data, a simpler model will be selected, but as more information becomes available the system should rely on the more sophisticated models. The FCO should, however, also be able to configure how the system selects the combination of TEMs, thereby ensuring that the system supplements the FCO's analysis style and promotes timely, naturalistic decision making.



(a) *Closest point of approach* (b) *Course related*

(c) *Attack Angle* (d) *Shortest time to DA*

FIGURE 3.2: *Graphical illustration of the working of deterministic TE models (adapted from [168]).*

It is important to note that the threat values returned by the TEMs should only be in a comparative manner — they do not represent absolute estimate values. A threat value of 0.5 may, for example, not indicate the same level of threat from one engagement to the next. As a result, the same TEMs should be active with respect to all aircraft. To illustrate, the stochastic models cannot be active for some threats, while only flagging models are active for other threats. The threat values resulting from different TEMs are not directly comparable — a flagging model will allocate a binary value of 0 or 1, whereas the deterministic and stochastic models will output a threat value on the interval $[0, 1]$.

Only flagging and deterministic models are considered further in this thesis, since the information required to sufficiently test the stochastic models are not readily available. As will be demonstrated in the later chapters, however, the deterministic models do provide an effective means of distinguishing between different levels of threatening aircraft, thereby making these models a valuable aid to FCOs.

As illustrated in Figure 3.1, Roux and Van Vuuren [168] proposed that the final component of the TE subsystem should be a *Threat Evaluation Fusion Model* (TEFM) component. The threat values returned by the TEMs, deterministic and stochastic, are threat values for each

aircraft, with respect to each DA, per TEM. The WA subsystem, however, requires one threat value per aircraft which represents a system-wide threat value for that aircraft. It is therefore required to fuse the set of threat values returned by the TEMs for a particular threat into a single system threat value for that aircraft. Different methods have been proposed to fuse these threat values [78, 104, 167]. These methods vary from consensus ranking procedures and value function procedures to goal-oriented approaches. None of the previous members of the TEWA Centre of Expertise at Stellenbosch University has, however, sufficiently demonstrated the functioning of this fusion process, hence thesis Objective X (of §1.4). More detail on the implementation of the TEFM component will be provided in a later chapter of this thesis.

After conclusion of the TE process, the results of the TEFM component, together with the current air-picture and other relevant information, are presented to the FCO through the HMI in order to provide germane decision support to the operator.

3.2.2 Weapon Assignment

The WA subsystem design was initiated by Potgieter [155] in 2008. He developed a high-level, generic representation of a workable system architecture which utilizes the output information from the TE subsystem. This design included an *Engagement Efficiency Matrix* (EEM) component and a *Weapon Assignment Model* (WAM) component which is similar to the TEM component of the TE subsystem. The EEM component filters the *Single Shot Hit Probability* (SSHP) values of the WSs to account for, among others, environmental conditions and terrain restrictions which will have a detrimental effect on the accuracy and/or reliability of the weapon systems.

The WAM component forms the central part of the WA subsystem and contains a variety of algorithms for solving the WA subproblem. The WAM component's purpose is similar to that of the TEM component within the TE subsystem. The WA models utilise the system threat values of each aircraft, together with the outputs of the EEM component, to propose high-quality solutions in respect of WS assignments to threats. These WS assignments are, in turn, accepted or adjusted by a operator. In order to populate the WAM, a number of heuristic rule-based and exact computation-based WA models were developed and compared, independently from the TE subsystem.

Lötter et al. [120] proposed a classification of four types of WAMs. In order of increasing complexity these classes are: (i) single-objective, static WAMs, (ii) multi-objective, static WAMs, (iii) single-objective, dynamic WAMs and (iv) multi-objective, dynamic WAMs. A graphical representation of these classes of WAMs is shown in Figure 3.3, where the level of grey-shading indicates the degree of complexity of the algorithm. The two classes of static WAMs both solve the WA problem for only the current time stage — suggesting an allocation of WSs to threats, without taking into account weapon setup times or SSHP values which change over time. The two dynamic WAMs, on the other hand, considers a pre-determined time continuum to schedule the best fire-windows for the engagement of threats by WSs. These models are explained in more detail in the following paragraphs.

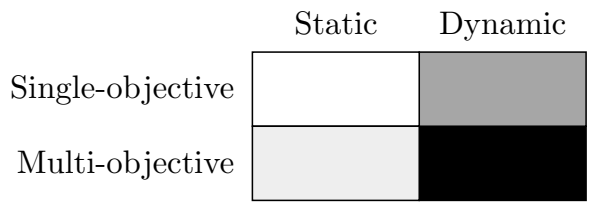


FIGURE 3.3: *Complexity levels of WAMs, with darker shades of grey denoting increased complexity [120].*

In 2009, Du Toit [59] expanded upon the WAMs by Potgieter [155] by reformulating the WA problem as a dynamic weapon target assignment problem with expected threat priority accumulation, allowing for the possible prediction of the future positions of an aircraft. Three existing techniques were reviewed and one newly developed technique was proposed which may be used to compute solutions to this novel, dynamic weapon target assignment problem.

Lötter *et al.* [116] further expanded on the work of Du Toit [59] in 2009 by modelling the weapon assignment problem as a multi-objective, static decision problem. A questionnaire was used to obtain feedback from military experts in order to elicit possible objectives for the WA problem. Two objectives were selected for use in their research. These objectives are: (i) minimizing the total cost associated with assigning weapon systems (ammunition expenditure) and (ii) minimizing the expected accumulated threat survival probability. Their WAM formulation was solved for a single scenario by implementing a multi-objective, evolutionary metaheuristic — the *Nondominated Sorting Genetic Algorithm II* (NSGA II) — which provided promising results by proposing high-quality weapon assignments to threats.

To supplement the work of Lötter *et al.* [116], Van der Merwe [215] formulated the WA problem as a vehicle routing problem with time windows in 2013, where vehicles (WSs) have to deliver commodities (ammunition) to customers (threats) within a prespecified time-frame (fire window). This new methodology added a dynamic scheduling element to the WA problem, thereby enabling the TEWA system to account for the weapon setup time which includes operator response time, reloading of ammunition, orientating the WS and the time required to compute the interception point(s) before engaging.

The weapon setup time will vary from WS to WS and is analogous to the vehicle travel time in a vehicle routing context. Furthermore, the EEM is analogous to the probability that a vehicle will successfully serve a customer during a first visit. Thus, in the formulation of Van der Merwe [215], the EEM is used in the objective function, instead of the weapon setup time. This formulation attempts to minimize the total expected survival probability of all the threats over the entire service window (total scenario time). The model generates a schedule of events (engagements) in an attempt to counter the threats as effectively as possible. A schedule is an indexed discrete-time intervals list of decision instants with associated events. To solve this new problem formulation, a hybrid metaheuristic approach of simulated annealing and tabu-search was suggested.

Lötter and Van Vuuren [117] finally completed the architecture design of the WA subsystem for a *Surface Based Air-Defence*[3] (SBAD) environment in 2014. Their design included the components depicted in Figure 3.4 — an *Engagement Quantization* (EQ) component, a WAM component and a *Weapon Assignment Solution Selection* (WASS) component. The EQ component is an expanded and improved version of the EEM component proposed earlier by Potgieter [155]. The WAM component is similar to the one presented by Potgieter, but allows for the four classes of WAMs of varied complexity shown in Figure 3.3. These different WAMs are aimed at allowing the FCO to tailor the TEWA system to a preferred GBAD system setup, before or during a combat engagement.

In order to ensure that only the necessary information is shown to the FCO, the WASS component facilitates selection of a combination of solution techniques for solving the adopted WAM and filters out dominated solutions. The FCO is then presented with a small set of Pareto-optimal trade-off solutions from which to select an appropriate response.

[3]The term *Surface* is used so as not to single out a ground-based, sea-based or snow-based environment.

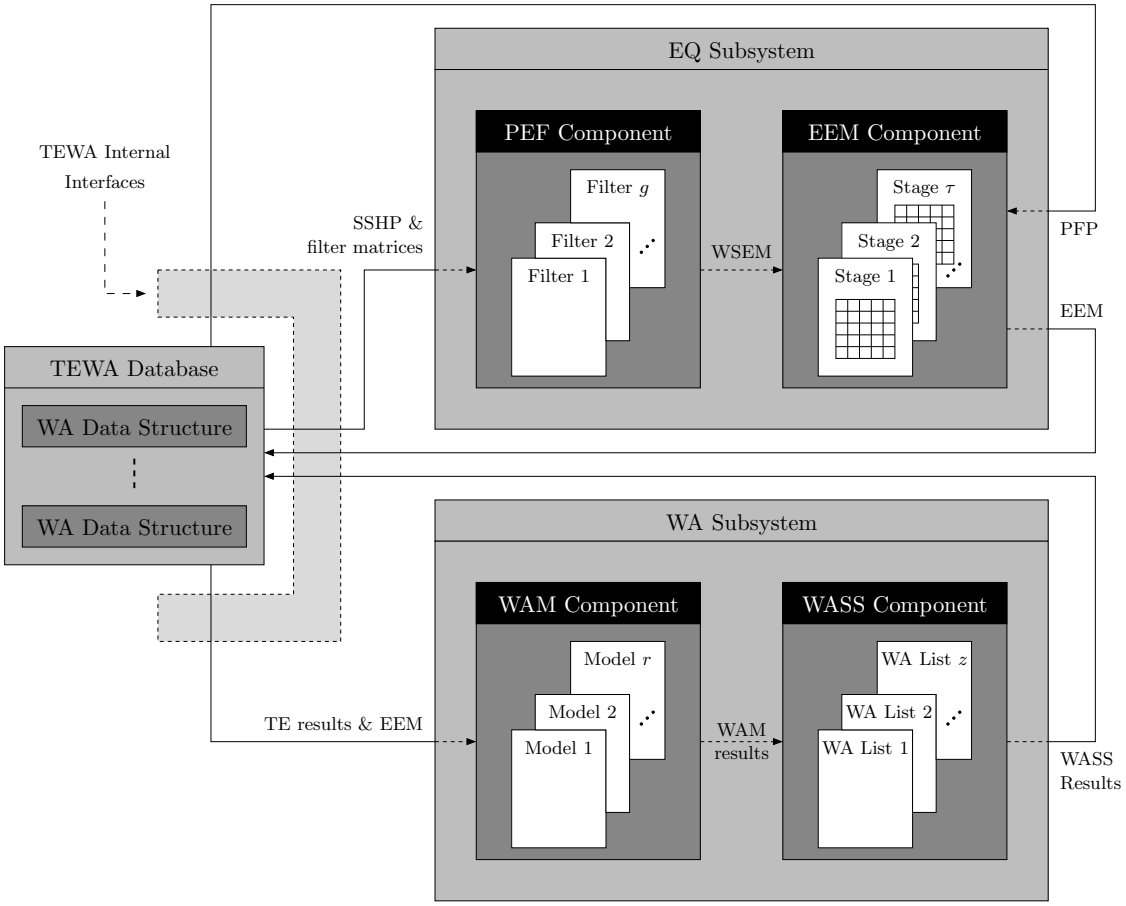


FIGURE 3.4: *Composition of the WA subsystem [120].*

3.3 International TEWA Research

The availability of detailed international research on TEWA-related subjects is quite limited. In recent years (2003+), however, an increasing number of organisations have published TEWA-related articles. Most of these articles are essentially high-level system overviews, and generally do not provide detailed information on the algorithms forming the core of a TEWA system. It is possible that many of the complicated TEWA-related sub-problems have already been solved in the past, but are kept confidential for national security and competitive reasons. This section contains a brief literature survey of recent and current TEWA research reported in the open literature.

One of the more established researchers in this field is Liebhaber, who is a member of the Pacific Science and Engineering Group [113, 114], who focuses on human-machine interfaces for highly complex systems (within the air-defense environment). Another researcher who does TEWA-related work, is Falkman [69], based at the University of Skövde whose projects are mainly funded by SAAB. There are also several researchers from the Defence Research and Development Group in Canada [22, 149] with a similar focus. The Johns Hopkins APL Technical Digest has also published various high-level articles on the subject of air-defense. More recently, however, their focus has shifted towards optimisation for missile defense [100, 186]. Another noteworthy organisation, who does not publish many articles in the open literature, but has been working in TEWA-related fields for some years, is the Lincoln Laboratory of the Massachusetts Institute of Technology.

Among other responsibilities, the Lincoln Laboratory develops and assesses integrated air-defense systems for defense against various types of air threats. Their emphasis is on rapid prototyping sensor and system concepts, as well as on algorithms. They work closely with the US Government and defense industry in order to ensure a rapid concept-to-delivery cycle. Unsurprisingly, none of this work is found in the open literature. It is, however, anticipated that their research is well ahead of its time, considering the complexity of the United States' integrated air-defense systems. There is one noteworthy document titled "*The Ground Environment Problem in Air Defence: An Appraisal of the Lincoln Transition System*," dated 1954, which details the shortcomings of the US's Lincoln Transition System — the predecessor of the cold-war era *Semi-Automatic Ground Environment* (SAGE) [202, 203]. This document was classified as "Secret" at the time, but has since been declassified, reaffirming the nature of the work done at Lincoln Laboratory.

3.3.1 Threat Evaluation

The parameters employed in TE research are very similar across institutions. The basic parameters, as suggested by Liebhaber [113], are predominantly incorporated into TE algorithms. The only difference is how these parameters are combined to obtain a holistic view of the level of threat posed by aerial entities to the defended system. Some threat evaluations models are based on fuzzy logic [105], others on artificial neural networks [9] and some on Bayesian networks [70, 94, 143]. Johansson and Falkman [95] evaluated the fuzzy logic approach and Bayesian network approach to TE and concluded that a hybrid of these two approaches seems promising and viable for future research.

The outputs of TE algorithms are generally on a scale from 0 to 1 (or sometimes 0–100), identical to the approach suggested by Roux and Van Vuuren [168].

3.3.2 Weapon Assignment

In WA[4] related research, different models have been proposed for the WA problem (similar to the models described in Section 3.2.2) as well as a variety of solution methodologies for solving these combinatorial optimisation problems. The WA problem is nondeterministic polynomial time-complete; for a proof see [115]. Because of the time constraint placed on solving this problem in a highly-dynamic combat environment, heuristic methods are generally preferred to find good solutions (not necessarily optimal) in the limited time frame.

No exact methods exist for solving even small-sized WA problem instances (for example, containing 20 WSs with 20 threats) exactly [2]. Although heuristic methods have been proposed for solving this NP-hard problem, without exact algorithms for solving realistic WA problem instances optimally, it is difficult to analyse the performance of proposed heuristic algorithms. Therefore, the quality of solutions generated by a wide variety of heuristic algorithms are currently compared in order to evaluate the performance of a specific algorithm. Some of the algorithms suggested to solve WA problem instances include particle swarm optimisation [25, 39], ant colony optimisation [40], a large-scale neighbourhood search algorithm [2] and different genetic algorithms [111]. The genetic algorithms and the particle swarm optimisation algorithms seem to excel in terms of producing reliable solutions rapidly, for many current WA models. The TEWA Centre of Expertise at Stellenbosch University, also proposed various promising hybrid algorithms [155, 215].

[4]WA is sometimes referred to as *weapon allocation* or the *combat power management problem* in the international literature.

3.3.3 Human-Machine Interaction

During the early 1990s, the *Tactical Decision Making Under Stress* (TADMUS) programme in the United States of America involved research in support of improved DSS design by integrating cognitive theory and human-machine interaction technology [133]. This programme attempted, in essense, to design a DSS according to a "naturalistic" model for decision making in environments that are characteristically highly complex, short-fused and dynamic. Some of the TADMUS programme's results have since been declassified, and include discussions on limitations that were present in a preliminary designed DSS [87]. This research programme was most likely the foundation for a host of related research that followed.

Several other institutions have conducted similar research, often using the results of the TADMUS programme as a starting point. Research on the enhancement of SA in the military air domain may be categorised into the following three broad categories, as suggested by Oxenham [148]:

Air picture enhancement. Research in this category is focused on the use of contextual information to enhance the air-picture available to operators — in particular, the management of target tracks [10]. This research forms part of the sensor-management research field, as described in Section 2.4.2. Furthermore, emphasis is also placed on the association of target tracks with airline tracks in order to prevent fratricide incidents [147].

Automated situation and impact assessment. Since the automation of unfolding situations and the assessment of possible impacts involve both human-related factors and technical concerns, research within this category is rather diverse. It includes, among others, philosophical issues related to the design of a DSS [170, 107], as well as the establishment of rigid rules for use in impact and situation assessment [171, 46].

Human factors. The incorporation of human-related factors into the DSSs is an attempt to promote naturalistic decision making and identifying human-machine interfacing limitations. Limitations are any DSS functionality that restricts the flawless operation of an user-friendly HMI. The domain of naturalistic decision making has focused on the methodologies operators apply when evaluating threats and prioritising targets for the purpose of developing cognitive-based DSSs [57, 113, 139]. HMI limitations, on the other hand, has focused on ways of enhancing the visualisation of data and streamlining the access of information to the human operators [29, 150, 176].

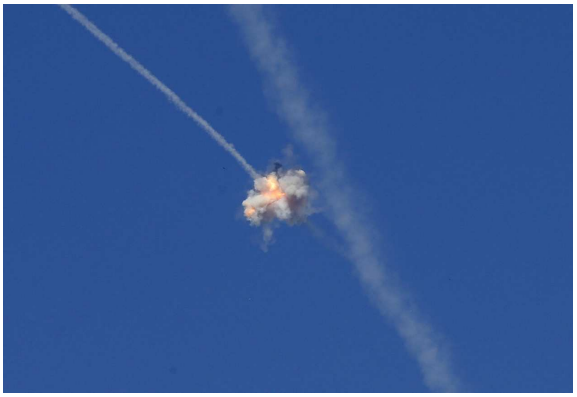
Many of results of the aforementioned HMI-related research emanated from the evaluation of existing systems. For example, Brown *et al.* [29] provided a detailed evaluation of a C2 system in aid of an anti-air warfare commander. Special emphasis was placed on the human-machine interaction and various suggestions were made to improve the existing system.

3.4 Existing TEWA Systems

Several TEWA DSS are in use around the world, but, the inner workings of these systems are typically kept secret for security and competitive reasons. Some of these systems are described very broadly in this section to provide some background on the application environment of a TEWA DSS. This discussion is not exhaustive, but is instead aimed at providing the reader with a practical overview of existing TEWA DSSs in actual use today.

With the ongoing Israeli-Palestinian conflict, Israel's *Iron Dome* system has become renowned in the popular news. The Iron Dome is an all-weather anti-RAM (*Rockets, Artillery and Mortars*) mobile defense solution with an advertised range of 70 km [158]. The system identifies a RAM threat and analyses the threat's trajectory. If the predicted impact point is within the protected area, an interceptor missile is launched to intercept the threat. The TE process therefore consists of calculating the impact point of RAM threats in relation to DAs. The Iron Dome system has been selected by Israel's Defence Ministry as the most comprehensive defense system against a wide range of (mainly RAM) threats at a relatively low cost.

(a) *A single Iron Dome battery firing an interceptor.*



(b) *Interception of a rocket by the Iron Dome.*

FIGURE 3.5: *The Iron Dome system in operation [82].*

Another TEWA DSS is the *Phased Array Tracking to Intercept of Target* (PATRIOT) system. This system was originally developed to counter Soviet fixed-wing and rotary-wing aircraft [202]. It replaced the Nike-Hercules missile system, which was the primary air-defense system of the United States of America for more than 25 years, predominantly during the cold war era [18]. The PATRIOT system is a stand-alone mobile air-defense solution, in the sense that all sensors and C2 equipment are part of the mobile installation. A graphical representation of the physical elements constituting a PATRIOT system is given in Figure 3.6. Because of this mobility, the PATRIOT system is utilised in the United States' Aegis ballistic missile defense system. Aegis is the naval, multi-mission component of the greater missile defense system [210]. There is, however, some controversy surrounding the PATRIOT system's success on the battlefield [79]. Nonetheless, the Patriot system is still undergoing continuous research [213] and, as such, will most likely remain the standard for other integrated air-defense systems in the foreseeable future.

The US Missile Defence Agency also demonstrated the ability of the Aegis destroyers to coordinate a counter response to a multitude of incoming threats [84]. The algorithmic models are collectively known as Lockheed Martin's Distributed Weighting Scheme, which utilizes TE and WA-related algorithms to determine the best vessel to engage an incoming threat. It would seem that the objective function is multi-objective and dynamic in the sense that cost is minimised and the survival probability of the threats are minimized and WSs are scheduled to engage threats at specific time-stages in order to ensure coordination among the three Aegis destroyers [71].

The main difference between the Iron Dome system and the PATRIOT system lies in their intended use. Whereas the Iron Dome is mainly a stationary system and the area that it protects is well-defined, the PATRIOT system is designed to adapt easily to changing scenarios and is more mobile [18, 158]. Another factor related to the difference in these two systems' intended use involves the difference in their unit cost per interceptor. The PATRIOT system's

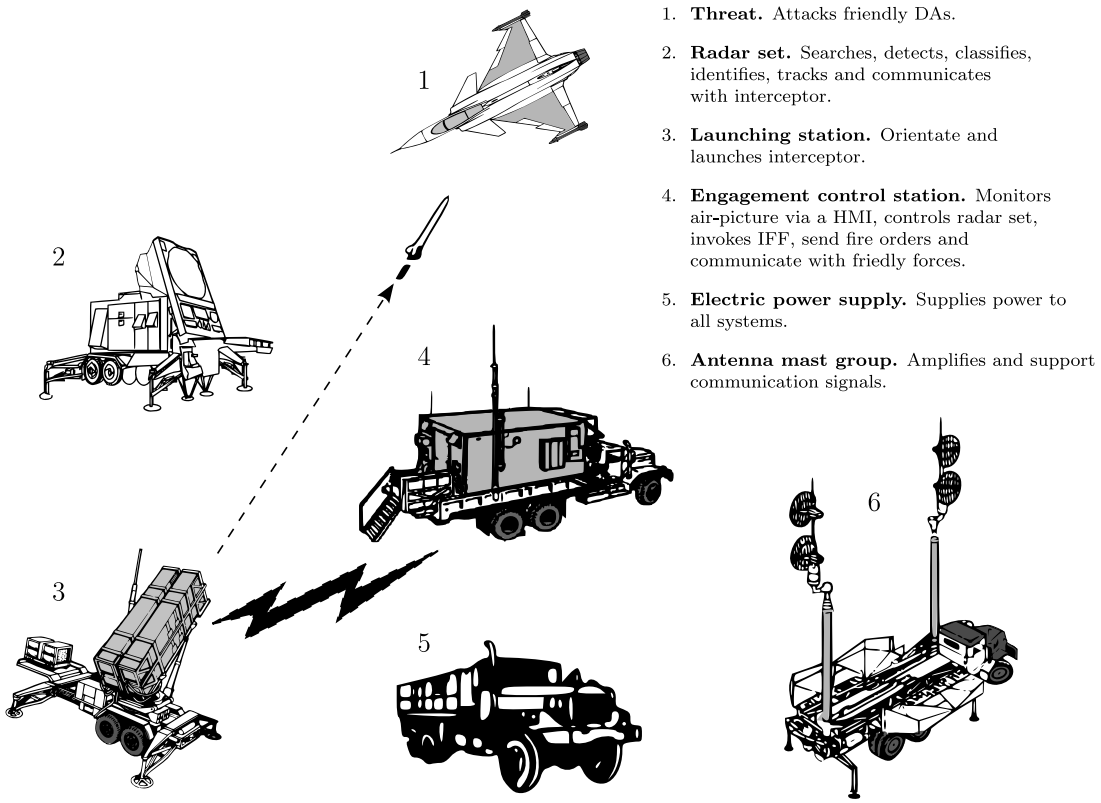


FIGURE 3.6: *Basic PATRIOT air-defense system operation (compiled from the information provided in [151]).*

interceptors cost approximately $3–4 million each [152], whereas the Iron Dome incurs a running cost of approximately $50–90 thousand [83] per interceptor. This major difference in unit cost suggests a significant difference in the intended threats to be countered by the two systems. The PATRIOT system is designed to engage all major aerial threats — including RAM threats, fixed-wing aircraft and ballistic missiles, while the significant cost of a single interceptor makes it inappropriate to counter smaller RAM threats. The Iron Dome, on the other hand, is deployed mainly to protect Israel's civilian population from RAM threats; the Iron Dome is not designed to counter long-range threats such as ballistic missiles or fixed-wing aircraft.

Another noteworthy air-defense system, similar to South Africa's SABLE system and therefore the most likely direct competition, is the Russian S-350E Vityaz system currently under development. The first deliveries are expected in 2016 to the Russian Army [13]. This system is advertised as a highly mobile system, with communication equipment, a fire control station and surveillance radar included as part of the system. Almaz-Antey developed this system to replace the renowned but aging S-300 air-defense systems.

The last integrated air-defense system considered in this section, which seems to be the most of exhaustive of existing alternatives, is the *Thales Advanced Air Defence* system. This system forms an integrated set of systems designed by Thales[5], Raytheon[6] and other third party manufacturers, with the aim of supporting timely decision making and effective responses for the protection of military forces, key assets, nations and citizens around the globe [199]. This system is being implemented to protect more than 10 million square kilometers of European airspace

[5] *Thales* has been a global supplier of world-class air-defense related solutions for the past forthy years [199].

[6] *Raytheon* is a global technology and innovation company specializing in the defense, security and defense-related markets [159].

in the foreseeable future. Thales follows a philosophy of designing their defense systems around cyber-secure open architectures, thereby ensuring highly flexible and robust systems. Thales' air-defense capability includes a fully tailored air-defense system with a C2 system, weapon coordination system, surveillance radars and mobile WSs.

With the above-mentioned TEWA-related systems in use around the world, companies aim to improve the algorithms which form the critical cores of these systems. Considering the competition between the companies designing these systems and the security issue, it is not surprising that no details are available about the algorithms and software architectures inside these systems.

3.5 Chapter Summary

This chapter opened with a summary of the current state of the South African GBAD project (§3.1). After introducing the South African GBAD programme, research related to this programme is reviewed from a domestic (§3.2) as well as an international perspective (§3.3). Both these sections contain subsections which distinguishes the available literature on the basis of WA and TE related work. This chapter closes with an overview of existing TEWA systems (§3.4) in order to provide some context in respect of the practical areas of this research area.

From the material in this chapter it is clear that extensive research has been conducted locally on the WA and TE subsystems, but a major problem is that these developments, for the most part, are done in isolation. There are a number of complicating factors with respect to the practical implementation of the proposed system architectures and models. Some of these questions may be derived from the aforementioned literature review and include, but are not limited to, the data fusion process, the solution methodology to implement, handling of uncertainty within the GBAD environment and evaluating the performance of the system as a whole. Although some of these limitations are addressed in this thesis, further work will still be required in order to translate the theoretical knowledge to practical implementation.

As mentioned previously, the main purpose in this thesis is to integrate and demonstrate the workability of the research done at the TEWA Centre of Expertise at Stellenbosch University. This does, however, not mean that all international research will be ignored. The work reviewed in this chapter will be used to facilitate the aforementioned integration and demonstration.

PART II

DEVELOPMENT AND INTEGRATION

CHAPTER 4

Simulation Development

The question of whether a computer can think is no more interesting than the question of whether a submarine can swim.

— Edsger Dijkstra

Contents

This chapter opens with an overview of the concerns typically encountered when modelling military systems as well as a discussion on the different classes of widely-used simulation approaches. The adoption of a specific simulation paradigm is also motivated. Thereafter, the steps of a typical simulation study are reviewed and explained. The core of the chapter details the implementation and assumptions made in this thesis in respect of the implementation of all the TEWA elements (sensor systems, DAs, WSs and threats) described in Chapter 2. The selected simulation environment is finally motivated in the final parts of this chapter, providing a high-level overview of the programme flow of the simulation model developed in this thesis.

4.1 Simulation of Military Systems

Many real-world systems may be modelled almost completely as some mathematical structure or concept, which may then be manipulated by established mathematical theory in order to analyse or optimise the underlying real-world system [98]. In many cases, however, the system parameters are not known with certainty or are variable, and the interaction between subsystems results in a too complex system to model according to purely mathematical relations. This is generally the case with TEWA systems for which a different modelling approach is required in order to facilitate effective analysis and optimisation of the system.

The hallmarks of a complex system are self-organisation, adaptation and emergence [146]. There is often confusion between the notions of a *complex* system and a *complicated* one. A system that consists of a large[1] number of components may be classified as a complicated system. The functionality of this type of system is generally characterised by the sum of its parts, which is not the case for a complex system. Another fundamental difference is that complicated systems often do not adapt to changes. Complex systems, on the contrary, are characterised by their robustness in terms of adapting to changes in the environment within which they function. A system may therefore be complicated, but not necessarily complex. There are several added concerns when attempting to design a complex system, when compared to a complicated one.

Modelling and Simulation (M&S) is a methodology that is used to understand the functioning of complex real-world systems. Most of the current military systems are very software-intensive and network-enabled, containing complex integrated subsystems. The complexities that arise in this context are often the result of synergies between the components in the system. M&S entails developing a so-called simulation model of which the inputs may be altered in order to investigate how the changes are likely to affect the outcomes of the underlying real-world system; thereby accounting for the majority of the complex interactions in the system. A simulation model may be considered as a set of rules (*e.g.* flow charts, state machines or computer code) that define how a system changes over time from its present state as a result of a variety of internal and external influences. *Simulation* may therefore be seen as the process of *modelling* a system as it progresses through state changes (continuously or in a discrete fashion) over time.

The environment in which military systems operate are generally more demanding and dynamic than those of their civilian counterparts. The simulation models used in a military environment may therefore be seen as different from those used for commercial purposes [104]. Some of these differences include:

- the fact that military simulation models are mostly highly classified and exist as black-box systems;
- a realisation that the modelling of WS capabilities and the doctrine regarding their utilisation are not typically used in other M&S environments;
- acknowledgement that algorithms employed in a military context are generally closely regulated in order to prevent reverse engineering by potential opposing forces, and
- the fact that some of the equations and system architectures typically used in a military environment are usually not included in commercial M&S or available in the open literature.

[1] The exact meaning of "large" depends on the application.

Consequently, the military community relies heavily on home-grown computational models to gain insight into the performance of a system during the early life-cycle stages, thereby reducing the risks associated with the final system [136]. Such models are usually applied whenever prototyping with the real system is either unjustifiably expensive or is impossible because a system is in an early life-cycle stage, and thus still largely non-existent. M&S may be seen as a means to optimise systems prior to implementation and thereby reduce the risks associated with the final system.

M&S has been applied extensively and successfully to a wide range of military problems, such as wargaming, logistics, communication and acquisition projects [17]. In general, it is assumed that for complex problems in which the time-continuum is of central importance, such as for TEWA systems, simulation modelling is the superior modelling alternative. For these reasons, a simulation model is also employed in this thesis to evaluate the performance of a TEWA system.

It is clear that M&S forms an important aspect of military-system analysis because of the benefits it provides above other approaches. M&S is also expected to continue playing an ever-increasing role in military-related research and development as defence budgets decrease and the complexity — together with the associated risks — of military systems increase.

Ignizio [90] clarified the importance of understanding a system before using computational methods (simulations) to evaluate the system, by stating the following:

> *"As we increasing learn about a system, its complexity decreases and our understanding increases. As the complexity decreases, the precision afforded by computational methods becomes more useful in modelling the system"* [90].

The complex nature of military systems, together with the general unavailability of detailed underlying information, poses a significant challenge from an academic modelling standpoint. As a result, it is unavoidable that system designers will spend a great deal of time or effort in redesigning something that already exists or, similarly, solving problems that have already been solved.

4.1.1 Classification of Simulation Models

According to the *Defence Science Board*[2] [211], distributed simulations in a military environment may be classified into three groups, namely *live*, *virtual* and *constructive* simulations. Although there is no definite distinction between these groups, the following descriptions provide some clarity as to the overall extent of this categorisation:

Live simulations include human operators interacting with real-world systems in the context of simulated scenarios. The operational testing of systems, field exercises and wargaming (large-scale military exercises) are typical examples of live simulations.

Virtual simulation involves real operators interacting with a simulated environment. Examples of virtual simulation include aircraft and tank simulators. This is often referred to as *Operator-in-the-Loop* (OIL) simulation (*i.e.* operators interacting with the simulated environment through HMIs) [62]. The importance of virtual simulation has increased in recent years with the advent of more capable computers, especially in the areas of graphics technology and artificial intelligence [156].

[2]The *Defence Science Board* is a committee of civilian experts, established in 1956, who are responsible for advising the US Department of Defence on technical and scientific matters [51].

Constructive simulation refers to simulations where the human operators may (or may not) interact with a model of a system, and everything else is simulated. Constructive simulation is generally used for training operators as well as for analytical decision-making (*e.g* in trade-off analyses or for planning purposes in respect of the future direction of a project). Constructive simulation was, for example, used in a statistical analysis and evaluation of emergent behaviour, at battery level, for the South African *Virtual GBAD Demonstrator* (VGD) [136].

According to Johnson *et al.* [97] the above classification may be problematic for two reasons. First, there is no clear distinction between these categories, since the degree of human-involvement and equipment realism is variable. There is also no classification for the case where simulated operators (people) interact with real equipment. An additional class of *Smart simulation* was suggested by Johnson *et al.* [97], but has not been fully endorsed in the literature. The differences between the above-mentioned classes of simulations are illustrated graphically in Figure 4.1.

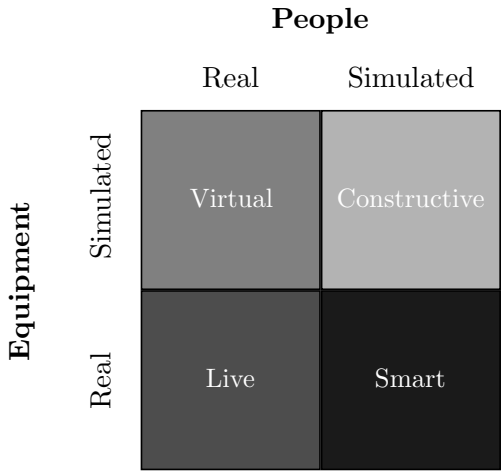


FIGURE 4.1: *Matrix of different simulation and model classes [97].*

The simulation developed in this thesis resides largely in the class of constructive simulations. For the final validation stages, however, it is still required to execute live simulation so as to sufficiently evaluate, and gain confidence, in the TEWA models clarified in later chapters. Model-qualification cannot be achieved solely within a virtual-environment, because of the critical role of the operator and the effects of real-world conditions on the outputs of the TEWA system. These effects cannot be modelled fully within a virtual environment.

The types of simulation models described above are based on the extent of human-interaction as well as the role of real-world systems in the simulation. The classification does, however, not provide clarification on how the simulation model changes states through time. TEWA systems are characteristically dynamic systems and the evaluation of this kind of system in a simulation environment is a typical operations research problem. Two popular approaches to addressing this kind of problem are *Discrete-event Simulation* and *Systems Dynamics* [196].

Discrete-event simulation models a system as a set of individual components (entities) moving through a series of processes and discrete stages. System dynamics models[3], on the other hand, is a pseudo-continuous alternative in which various stocks and flows are adjusted over time and

[3] *System dynamics* emerged from the discipline of servomechanisms engineering, and not general systems theory or cybernetics as the name suggests [163]. The results generated by incorporating this methodology may, however, still be useful in gaining an understanding of the system's performance and identifying limitations.

the outcome of the system observed. Both these approaches are used to support learning and decision-making [164]. In the case of a TEWA simulation, which may also be used for training purposes, it is important to consider the perspectives of end-users of the simulation model. Tako and Robinson [196] conducted an empirical study which compared the two aforementioned simulation models. Their summarised results are provided in Table 4.1.

TABLE 4.1: *Comparison between discrete-event simulation and system dynamics [196].*

Criteria	Discrete Event Simulation	Systems Dynamics
Means of understanding		
Transparency	The operator does not understand the underlying mechanics.	Models (flows and links between them) are transparent.
Animation	Animation and graphics aid understanding.	No animation — static display aids understanding.
Complexity		
Level of detail	Emphasis on detail complexity.	Emphasis on dynamic complexity.
Feedback	Not explicit.	Clear to end user.
Validity		
Credibility	Both modelling paradigms are equally valid and may provide realistic results and aid communication to the operator.	
Usefulness		
Learning aid	Less frequent used as learning aids.	More frequent used as learning aids.
Strategic thinking	More commonly used for solving tactical and/or operational issues.	Aids in strategic thinking.
Communication tool	Both modelling paradigms are seen as good communication tools for facilitating communication with the end users.	
Results analysis		
Nature	Provides statistically valid estimates of system's performance. Results aid in improving system.	Provides full-picture of systems. Results aid conceptual understanding.
Interpretation	More difficult. Requires statistical knowledge.	Easily interpreted, little to no statistical analysis required.
Observation	Variations of results is explicit.	Mainly deterministic results, which convey casual relationships between variables.

Although some researchers argue that these two simulation approaches are fundamentally different, researchers have more recently (2004+) been inclined to consider them as complementary to one another [132]. There have been circumstances where discrete events have been modelled in a systems dynamics fashion. Coyle [132] noted two key differences between the two approaches: Discreet-event models generally include stochastic elements in the form of random noise, whereas in system dynamics, stochastic elements are normally included in the form of appropriate delays. Secondly, in discrete-event simulation, the model may be viewed as on open-loop structure, in contrast to system dynamics where feedback is explicitly defined in a closed-loop structure. In a TEWA simulation environment, open-loop structures (such as the transition from TE to WA)

are required, but the WA process also relies heavily on the use of feedback — changes in the conditions of the scenario may result in the switching of WS assignments which my affect how the system performs during the next time stage. From the comparison in Table 4.1 it is clear that there are also several similarities between the paradigms of discrete-event simulation and systems dynamics. It should therefore be possible, and may even be desirable, to have a simulation model with characteristics of both, as is the case with the TEWA simulation developed in this thesis.

4.1.2 Steps in a Typical Simulation Study

The exact steps involved in a simulation-study depend largely on the scope and details of the model being developed [108]. The steps shown by Figure 4.2, however, generally form part of most simulation studies. More details on these steps are provided in the process outlined below.

1. *Formulate problem and plan study.* Prior to commencing with the simulation development, it is required to clarify the study's objectives and scope. The objective and scope requirements are generally elicited from the project's stakeholders. If, however, these actors are not directly involved in the model building process, the requirements should be derived from the overall requirements of the project. Without these crucial details, is likely that the simulation study will not achieve its intended aim. For large-scale projects, this step also entails assigning personnel as well as setting budget and time constraints.

2. *Collect data.* The quality of the outputs from a simulation is limited by the quality of the corresponding input information. Apart from the input information, relevant data for simulation performance evaluation and simulation calibration must also be acquired, when possible. This is required for validation purposes in order to compare a simulation output with the typical output of the underlying real-world system.

3. *Build model.* This stage essentially consists of developing a conceptual model of the simulation. This conceptual model may either be of an information flow nature, indicating the various components and how they interact, or of a discrete chain-event nature, indicating the order in which events are executed and their influence on one-another. At this stage, the proposed model should be validated by inspection and advice from experts, if available, in order to ensure that the system designer's interpretation of the model requirements are in line with that of the stakeholders. Special emphasis should also be placed on ensuring that the model architecture is sound.

4. *Implement model.* This stage is where the conceptual model, developed in step 3, is implemented as a computerised simulation process. There exist various suitable programming languages, as well as purpose-specific software packages, for implementing a simulation model. Software languages are usually more readily available in comparison with purpose-built software packages. The use of modern specialised software packages may, however, greatly reduce the development time and improve the model's realism. MATLAB® was adopted as simulation environment in this thesis. This choice is motivated later in the thesis.

5. *Verify model.* This step involves testing the implemented model in order to ensure that it performs according to the requirements prescribed in the conceptual model. Verification is, in essence, an iterative process of debugging and testing in order to assess whether the implementation is performing as expected.

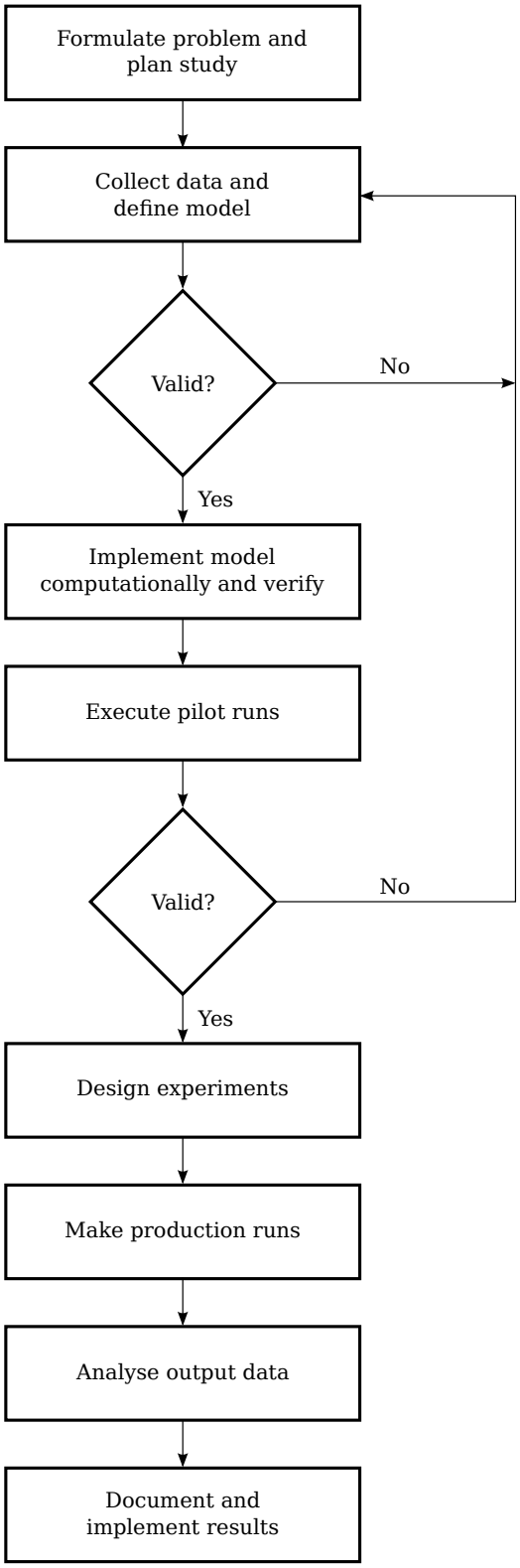


FIGURE 4.2: *Typical steps in a simulation study [108].*

6. *Validate model.* Validation is the process of determining whether an implemented model is an adequately accurate representation of the real, underlying system according to a desired level of fidelity. This can be achieved by comparing the simulated model's results with that of the real-world system, thereby allowing the developer to qualitatively determine the model's credibility in terms of producing "realistic" results. If real-world results are not available, the model can be validated with the help of domain experts.

7. *Design experiments.* Once the model is deemed a suitable representation of the real-world system, the different test scenarios must be finalised. For a Monte-carlo simulation, such as the one developed in this thesis, the required confidence limits and starting conditions should be specified prior to executing the experiments.

8. *Analyse results.* In order to determine the performance of the system under consideration, numerous production runs of the aforementioned experiments must be executed. By using the results of the experiments, system optimisation may ensue. In addition, sensitivity analysis is often performed in respect of the input parameters in order to better understand the limitations of the system and identify further optimisation opportunities.

9. *Document conclusions and results.* This step is rather self-explanatory, but the importance of this step is quite often underestimated [108]. The important information that must be provided includes a description of the assumptions made as well as the scope of the simulation, in order to clarify the limitations of the results. In addition to this information, the documentation should also promote traceability of the simulation functionalities. Traceability allows the reader to trace a specific feature of a simulation to a specified requirement. This should facilitate the system evaluators in identifying any unnecessary functions and/or missing functionality in terms of the pre-specified simulation requirements.

4.1.3 Verification, Validation and Accreditation

Verification, validation and accreditation is performed in order to determine the credibility of a model, or simulation, based on achieving its intended purpose [197]. Verification, validation and accreditation are three distinct, but interrelated processes that are followed to evaluate evidence in order to determine the extent to which the performance of the simulation reflects the real-world system. These three processes may be defined as follows:

Verification is the process of confirming that a model accurately represents the developer's conceptual description and specifications [212]. This confirmation is achieved by providing objective evidence that the *specified engineering requirements*[4] have been fulfilled [188]. Verification basically involves determining whether the results of a specific development stage satisfies the conditions imposed at the start of that stage (*e.g.* the simulation development stage).

Validation is the process whereby the degree to which a model is an accurate representation of the real world is determined, from the perspective of the end-users. Validation[5] therefore confirms, by providing objective evidence, that the model is adequate for its specific *intended use* or application [188]. An example of validation may be found in the Aegis

[4]Engineering requirements specify what a system must *do*; not what it must *be*. It is generally accepted that engineering requirements must be discriminatory, measurable, orthogonal, universal and external [208].

[5]Validation is mainly used to confirm customer requirements which, contrary to engineering requirements, specify what a system must *be*; not what it must *do*.

BMD system which underwent a pre-mission analysis during which the system received *scenario certification* [186, p.11]. Scenario certification entails, as the name suggests, that the system can safely execute the stated mission for the specific tested scenario. Similar tests would be required before industrialising a TEWA system.

Accreditation is the official certification of a model or simulation [178]. The accreditation of a model certifies that the model is acceptable for a specific purpose. Therefore, accreditation does not apply to a class of models or a suite of applications — a separate accreditation is required for each application. Accreditation should be undertaken by an independent third-party.

4.2 Representation of Simulation Model Entities

The details regarding the representation of each of the TEWA system components, such as sensors, DAs, WSs and threats, are detailed in this section within the context of the simulation developed in this thesis. Several assumptions were also made in order to facilitate development of the simulation. These assumptions are also clarified and, where needed, explained in this section.

For the purpose of this thesis, the TEWA DSS is modelled as an automated decision making system, instead of a semi-automated process as proposed by Roux [167]. This choice is mainly because it would be intractable to accurately model the complex system of a human operator, which forms a critical component of the semi-automated system. Secondly, the main goal in this thesis is to evaluate the synergetic effects between the TE and WA models, and not the effects of human operator decisions. Nonetheless, it is still important to account for the effects the TEWA system may have on the decision making abilities of the operator throughout system development. This is the topic of an entire chapter later in this thesis.

4.2.1 Sensor Systems

The detection of threats is performed by sensor systems which have certain limitations. The sensors are assumed to perfectly detect and track all threats. Hence, there is no notion of an undetected threat or surprise encounter in the simulation designed in this thesis.

All sensors require a certain time to establish a track position after the detection of a threat which, ultimately, limits the length of the TEWA cycle time. In addition to the detection rate (see Table 2.1), the modelling of the sensor systems should also account for the time to generate the precision track, which is displayed on the HMI. For mechanically-scanned sensors, with medium accuracy, it should be sufficient to model the sensors with an update rate of one second. The sensor systems will therefore update threat positions and kinematic data every second, which is also the TEWA cycle time. During each cycle, TE is conducted in respect of the target track and WA reconsidered in order to determine whether better allocations have become possible.

It is also assumed that the sensors are able to perfectly observe the outcome of a WS engagement. At each stage during which WSs are assigned, the outcome is perfectly observed and the new system state forms the input to the TEWA decision system during the next decision making cycle. In reality, however, there is also a limited number of sensors which need to be assigned to threats (similar to WSs), as described in §2.4.2. This is a separate optimisation problem which is beyond the scope of this thesis.

4.2.2 Defended Assets

When developing a simulation model of a TEWA system, it is important to include a realistic representation of the DAs. The representation of the DAs has a significant impact on the performance of a TEWA system since their priority values influence the perceived threat values of the aircraft which, in turn, are used in WA subsystem for the allocation of WSs.

As explained in §2.4.5, each asset has a priority value assigned to it by the defending force. This priority value quantifies the relative importance to the defending force of protecting the asset in question. Several variables may influence the priority of a specific asset, such as its repairability R, vulnerability V and vital importance I. According to Labuschange [106], the priority of each DA may be determined according to the formula,

$$P = I + R + V.$$

This formula is considered sufficient for practical purposes within the air-defense domain. It is, however, important to understand its limitations. The formula is not scientifically justifiable because no units of measurement exist for the three aspects contributing to a DA's priority. It is furthermore not clear why the values of I, R and V should be included in a purely additive manner in the formula. The formula is therefore not homogeneous and it is, as a result, not mathematically justifiable.

Du Toit [59] took a fresh look at DA priorities and proposed a method for quantifying DA priorities based on objective evaluations of asset characteristics (R, V and I) by experienced air-defense personnel. The method was, however, not successfully applied to obtain reliable priority values for real-world DAs. Du Toit [59] did, however, provide average priority values as elicited from 19 respondents, but the standard deviation of the responses, together with feedback from domain experts, resulted in some of the priority values not being considered as relevant and fears that they may even lead to unnecessary confusion or ambiguity. The study by Du Toit was, nevertheless, consulted as guidance to select the priority values (scaled to the interval $[0, 100]$) shown in Table 4.2 for use within the simulation environment designed in this thesis.

TABLE 4.2: *Example DA priority values used in simulation environment.*

DA Type	**Priority value**
Hangar	50
Radar	70
Command Centre	90
Stationed Aircraft (unprotected)	60
Stationed Aircraft (protected by shelters)	40
Fuel Tank	80

For the purpose of this thesis, all DAs are modelled as point assets, in the sense that no dimensions are used to define a DA. Therefore, the orientation TE models suggested by Heyns [78] are not implemented. Furthermore, it is assumed that any DA may be destroyed by a single hit from a hostile aircraft. There exists, as such, no notion of partial-damage to an asset in the simulation model put forward in this thesis.

4.2.3 Weapon Systems

The main aim of a WS is to fire missiles or gun-projectiles at an airborne threat in order to destroy it. For more information on the typical types of WSs in a GBAD environment and their

characteristics, the reader is referred to §2.4.4.

Besides the SSHP curve, each WS is associated with a specific weapon setup time. The weapon setup time includes reloading of ammunition, orientating the WS and the time required to compute interception points before engaging. For simplicity, each WS is assigned a setup time of four seconds in the simulation model. This setup time serves as a delay that must pass before a new WS assignment is possible for the currently assigned WS.

There are three different types of WSs available within the simulation environment — *Close-in Weapon Systems* (CIWS), *Very Short-range Air Defenses* (VSHORADs) and *Short-range Air Defenses* (SHORADs). These WSs have several range-related specifications that can severely influence the outcomes of the simulation. The WS specifications shown in Table 4.3 are assumed in the simulation model *cf.* [13, 53, 121]. The maximum effective ranges were determined from the aforementioned documents, but no information could be found in respect of the minimum possible ranges of WSs. These values were therefore determined by intuition and in order to obtain a "realistic" SSHP value function.

TABLE 4.3: *Example specifications of typical ground-based WSs [13, 53, 121].*

WS Class	Ammunition Type	Max Effective Range	Min Possible Range
CIWS	35 mm Rounds	±4 km	±0.2 km
VSHORAD	Missiles	±6 km	±0.9 km
SHORAD	BVR Missiles	±20 km	±1.2 km

Each WS is characterised by a SSHP matrix, containing SSHP values for different engagement ranges and zones within the WS's surrounding space. In the simulation environment, this SSHP volume of a WS is characterised by a continuous SSHP function similar to the declassified form of the SSHP for a surface-to-air weapon shown in Figure 4.3.

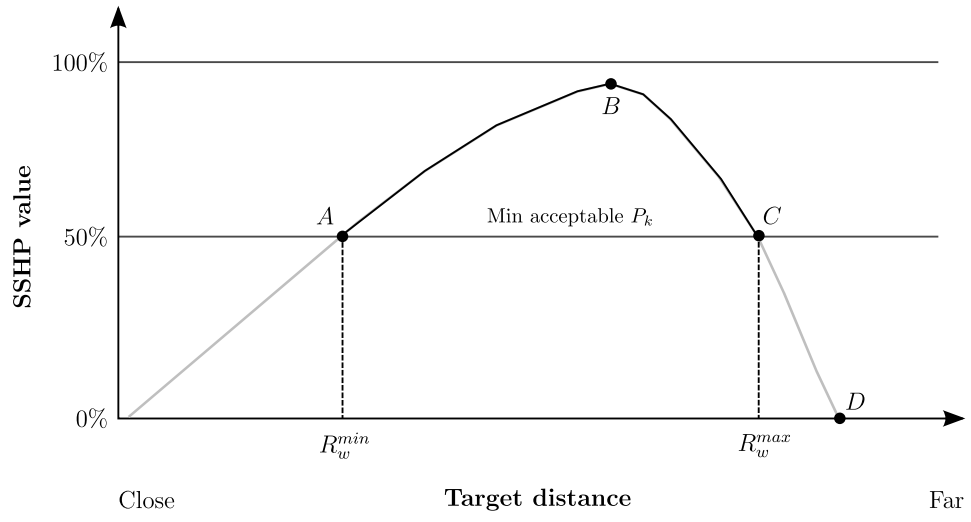


FIGURE 4.3: *Typical SSHP function for a missile-WS (adapted from [21]).*

In Figure 4.3, the vertical axis denotes the SSHP value (in this case, also the kill-probability) for the corresponding range on the horizontal axis. The threshold value for an engagement, which is set by the operators, is denoted by P_k. As such, a WS may only engage a threat if the threat is within a distance interval delimited by the earliest time for effective engagement R_w^{min} and latest time for effective engagement R_w^{max} (the dark section of the curve).

By testing different curves, it was determined that a third degree polynomial best represents the form of the curve shown in Figure 4.3. To construct a third degree polynomial SSHP function for each WS, four points need to be known on the curve. By assuming that the curve shown in [21] is according to scale, the following four coordinate relations were derived by observing the proportionality of the curve and assuming P_k has a value of 0.5 (as was the case in [21]). The four coordinates are indicated by the A, B, C and D points in Figure 4.3.

TABLE 4.4: *Coordinates for constructing the SSHP function of a missile-WS.*

Symbol	x-coordinate	y-coordinate
A	R_w^{min}	0.50
B	$\left(R_w^{max} - R_w^{min}\right)\frac{3}{5} + R_w^{min}$	0.85
C	R_w^{max}	0.50
D	$\left(R_w^{max} - R_w^{min}\right)\frac{1}{5} + R_w^{max}$	0.00

In Table 4.4, the x-coordinate refers to the range from the WS and the y-coordinate to the SSHP value of a WS for that specific range. The variables, R_w^{min} and R_w^{max}, denote the minimum and maximum effective range of WS w, respectively. The minimum and maximum effective ranges of the WSs are the values shown in Table 4.3. The coordinates shown in Table 4.4 are used for the construction of the third-degree SSHP curve of each WS. This curve, in turn, is used in both the objective function for WA as well as determining the result (hit or miss) of a missile-WS (VSHORADs and SHORADSs) engagement.

There are, however, also gun-WSs available in the simulation environment — the CIWSs. The SSHP curve of a gun-WSs differs significantly from those of the missile-WSs [121]. Şandru and Rąadulescu [177] suggested the use of the Raleigh formula for modelling gun-based WSs. They observed that the SSHP of a gun-WS decreases drastically as the range between a threat and the WS increases. Hence they suggested the expression

$$z = 1 - \exp\left[\frac{-k \cdot T_a}{(\Delta \cdot X)^2}\right], \tag{4.1}$$

for the SSHP z of a gun-WS round, where T_a denotes the projection of the threat surface area into the target plane, Δ is the total error (expressed in mRad), X denotes the range to the threat and k is a correction coefficient[6]. For a burst of ammunition, as is generally the case with gun-WSs, the SSHP (P_k) may be determined as

$$P_k = 1 - e^{S \cdot \ln(1-z)}, \tag{4.2}$$

where S denotes the number of rounds in a burst. In order to determine realistic values for these variables, the plot of the SSHP curves shown in [177, p.188] were used iteratively to determine suitable values to be used within the simulation. The resulting values are provided in Table 4.5.

Within the simulation, the success of an assignment is determined with an uniform random distribution on the interval $[0, 1]$. If the generated random number is smaller than the current SSHP value for the considered WS-threat pair, the engagement is successful; otherwise it is a miss and re-engagement would be required. A successfully hit is also modelled as a kill. This assumption, however, has its limitations.

[6]The purpose of this correction coefficient is most probably for calibration purposes. The intended purpose, however, is not explicitly stated by Şandru and Rąadulescu [177].

TABLE 4.5: *Constants used within the Raleigh function (4.1)–(4.2) for a gun-WS.*

Variable	Value
$k \cdot T_a$	110
Δ	0.05
S	30

Although in the simulation put forward in this thesis a successful hit is modelled as a kill, this in not necessarily the case in practice. A successful hit does not necessarily guarantee that the target will be incapacitated to any degree — it may not even be damaged [121]. This necessitates the requirement for a *vulnerability measure*[7] for threats. Vulnerability measures are generally provided by the manufacturers of WSs and are sometimes used in the survivability analysis of fighter aircraft [16]. Notwithstanding, because of the sensitive nature of these measures, there is not sufficient information available in the open literature to incorporate vulnerability measures in an academic simulation. More context on this matter is, however, provided below for the sake of completeness and to assist future work.

In quantitative terms, vulnerability is a conditional probability, defined as the probability of a kill given a hit anywhere on the target [121]. This notion of conditional probability is illustrated by the Venn diagram shown in Figure 4.4.

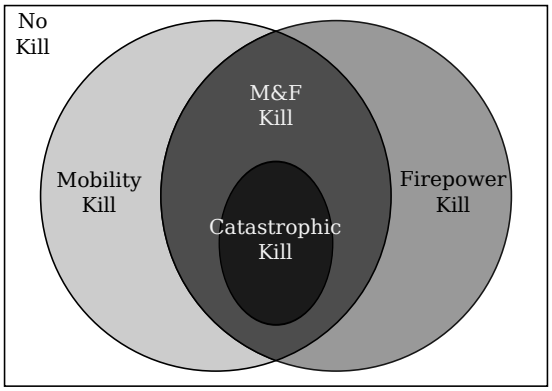


FIGURE 4.4: *The different outcomes of a successful WS hit (adapted from [54]).*

In Figure 4.4, the area of the *No-kill* zone may represent the possibility of a miss or a hit with no apparent damage. For the other areas, similarly, the proportional size of the area represents the possibility of the indicated event occurring. *Mobility-kill* refers to a hit where the threat is unable to return to base, but may still launch its ordnance at DAs. A *Firepower-kill*, on the other hand, refers to the case where the threat's WSs and/or sensor systems are destroyed, thereby preventing it from engaging DAs, but it may still be able to return to base for repairs. The *Mobility- and Firepower-kill* (M&F) case is the union of the aforementioned two cases. The last case, *Catastrophic kill*, is where the threat is disintegrated upon impact. This latter case is how each WS hit is modelled in this thesis.

In order to obtain more realistic results, it would be meaningful to design an accurate vulnerability measure for different threat-WS combinations and incorporate it in the simulation model at a future point in time. This measure may then specifically be incorporated into the objective function for WA or the threat values.

[7] Vulnerability refers to the ease with which a target can be damaged or destroyed.

4.2.4 Threats

The realistic modelling of threats is an important concern in order to obtain meaningful results from the simulation model presented in this thesis. As may be gathered from the previous chapters, implementing a realistic representation of threats is highly complicated, because of all the factors that must be accounted for. Some main considerations were identified and implemented at a sufficient level of detail. The main considerations when modelling the threats at a sufficient level of fidelity, include:

- incorporating aircraft attribute data, such as weapon envelopes and aircraft types in the decision making process,
- including precision track generation, while still accounting for measurement inaccuracies and pilot deviation from standard attack manoeuvre, and
- forecasting the future positions of threats in order to schedule the engagement of WSs.

The incorporation of these considerations in the simulation environment are detailed in the remainder of this section.

Threat Formative Element Combination

The information contained in the so-called *Formative Element Combination* (FEC) associated with an aerial threat, such as aircraft type, origin, most-probable *Weapon Deployment Profile* (WDP) and weapon envelope are contained in a structure variable in the simulation environment. Only the aircraft type, weapon envelope and IFF response results are included in the FEC implemented in the simulation model. The information contained in the FEC, excluding the IFF response, is solely used for operator situation assessment by presenting the FEC information to the operator on the HMI. A negative IFF response or the absence of an IFF response is a prerequisite for engagement as stated by the ROEs [42]. In future work, however, the other information contained in the FEC may perhaps also be used for the implementation of the TE and/or WA processes directly.

In 2013, Van Staden [216] developed a framework for the classification of FECs and, more specifically, the identification of the aircraft WDP. A *Hidden Markov Model*[8] was suggested to be used for the purpose of identifying the unknown WDP by observing current and previous data-points of the threat's track. This method is not implemented in this thesis, because of the unavailability of the detailed information required for these stochastic TE models.

Track Generation

Describing the motion of a threat necessitates a mathematical description of the path that it follows in order to determine its position at a specific time. The required class of functions employed for this purpose has to be capable of realistic representations of a wide variety of threat tracks without requiring excessive computational power [60]. In addition, the resulting tracks also have to be continuously differentiable so that quantities such as velocity have meaning. Not all curves are suitable for modelling the track of an aircraft. The representation of threat tracks

[8] A hidden Markov model is a statistical pattern recognition technique in which the system being modelled is assumed to be a Markov model with unobserved states [216].

is a critical driver of system performance and it is therefore of utmost importance to model this component at a sufficient level of fidelity [65].

In 2009 Du Toit [60] identified B-Splines[9] as an effective curve scheme aimed at modelling rigid body kinematics and for interactive curve design. A cubic spline[10] is used to connect each of the five segments of the six input coordinates. Du Toit also detailed the difficulties associated with using a dynamic model[11] to describe the movement of an entity. Judd *et al.* [99] also successfully implemented B-Splines for the path planning of unmanned aerial vehicles. The suggestions provided in [60, 99] are used collectively in this thesis for the generation of threat trajectories from a set of input coordinates. The successful implementation of the track generation mechanism was a major milestone during this thesis.

Another consideration for the realistic modelling of aircraft tracks is the variation in pilot behaviour (when repeatedly performing an attack manoeuvre) as well as the inaccuracies resulting from sensor system measurements. These effects are accounted for in the track generation mechanism employed in this thesis by superimposing random spheres on the input coordinates, as suggested by Du Toit [60]. In the simulation results reported in this thesis, a distance of 10 m is used as the diameter of the random-sphere. For each simulation run, the coordinates of waypoints used for the generation of the track, may lie anywhere within this 5 m radius sphere (these coordintates are generated according to a normal distribution). An exaggeration of this concept is illustrated by Figure 4.5. Four replications of the track generation of three aerial threats are shown in the figure.

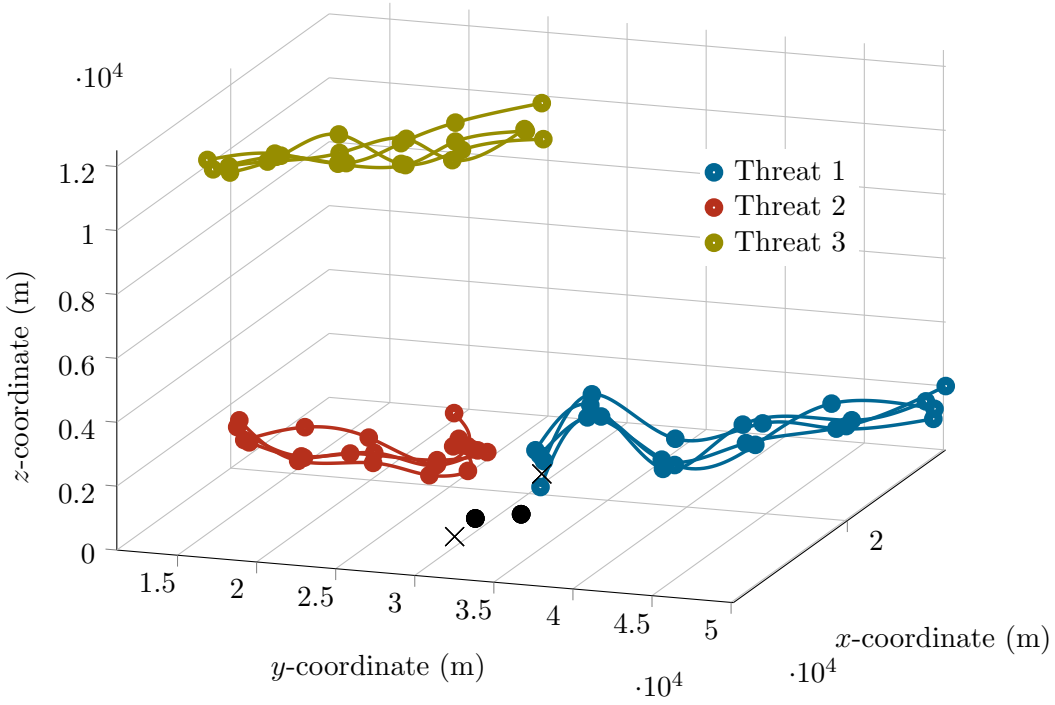


FIGURE 4.5: *Exaggeration of random noise superimposed on the waypoint coordinates used for aerial threat track generation purposes.*

[9] *B-splines* are a subset of the class Bézier curves which are parametric polynomial curves specified by additional control points which determine the shape of the curve [60, ch.4].

[10] A cubic spline refers to the case where the piece-wise polynomials are composed of fourth degree segments joined by specified knot conditions.

[11] Du Toit [60] referred to a dynamic model which specifies the trajectory of an entity's centre of mass and its orientation as functions of time.

Flight Path Prediction Module

The final component required to model the threats is a *flight path prediction module*. The inclusion of a flight path prediction component allows the TEWA system to suggest a chronological engagement order list, as is required by the dynamic WA models. The flight path prediction model is mainly used to forecast the changes in SSHP values as threats traverse the 3D space surrounding the DAs, thereby allowing for more informed decision-making during the formulation of possible countering strategies.

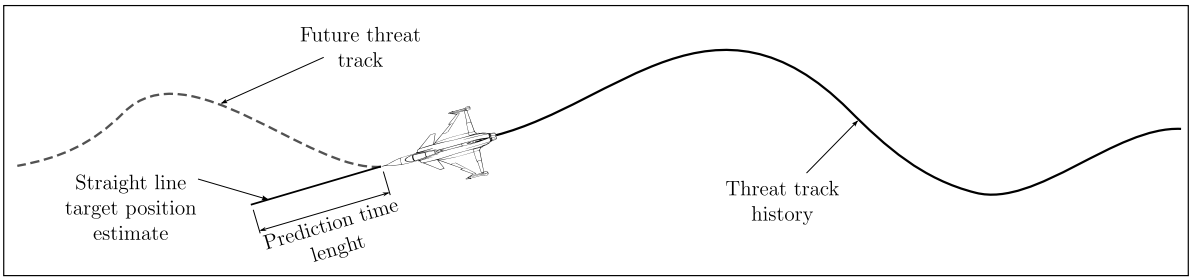


FIGURE 4.6: *Flight path prediction method employed.*

The method employed for the flight path prediction in this thesis is *current vector extrapolation*, which assumes that an aircraft will continue flying in a straight line at its current velocity. This concept is illustrated in Figure 4.6. The speed of a threat is, as such, assumed to be constant between two successive waypoints at which position and velocity vectors are estimated.

More specifically, the current heading vector of the threat is calculated by using its current position and its previous position. This heading vector $\vec{P}$, is then normalised so as to obtain the heading unit-vector

$$\vec{u} = \frac{\vec{P}}{\|\vec{P}\|}.$$

The velocity vector is calculated as,

$$\vec{V} = V \cdot \vec{u},$$

where V is the speed of the aircraft as measured by the sensor systems. By assuming constant velocity, the future threat position after t time units is estimated as,

$$\xi_f = \xi_c + \vec{V} \cdot t,$$

where ξ_f denotes the future position of the aircraft in Cartesian coordinates, and ξ_c denotes the current position as measured by the sensor systems. $\vec{V_t}$ represents the aircraft velocity, which is assumed constant over the prediction timeframe of length t. The predicted position of a threat is used to calculate new SSHP values for the threat at the future positions. These newly calculated SSHP values are used within the objective function in order to schedule the engagement of the threat by WSs.

The length of the prediction timeframe is pre-specified by the operators, since there is a certain trade-off analysis required to determine an optimal prediction time length. To illustrate — the longer the time length, the less accurate the prediction may represent the real situation and the longer computations will take, but more coordination will, however, be possible between WSs. Allowing a longer prediction time-length may also result in the WA algorithms prolonging the assignment of WS in order to benefit from increased SSHP values when the threats are closer to the WSs. This approach may result in unnecessary risks and warrants the inclusion of a weighted utility function which prioritises earlier engagements, as explained in a later chapter.

4.3 Simulation Software Environment

Before commencing with simulation development, it was required to select a suitable software environment for the simulation. Some of the software environments considered for this purpose include, Java [145], C++ [32], R [201], MATLAB [127] and the STAGE [189] simulation environment.

Initially, the STAGE simulation environment seemed very promising. STAGE is a simulation package that offers an extendable, standards-based environment which allows the developer to create more sophisticated scenarios in a shorter amount of time. The problem, however, was the high cost of the simulation software. After correspondence with Behrens [20] (CEO of Cybicom Africa Technologies), the developers of STAGE provided discount for a student license, but the cost (R88 122) was still excessive. According to Behrens the Cybicom Atlas Defence (the local reseller for Presagis products) helped to develop the classroom trainer for South Africa's GBAD system. Cybicom plan on using STAGE for the next stage in the GBAD program.

Because of the high cost of the STAGE simulation package, the decision was made to select MATLAB, which is available at Stellenbosch University. MATLAB, short for matrix laboratory, is a high-level language and interactive environment widely used in industry and at academic institutions. MATLAB was deemed a suitable environment for several reasons. Since many of the variables that are required to model a TEWA system are easily represented in matrix form, the ease with which MATLAB can manipulate matrices, is a major advantage. The additional toolboxes available within the MATLAB environment, such as the optimisation and spline toolboxes, also proved useful during software development; something that the alternatives lack. It was concluded that MATLAB provides a considerable advantage in terms of rapid prototyping and understandability of implemented models, especially in comparison with the lower-level alternatives of Java and C++.

In addition, MATLAB is also the industry standard for defense-related simulations in South Africa [44] — Denel Dynamics is the largest user of Matlab in South Africa and Reutech is also an avid user. The use of MATLAB is furthermore ideal for shortening the development-to-implementation cycle [52]. Reutech engineers developed a methodology to start system-level verification earlier during the development process and thereby speed-up the development cycle. Basically, the methodology relies on generating *Hardware Description Language* (HDL) code automatically, instead of by hand. The components of a signal processor was programmed in MATLAB after which the code was converted to HDL. The HDL code is finally transferred to a FPGA (Fully-Programmable Gate Array) which, in turn, is integrated into the appropriate hardware. Hence, opting for MATLAB should ease the transition from an academic environment to industry, if the stakeholders of the project require to pursue this route in the future.

4.4 Simulation Model Architecture

This section provides the reader with a high-level overview of the information flow of the TEWA simulation model developed in MATLAB. After reading this section, the reader should be better equipped to understand the simulation source-code in Appendix C as well as the detailed explanations of the TE and WA subsystems which follow in the next chapters. The source-code is included in this thesis with the main goal of assisting future work. As such, the source-code may be ignored by the reader with no break in continuity or lack of understanding.

The simulation developed in this thesis may be classified mainly as a discrete-event Monte-Carlo simulation. The discrete-event simulation is selected, not necessarily because the system cannot

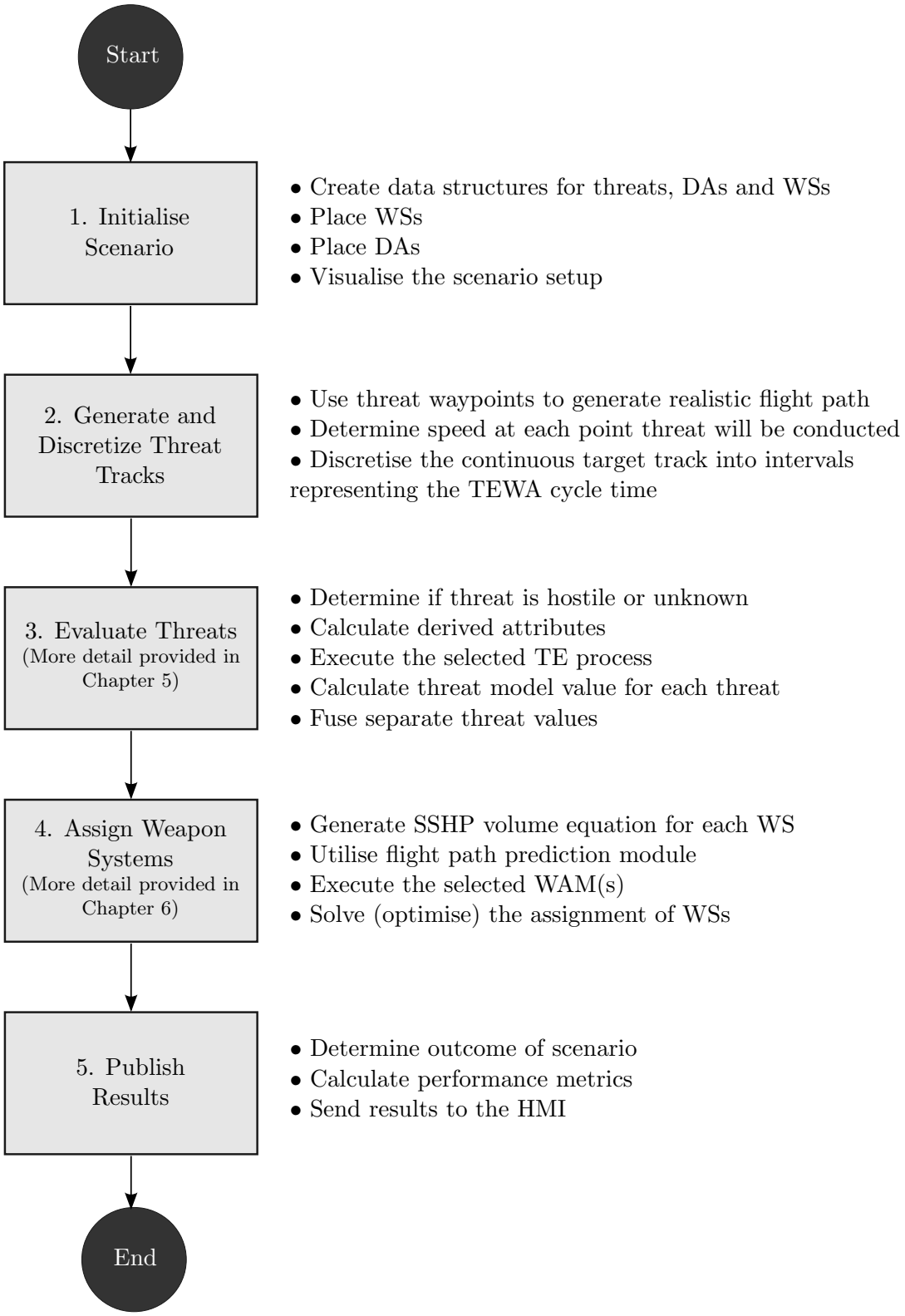


FIGURE 4.7: *Overall simulation environment logic flow.*

be modelled as a complex continuous process, but rather by virtue of the fact that the technologies employed (update rate of sensors) limit the observation of the *Area of Responsibility* (AOR) to a number of time-indexed, discrete events.

The architecture of the main programme has an open-loop structure (*i.e.* feedback is not apparent) as may be seen in Figure 4.7. It is important to note, however, that this is only possible within a simulation environment, mainly because the threats are not reactive in the sense that they will react (execute evasive manoeuvres) if a WS fires upon them. The threat tracks are therefore fixed from the beginning of the simulation. Consequently, the threat values are calculated for an entire scenario and the resulting threat value matrix is fixed for the whole scenario timeframe, thereby eliminating the need for feedback which would overcomplicate the software development.

A high-level layout of the logic flow in the simulation is depicted in Figure 4.7. More details regarding the implementation of these sub-routines are provided below, using the same numbering as in the figure:

1. *Initialise Scenario.* The initial stages of the programme includes creating data structures for all the threats which contain their attributes and kinematic data. Each threat track is defined by a set of input coordinates[12] as provided by the user; the velocity at each of these coordinates and, lastly, the most probable WRL, which is defined by a FTTF and LTTF. If the WDP is known, this is also included, but only for evaluation and traceability purposes — it has no logic role in the programme.

   The parameters of the WSs and DAs are also initialised during this stage in order to construct a simulation scenario. The location of each WS and DA is defined by x and y coordinates. The z coordinate is assumed to be zero. The DAs also have priority values assigned to them, as defined in Table 4.2. The WSs, on the other hand, have a maximum range, minimum range, setup time, ammunition count and cost associated with them. These initial data structures contain the sole inputs for the rest of the simulation.

   A scenario generation algorithm was suggested by Van der Merwe [215], in which a deterministic approach is followed to place DAs and a heuristic is used to place WSs in order to sufficiently protect the DAs. In reality, however, the placement of DAs and WSs are typically characterised by various constraints (*e.g.* terrain conditions, building and safety regulations and even commander intent), though it is rarely possible to quantify these exactly. The placement of airfields have different requirements than, for instance, a command centre, and the evaluation of scenarios that are not realistic will not result in meaningful results. This simulation therefore requires manual input to define the positions of WSs and DAs, thereby allowing for the testing of certain specific scenario set-ups only.

   Furthermore, as explained in §4.2.4, stochastic noise is added to the input waypoint coordinates of aerial threats in order to simulate pilot behaviour and sensor measurement inaccuracies. This random noise is added to the set of input coordinates before the tracks are generated using B-splines. These input values are finally visualised using MATLAB's visualisation toolbox.

2. *Generate and Convert Track to Discrete Time.* By utilising the input coordinates, B-splines are generated to serve as tracks for the threats. The default functions used for spline generation, as provided by *Matlab*, proved inadequate since the input coordinates of a specific threat are not necessarily unique or monotonic. An alternative toolbox was

[12] Each input coordinate is specified by a x, y and z coordinate, but may be interpreted, respectively, as latitude, longitude and altitude, without loss in meaning.

therefore implemented as may be seen in Appendix C. These splines are subsequently discretized according to the number of TEWA cycles as specified by the user — using the piece-wise polynomials of the splines in order to determine function values (threat positions) for specific points in time.

3. *Evaluate Threats.* All results of the previous two stages are directly sent to the TE sub-routine for further evaluation. Feedback to the track generation module is therefore not required in the simulation environment, as will typically be the case in practice. This stage is further clarified in the following chapter.

4. *Assign Weapon Systems.* This stage is essentially where the reactive WA process is executed. More detail on this stage is provided in Chapter 6.

5. *Publish Results.* The final stage of the simulation involves saving the results by printing them to files and calculating the relevant performance metrics. The saved information may then be used for further analysis in order to evaluate the performance of the various algorithms and models. Finally, the information for the specific scenario is sent to the HMI environment to be used for visualisation and for validation purposes in conjunction with domain experts.

4.5 Validation and Verification Strategy

An important problem facing any real-world simulator is that of trying to determine whether the model is an accurate representation of the system being studied [108]. As described in §4.1.3, verification and validation are essentially two independent processes that are uses collectively to assess whether a product (simulation) meets the initial customer requirements and engineering specifications and, in turn, fulfils its intended purpose. Both these processes were diligently executed throughout the development of the simulation in an iterative, evolutionary approach as described in the following sub-sections. The difference between these two processes, with reference to a simulation context, is illustrated in Figure 4.8.

4.5.1 Verification

Verification is an iterative process that must be executed throughout simulation development. For the simulation developed in this thesis, a bottom-up approach was followed where each simulation sub-routine (*e.g.* WA and TE) is first tested in isolation before integrating the various sub-routines; thereby effectively a evolutionary development process. By so doing, traceability is ensured because the interactions that affect the overall system functionality is kept to a minimum, thereby effectively assisting the traceability of detected simulation logic fallacies.

An effective tool that may be employed for verification purposes is the use of a so-called *trace*. A trace is effectively a print-out of the simulation's current state during a discrete point in time [108]. According to Law and Kelton [108], a trace is one of the most powerful techniques that may be used to debug a discrete-event simulation, thereby effectively verifying correct operation of the various simulation sub-routines. Traces were extensively employed in order to verify the TE, WA as well as all supporting sub-functions.

In past projects [78, 104], the TE subsystem has, to a certain degree, already been tested in isolation. As such, the results from previous studies were used in order to verify correct functioning of the TE sub-routine developed in this thesis. The WA sub-routine as implemented

in this thesis has, however, not been tested in the past. The WA sub-routine was developed in an evolutionary approach — individual constraints were added to the optimisation routine in a systematic manner in order to ensure correct functioning before increasing the WA sub-routine's complexity. Traces were employed in order to assess correct functioning of the WA sub-routine during each development stage. For example, traces were used for assessing the value of the objective function and determining whether selected WS assignments are indeed feasible.

The graphical 3D visualisation of the GBAD scenario, as shown in Figure 4.9, forms an integral part of the simulation verification process. Such a visualisation enables the system designer to intuitively inspect and identify limitations present in the logic of the developed simulation. The visualisation was, for example, used to determine whether the best positioned WS does, indeed, engage the closest threat. Furthermore, according to Sargent [178], graphical visualisation makes it easier to also detect anomalies within the simulation (*e.g.* a WS engaging a threat that is out of range or, alternatively, identifying a threat that is performing unrealistic manoeuvres).

4.5.2 Validation

Similar to verification, validation of the simulation model described in this chapter was performed throughout the model's development stages. Validation, effectively, determines that the simulation model is a sufficiently adequate representation of the real-world system so as to achieve the project objectives as given in §1.4.

The simulation model was tested for continuity by altering certain input parameters and observing the response — if the output response does not correspond to the author's intuition, further investigation was launched in order to find solutions. Generally, validation is achieved by comparing a simulation's output to that of the real-world system which the simulation is imitating. For a TEWA system, however, such information is not readily available or even non-existent. An alternative is therefore to use so-called *face validation*. Face validation entails presenting the simulation results together with the underling assumptions to a military expert who, in turn, assess the relevance and adequacy of the simulation output.

Since the developed simulation framework is a proof-of-concept model, a low level of validation fidelity is adequate. The results presented in the worked example later in this thesis may serve as additional proof of the correct functioning of the simulation and its adequacy in terms of meeting the project's objectives.

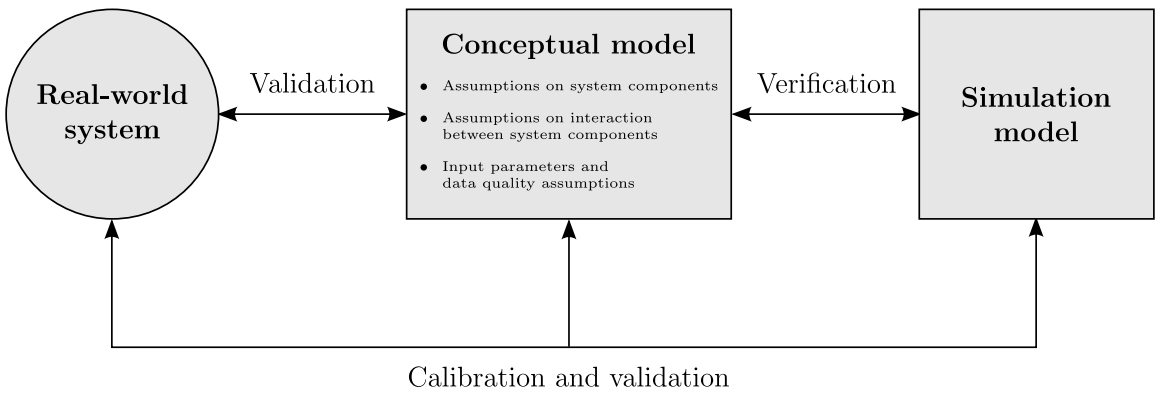


FIGURE 4.8: *The difference between validation and verification.*

4.6 Illustrative Example

A worked example is introduced in this section in order to be used throughout the remainder of the thesis for illustrating and clarifying the concepts implemented in the simulation environment. A hypothetical GBAD scenario is illustrated in Figure 4.9. In this figure the threat paths of three threats are indicated by the three lines, the FEC of the three threats are shown in Table 4.6. The positions of two DAs are depicted by black dots and their positions are listed in Table 4.7.

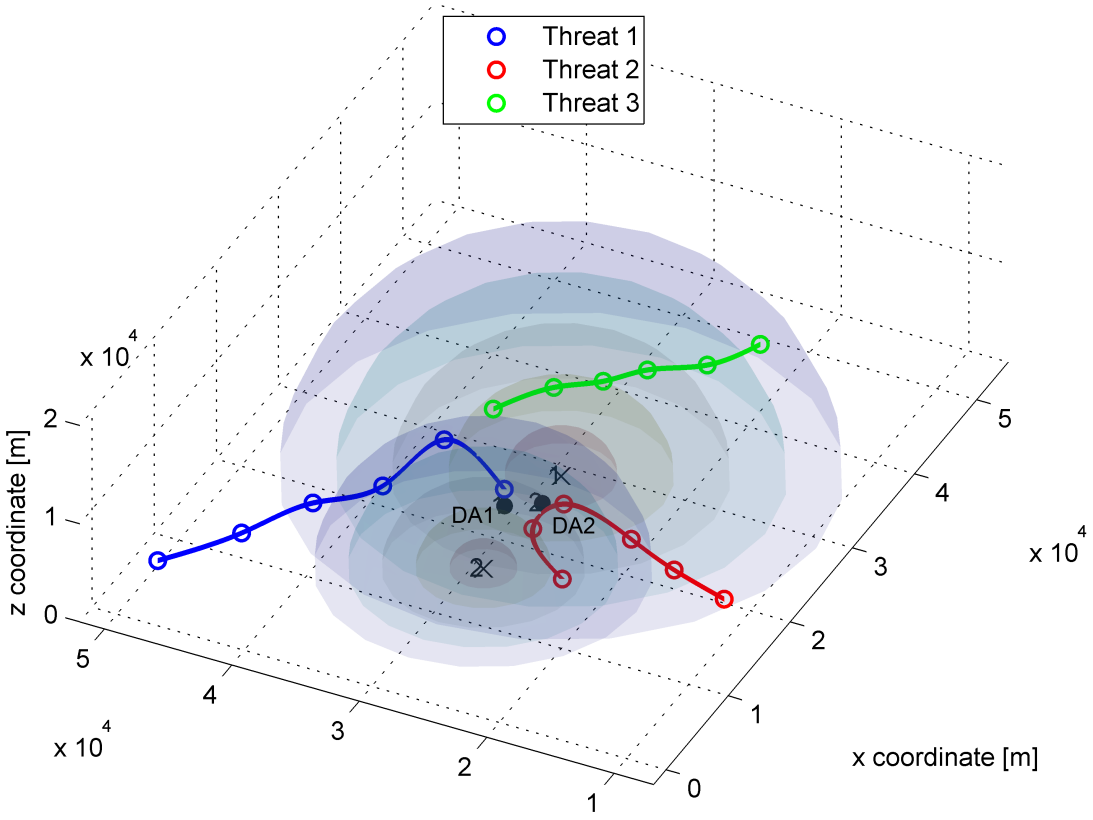


FIGURE 4.9: *Hypothetical ground-based air defense scenario.*

The differently coloured domes represent the SSHP distribution volumes of two ground-based WSs which are indicated by the two crosses; their properties are indicated in Table 4.8. The outside dome has a SSHP of 0.5, whereas the other coloured domes do not have particular SSHP values associated with them, irrespective of their colour. These domes were mainly used for verification purposes and to aid understanding. The SSHP functions, as described in §4.2.3, are used to populate the SSHP matrix.

Although the threat models are unaware of this, Threat 1 is executing a pitch-and-dive attack manoeuvre in respect of DA 2. Threat 2, on the other hand, is executing a toss-bomb attack manoeuvre in respect of DA 1. The basic principles on which these attack manoeuvres rely are illustrated in figures provided in Appendix B. Finally, Threat 3 is a passenger aircraft passing over the conflict zone and therefore poses no real threat. Threat 3 is only present to ascertain the response of the TE algorithms in the case where an aircraft is not attacking the DAs. This hypothetical GBAD scenario is used throughout the remainder of this thesis in order to demonstrate the working of the underlying TE and WA algorithms.

TABLE 4.6: *Threat-related data for the illustrative example.*

Information type	**Data**
Threat 1	
Weapon Deployment Profile	{Pitch and Dive}
x-coordinates	$\{6\,320, 10\,994, 15\,845, 20\,716, 25\,043, 24\,056\}$
y-coordinates	$\{49\,556, 45\,825, 42\,344, 38\,904, 36\,174, 31\,929\}$
z-coordinates	$\{1\,524, 1\,524, 1\,523, 1\,549, 3\,833, 1\,492\}$
Speed points	$\{180, 180, 240, 240, 300, 240\}$
WRL	$[75, 100]$
Threat 2	
Weapon Deployment Profile	{Toss Bomb}
x-coordinates	$\{19\,409, 21\,084, 22\,785, 24\,053, 21\,044, 16\,378\}$
y-coordinates	$\{11\,355, 16\,027, 20\,762, 26\,285, 27\,254, 23\,519\}$
z-coordinates	$\{1\,091, 1\,091, 1\,092, 1\,590, 1\,075, 1\,093\}$
Speed points	$\{200, 200, 240, 200, 200, 200\}$
WRL	$[50, 70]$
Threat 3	
Weapon Deployment Profile	{None}
x-coordinates	$\{36\,190, 33\,280, 29\,940, 27\,810, 25\,360, 21\,330\}$
y-coordinates	$\{17\,550, 20\,120, 22\,920, 25\,370, 28\,050, 30\,850\}$
z-coordinates	$\{11\,987, 11\,897, 11\,887, 11\,877, 11\,877, 11\,897\}$
Speed points	$\{190, 200, 240, 240, 240, 240\}$
WRL	$[\text{N/A}]$

TABLE 4.7: *DA-related data for the illustrative example.*

	DA 1	DA 2
DA type	Command Centre	Hanger
x-coordinate	23 800 m	25 250 m
y-coordinate	31 200 m	28 940 m
z-coordinate	0 m	0 m

TABLE 4.8: *WS-related data for the illustrative example.*

	WS 1	WS 2
WS type	SHORAD	VSHORAD
x-coordinate	28 800 m	16 300 m
y-coordinate	29 200 m	29 200 m
z-coordinate	0 m	0 m
Setup time	4 s	4 s
Ammunition	8	5

4.7 Chapter Summary

In §4.1 the notion of M&S was introduced and the differences between military and commercial simulations were clarified. Several considerations were also mentioned for the evaluation of military systems. The rationale for the use of simulation as a test bed for a TEWA system was motivated and several difficulties in this respect were elucidated. The section not only formed the background for the rest of the chapter, but also serves as background for the performance evaluation chapter to follow in a later part of this thesis. The different classifications of simulation models were described in §4.1.1. Simulations were first categorised according to the standard used to classify military simulations, after which two simulation modelling approaches — discrete-event simulation and systems dynamics — were compared. In §4.1.2, a framework was introduced to provide some structure to the simulation process. This section concluded with a clarification of three generally misused concepts: verification, validation and accreditation in §4.1.3.

The representation of each of the TEWA elements (sensor systems, DAs, Wss and threats) within the simulation environment was detailed in §4.2. The assumptions made in respect of sensor systems were described in §4.2.1, after which the representation (priority values) and assumptions related to the DAs were addressed (§4.2.2). In §4.2.3, the modelling of WSs was considered, with special emphasis on the construction of a SSHP function and the notion of vulnerability. The modelling of the threats was finally clarified in §4.2.4, with reference to the construction of an FEC, the generation of threat tracks and flight path prediction.

A motivation was provided in §4.3 for the simulation environment, MATLAB, selected for the purpose of model development. In §4.4, a high-level overview was provided of the logic flow in the simulation developed in this thesis. Each of the stages in the simulation environment were also introduced. The chapter closed with the introduction of a hypothetical GBAD scenario, to be used for clarifying the WA-related and TE-related concepts throughout the remainder of this thesis in §4.6.

CHAPTER 5

Threat Evaluation Implementation

Although our intellect always longs for clarity and certainty, our nature often finds uncertainty fascinating.

— Carl von Clausewitz

Contents

TE is essentially the ongoing process of identifying, accessing and prioritising aerial entities which pose a threat to the defended system. This chapter opens with an overview of TE so as to better clarify the purpose and aim of the TE subsystem. The core processes and models of the TE subsystem are also introduced. After a basic understanding of TE has been acquired, the working of its constituent TE models are explained in some detail. A newly developed TE model fusion output process is subsequently explained and demonstrated. The chapter closes with a logic flow diagram of the TE simulation architecture adopted for system evaluation purposes in this thesis.

5.1 Threat Evaluation Overview

The TE process is a crucial C2 function during any GBAD scenario [169]. TE consists of determining the level of threat posed to the defended system by aerial threats and, consequently, the priorities associated with these threats in terms of their engagement by WSs within the *Area*

of Responsibility (AOR) [91]. This task is a highly complex task that requires making useful interferences under tight time and serious uncertainty constraints.

Prior to executing the process of TE, input information is required from ground radars and associated sensors. These sensors are responsible for detecting, tracking and identifying potential threatening aerial vehicles [78]. The TE subsystem utilizes the kinematic, tracking and attribute data collected by the sensors, together with pre-deployment information, in order to estimate the level of threat posed by each aerial vehicle. The output of this process is ultimately a system threat value for each threat.

Different measured attributes are taken into consideration when determining these threat values. These attributes may be subdivided into three classes [91, 149, 190]:

Opportunity parameters quantify the extent to which certain preconditions in the environment are met in order for the threat to successfully launch an attack. Truter and Van Vuuren [206] refer to this class of parameters as "proximity parameters" in order to contextualise this parameter class in a TEWA environment, since opportunity is generally measured in terms of the proximity of a threat in relation to a DA. Opportunity is nonetheless used here for the sake of generality. A threat that is far away from a DA will not be classified as an imminent threat to that DA, when compared to threats that are closer to the DA. A widely used example of such a parameter, is the range to the *closest point of approach* of a threat with respect to a DA [168] or distance to weapon release.

Capability parameters attempt to quantify a threat's ability to cause damage to a DA. In order to calculate this value, it is required to know specific characteristics of the attacking aircraft. Examples of capability parameters include the threat type, its weapon envelope and its fuel capacity.

Intent parameters aim to quantify the will and determination of a threat to cause damage to a DA. Of the three parameter classes, intent is the most difficult type of parameter to estimate, but certain measured threat attributes may nevertheless be used to estimate a threat's intent [169]. One method in which intent is estimated is through recognition of known attack manoeuvres from an aircraft's measured track.

Examples of the above-mentioned classes of parameters are implemented in a number of different algorithms of varying complexity and sophistication that typically run concurrently within the TE subsystem, each assigning threat values to each threat with respect to each DA separately. This results in several threat values for each threat. In the case where certain necessary sensor data are not available, the TE system will select scaled-down TE models which are able to estimate threat values in the absence of very detailed threat data [120]. The different threat values for each threat-DA pair are then fused together to obtain a single prioritised list of system threat values (*i.e.* a threat value for each threat, typically found on a consensus basis, taking into account the results contributed by all the TE models) [120]. These threat values are used by the WA subsystem in a bid to optimise the utilisation of available resources (WSs and ammunition) when weapon assignment decisions are made for engaging the aerial threats.

Roux and Van Vuuren [168] proposed three levels of TE models of varied complexity. This multi-level model infrastructure is illustrated graphically in Figure 5.1. In order of increasing complexity, they are flagging models, *Deterministic Models* (DMs) and stochastic models. Flagging models are binary in nature and are activated when certain threshold violations occur (*e.g.* when sudden increases in altitude or the dropping of paratroopers are observed). Stochastic

TE Model	Type	TE Approach
Flagging of change in aircraft behaviour	Qualitative (Binary)	Continuously, based on threshold violations
Aggregation of results from aircraft deterministic behaviour	Quantitative (Value-based)	Default approach, in the absence of sufficient pre-deployment intelligence
Factoring in of results from probability-based models		Phased in as aircraft kinematic behaviour and attributes are recognised

FIGURE 5.1: *The multi-level modelling approach to TE where darker shades indicate increased levels of sophistication and increased levels of information required (adapted from [168]).*

models are probability-based and require detailed information on enemy arsenals, threat types and doctrine. The focus in this thesis is only on flagging models and DMs.

DMs utilise the measured kinematic data from sensors and calculate derived attributes which are collectively used to estimate an aircraft's threat value. The estimation criteria used by DMs may include the time to weapon release, or any course, heading or distance-related measure. For the implementation of these models, basic pre-deployment information, such as DA positions, importance values of the different DAs and, in some cases, their orientations as well as the maximum turn radii of attacking aircraft, are required [169]. As a result, the exact input information required depends on the specific DM implemented.

5.2 Implemented Threat Evaluation Models

Various TE models are described in some detail in this section. The focus here is on the DMs. The background required for the development of these models derived from the work of Heyns [78]. The models have, however, never been implemented in a 3-D simulation environment and therefore require adaptation. In addition, the value-based TE-model output fusion approach, as applied in this thesis, has not been implemented successfully in the South African context.

There are three general requirements that a DM should adhere to in order to be successfully implementable in practice [207]. First, the DM should be *robust* in the sense that it should not be too sensitive to measurement errors by the sensors. The results of the models should therefore not change drastically if small measurement errors occur. Secondly, a DM has to be *generic*. This is required in order to ensure that the DM does not discriminate between specific types of threats in a certain environment — the DM should be able to estimate intent for all types of scenarios and environments objectively. Lastly, the calculations carried out in a DM should be *computationally simple*. Since time is an extremely valuable commodity during an aerial attack, it is of utmost importance to ensure that the algorithms execute quickly. There is generally a trade-off between algorithm complexity and execution time. It should be noted that the DMs implemented during the course of work towards this thesis function in conjunction with other

lower-level algorithms employed for the purposes of interception point calculations, radar image processing and aircraft property extraction [24]. Rudimentary but sufficiently accurate models are expected to be highly beneficial in order to ensure successful system functioning.

5.2.1 Slant Distance TE Model

Perhaps the most basic manner of measuring opportunity involves use of the Euclidean distance between a threat and DA. Such an approach is adopted in the *slant distance TE model*.

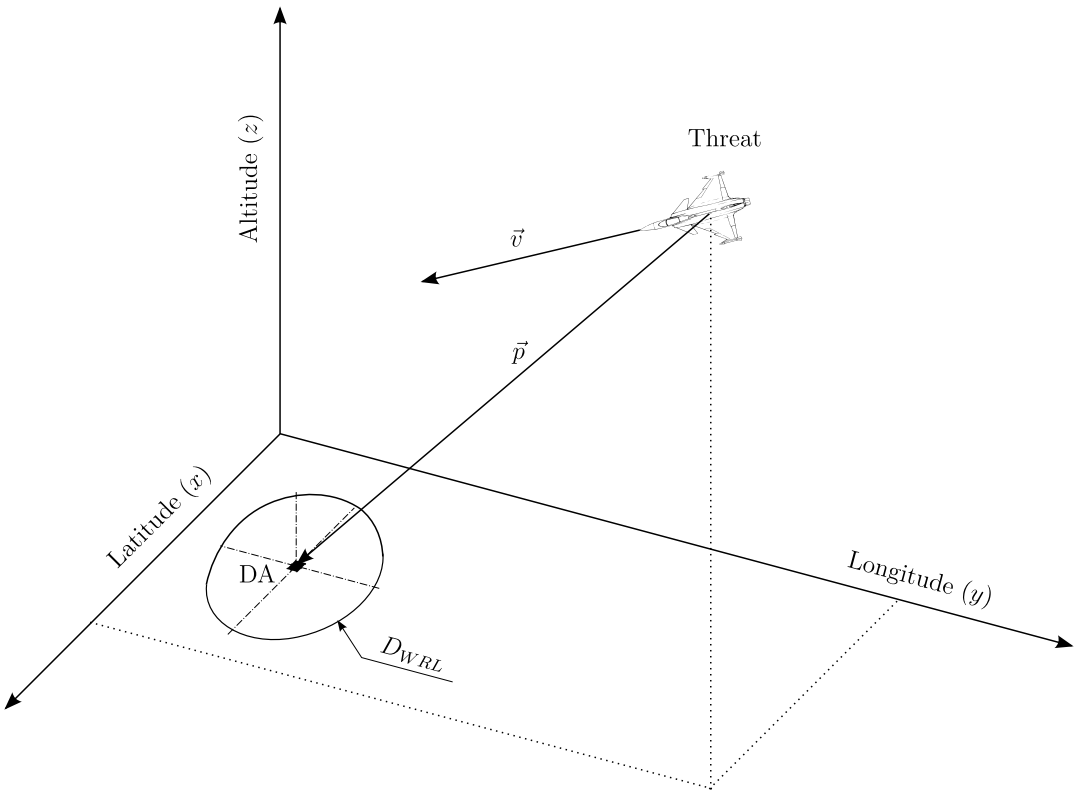


FIGURE 5.2: *Working of the slant distance-related TEM.*

Slant distance becomes more important as the threat closes in on a DA. As soon as a threat's distance from a DA is less than the stand-off range of the threat's most probable weapon system, the distance model becomes very important since the aircraft's weapon can be launched at any moment. It was therefore decided that two derived attributes should be utilised in the calculations of the slant distance model: (i) the Euclidean distance between a threat-DA pair and (ii) the most-probable stand-off range of the threat's ordnance. The parameters utilised for the calculation of this DM are depicted in Figure 5.2.

The position $\mathcal{T} = (x_1, y_1, z_1)$ of the threat and the position $\mathcal{D} = (x_2, y_2, z_2)$ of the DA are described in terms of Cartesian coordinates. The distance between the threat and the DA is given by the Pythagorean formula

$$\|\vec{p}\| = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}. \tag{5.1}$$

This distance is then used together with the most-probable threat ordnance stand-off distance

D_W and the area of responsibility radius A_R in order to calculate the distance threat value

$$V_d = \begin{cases} 1 & \text{if } \|\vec{p}\| \leq D_W \\ \frac{\|\vec{p}\| - A_R}{D_W - A_R} & \text{if } D_W < \|\vec{p}\| \leq A_R \\ 0 & \text{if } \|\vec{p}\| > A_R. \end{cases} \tag{5.2}$$

When implementing this model in conjunction with other models, there are some considerations to be aware off. When a threat is far away from a DA, the speed at which it approaches should be a better variable by which to distinguish between the level of threat it poses to the DA. Time-to-weapon release-related parameters should therefore be better predictors of opportunity at large ranges.

5.2.2 Course-related TE Model

The course of an aircraft with respect to a DA may be used as a suitable measure to quantify the level of intent as well as opportunity that the threat poses to a specific DA at a specific time-stage. As mentioned in §2.3, an aircraft's course is increasingly directed towards a DA during the later stages of most weapon deployment profiles as the aircraft readies for weapon-release. It is therefore reasonable to assume that, when the course of a threat with respect to a DA changes, the aircraft's threat value should change accordingly.

The course-related model developed by Heyns [78, p.33] is a 2-D model. In this thesis, the TEWA system is, however, tested in a 3-D environment. This necessitates the need for a formula for determining the 3-D course angle.

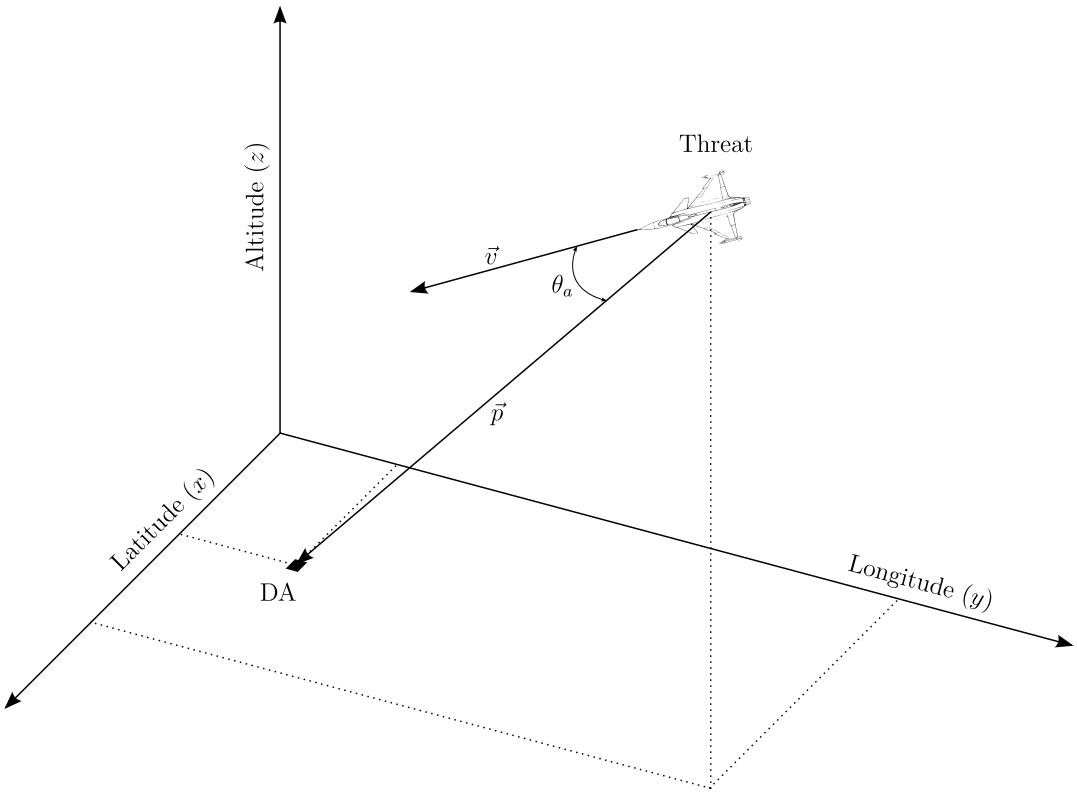


FIGURE 5.3: *Working of the course-related TEM.*

More specifically, the attack-angle of an aircraft relative to a DA has to be determined in 3-D space. The attack angle is given by

$$\theta_a = \arccos \frac{\vec{p} \cdot \vec{v}}{\|\vec{p}\| \cdot \|\vec{v}\|},$$

where $\vec{p}$ and $\vec{v}$ denote the position and velocity vectors, respectively, as indicated on Figure 5.3. For small angles and in applications where the processor is limited to a smaller bit-size architecture, this formula may, however, result in overflow/underflow since the value of the cosine function is close to one for small angles. If the resolution of the software architecture is not large enough, calculation errors may therefore result. A similar type of overflow problem[1] resulted in the fratricide incident of 25th February 1991 when a Patriot missile system failed to intercept an incoming Scud during the Gulf War in Saudi Arabia [75]. An alternative, which should provide more stability involves use of the arctan function. Arctan employs both the sine function (the trigonometric function with the best resolution for small angles) for the cross product and the cosine function (good resolution around larger angles) for the dot product. Their use is therefore less sensitive to measurement inaccuracies. Although arctan calculates the angle correctly, it is not possible to distinguish whether the angle is in the first or third quadrant if the value is positive, and when it is negative, it may be an angle in the second or fourth quadrant. It is therefore required to observe the sign values of the sine and cosine functions separately in order to determine the quadrant of the angle uniquely. In the case where cosine is negative, π is added to the result. When calculating the angle for this application, only the smallest angle between the two vectors is of interest, resulting in the attack-angle

$$\theta_a = \arctan \frac{\|\vec{u} \times \vec{v}\|}{\vec{u} \cdot \vec{v}}. \tag{5.3}$$

After the angle θ_a has been determined according to (5.3), the angle in radians is utilized so as to calculate the course-related threat value as

$$V_c = \begin{cases} 0 & \text{if } |\theta_a| > \frac{\pi}{2}, \\ \cos\theta_a & \text{otherwise.} \end{cases} \tag{5.4}$$

Cosine is seen as a suitable function for determining the threat value, since for small angles the threat value is close to one whereas, the closer the angle is to diverting away from the DA, the closer the value is to zero. If, however, a threat is not approaching a DA with an approach angle of at least 90°, the threat value is set to zero.

When a threat closes in on a DA and starts manoeuvring, the movement generally becomes erratic and unpredictable [216]. As such, course is a good predictor of intent when a threat is at a medium distance. As soon as the course measurement shows a sudden increase or decrease, the threat should be flagged as exhibiting highly threatening behaviour. Also, when using pre-deployment information in conjunction with the DMs, it should be noted that course is a better intent estimator when an aircraft is employing gun systems, such as Vulcan cannons, since an aircraft needs to be directed towards a DA to make use of such weapon systems effectively. In contrast, this is not necessarily the case with guided-munitions which have the advantage of homing in on a target and are therefore not as sensitive to attack angle threat values as opposed to gun-munitions.

[1] The limited bit size architecture (24 bit) of the Patriot's micro-controller resulted in overflow errors when using the system's internal clock to determine the operational-time (*i.e.* the time since the system was last restarted). Since the system was in operation for around 100 hours, there was an error of approximately 0.34 seconds when calculating the operational-time. The operational-time is, in turn, used for the calculation of the interception point and, because of the high speed of a Scud missile, this inaccuracy had a significant effect when calculating the interception point. The scud failed to intercept the incoming missile, resulting in the loss of 28 soldiers lives and injuring 100 others. The interested reader is referred to [75] for more detail.

5.2.3 Closest Point of Approach TE Model

Closest Point of Approach[2] (CPA) is a widely-used criterion for predicting the threat level of an aircraft or missile [34, 198], as depicted in Figure 5.4.

The CPA is calculated by assuming a constant course from the current position. The orthogonal passing distance is used to calculate a threat value on a zero to one scale. Since CPA is only important when an aircraft approaches a DA, a prerequisite for this model is that the course-related threat value should be positive; otherwise this model assumes the value of zero. Because vectors are used for the calculation, it is possible that the CPA results in a high threat value although the threat is moving away from a DA if the course is not used as prerequisite.

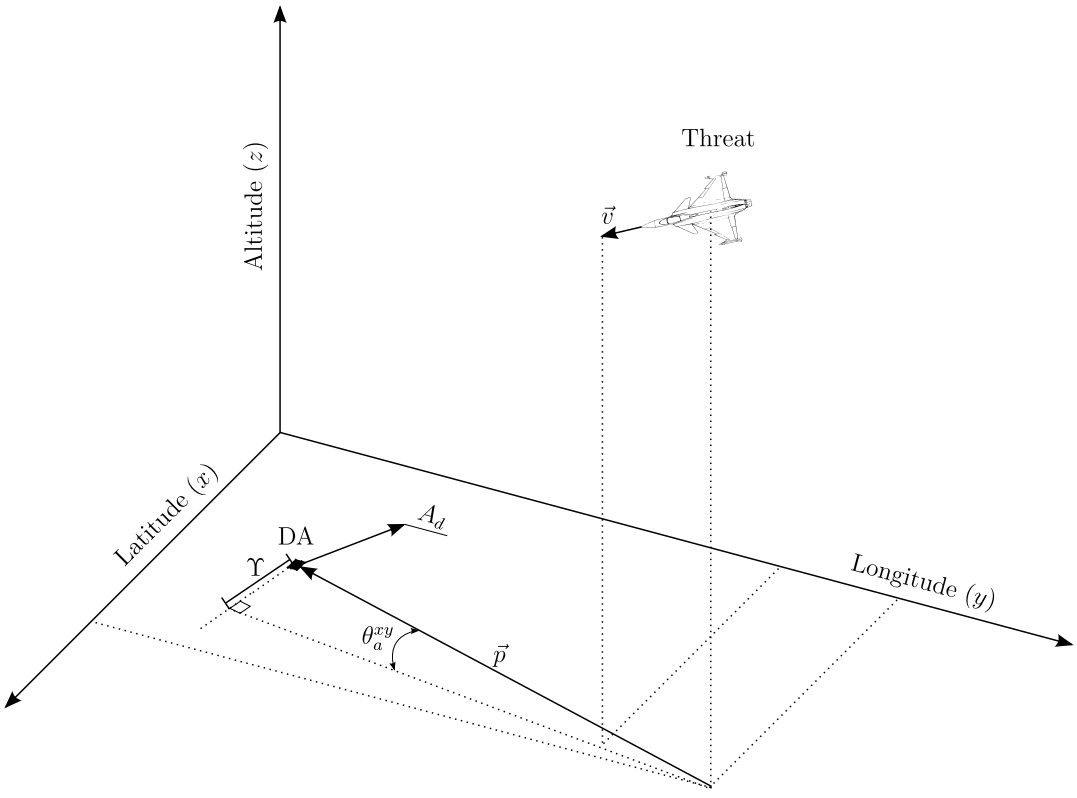


FIGURE 5.4: *Working of the CPA-related TEM.*

CPA is, in essence, a 2-D threat criterion. This is also advantageous because of the requirement that the DMs should be utility-independent when applying certain aggregation techniques [102, Chapter 10]. Only the x and y coordinates of a threat $\mathcal{T}$ and DA $\mathcal{D}$ are therefore used to determine the passing distance, as illustrated in Figure 5.4.

Prior to determining the passing distance value Υ, a projection of the threat's position onto the (x, y)-plane is used to calculate the length of the vector $\vec{p}$ according to (5.1). Similarly, the angle θ_a^{xy} is also determined in the 2-D plane according to (5.3). The Υ value is then calculated as,

$$\Upsilon = \|\vec{p}\| \sin \theta_a^{xy}.$$

Another important parameter in the calculation of the CPA-related threat value, is the *action distance*, denoted by A_d. The action distance is the maximum passing distance that is of concern

[2]The implemented CPA model is referred to as the *Projected Orthogonal Passing Distance* (POPD) model in non-military literature. CPA is used here in order to keep the terminology consistent for a military readership.

to the ground-based defenders. It is common doctrine that the action distance is related to the KOB, which is typically specified in the pre-deployment information database. For the purposes of the simulation in this thesis, the action distance is given by

$$A_d = 0.9 \times \text{KOB}.$$

The resulting CPA threat value is then given by

$$V_p = \begin{cases} 1 - \frac{\Upsilon}{A_d} & \text{if } \Upsilon \leq A_d, \\ 0 & \text{otherwise.} \end{cases} \tag{5.5}$$

In contrast to course, which is important in the prediction of intent when an aircraft is employing gun systems, the CPA becomes important when bomb weapon systems are employed. In the absence of guidance technology, an aircraft's projected trajectory should pass directly over a DA for an accurate bomb-strike.

5.2.4 Altitude-related TE Model

Altitude, as well as sudden changes in altitude, may effectively be employed to distinguish non-threatening targets (commercial or surveillance aircraft) from threatening (attacking) aircraft. During air attacks, it is general practice that an aircraft approaches a target from far away by flying low, in order avoid radar [167]. Similarly, before the aircraft releases ordnance, sudden changes in elevation are typically witnessed. Furthermore, fighting aircraft generally fly at lower altitudes when compared to surveillance aircraft and commercial aircraft [195]. Elevation is not an ideal deterministic criterion for predicting threat level on its own, but combining elevation with other TE models may provide a more accurate estimate of intent.

The principle by which an implemented altitude-related TEM may function is illustrated in Figure 5.5. As may be seen in the figure, the threat value is set equal to one if the altitude is lower than some lower-bound A, and set to zero as the altitude exceeds that of commercial and surveillance aircraft (the upper bound B in the figure). Between these two extremes, the threat value decreases linearly as a function of altitude Ξ, as captured by the piece-wise linear function,

$$V_a = \begin{cases} 1 & \text{if } \Xi \leq A, \\ \frac{-\Xi}{B-A} + \frac{B}{B-A} & \text{if } A < \Xi \leq B, \\ 0 & \text{if } \Xi > B. \end{cases} \tag{5.6}$$

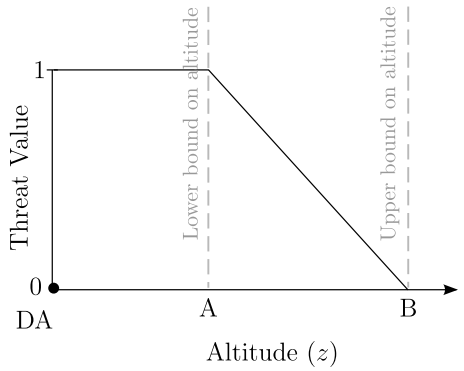


FIGURE 5.5: *Working of the altitude-related TEM.*

According to [45], typical bombing altitudes during the Second World War were generally in the order of 3 000 m and 7 000 m, but more regularly between 5 000 m and 6 000 m. The major predictor for bombing altitude is the carrying capability of the aircraft — the heavier the payload, the more lift is required to keep the aircraft airborne [121]. Also, the higher an aircraft flies, the less dense the atmosphere is, which results in less oxygen for combustion in the engine, thereby resulting in less thrust. These types of arguments may be used to determine suitable values for the bounds A and B in (5.6). Because of a lack of better information, the typical bombing altitudes during the Second World War are used in this thesis to quantify the altitude-related threat level posed by aircraft. The lower bound A and upper bound B assume values of 5 000 m and 10 000 m, respectively.

Example 5.1. *Consider the working of the DMs described above with reference to the scenario introduced in §4.6. The threat tracks in Figure 4.9 correspond to a scenario spanning 120 seconds. For the purpose of this example, as well as for the subsequent examples in this chapter, the TEWA cycle is repeated every second (*i.e. *the real-time data updates are assumed every second).*

The threat values produced by the four DMs described above are shown graphically in Figure 5.6. By comparing these threat values to the illustrated scenario, it may be seen that the DMs do indeed function as expected. Both V_a values for a specific threat correspond throughout since the DAs are all placed at an elevation of 0 m. □

5.3 Data Fusion Processes

The purpose of the TE fusion component within the TEWA cycle is to combine the results from the various DMs. This fusion process must be achieved in a manner that is not only mathematically tractable, but also practical for use in real-world military applications.

All DMs produce threat values on the real interval $[0, 1]$. The aim of the fusion process should therefore be to construct a value-based prioritised list of threats, based on their corresponding system threat values. The adjective *value-based* in this context refers to a cardinal prioritised list, as opposed to an ordinal list. Multiple techniques exist for this purpose in the field of *Multi-Attribute Utility Theory* (MAUT) [102]. These different techniques may be classified as *value measurement models*, *goal aspiration models* or *outranking models*. The reader is referred to [104] for a detailed comparison of these methods.

Value-based measurement models are the only MAUT models in which a numerical preference score is calculated pertaining the degree to which a certain alternative may be preferred above another. The results are therefore quantitative in nature, facilitating retention of the level of preference of one alternative over another during the fusion process.

When utilising the DMs of §5.2, a threat value is determined for each threat-DA-DM triple. If there are, for example, three incoming threats, two DAs to protect, and four different DMs, then a total of 24 different threat values will therefore be calculated. The output of the TE subsystem should, however, be a single system threat value for each threat in order for the weapon assignment subsystem to function effectively.

The purpose of the data fusion process proposed in this thesis is to fuse together these different threat values — which are distinguished according to threat, DA and DM — so as to obtain a single system threat value per threat. Different hierarchies of threat values are shown in Figure 5.7. Since the DMs are typically configured differently, the first step should be to obtain a threat value with respect to each threat-DA pair (*i.e.* to fuse together the threat values obtained by the different DMs, denoted by step 1 in the figure). After obtaining the theat-DA threat

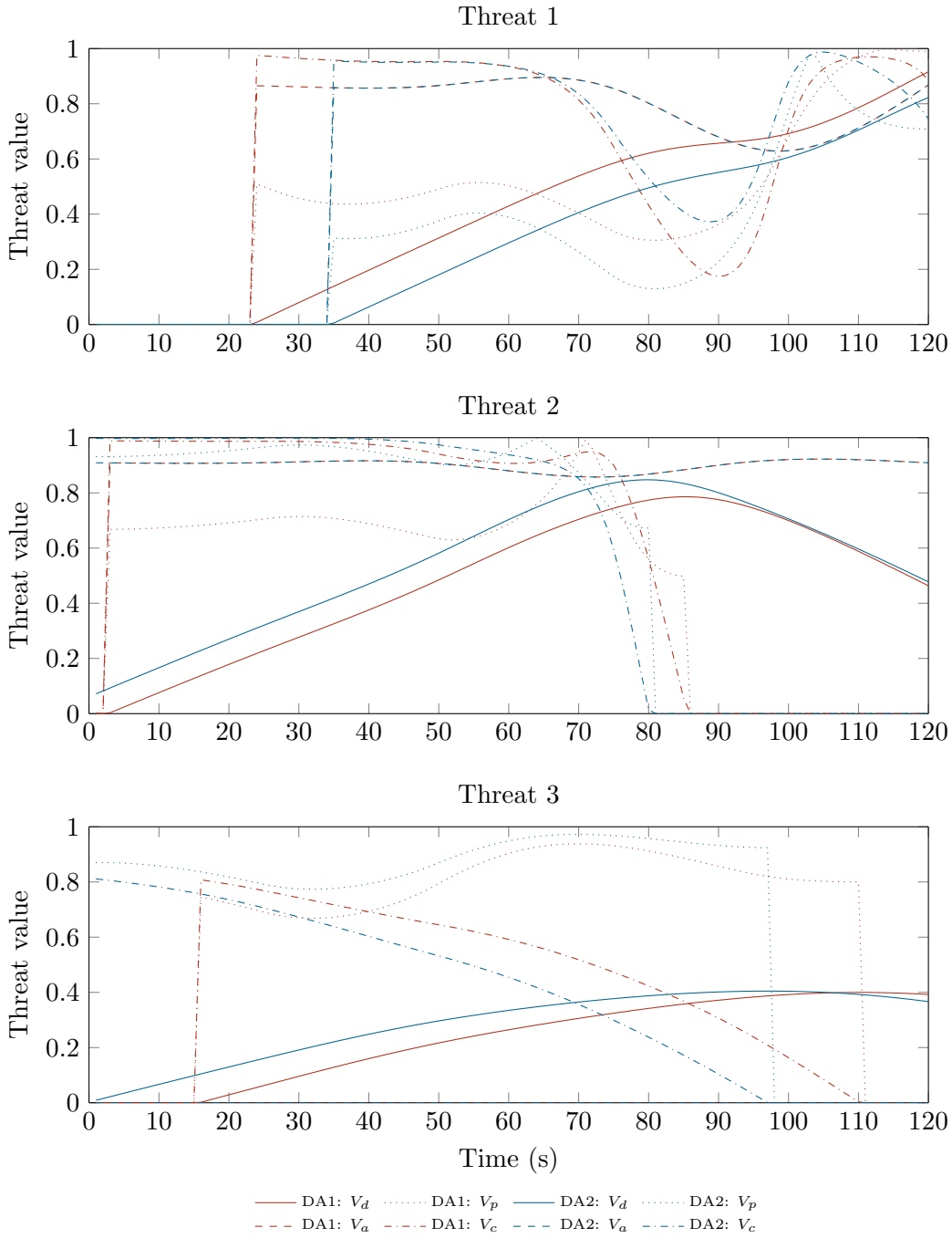


FIGURE 5.6: *Original threat values, distinguished in terms of threat, DA and DM.*

list thus fused, the importance weights of the DAs may finally be used to fuse together a single system threat value for each threat, denoted by 2 in the figure.

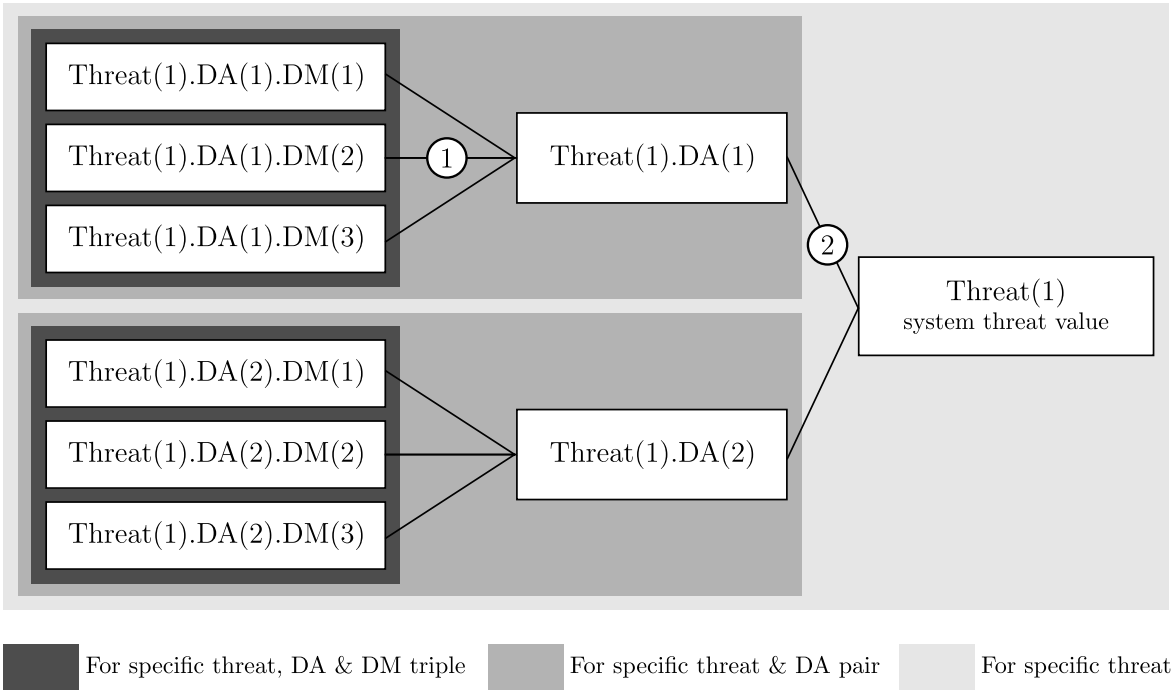


FIGURE 5.7: *Relationships between different threat values. Fusion process (1) involves the use of multi-attribute utility functions within an aggregated value function tree in order to obtain an individual threat value per DA. Fusion process (2) involves use of the additive weighting method, where the weighting coefficients are the normalized importance values of the DAs.*

5.3.1 Computation of Threat-DA Pair Threat Values

The evaluation of threat values according to the DMs described in 5.2 is updated each time new information is received from the sensor systems; the duration of this update-cycle is known as the TEWA *cycle time*. Consequently, the deterministic threat values are updated throughout the engagement as the threats approach the DAs.

This TE process calls for a fusion model which is dynamic in nature. Furthermore, as threats approach the DAs, certain DMs become more accurate or relevant in estimating an aircraft's threat level. For fixed-wing aircraft, for instance, when threats are far from the DAs, the time-to-weapon-release may be a better predictor of threat level than distance-related threat measures. To illustrate, at long ranges when two threats are executing the approaching phases of their attack profiles at the same distance from the DAs, time should be a better predictor of threat level. Although they are the same distance from the DAs, the threat with the higher velocity poses a more imminent danger to the DAs, since it would be able to release its weapons earlier. In contrast, if threats are entering the manuevre phases of their attack profiles (in anticipation of weapon release), distance-related measure ought to be a better predictor of threat level, since time-related measures become increasingly difficult to predict accurately during these final phases of the attack profile. These are typical concerns that the TE model output fusion process has to account for. The fusion method must therefore involve expert opinion during its development phase.

For this step of the fusion process, two methods were considered — the construction of an individual utility function for each DM which is based on expert inputs obtained from the *Analytical Hierarchy Process* (AHP) or, alternatively, the construction of a multi-attribute utility function which provides a threat value for a specific DM combination, also based on expert inputs. These two approaches are described below.

Multiple Weighting Utility Functions

A possible solution to the above-mentioned fusion problem is the construction of multiple weighting utility functions. It is proposed that these utility functions should solely be a function of range to a DA, thereby allowing the fusion process to utilise this range for determining the importance weighting of a specific DM. A separate weighting function is therefore required for each DM. The different weighting functions should be used to determine the relative importance of a DM at a specific range. The resulting weightings may then be applied in the additive weighting method to aggregate the different threat values. The end result is a single threat value for each threat-DM pair, where the fused threat value depends on the threat values of the DMs combined.

These DM utility functions may be constructed by the inputs provided from the *Analytical Hierarchy Process* (AHP) which is often used to elicit domain expert knowledge. The AHP is a method, dating from 1980, for facilitating complex multiple-criteria decision analyses where the exact criteria for preferring one alternative to another are difficult to quantify [7], as is sometimes the case with DMs.

The AHP provides a hierarchy of decision items or alternatives using pairwise comparisons of decision items in matrix form. These pairwise comparisons produce a weighting score for each decision item which indicates a specific decision item's importance relative to the alternatives. For this application, the decision items (alternatives) are the different DMs. The AHP should be applied for several different distances in order to determine how the relative importance values of the different DMs depend on distance. The problem structure for the proposed AHP is depicted in Figure 5.8.

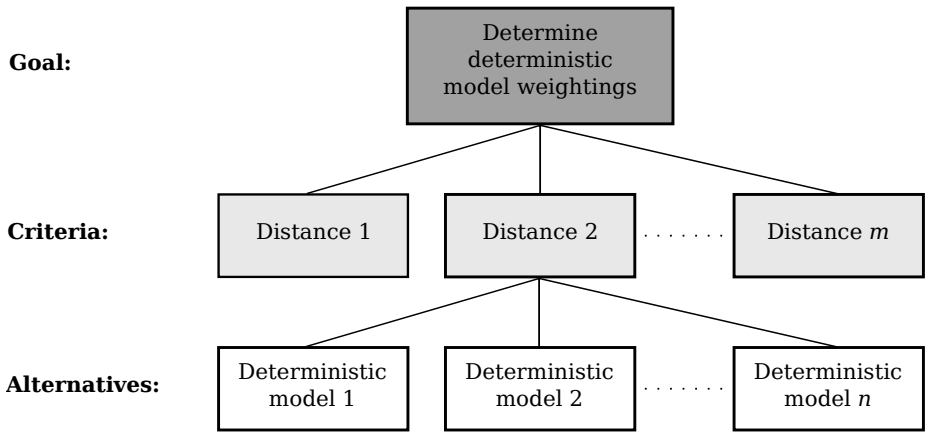


FIGURE 5.8: *AHP problem structure.*

As may be seen in Figure 5.8, the decision criteria are discrete threat-to-DA distances, but the weighting function must be continuous in order to apply the resulting fusion approach effectively. One method of obtaining a continuous function is to apply curve fitting regression methods to the discrete weighting points (DM importance weights for each of the distances) resulting from the AHP. The resulting curve will then be a function of a threat's distance from a DA, providing a weighting importance value for the relevant DM. These functions may be calibrated for a specific scenario by considering historical test data.

After eliciting the importance weighting functions of the respective DMs, these weights may be used within an aggregated value function tree in order to determine the resulting threat-DA threat list values. A higher order hierarchical aggregation model may be used, as was suggested by Kok [104], where decision criteria are considered inherently different and should be aggregated first separately and then later in combination so as to obtain the value at the top of the aggregation tree. This concept is illustrated in Figure 5.9, but should, however, be carefully considered because of the interdependency between criteria.

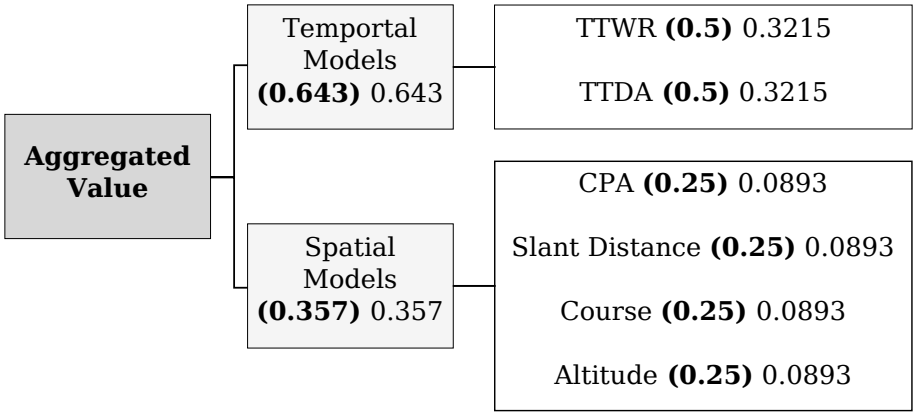


FIGURE 5.9: *Aggregated value function tree structure.*

Apart from arranging independent criteria in the hierarchical aggregation tree, interdependent criteria may also be grouped together with similar criteria which may be aggregated separately [222]. The course-related threat measure exhibits a causal and associative interdependency with CPA-related threat value, since the CPA is directly influenced by the aircraft's course. Similarly, *time-to-weapon-release* (TTWR) and *time-to-DA* (TTDA) DM threat values exhibit strong interdependence. It is therefore sensible to categorise these criteria into two interdependency classes — temporal models and spatial models. This concept is illustrated in Figure 5.9. The AHP should be applied separately for the temporal and spatial models, for each of the distances considered. It is also required that the AHP should be repeated for each interdependency class in order to obtain separate weightings.

In Figure 5.9, the boldface values represent weightings returned by hypothetical utility functions, whereas the normal-font values represent normalised importance weights. By following this hierarchical aggregation tree approach, the resulting fusion model is expected to be more robust in respect of fluctuations in threat criterion values due to the top-down cumulative weight distribution and the normalisation of criterion-interdependency weights.

Since the different weighted utility functions are aggregated to form an all-encompassing threat-model value, there are certain conditions that must be satisfied. The relative importance value depends on expert preference with respect to the attributes, and is based on the assumption

of preferential independence. The different DMs must therefore satisfy the conditions of utility independence, preferential independence and additive independence [102]. Consequently, if distance is used as the criterion for evaluating alternatives, then a DM, such as a slant distance-related DM, may not form part of the set of alternatives or, for that matter, any model that exhibits dependence on distance. This fusion approach is therefore at best risky, since the possibility exists to make serious mathematical modelling mistakes and hence caution should be applied when implementing this method.

The AHP is, nonetheless, still the preferred method of choice for eliciting expert opinion, since it allows for the direct integration of operator training in the models through the use of expert opinion so as to obtain initial weighting functions. In addition, the target group (military experts) is not required to have any background knowledge of decision making theory or matrix algebra for the system designer to be able to employ this method effectively. No alternative MCDA method could be found which produce a cardinal ranking list (based on threat values) instead of an ordinal list. A major problem arises, however, in respect of inconsistent inputs, which is common in subjective human judgement situations.

These inconsistencies arise when three or more items are compared. The level of inconsistency may be quantified using a confidence measure which may, in turn, be used to quantify the scope of inconsistencies present [222]. The natural incorporation of such an inconsistency index into the AHP is one of the reasons why the AHP is a popular method for eliciting expert knowledge. This inconsistency index may serve as a threshold value for accepting the expert inputs. If the threshold value is not reached, the experts may be required to repeat the questionnaire on which the construction of the fusion model is based.

In an attempt to limit the amount of personal subjectivity, it is advocated that a group of military experts attend a workshop and discuss the different alternatives for each of the criteria with the goal of reaching group consensus in respect of input values for the AHP.

Despite the limitations of the AHP, this methodology has been applied with great success in the context of various high-risk applications [214]. Within the TE domain, the AHP has been applied successfully to identify threats and human errors affecting aviation safety [38]. In the defense domain, the AHP has been applied successfully in performing trade-off analyses for choosing between weapon systems and evaluating defense architecture frameworks [8].

Multi-attribute Utility Function

An alternative to the aforementioned multiple weighting utility function method, is the construction of a single multi-attribute utility function. Such a multi-attribute utility function should provide a single threat value for each threat-DM pair, where the fused threat value depends on the threat values of the DMs combined. For illustrative purposes, three spatial DMs are considered — the slant distance model, the CPA model and the altitude-related model, all described in §5.2. These three models were specifically chosen in order to adhere to the requirements of utility- and preferential independence when constructing a multi-attribute utility function.

In order to construct the utility function, it is first required to quantify the preferences of the end-users. To this end, the threat-value interval may be subdivided into three intervals; the midpoints of these intervals may then represent respectively low (0.16), medium (0.5) and high (0.83) threat values. This concept is illustrated by Figure 5.10.

The intervals in Figure 5.10 are suggested in order to facilitate effective elicitation of end-user preferences. It is anticipated that domain experts should be able to provide an appropriate or expected combined threat value when the threat value of the altitude and passing distance DMs

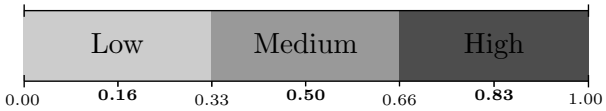


FIGURE 5.10: *Threat value intervals.*

are low, for instance, but that of the slant distance DM is high. This elicitation process may be repeated for all twenty seven combinations of the DM threat levels in order to obtain a combined or characteristic threat value for each triple.

In an attempt to limit the amount of personal subjectivities of domain experts and to address the difficulties that arise when aggregating individual preferences, it is again advocated that a group of military experts attend a workshop and discuss the different alternatives for each of the criteria, with the goal of reaching group consensus in respect of input values for construction of the utility function.

Hypothetical threat values for the aforementioned combinations were selected for use in a proof-of-concept example. MATLAB was used to fit various functions through the resulting twenty seven data points. The data points that were used for the curve fitting are shown in Appendix A. It was determined that a third-degree, three-variable polynomial provides the best fit. The resulting function,

$$\Gamma\left(V_d, V_p, V_a\right) = \sum_{i=0}^{3}\sum_{j=0}^{3}\sum_{k=0}^{3} a_{ijk} V_d^i V_p^j V_a^k, \tag{5.7}$$

returns a fused characteristic threat value in the real interval $[0, 1]$ for a specific threat-DA pair. In (5.7), the symbols V_d, V_p and V_a denote the slant distance, passing distance and altitude DM threat values with respect to the considered DA, respectively. The values of the coefficients a_{ijk} are shown in Table 5.1. All values not in the table assume a value of zero.

TABLE 5.1: *Coefficients for the multi-attribute utility function (5.7).*

$a_{000} = -0.12$	$a_{100} = 1.3$	$a_{200} = -1.2$	$a_{300} = -0.48$	$a_{110} = -0.74$	$a_{210} = 0.52$	$a_{101} = -1.8$
$a_{201} = 1.5$	$a_{010} = 0.45$	$a_{020} = 0.31$	$a_{030} = -0.21$	$a_{001} = 0.42$	$a_{002} = 0.8$	$a_{003} = -0.54$

The function in (5.7) may be visualised as a three-dimensional, colour-shaded cube defined by three DM threat values (independent variables) along the cube axes and the fused threat value (dependant variable) in colour. This function is therefore shown as a series of two-dimensional, coloured slices[3] in Figure 5.11. In the figure, the vertical-axis denotes the altitude-related threat value V_a, described in §5.2.4, while the horizontal-axis denotes the value of the passing distance threat value V_p described in §5.2.3. The four slices are distinguished in terms of the value of the orthogonal distance threat value V_d, described in §5.2.1, which is kept constant for each slice. The colour-map on the right indicates how the colours in the slices translate to the calculated fused threat value. The two points indicated — A and B — are used as reference points to facilitate an interpretation of the figure.

The four slices were obtained by keeping V_d at a constant value, as indicated at the top of each slice. The different constant values, which define the cutting planes, were determined as follows: First low and high threat values of respectively 0.1 and 0.9 were selected. The resulting threat interval $[0.1, 0.9]$ was then partitioned into three equally-spaced intervals in order to obtain the last two values used for the visualisation — 0.37 and 0.63.

[3] A slice is a "cross-section" of the dataset. Any kind of surface may be used to slice the data, however, in this case, the cutting surface is the plane that results from keeping on of the DM threat values constant.

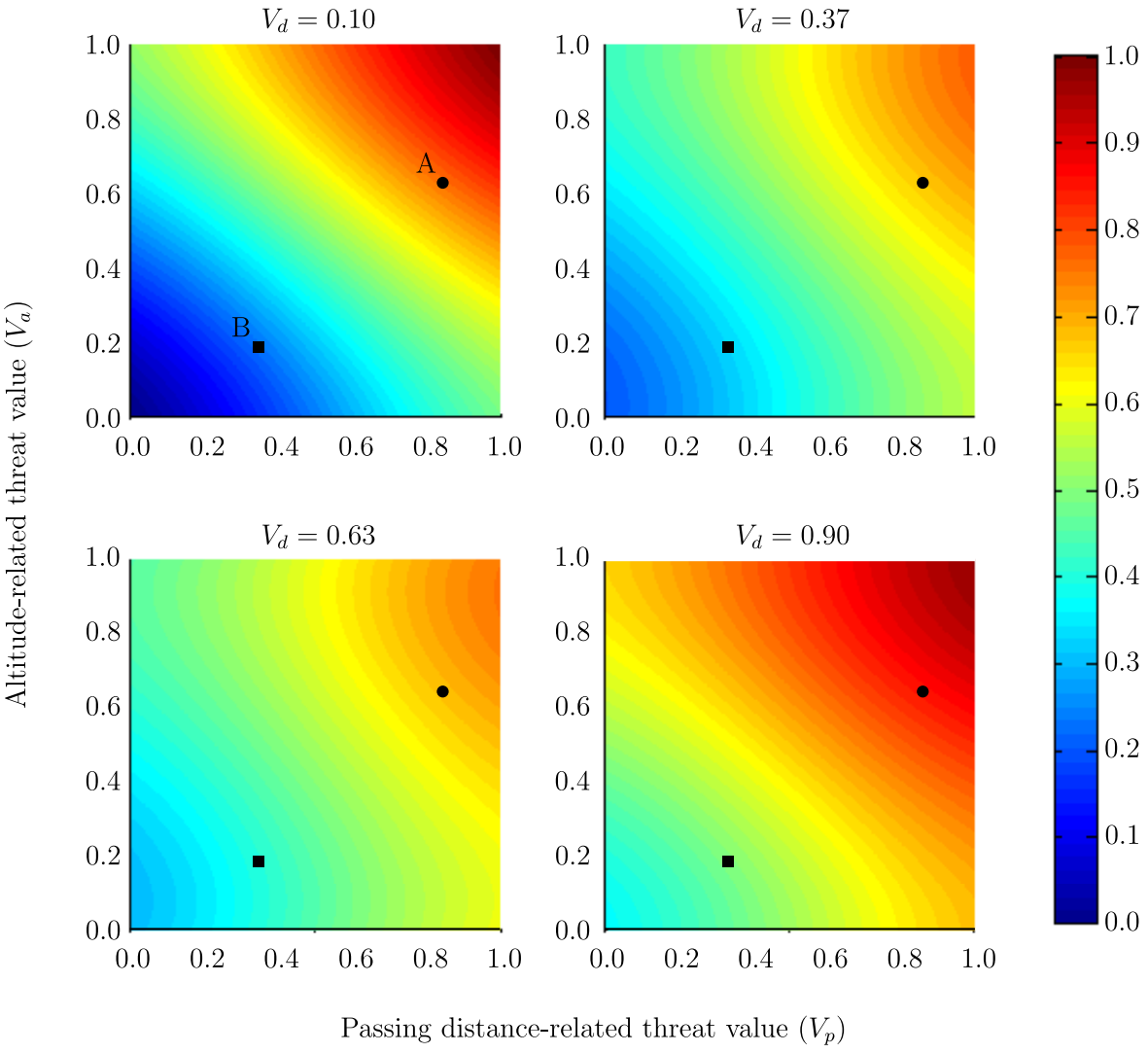


FIGURE 5.11: *Four slices of the multi-attribute utility function (5.7).*

For the first slice, $V_d = 0.10$, the threat is still far from the DAs since the distance-related threat value is low. Since the threat is far away (currently in the approach attack phase), the distance-related threat measure is not as important as the alternatives. Although speed is not implemented in these models, the speed at which a threat approaches should be a better DM by which to distinguish between the level of threat of different threats, since a faster threat will reach the DAs sooner. During these stages, a high V_a value may signal that the threat is attempting to avoid sensor systems and/or long-range WSs, whereas a high V_p value may indicate the early intent of the threat. These last two threat values therefore collectively determine the fused threat value. It would seem that both the V_a and V_p values contribute equally to the fused threat value — *i.e.* if both values are 0.4 (above point B), the fused threat value would also be approximately 0.4. The diagonal or 135° colour-distribution of the area corroborates this observation.

As the threats approach the DAs, the relative importance of the values of V_d increases. This may be seen from the monotone distribution of the colours; the dominant colour is close to the colour defining V_d. When comparing the relative importance of V_a and V_p, it may be seen that at this stage, the fused threat value depends more sensitively on the V_p value. The shape of the blue zone at point B clarifies this — the fused threat value has more resolution along the horizontal axis than along the vertical axis where the colours are more monotone. The horizontal axis threat value V_p therefore has a stronger influence on the resulting threat-DA threat value. For the slice in which $V_d = 0.63$, the fused threat value relies even more heavily on V_d (monotone colour distribution corresponding to the value of V_d) while the V_a value is gaining more priority (the change in gradient of the colour zone close to point B).

The last slice, where $V_d = 0.9$, corresponds to the situation where the threat is very close to the DAs. The overall colour of the slice is therefore more towards the high threat value red spectrum of the colour map. During this stage of the engagement, the time available to make good decisions decreases drastically since the threats are close to their WRLs. Although the fused threat value is mainly higher during this stage, very low values of V_a and V_p still result in a lower fused threat value, since the threats are generally not able to release their ordnance if their altitude is too high and they are not aligned towards a DA. The V_d value is, however, still the most important factor during this stage, since the DAs are closing in on the stand-off range of a threat's ordnance and, as a result, may be attacked at any moment.

Example 5.2. *Suppose the threat values shown in Figure 5.6 are to be fused together in order to obtain a single threat value for each threat-DA pair. Only three spatial DMs are implemented — the slant distance model, the CPA model and the altitude-related model. The course model is not included in the fusion process, since the inclusion of the course DM will violate the condition of utility independence, which is required in order to apply the multi-attribute utility function. The course DM exhibits a strong utility dependence with respect to the CPA DM, since both DMs quantify the heading of a threat relative to a DA. The resulting fused threat values are shown in Figure 5.13 as a function of time. These threat values correspond to fusion step 1 in Figure 5.7.* □

5.3.2 Threat-DA Threat Value Scaling

Another consideration in the fusion process, not described above, is the scaling of threat values. TE is typically conducted on all threats within the AOR which are identified as hostile or unknown. Doctrine often requires that if a threat enters an area enclosed by a pre-specified distance radius from a DA — here referred to as the *Keep-Out Boundary* (KOB) — the threat must be classified as highly threatening with respect to the DA considered. If the assumption is

made that threats are classified according to three priority categories, namely low, medium and high (as illustrated in Figure 5.10), then the scaling should ensure that any threat is classified as highly threatening when it enters the KOB.

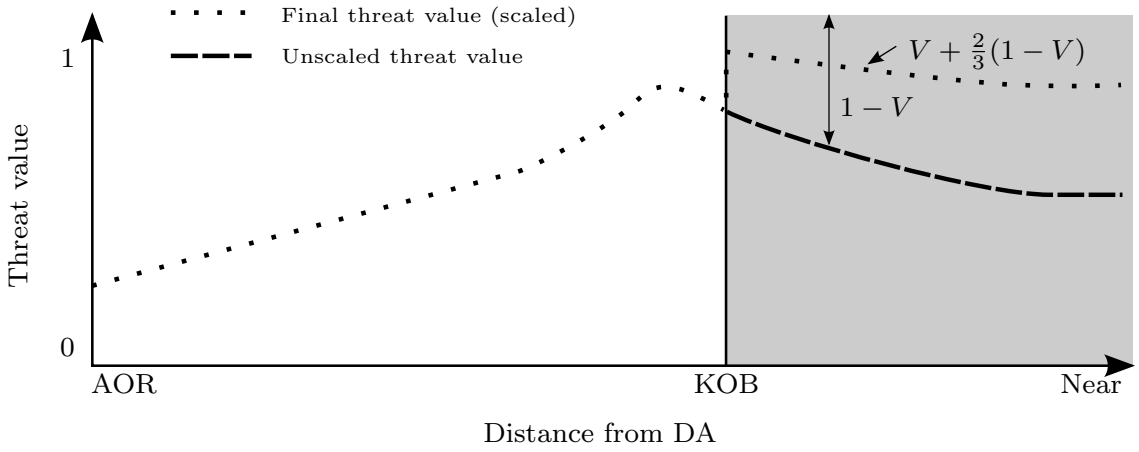


FIGURE 5.12: *Suggested threat value scaling method when threats cross the KOB.*

The scaling should therefore make provision for increasing the threat priority of low and medium threats to high priority when a threat crosses the KOB. It is advocated that the threat value V of a threat crossing the KOB should be increased by $\frac{2}{3}(1-V)$, as illustrated in Figure 5.12. This scaling of the threat values will always ensure that the low and medium threats are classified as highly threatening threats when inside the KOB. This is true if the threat values V of the classes of low, medium and high priority threats occupy the ranges $0 - \frac{1}{3}$, $\frac{1}{3} - \frac{2}{3}$ and $\frac{2}{3} - 1$, respectively. In practice, however, the scaling value (suggested here to be $\frac{2}{3}$) should be agreed upon by domain experts. The result of this process is a scaled threat-DA threat list.

Example 5.3. *Suppose the $\frac{2}{3}$–KOB scaling described above is to be applied to an unscaled fused threat value. The pre-scaled threat values depicted in Figure 5.13 result in the threat values shown in Figure 5.14 are after the scaling process.* □

5.3.3 Computation of System Threat Values

After calculating a single threat value for each threat-DA pair, it is required to fuse these values together in order to obtain a system threat value for each threat (fusion step 2 in Figure 5.7).

Threat values are typically segregated into *target-based* and *asset-based* formulations. For target-based problem formulations, a numerical value is typically attributed to each threat, depending on the threat attributes. These numerical values are the threat values resulting from the TEMs. The limitation, however, of target-based formulations of a system threat value is that these types of threat values are generally designed for defending a single asset, unlike the area-based defence requirement of a typical GBAD environment. In the alternative approach, when adopting *asset-value* based formulations, DAs are ranked according to a relative importance value, as described in §2.4.5. By employing these DA importance values, the total cumulative survival probability of the threats — weighted according to the DA to which they pose a threat — is minimised. Asset-based formulations have proved to provide superior results for area-based defence (as is the case with NCW in a GBAD environment) compared to target-based formulations [123]. More pre-deployment information and processing power is, however, required in asset-based

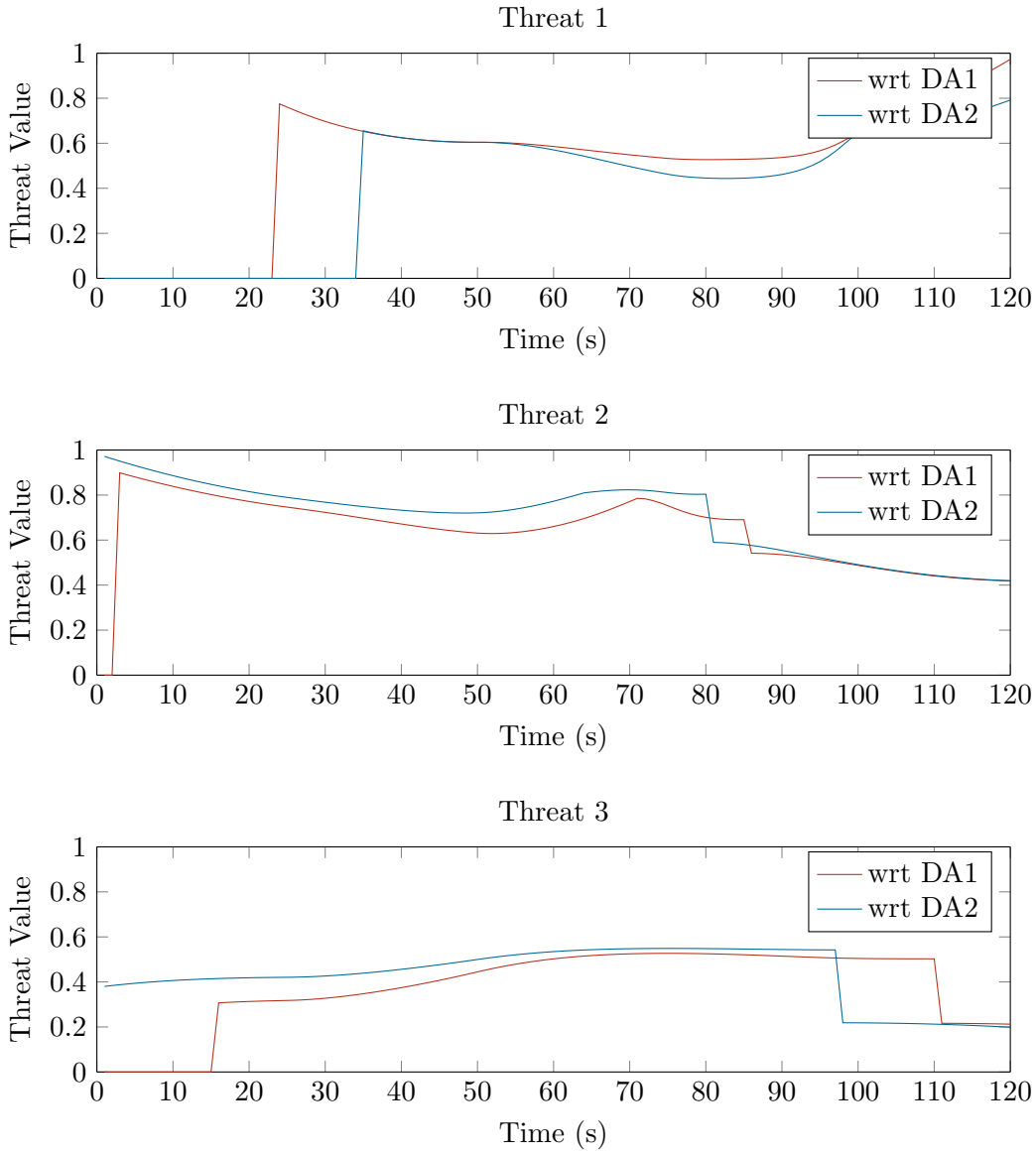


FIGURE 5.13: *Threat-DA threat values before scaling.*

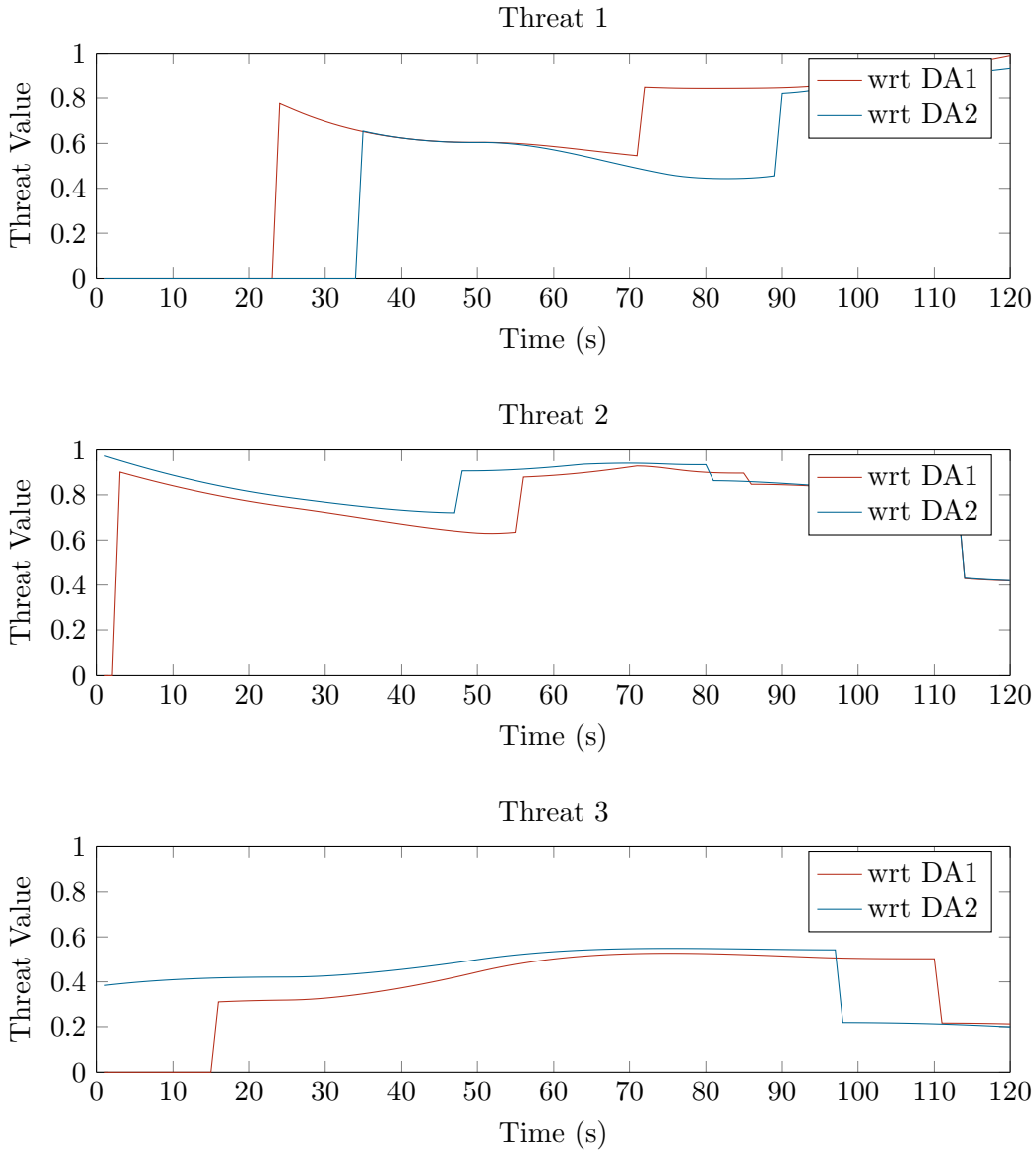


FIGURE 5.14: *Threat-DA threat values after scaling.*

formulations, than in target-based formulations. This asset-based formulation is, nonetheless, still more suitable for a networked GBAD environment where different assets require different levels of protection.

Du Toit [59] suggested a method for converting a target-based formulation to an asset-based formulation by defining threat values in terms of DA priority weights. The relative DA priority weights should be determined prior to system implementation and should therefore be stored in the knowledge base. These priority values quantify the relative importance to the defending force of protecting the DAs in question. Several variables may influence the priority of a specific DA, such as its repairability, vulnerability and vital importance [167].

Repairability of a DA refers to its ability to recover from damage inflicted to it, and is usually determined based on the manpower, equipment and time required to repair the asset to a functioning state. *Vulnerability*, on the other hand, refers to the extent to which an asset is susceptible to damage and surveillance during an attack. Armour, position, countermeasures and camouflage are factors which influence a DA's vulnerability. Finally, *vital importance* is the degree to which the mission's success relies on a specific DA. Assessing the impact that the destruction of an asset will have on the mission's success is one way of determining its vital importance value. A command centre, for example, is more important in respect of ensuring mission success (for maintaining command and control superiority) than a redundant (backup) sensor system.

The threat-DA threat values may be fused together based on the DA priority values by an additive weighting function to obtain the system threat value

$$\Lambda_t = \sum_{k=1}^{n_D} \Gamma_{k,t} \cdot \psi_k, \tag{5.8}$$

associated with threat $t \in \{1, \ldots, n_T\}$, where ψ_k denotes the normalised priority value of DA $k \in \{1, \ldots, n_D\}$. In (5.8), $\Gamma_{k,t}$ represents the (fused) threat value of the threat t and DA k pair, while the total number of DAs is denoted by n_D and the total number of threats by n_T.

Example 5.4. *After obtaining the scaled threat-DA threat value list, these threat values may be fused together using the normalised importance weights of the DAs. The additive weighting method described in §5.3.3 was used for obtaining the system threat values shown in Figure 5.15. These threat values may be used for ground-based weapon assignment purposes.*

From Figure 5.15 it is clear that the threat values provide realistic threat estimates of the various threats. The sudden rises and drops in threat values occur when certain models are "switched on" or "switched off." For example, a threat must be within the pre-specified AOR radius before TE is conducted in respect of that threat. Similarly, the passing distance-related DM is also only active if the threat is heading towards a DA. It is worth mentioning that Threats 1 and 2 are scheduled to release their weapons at approximately times 90 and 60, respectively. It is therefore encouraging to note that the fused threat values of these threats are at their highest levels close to the weapon release stage. Finally, it is heartening that the threat values associated with Threat 3 which is, in fact, not exhibiting threatening behaviour, is significantly lower than those of Threats 1 and 2.

Rapid changes in the threat values may, however, be of concern to the effective functioning of the TEWA system, as described by Lötter and Van Vuuren [118]. The rapid changes in threat values, observed at times 3, 22, 37, 58, 72, 91 and 116 in Figure 5.15, may result in switching of weapon assignment recommendations. This type of switching is a typical emergent property of TEWA systems and is something that must be fully understood before implementing such systems. □

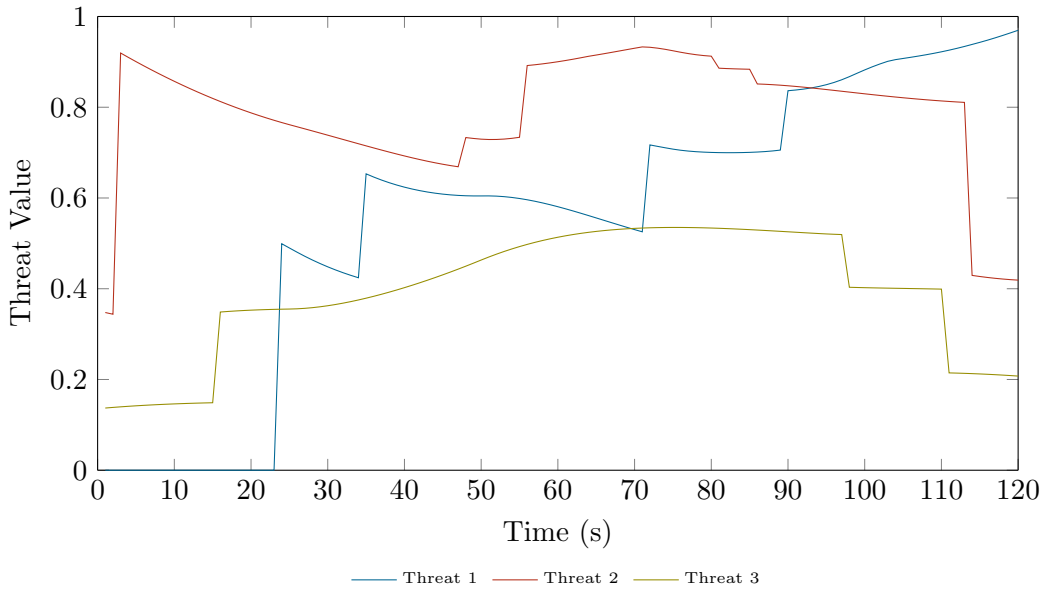


FIGURE 5.15: *System threat values as a function of time.*

5.4 Threat Evaluation Simulation Architecture

The mathematical modelling approaches and fusion methodologies described in the aforementioned sections may be implemented in the TE process within a simulation environment. An overview of the programme logic flow in such a simulation environment is depicted in Figure 5.16 with specific reference to the methods explained in the preceding sections. This logic flow diagram shows how the inputs to the TE subsystem are utilised in order to obtain a single system threat value per threat. This sub-routine should be executed for each TEWA cycle. The only inputs required for the calculation of these threat values are the positions of the DAs as well as the velocities, current- and past tracks of the threats.

5.5 Chapter Summary

In §5.1 the TE process was introduced with reference to the different classes of core parameters and processes constituting the TE subsystem. The TE overview forms the background for achieving an understanding of the four DMs described in §5.2. These DMs include a slant-distance related DM (§5.2.1), a course-related DM (§5.2.2), a passing distance-related DM (§5.2.3) and an altitude-related DM (§5.2.4). The slant distance-related DM of §5.2.1 calculates a threat level which depends on the slant-distance of a threat from a DA as well as the most probable stand-off range of the threat's ordnance. The course-related DM of §5.2.2 calculates the heading of a threat relative to a DA and this value is used, in turn, to calculate the threat value. The passing distance-related DM of §5.2.3 is in essence the widely used CPA model, whereas the altitude-related DM of §5.2.4 is a novel model designed to quantify a threat-level in terms of an aircraft's altitude relative to a DA.

The implemented data fusion process, which to a large extent influences the performance of a TEWA system, was explained in §5.3. The first step of the fusion process, as explained in §5.3.1, is to fuse the threat values obtained from the different DMs together in order to have a single threat value per threat-DA pair. Two methods were proposed for accomplishing this

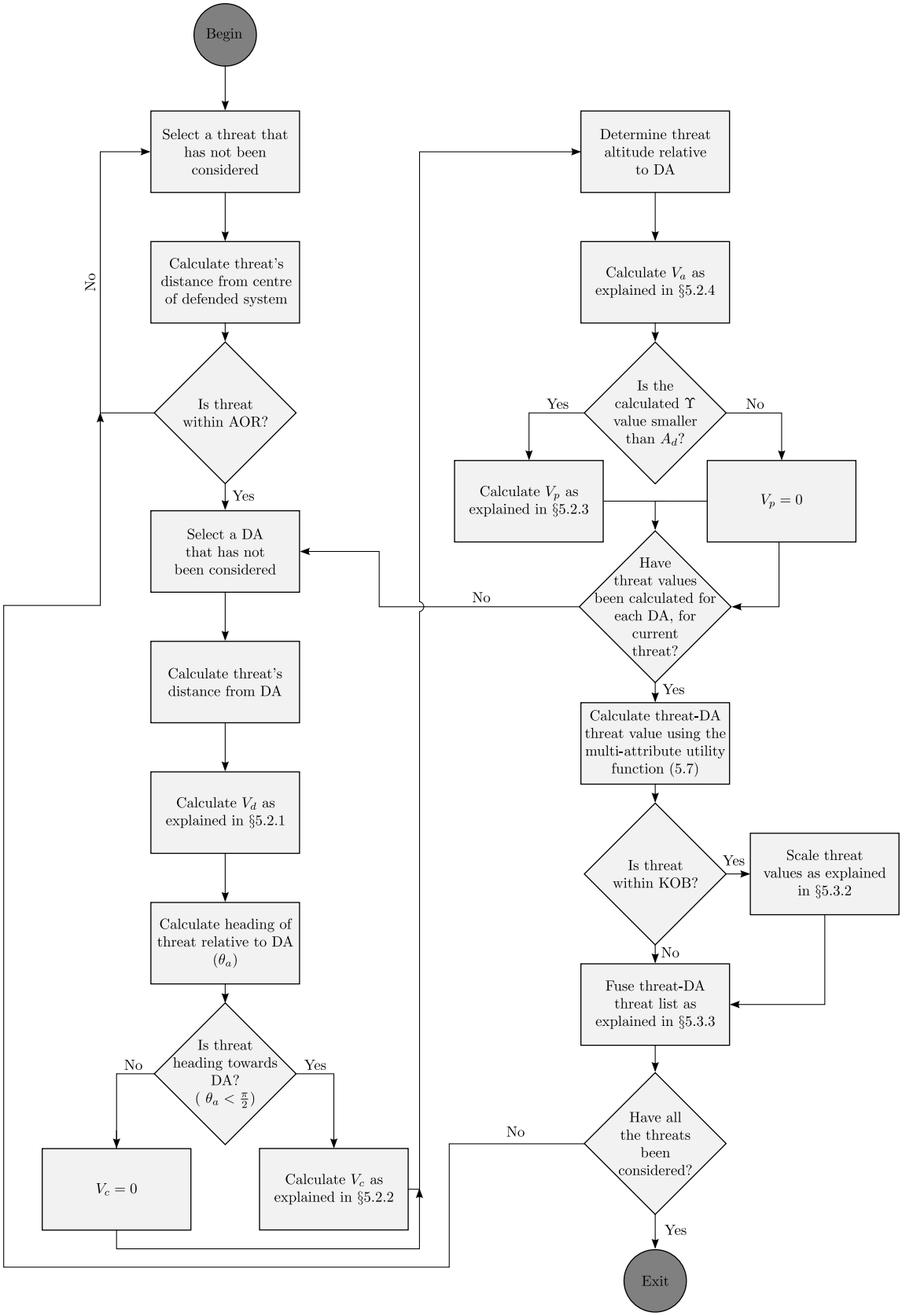


FIGURE 5.16: *Logic flow of the TE process in the simulation performance evaluation framework developed in this thesis.*

first step of the fusion process — a combination of weighted utility functions (§5.3.1) and a multi-attribute utility function (§5.3.1) where the attributes are the various DMs. An example was also provided in §5.3.1, demonstrating the working of the multi-attribute utility function. After determining the threat-DA threat list by means of the multi-attribute function, a scaling method was implemented if the threats are in close proximity to the DAs (§5.3.2). The final stage of the fusion process was explained in §5.3.3 where the threat-DA threat list is fused by employing the priority weights of the respective DAs. A single threat value is thus obtained per threat which may be used for WA purposes.

The aforementioned TE models and fusion processes were all implemented in MATLAB in the context of a discrete-event Monte-carlo simulation which may be used for performance evaluation purposes. The implemented logic-flow of the TE subroutine is depicted in §5.4.

The stochastic models developed by Roux and Van Vuuren [168] are not implemented in this thesis, because these algorithms require detailed and accurate information on the WSs, aircraft in order to be implemented effectively. As such, the complexity of these algorithms would not add, at this stage, any practical value to the current research project. Roux and Van Vuuren [168], in fact, also suggested the use of the less sophisticated flagging models as explained in §5.1. These simpler models are, however, binary in nature and are solely used for improving operator situation awareness; they are not utilised within the WA algorithms *per se*. The flagging models should be used in conjunction with the more sophisticated TE models. In practice, these models may be used for WA purposes in the case where the DMs or stochastic models are not deemed sufficient because of information shortages. Some of the flagging models, as suggested by Kok [104], were implemented, but do not serve any practical purpose within the simulation performance evaluation environment and, consequently, do not affect the performance of the TEWA system as a whole.

CHAPTER 6

Weapon Assignment Implementation

Opportunities multiply as they are seized.

— Sun Tzu

Contents

Weapon Assignment (WA) is informally defined as the process of reactive allocation of weapon resources (ammunition and WSs) to counter identified threats [72]. The inputs to the WA subsystem of a TEWA system is the system threat values calculated by the TE subsystem, as explained in Chapter 5. The WA subsystem of a TEWA DSS in the context of GBAD is, subsequently, responsible for proposing high-quality assignment proposals of available ground-based WSs to engage aerial threats over some specified time-frame [119].

This chapter opens with an overview of the WA problem and the different classes of model formulations for this problem. Static and dynamic model instances are described in some detail and example formulations are provided. After an understanding of the WA process has been acquired, the different solution approaches traditionally adopted to solve the WA problem are introduced. Furthermore, the difficulties associated with selecting an appropriate solution approach for the WA problem are also explained after which an overview is provided of the genetic algorithm implemented for the purpose of solving the WA problem in this thesis. The core of the chapter details the implemented WA process, emphasising the objective function, and the constraints of the WA model as well as the logic flow of the simulation performance evaluation framework in the context of WA.

6.1 Weapon Assignment Problem Formulations

The WA[1] problem was initially informally described by Flood [134] in 1957, after which the first mathematical formulation of the problem was presented by Manne [125] in 1958. This problem is a fundamental, well-studied problem in the field of defense-related operations research, and is generally seen as the "heavy weight" problem in this domain [81]. This problem is typically formulated as a non-linear integer programming problem.

Benaskeur *et al.* [21] argued that all the existing WA problem formulations make different simplifying assumptions in respect of the WA modelling paradigm. The different subsequent WA solution approaches also have different advantages and limitations, depending on the simplifying assumptions to make the problem more tractable. The majority of WA models consist of an adaptation of the well-known assignment problem, while abstracting important aspects of the WA problem such as forecasting and non-homogeneous WSs [22]. Benaskeur *et al.* [21] therefore argued that the complexity of current WA problems cannot be sufficiently described by virtue of the well-known assignment problem. They referred to the WA problem as the *Combat Power Management* (CPM) problem so as to account for all the generally ignored factors and complexities in the problem description.

Although the WA simulation framework put forward in this thesis may be used and interpreted in the context of the CPM problem, the problem is still referred to as *the WA problem*, since all the previous local TEWA-related work on which this thesis builds used this terminology. The problem solved in this thesis accounts for, among other factors, different types of WSs with explicitly defined SSHP volumes, range-restricted WSs as well as the notion of *Fire Orders* (FOs), which includes a flexible WA engagement scheduling element. All of these factors point toward a divergence from the classic assignment problem. Unfortunately, other complexities such as the uncertainty associated with target detection and SSHP values which depend on specific threat-WS combinations — all of which form part of the CPM problem — are not accounted for in the models implemented in this thesis. Such extensions are left for future work.

The WA problem may be categorised according to the WA paradigm adopted — the two problem-classes generally referred to in literature include *static* and *dynamic* formulations. WA can, however, further be classified according to whether a single objective or multiple objectives are pursued, as explained in §3.2. The scope of this thesis is limited to the single-objective case. For a more detailed explanation of these different types of WA models, the reader is referred to [119].

The overall goal of all these different WA formulations are, in fact, the same — ensuring that the DAs survive the GBAD scenario. To this end, model formulations may be subdivided according to two alternative views: (i) an *asset-based* formulation and (ii) a *target-based* (threat-based) formulation, as introduced in §5.3.3. In the first case, the overall goal is achieved by maximising the survivability of the DAs. In target-based formulations, on the other hand, the cumulative survival probability of the threats, weighted by their respective system threat values, is minimised. Recall from §5.3.3 that in a networked GBAD environment, asset-based formulations are preferred since different DAs require different levels of protection. Nevertheless, target-based formulations are more suitable when the intent of the threats are unknown or difficult to predict [50], as is the case in a typical GBAD environment. For this reason, the conversion of a target-based to asset-based formulation, as suggested by Du Toit [59, p.85], is

[1] The WA problem is often also referred to in more detail as the *Weapon Target Assignment* (WTA) problem. The WTA problem is, however, only referred to as the WA problem in this thesis in order to decrease the number of acronyms and thereby enhance readability. In recent literature, the problem has also been referred to as the *Combat Power Management* (CPM) problem [21].

adopted for WA purposes in this thesis, as detailed in §5.3.3.

Although the implemented problem is not, in essence, an instance of the classical assignment problem with a fixed set of constraints and modelling approach, various WA problem formulations are nevertheless reviewed. The rationale inherent to these mathematical models has inspired the formulation of the WA models adopted for implementation in this thesis.

6.1.1 Static Model Formulations

In the static version of the WA problem, the inputs to the problem are fixed and a solution is only calculated for a single instant or time-stage. The positions of all the threats are known in these formulations, as well as the positions of the WSs. The WSs are, in turn, allocated to threats for engagement during a single time-stage. The WA problem in this context therefore does not require the use of forecasting methods, thereby most likely reducing the effectiveness of the resulting solutions.

Various models have been proposed in the literature to represent this case of the WA problem [3, 111, 128]. A qualitative evaluation of the relative performances of the different WA models in this class was carried out by Potgieter [155], Du Toit [59] and Kok [104]. Based on the discussions contained in their theses, the formulations of the most promising static WA models are provided here.

The most basic mathematical model of the WA problem, and probably the most studied, is the *Static Weapon Assignment Problem* (SWAP) as formulated by Manne [125]. This assignment problem has been adapted in numerous ways to focus on different elements within the larger CPM problem. Some of these alternative formulations include the SWTA problem with uniform WSs, the SWTA problem with range-restricted weapons and the SWTA problem with multiple types of WSs [59, 215].

In the SWAP formulation, V_j denotes the priority of eliminating threat j, while p_{ij} is the SSHP associated with the threat j and WS i combination. Since a hit is assumed to be a kill in this formulation, the complement of p_{ij} is seen as the survival probability of threat j. The number of threats and WSs are denoted by n_t and n_w, respectively. The binary decision variable x_{ij} assumes the value 1 if WS i is assigned to threat j, or the value 0 otherwise. In the classical SWAP formulation, the objective is to

$$\text{minimise} \quad \sum_{j=1}^{n_t} V_j \prod_{i=1}^{n_w} (1 - p_{ij})^{x_{ij}} \tag{6.1}$$

$$\text{subject to the constraints} \quad \sum_{j=1}^{n_t} x_{ij} = 1, \qquad i = 1, \ldots, n_t, \tag{6.2}$$

$$x_{ij}(\tau) \in \{0, 1\}, \qquad i = 1, \ldots, n_t, \quad j = 1, \ldots, n_w. \tag{6.3}$$

Here constraint set (6.2) ensures that each threat is allocated exactly one WS while constraint set (6.3) ensures the binary nature of the decision variables. It is therefore assumed that all the WSs have to be assigned to engage threats and that there is no limitation on the number of WSs assigned to a specific threat, thereby decreasing the survivability of a high-priority threat. This formulation, together with the aforementioned alternative formulations, served as the foundation for the static WAM adopted in this thesis, as described later in this chapter.

6.1.2 Dynamic Model Formulations

Dynamic formulations of the WA problem are essentially multi-stage problems with feedback where the results of the current stage serve as inputs for solving the next stage. Temporal quantities are therefore incorporated into the constraints and objective function, thereby complicating the formation when compared to the static case. In the dynamic case, the most effective time-stage is determined from a number of possible future time-stages during which a WS is required to engage a threat. The outcome of the engagement is then assessed in order to determine the strategy for the next period. The dynamic nature of these models imply that the WA problem is solved for the current time-stage as well as for a pre-specified number of future time-stages, known as the *prediction time-frame*. This class of models therefore includes a scheduling element of when a WS should engage a threat in order to optimally utilise available resources (WSs and ammunition) in addition to the assignment component of static WAMs.

According to Hosein [81], dynamic formulations approximately double the effectiveness of WA models in terms of protecting assets. In Hosein's WAM formulation, the kill-probability of a weapon-threat pair depends solely on the asset to which the threat is directed. Under this simplifying assumption, the computational complexity of a solution algorithm may be reduced significantly while still maintaining a cost-performance advantage when compared to static problem formulations [81]. The effectiveness of a specific problem formulation is, however, strongly dependent on the problem type (*i.e.* the properties of the considered GBAD scenario and the size of the problem — number of threats, DAs and WSs) as well as the underlying assumptions. Because of a number of unaccounted factors, it is not possible to assume that the advantage of approximately doubled effectiveness will, in fact, translate to the real-world system for all scenarios.

The most basic form of the dynamic problem is the *Shoot-Look-Shoot* instance formulated by Hosein [80] in 1990. In practice, this formulation has been shown to increase effectiveness of protecting DAs by reducing the number of unnecessary rounds fired [155]. In this formulation, the outcome of assigning WSs to threats is perfectly observed and the resulting outcome is then used in order to solve the problem for the next stage. The formulation does, however, not include a scheduling element in the sense that a specific time-stage is suggested to the FCO for assigning a specific WS to a threat. As such, the formulation essentially entails solving the static problem instance for each time-stage; the only advantage being the incorporation of feedback (past events in the scenario are used in subsequent problem instances) and the improved effectiveness resulting from the perfect observation of a WA engagement result.

Two alternative dynamic formulations, which include a scheduling element, are the *Expected Threat Priority Accumulation* (EVA) model formulation by Du Toit [59] and the DWTA model with *Time Windows* (TWs) by Van der Merwe [215]. These model formulations are reviewed in the remainder of this section.

Expected Threat Priority Accumulation

In 2009, Du Toit [59] formulated a comprehensive dynamic WA problem. This mathematical model includes the traditional DWTA problem objective function of minimising the priority-weighted sum of the survival probabilities of aerial threats, subject to several temporal constraints so as to add more realism.

In Du Toit's formulation the time continuum is discretized into time-stages based on the duration of the TEWA-cycle time. During each stage, the number of threats together with their current and past positions, are known. In order to include a scheduling element, the future positions

of threats are determined by employing a flight path prediction model, as described in §4.2.4 which, in turn, introduces another element of uncertainty.

A particular problem investigated by Du Toit was how to incorporate preference for earlier engagements so that the solution does not always prefer later engagements during which the SSHP are usually higher (since the threats are typically nearing the WSs). Later engagements have a larger uncertainty associated with them because of the inaccuracy of the flight path prediction model and the postponement of aerial threat engagement is generally dangerous. A stage utility function[2] was suggested for use as a mitigation strategy for this purpose.

The type of stage utility function employed may, however, significantly impact on the proposed assignments (positively or negatively, depending on the GBAD scenario). Du Toit compared three different utility functions — a constant function, a linearly decreasing function and an exponentially decreasing function. The resulting evaluation concluded that the resulting assignments are strongly dependent on the particular stage utility function implemented. An alternative objective function (6.4) was also suggested, where the utilities of earlier engagements are scaled according to the number of prediction time-stages.

In Du Toit's model [59], $\Lambda_j(\bar{\tau})$ denotes the system threat value of threat j during time-stage $\bar{\tau}$, while $Q_{ij}(\tau)$ denotes the survival probability of threat j when engaged by WS i during time-stage τ. The number of threats and WSs during the first time interval are denoted by $n_t(\tau_\alpha)$ and $n_w(\tau_\alpha)$, respectively. The last time-stage interval is denoted by τ_ω while $\bar{\phi}_i$ denotes the number of initial intervals during which WS i is unavailable (*i.e.* its setup time). The decision variable $x_{ij}(\tau)$ assumes the same meaning as before with an added time-index. The objective in Du Toit's model is to

$$\text{minimise} \qquad \sum_{\bar{\tau}=\tau_\alpha}^{\tau_\omega} \sum_{j=1}^{n_t(\tau_\alpha)} \Lambda_j(\bar{\tau}) \prod_{\tau=\tau_\alpha}^{\bar{\tau}} \prod_{i=1}^{n_w(\tau_\alpha)} Q_{ij}(\tau)^{x_{ij}(\tau)} \tag{6.4}$$

subject to the constraints

$$\sum_{j=1}^{n_t(\tau_\alpha)} \sum_{\tau=\tau_\alpha}^{\tau_\alpha-1+\bar{\phi}_i} x_{ij}(\tau) = 0, \qquad i = 1,\ldots,n_w(\tau_\alpha), \quad \tau = \tau_\alpha,\ldots,\tau_\omega, \tag{6.5}$$

$$\sum_{j=1}^{n_t(\tau_\alpha)} \sum_{\tau=\tau_\alpha+\bar{\phi}_i}^{\tau_\omega} x_{ij}(\tau) \leq 1, \qquad j = 1,\ldots,n_t(\tau_\alpha), \quad \tau = \tau_\alpha,\ldots,\tau_\omega, \tag{6.6}$$

$$x_{ij}(\tau) \in \{0,1\}, \quad i = 1,\ldots,n_w(\tau_\alpha), \; j = 1,\ldots,n_t(\tau_\alpha), \; \tau = \tau_\alpha,\ldots,\tau_\omega. \tag{6.7}$$

Here, constraint set (6.5) ensures that no WS engagements are scheduled during the initial $\bar{\phi}_i$ intervals, while constraint set (6.6) ensures that a WS is only assigned once during the remaining TEWA-cycles. The final constraint set (6.7) ensures the binary nature of the decision variables.

[2] A stage utility function is a monotonically decreasing function that assigns greater importance (utility) to earlier engagements [59].

Weapon Assignment Problem with Time Windows

Van der Merwe [215] further expanded upon the family of dynamic models by incorporating the notion of a fire window. Her formulation was briefly introduced in Section 3.2.2, but is repeated here and expanded upon.

Van der Merwe [215] formulated the WA problem as a vehicle routing problem with time windows in 2013, where vehicles (analogous to WSs) have to deliver commodities (analogous to ammunition) to customers (analogous to threats) within a pre-specified time-frame (analogous to a fire window). This novel formulation enables the TEWA system to account for the weapon setup time which includes operator response time, reloading of ammunition, orientating the WS and the time required to compute the interception point(s) before engaging.

The weapon setup time typically varies from WS to WS and is analogous to the vehicle travel time in a vehicle routing context. Furthermore, the EEM (SSHP values) contains values that are analogous to the probability that a vehicle will successfully serve a customer during a first visit. Thus, in the formulation of Van der Merwe [215], the EEM is used in the objective function, instead of the weapon setup time. This formulation attempts to minimise the total expected survival probability of all the threats over the entire service window, again weighted by their respective threat value. The model generates a schedule of events (engagements) in an attempt to counter the threats as effectively as possible. A schedule is an indexed discrete-time interval list of decision instants with associated events. To solve this new problem formulation, a hybrid metaheuristic approach of simulated annealing and tabu-search was suggested.

In the model of Van der Merwe [215], d_i denotes the number of time-stages that WS i requires to "travel" between concequtive engagements. The value of d_i may therefore be seen as the weapon setup time of WS i as described above. The *First-Time-to-Fire* (FTTF) e_{ijk} and *Last-Time-to-Fire* (FTTF) ℓ_{ijk} is associated with WS i when assigned to engage threat j during the WS-threat pair's k^{th} fire window. The length of a WS-threat pair's k^{th} fire window — measured in time-stages — is denoted by s_{ijk}. The value of a parameter f_{ij} indicates the number of distinct fire windows for the considered WS-threat pair. Furthermore, p_{ijk} denotes the SSHP value[3] associated with fire window k of WS i and threat j.

The decision variable x_{ijk} assumes the value of 1 if threat j is engaged by WS i during the k^{th} fire window; otherwise the value of 0 is assigned to the decision variable. As in the vehicle routing problem with time windows, a binary axillary variable y_{ihj} is employed which serves as a "vehicle flow variable." This variable takes a value of 1 if threat h precedes threat j in a sequence of engagements by WS i, or a value of 0 otherwise. The objective is therefore to

$$\text{minimise} \quad \sum_{j=1}^{n_t} \Lambda_j \prod_{i=1}^{n_w} \prod_{k=1}^{f_{ij}} (q_{ijk})^{x_{ijk}} \tag{6.8}$$

[3]In order to efficiently quantify the SSHP value associated with a specific fire window is a complicated task, since the WS will hit a target during a singe time stage. It is therefore not suitable to use an average or median value as a merit value for a specific fire window in practice, as was used for demonstration purposes by Van der Merwe. A more accurate performance metric of a fire window will need to be investigated. This concern is, however, left for future work.

subject to the constraints

$$\sum_{i=1}^{n_w}\sum_{h=0}^{n_t+1} y_{ihj}(\tau) \leq \kappa, \qquad j = 1,\ldots,n_t, \tag{6.9}$$

$$\sum_{h=1}^{n_t+1} y_{i0h} = 1, \qquad i = 1,\ldots,n_w, \tag{6.10}$$

$$\sum_{h=1}^{n_t+1} y_{i,h,n+1} = 1, \qquad i = 1,\ldots,n_w, \tag{6.11}$$

$$\sum_{h=0}^{n_w+1} y_{ihj} - \sum_{h=0}^{n_w+1} y_{ijh} = 0, \qquad i = 1,\ldots,n_w,$$
$$j = 1,\ldots,n_t, \tag{6.12}$$

$$\sum_{k=1}^{f_{ij}} x_{ijk} = \sum_{k=1}^{f_{ihj}} y_{ihj}, \qquad i = 1,\ldots,n_w,$$
$$j = 1,\ldots,n_t, \tag{6.13}$$

$$\sum_{k=1}^{f_{ih}}(e_{ihk}+s_{ihk})x_{ihk} - \sum_{k=1}^{f_{ij}} e_{ijk}x_{ijk} \leq (1-y_{ihj})M, \qquad i = 1,\ldots,n_t,$$
$$j = 1,\ldots,n_t, \tag{6.14}$$

$$j = 1,\ldots,n_t,$$

$$\sum_{j=1}^{n_t}\sum_{k=1}^{f_{ij}} x_{ijk} \leq A_i, \qquad i = 1,\ldots,n_t, \tag{6.15}$$

$$y_{ihj} \in \{0,1\}, \qquad i = 1,\ldots,n_w,$$
$$j = 1,\ldots,n_t, \tag{6.16}$$
$$h = 1,\ldots,n_t,$$

$$x_{ijk} \in \{0,1\}, \qquad i = 1,\ldots,n_w,$$
$$j = 1,\ldots,n_t, \tag{6.17}$$
$$k = 1,\ldots,n_t.$$

In this formulation, constraint set (6.9) ensures that a maximum of κ WSs are assigned to engage any threat during the forecasting time-frame. Constraint set (6.10), on the other hand, ensures that WS i leaves its idle state (denoted by 0) exactly once if it is assigned to engage threats. Similarly, constraint set (6.11) ensures, if a threat is engaged by WS i, that the WS returns to its idle state to be available again afterwards. Constraint set (6.12) ensures, that if a threat it engaged by a WS, that the WS "leaves the threat" in order to be available for other assignment again later. Constraint set (6.13) ensures, if threat j precedes threat h when engaged by WS i, that threat j is engaged during exactly one fire window. If, however, threat h is engaged exactly before threat j, constraint set (6.14) ensures that no conflicts are present during the current time scheduling horizon — the time-stage during which the engagement of threat h starts plus the time it takes to engage threat h plus the time it takes to transition between threats h and j, does not overlap with the start of the engagement of threat j. The constant M in (6.14) denotes a large number. Constraint set (6.14) ensures that the engagement of threat j by WS i is, indeed, within a possible fire window. Constraint set (6.15) ensures that the capacity (ammunition supply) of WS i is not exceeded. Finally, constraint sets (6.16)–(6.17) ensure the binary nature of the axillary and decision variables.

As may be seen from the above explanation, this formulation is highly complex and for the purposes of practical implementation, there are still some unresolved concerns when implementing the formulation within a discrete-event simulation. Nevertheless, the ideas and constraints of this formulation inspired the dynamic WA model implemented in this thesis. Two of the main reasons why this exact formulation was not implemented are detailed below.

The dynamic case of the WA problem is highly complex and, as suggested by [59, 80], the formulation should therefore be kept as simple as possible. The WA problem with fire windows is a reformulation of the vehicle routing problem with time windows where the problem is generally solved for a fixed set of problem parameters. In the formulation of Van der Merwe [215], the "service window" was seen as the total scenario time for illustrative purposes, whereas, in the case of a discrete WA simulation, the problem needs to be solved during each TEWA-cycle separately. As such, a number of unforeseen complexities arise during implementation. The model formulation of Van der Merwe [215] does not account for changing numbers of threats (or WSs) during the scenario. There is, as such, no apparent incorporation of feedback within the formulation. If, however, the formulation is adjusted to include this, the algorithmic complexity of solving the model is expected to increase significantly, thereby increasing implementation complexity as well as increasing the computational resources required to solve the problem within an acceptable timeframe. For each TEWA cycle, the fire windows need to be re-calculated and the problem re-solved. The problem may, however, be solved again before the fire windows have passed, as such, a form of memory has to be incorporated to keep track of engagement times and prevent conflicts during the assignment of future forecasts — all of which must be accounted for. This increased complexity is generally accompanied by an increased time to solve the problem and also, increased risks of failure. The value of the increased complexity is not justifiable in the opinion of the author.

The notion of fire windows is the main advantage of Van der Merwe's formulation when compared to other dynamic formulations. When testing this formulation in a simulation environment where automatic decision making is employed, as opposed to semi-automatic decision making, the advantage diminishes. To illustrate, during every TEWA-cycle when the problem is solved, an engagement fire window is generated for each available WS. These fire windows are defined by an FTTF and an LTTF. Consequently, the FCO has to determine a specific (single) timestage during a possible fire window to engage threats. The fire windows essentially constitute all the possible stages during which a FCO may assign a WS to a threat. The threats are, as such, still effectively engaged during a single time-stage although the output is a fire window which consists of numerous timestages. The main advantage of this formulation is therefore the increased freedom of choice to the FCO. This freedom of choice advantage is not present during an automatic decision making process, where the TEWA system decides on the best engagement TEWA-cycle. The increased complexity resulting from the use of fire windows in order to schedule engagements is therefore also not the preferred method for an automatic decision making TEWA simulation.

6.2 WA Solution Approaches Comparison

The main local research into possible solution approaches for dynamic and static WA problem formulations was conducted by Potgieter [155] and Du Toit [59]. Potgieter mainly focussed on rule-based heuristics, whereas Du Toit experimented with more computationally expensive approaches and did some benchmarking in respect of the evaluation of different solution approaches. Kok [104] also suggested the use of a so-called random assignment approach and considered exhaustive enumeration for finding lower-bound and upper-bound solutions.

According to Hossein [80], there are several properties that complicate the exploration of the solution space of a WAM. These factors should be understood before selecting a suitable solution strategy, and include the following considerations:

- Simple versions of the WA problem have been shown to be *NP-complete*[4]. As such, there exist no efficient methods to solve these problems optimally; complete enumeration of all possible allocations (either explicitly of implicitly) is the only method for finding optimal solutions with certainty [80].
- The objective functions in all the classes of WA problems are *non-linear*. Consequently, well-known and thoroughly researched linear algorithms, such as the simplex method, cannot be used to solve these problems. The system designer typically has to resort to heuristic methods.
- WA problem formulations are typically *discrete* in the sense that only *binary* assignments (decision variables) are allowed in a feasible solution. In general, these classes of integer problems are hard to solve optimally compared to their continuous real-interval counterparts.
- The WA problem is finally *stochastic* in the sense that kill-probabilities and track generations have probabilities associated with them. This non-deterministic nature increases the complexity of evaluating WAM solution methodologies objectively.

These properties exemplify that finding optimal solutions in real-time to dynamic (or even static) WA problem formulations is a very hard problem, and effectively rule out any possibility of developing an efficient, exact solution method. In 2007, however, Ahuja *et al.* [2] considered a static instance of the WA problem and developed an exact solution approach. The structure of the WA problem was used to develop different lower-bounding schemes. By utilising these lower-bounding schemes within a branch-and-bound paradigm, they obtained exact results in a small time-frame — approximately 0.187 seconds for 100 WSs and 100 threats — by employing their so-called construction heuristic [2]. The static problem instance may therefore be considered as a well-studied problem which can be solved in real-time when using the right solution methods. The dynamic instance, on the other hand, does not currently benefit from the same level of study.

The fundamental difference between the WA problem and related optimisation problems in the literature further complicates matters with respect to finding an effective solution approach for dynamic WA problem formulations. There are, for example, publicly available benchmark problems for well-established problems such as the *Travelling Salesman Problem* (TSP) [154], but this is not the case for the WA problem. The aim in the TSP is the same for everyone and, as such, researchers have proposed various well-defined scenarios to be used for the comparison of solution approaches (algorithms). For the WA problem, on the other hand, different scenarios with different assumptions are generally used, thereby preventing the objective comparison of results emanating from different studies. Unsurprisingly, there does not exist any benchmark WA repositories as are available for the TSP [205]. It is therefore difficult to select the best solution approach for WA problems.

In the remainder of this section, previous work related to solving WAMs completed during the period 2006–2010 by TEWA students at Stellenbosch University is briefly summarised and updated by incorporating the relevant literature. Numerous solution approaches are reviewed and

[4]For a definition of the notion of NP-completeness, the reader is referred to [112].

qualitatively evaluated. The section concludes with a motivation as to why a *genetic algorithm* was chosen as the preferred solution approach for the static as well as dynamic WA problem in this thesis.

6.2.1 Exhaustive Enumeration

According to the exhaustive enumeration approach, an optimisation problem is solved by sequentially evaluating the objective function value for all feasible WS-threat pairs in order to determine the best solution. Evidently, the objective function value obtained by this method, if the search is completed, indicates a globally optimal solution. This method is also sometimes referred to as a "brute force" or "exhaustive search" approach [104].

Because of the tremendous computational requirements of this method and the time constraints typically prevalent in a GBAD environment, this method would not be suitable for real-time application in a TEWA context. If this method were to be applied, the time constraint could cause the solution process to exit prematurely, thereby only proposing low-quality feasible solutions, while other solution approaches would provide better results.

In small scenarios, however, it may be possible to use this technique to find an optimal solution in real time. Unfortunately, the computational requirements — and correspondingly the time required — increases exponentially as the problem size increases [104]. The exhaustive enumeration method should nevertheless be considered as a possible method to compute optimal solutions for less expensive problems in order to compare the quality of other solution approaches.

6.2.2 Branch-and-Bound

Besides the explicit exhaustive enumeration approach, the well-known (implicit) branch-and-bound enumeration [109] method may also be applied to WA problem formulations in order to find globally optimal solutions [166]. The branch-and-bound method renders unnecessary various enumeration steps (branches), thereby not requiring the enumeration of all possible solutions and consequently improving on the computational efficiency of the method of exhaustive enumeration described above. Since the size of the branch-and-bound search tree typically grows exponentially as a function of the number of decision variables, the standard form of this method is also limited by being applicable to small problem instances only if small computational budgets are available.

In 2007, Ahuja *et al.* [2] proposed the first implicit enumeration algorithm for solving moderately sized instances of the WA problem optimally. Their solution approach was introduced in §6.2. Van der Merwe [215] also implemented a branch-and-bound solution approach in order to solve a single, simplified case of the DWTA-TWs optimally — no feedback information or historical information were, however, incorporated into the solution methodology. This method may be seen as a good alternative approach for finding optimal solutions to the WA problem. An effective lower-bounding scheme is, however, essential to ensure efficient execution of this approach [215].

6.2.3 Random Assignment Approach

The random assignment approach attempts to simulate a total C2 breakdown where no coordination between WSs is possible. This random assignment approach was introduced by Olwell and Washburn [144]. The method functions under the assumption that WSs have entered a

"fire-at-will" status and the situation therefore effectively constitutes a random assignment of WSs to threats. The subset of available WSs still depends on whether the various WSs are capable of engaging a threat (*i.e.* SSHP constraints, line-of-sight constraints and ammunition availability are adhered to).

This method was suggested to be used as a lower-bound for WA algorithmic evaluation purposes. Following this approach, however, has its drawbacks. Since random assignments are made, there is no certainty regarding the quality of the random solution. A possible solution would be to execute numerous iterations in a bid to determine the smallest objective function value. It is possible (although unlikely) that a solution thus generated is actually a good one. For a TEWA DSS, it is more suitable to use an operator's suggestions as a lower-bound in order to determine whether the TEWA DSS does, indeed, improve upon the operator's decisions. Nonetheless, the random assignment approach method is still suitable for evaluation of the effects of C2 breakdown on the system's performance. Because of the random assignment approach, it is, however, important to note that the results will then depend on the specific GBAD setup (positions and number of DAs and WSs) and not necessarily on the TEWA algorithms.

6.2.4 Greedy Rule-based Algorithms

As mentioned, the prototype WA static problem is known to be NP-complete and, as such, is difficult to solve using exact methods. In contrast, most current TEWA systems likely function on a target-by-target basis using some form of greedy algorithm. A greedy algorithm typically generates a feasible solution to a combinatorial optimisation problem by making a locally optimal choice at each iteration during its execution with the hope of eventually finding a good solution.

In small problem instances, there may be a fair probability of finding optimal solutions. A greedy algorithm is, however, not guaranteed to find optimal solutions and often fails dismally in multi-threat scenarios [204]. In large problem instances, on the other hand, the large search space increases the likelihood of not finding a close-to-optimal solution [155]. The main advantage of a greedy heuristic solution approach is that it is typically very fast and easy to implement. Because of the time constraint coupled with the complexity of the WA problem, heuristic algorithms are generally preferred over exact ones [30] in the context of a TEWA system.

According to Johannson [93], greedy algorithms are best for large-scenarios (*e.g.* larger than ten threats and ten WSs) with a strong defense (*i.e.* a large WS to incoming threats ratio) [96]. It is, however, difficult to predict whether this observation will remain valid if a different modelling approach is applied. The modelling of threats, WSs and DAs may significantly alter the outcome of such a statement.

A popular variation of the greedy algorithm — the maximum marginal return algorithm — is known to return an optimal solution if all the WSs are identical. The algorithm functions by iteratively assigning a WS that achieves maximum improvement in the objective function value. A pseudocode description of this greedy approach is provided by Algorithm 6.1.

6.2.5 Metaheuristic Solution Approaches

From the discussion in Section 3.3.2, it is clear that various metaheuristic algorithms have been proposed for solving the different WAM formulations. The suggested algorithms include particle swarm optimisation [25, 39], ant colony optimisation [40], a large scale neighbourhood search algorithm [2] and different genetic algorithms [111]. Comparatively, the genetic algorithms and

Algorithm 6.1: Maximum marginal return algorithm.

Input : WS-threat pair SSHP values P_{wt}, and system threat values Λ_t
Output: WS assignments to threats

```
1  Initialise variables;
2  for w ← 1 to ‖W‖ do
3      HighestValue ← −∞;
4      AllocatedTarget ← 0;
5      Allocate WS so as to maximise reduction of threat priority;
6      for t ← 1 to ‖T‖ do
7          Value ← Λ_t × P_wt;
8          if Value > HighestValue then
9              HighestValue ← Value;
10             AllocatedTarget ← t;
11 Assign each WS w to their allocated threat t.
```

the particle swarm optimisation algorithms seem to excel in obtaining reliable solutions rapidly in the context of all forms of current WA models in the open literature.

Genetic algorithms are mainly used to solve more complex instances of the WA problem (*i.e.* instances containing more WSs and threats), but also provide near-optimal solutions to smaller problem instances. For smaller problem sizes, however, greedy rule-based algorithms are generally preferred [96]. Furthermore, a multi-objective, static WA problem formulation has been solved using the NSGA-II multi-objective genetic algorithm by Lötter and Van Vuuren [117] within a feasible timeframe. Because of these reasons, together with the motivation in §6.2, the single-objective, dynamic WA problem is also solved in this thesis using a genetic algorithm.

6.3 Genetic Algorithmic Implementation

A genetic algorithm is a method that may be used to solve unconstrained or constrained optimisation problems. The solution methodology was inspired by Darwin's theory of evolution [124], or more specifically, the key mechanism of evolution — natural selection.

This metaheuristic begins with a set of solutions which is collectively referred to as a population. The algorithm functions by modifying individual solutions from the current population and uses them as parents to produce children for the next generation. The parent solutions contribute their *genes* (the values of the decision variable vector) to their children. Generally, the algorithm selects parents that have better fitness values when forming the next generation — similar to the notion of natural selection where only the strongest individuals survive.

The genetic algorithm implemented in this thesis creates three types of children namely, *elite children*, *crossover children* and *mutated children*. The underlying mechanisms whereby these children are created are illustrated in Figure 6.1. Elite children are the candidate solutions with the best fitness values. These solutions are carried over to the next generation without any changes to the individuals. Crossover children, on the other hand, are created by combining certain segments of the binary decision variable vector from a pair of parents. Lastly, mutation children are created by introducing random changes (mutations) to a single parent. These mutations are achieved by inverting certain values at random positions of the binary decision variable vector.

(a) *Elite child* (b) *Crossover child* (c) *Mutation child*

FIGURE 6.1: *Three types of children used to populate the next generation in the generic algorithm.*

In the search space, the notion of elite children allows the algorithm to focus on areas in the search space which contain good solutions, whereas the incorporation of crossover combines the genes of parents in the hope of obtaining potentially superior children. To ensure that the algorithm does not converge prematurely to a weak local optimum, mutation adds diversity to the population, thereby improving the thoroughness of the search and increasing the chances of finding a global optimum. MATLAB utilises the so-called MI-LXPM real-coded genetic algorithm developed by Deep *et al.* [50] for solving integer problems. The reader is referred to [50] for a detailed description of the use of the crossover and mutation parameters. The selection process of a new generation is outlined follows:

1. A fitness score of each of the members of the current population is determined in order to rank them.

2. This raw fitness score is then scaled in order to convert it to a usable range of values for the purposes of selecting parents from whom children are produced.

3. The number of *elite children* is specified by the user. The top n solutions (where n is the specified elite number), based on fitness score value, are directly passed over to the next generation.

4. The number of elite children is subtracted from the initial population size. This new population size is used to determine the number of crossover children required, which is specified in terms of a percentage of this new population size.

5. After generating the crossover children, the rest of the population is filled by generating mutation children.

6. This new population of children then forms the next generation. Infeasible solutions are penalised by employing a penalising function and may therefore form part of the next generation, but are relatively unlikely to be selected for crossover (compared to feasible solutions).

The extent to which the mechanisms depicted in Figure 6.1 are utilised to form new populations, is a very important, problem-specific consideration. Careful consideration of these parameters is therefore crucial to ensure that good solutions are obtained in a limited time frame.

In order to determine a good combination of these genetic algorithm parameters, an experiment was conducted to determine the sensitivity of the simulation outcome with respect to the aforementioned population parameters. Since there are an infinite number of possibilities, a decision had to be made regarding the selection of a few parameter value combinations which could then be tested further.

Since the implemented simulation needs to adapt to changing scenarios, such as changes in the problem scale, it was decided to select the population size as a function of the number of decision variables. A larger population will generally improve solution accuracy, but increase computational time up to a certain point which depends on the problem structure [165]. It was therefore decided to select a population size that is sufficiently large. After testing various combinations, and gaining some intuition in respect of their effects on the speed and quality of solutions, the population size $\mathcal{P}$ is represented as

$$\mathcal{P} = \left\lceil \|x\| \times 1.2 \right\rceil,$$

where $\|x\|$ denotes the number of decision variables which, in turn, is the product of the number of threats, WSs and the length of the forecasting timeframe. The population size is therefore effectively 20% more than the number of decision variables.

Based on an analysis by Mathworks [126], a crossover percentage of 80–90% seems suitable for the majority of problem types. It was therefore decided to test crossover percentages of 80% and 90%. The elite count was taken as a default value of 10% of the population, but it was decided to test the effect that increasing and decreasing the elite-count has on the solution quality. Three different elite-count values were therefore tested — 5%, 10% and 15% of the population. The different combinations of population parameter values considered are listed in Table 6.1.

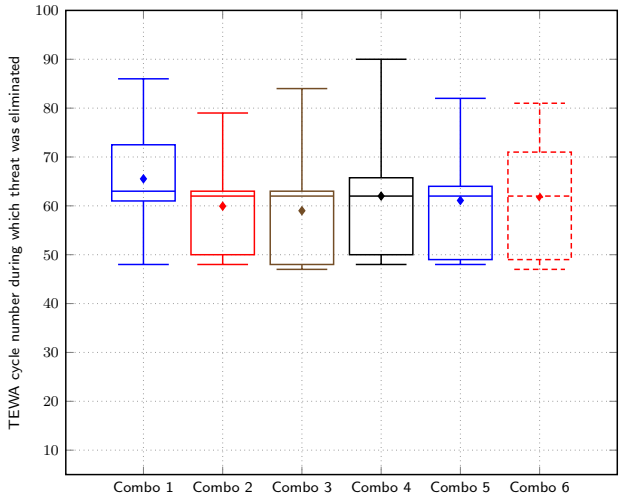
TABLE 6.1: *Different tested combinations of the genetic algorithm population parameter values.*

	Elite count	Crossover %
Combo 1	$\lceil 0.05 \times \mathcal{P} \rceil$	80
Combo 2	$\lceil 0.10 \times \mathcal{P} \rceil$	80
Combo 3	$\lceil 0.15 \times \mathcal{P} \rceil$	80
Combo 4	$\lceil 0.05 \times \mathcal{P} \rceil$	90
Combo 5	$\lceil 0.10 \times \mathcal{P} \rceil$	90
Combo 6	$\lceil 0.15 \times \mathcal{P} \rceil$	90

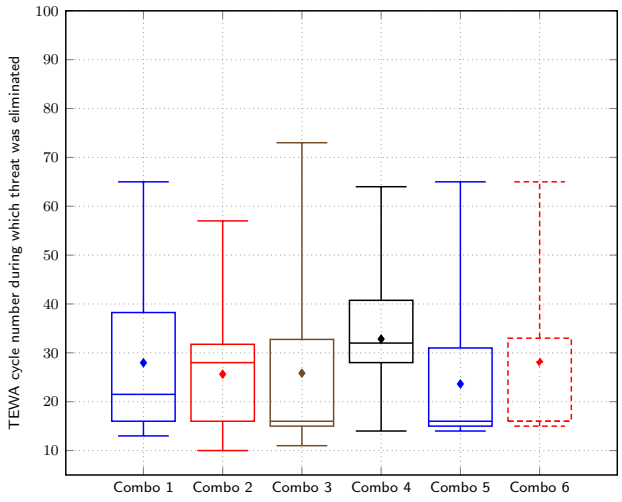
In order to assess the performance of these different population selection parameter value combinations, the dynamic WA problem was solved separately for these combinations in the context of the scenario introduced in §4.6. The outcomes of 30 scenario runs — and more specifically how early threats were successfully engaged — was used as the main performance criterion for this test. It was not possible to use the fitness function value for this purpose, since the problem changes continuously from one simulation run to the next because of all the stochastic elements. It would therefore be intractable to compare the fitness value for different simulation runs. The problem was solved for each iteration and the results used as inputs for the next computation cycle, as described in §6.4.1. The results of the analysis are visualised by the box-and-whisker plots in Figure 6.2.

From the plots in Figure 6.2 it is clear that there is a large variability in the results. This variability may be attributed to all the stochastic elements in the simulation — the random spheres for the generation of flight paths, the heuristic WA algorithm and — probably the largest contributor — the SSHP values of the WS-threat combinations. The outcome of each engagement was determined according to a uniform random distribution, as explained in §4.2.3. All these stochastic elements collectively contribute to the variability of the simulation results. It is possible that the simulation results are therefore, in fact, less sensitive to small changes in the genetic algorithm population selection parameters.

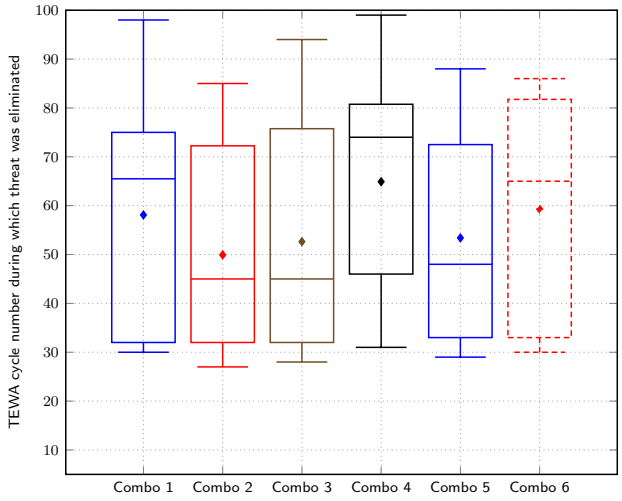
Combo 2 in Table 6.1 achieves the smallest spread when compared to the other combinations, for all the threats. Also, combo 2 generally achieves the earliest successful engagement for the



(a) *Threat 1*



(b) *Threat 2*



(c) *Threat 3*

FIGURE 6.2: *Box-and-whisker plot comparison of the different population generation parameter combinations for each threat. The horizontal axes denote the combination of parameters indicated in Table 3, while the vertical axes denote the time-interval during which the threat was successfully engaged and, correspondingly, destroyed.*

three threats. There is, however, not enough information available to ascertain with confidence that combination two is, indeed, always better than the alternatives. For this thesis, however, combo 2 was used in order to illustrate the functioning of the different algorithms and is deemed an adequate combination of parameter values for a proof of concept study such as this where the end goal is to illustrate the working of the TEWA algorithms of Chapter 5.

Furthermore, it is possible, since a relatively large population is used in the genetic algorithm, that the solution quality is less dependent on the other population parameters. If a smaller population is selected in order to adhere to restrictions placed on computational time, new population parameters will have to be selected. The ideal combination of these parameters is very problem-specific (*i.e.* the particular objective function and constraints adopted) and also depends on the instance size of the specific problem (*i.e.* the number of WS, threats and the forecasting time length) that has to be solved. If a metaheuristic is used to solve the WA problem in real-time, the properties of the metaheuristic should be tailored to the specific problem instance, and even updated in real time.

6.4 WA Model Implementations

Two WAMs were implemented in this thesis, of which only two formed part of the final simulation. The two models are an improved SWAP and a modification of the comprehensive dynamic model developed by Du Toit [59].

The implemented WA process functions as follows: The WA process utilises the system threat values obtained from the TE subsystem, as explained in Chapter 5. During every TEWA cycle the status of the GBAD scenario (threats destroyed, DAs destroyed, ammunition expenditure and WSs reloading) is updated. The status of the GBAD scenario forms the input to the subsequent iteration of computing. As such, the WAM formulation has to account for the possible changes of states of the simulation entities — *i.e.* introducing a feedback element and the notion of memory. The problem therefore has an increased complexity when compared to other similar (non air-defense) scheduling/assignment problems. The working of these implementations are described in some detail in the remainder of this section.

6.4.1 Implemented WA Models

The simulation model entities are represented in the simulation as explained in §4.2. How these models interact to achieve the required simulation performance evaluation functionality is detailed in this section. The results of the static and dynamic instances were compared in order to ascertain that the algorithms function as expected and all the implementation bugs have been corrected (verification). Only the dynamic implementation is described here since the static case is, essentially, a simplified version of the dynamic case. Both formulations have the same constraints, with the exception of the notion of a *Fire Order* (FO) which is lacking for the static case.

The Objective Function

Information required by the WA subsystem includes the data structures of the various WSs, as described in §4.2.3. The data structures of the threats, which indicate their positions throughout the scenario, and their WRLs, are also required. The SSHP values calculated by the flight-path prediction model, together with the current system threat value, are used to minimise the

accumulated survival probabilities of all the threats, weighted by their respective threat values. The probability of survival of a specific threat is calculated as the product of the complement of the SSHPs of the WSs assigned to the threat. The assumption is made that the events of a threat surviving multiple WS engagements are, indeed, independent events. The objective is therefore to

$$\text{minimise} \quad \sum_{t=1}^{n_t} \Lambda_j \sum_{\tau=1}^{F} \acute{c}(\tau) \prod_{w=1}^{n_w} Q_{tw}(\tau)^{x_{tw\tau}}, \tag{6.18}$$

where F denotes the forecasting time-window length. The variables n_t and n_w denote respectively the number of threats and WSs during the start of the current TEWA-cycle, as before. The probability of survival of threat t, when engaged by WS w, is denoted by Q_{tw}. Since a hit is modelled as a kill, as described in §4.2.3, the survivability of the threats may be seen as the complement of the considered WS-threat pair's SSHP value. The decision variable $x_{tw\tau}$ is a binary vector which assumes the value 1 if WS w is scheduled to engage threat t, τ time stages from the current TEWA-cycle, or the value 0 otherwise. The rest of the variables have the same meanings as before. The stage utility function $c(\tau)$ is given by

$$c(\tau) = a\mathrm{e}^{-bx}, \tag{6.19}$$

where the values of the coefficients a and b are functions of the forecasting time frame F and the required shape of the utility function. This stage utility function was deemed superior to the expected threat priority approach in (6.4), since it provides more flexibility to be tailored to the intuition of the operators. The expected threat priority objective function of Du Toit [59] did not provide any significant advantage when compared to a correctly calibrated stage utility function.

The normalised values $\acute{c}(\tau)$ are, in turn, utilised in objective function (6.18). This normalisation is achieved by dividing the weights for all the time stages in $c(\tau)$ by the total summed weight. This normalised weight then serves as the utility value. An exponential function was used to model the importance rating of selecting earlier engagements, since the relative utility may be seen as decaying exponentially for each interval. This is because later engagements are associated with numerous risks, such as increased uncertainty because of the implemented flight-path prediction model and delaying engagements unnecessarily, thereby providing the threats with more opportunity to engage the DAs. An exponential decaying utility function is therefore seen as more suitable than a linear decaying function.

Constraint Modelling

The WA problem is solved for each iteration by updating the states of the simulation elements — unavailable WSs, expended ammunition and destroyed threats. The forecasting time length during which WSs may be assigned is set by the operators. For each new iteration, the different distances between WSs and threats are recalculated and new corresponding SSHP values are determined. During each iteration, the solution must adhere to the following constraints:

1. A WS may be assigned a pre-specified maximum number of times during the scheduling horizon. This value is fixed beforehand for the scenario.

2. A threat may only be engaged a pre-specified maximum number of times by WSs over the entire scheduling horizon. This is required in order to avoid circumstances where

resources are exploited during the current time stage, thereby preventing the WSs to adapt to changes in the environment (*i.e.* newly detected threats, threats entering within range of the WSs) because of reloading and/or ammunition constraints — in essence, preventing the occurrence of a so-called overkill[5] event.

3. Each time a WS is allocated to a threat, regardless of the success of the engagement, a weapon setup time counter is started. This weapon setup time is WS-specific, and accounts for factors such as operator response time, reloading of ammunition, orientating the WS and, in some circumstances, the time required to compute the interception points before engaging. No new WS engagements are possible until this weapon setup time for the considered WS has passed.

4. Each WS also has a pre-specified amount of ammunition available. An engagement is therefore only possible if the ammunition supply of the WS under consideration has not been depleted.

5. There is a minimum SSHP requirement before a WS can engage a threat. This value therefore serves as a necessary prerequisite for engagement, and adherence to this prerequisite (or otherwise) has to be determined prior to assigning a WS to a threat. The SSHP value that is used to determine whether this constraint is satisfied, is calculated in conjunction with the flight-path prediction model as explained in §4.2.4. In order to determine the outcome of an engagement, however, the current SSHP value is recalculated, since the "actual" SSHP value will most probably differ from the predicted value.

6. A threat may not be engaged if it has already been destroyed.

The notion of an FO was used to account for constraints (3)–(6) above. An FO is unique for each WS and indicates which threat, and during which time stage, a WS should engage. A binary flag is also set for the specific WS to signify that the WS has an active FO. Constraints (1)–(2) above, on the contrary, do not require any feedback information and are therefore "fixed" throughout the scenario. Since scheduling is included in the modelling approach, the simulation has to, effectively, delay an engagement until the right time stage. When an FO is set for a specific WS, the WS may not receive a new FO until the current FO has been executed. The solutions obtained from running the optimisation routine (*i.e.* determining the decision variable values) is used so as to generate FOs.

Furthermore, the inclusion of an FO allows the solution to change from one iteration to the next if a significantly improving solution is found before the current FO is executed. This functionality may be achieved by incorporating a minimum fitness function threshold that must be exceeded before an improving solution is selected. For example, if 5% is set as the threshold, an FO is only cancelled and a new one selected if the fitness function increases by 5% or more. This work is, however, left for future performance evaluation studies since the selection of the tolerance value may significantly impact the outcome of a scenario thereby further complicating the identification and interpretation of meaningful results.

The outcome of a WS engagement is determined by using a uniform random distribution on the interval $[0, 1]$ and comparing this value to the SSHP of the WS-threat pair at the current time stage. Recall, from §4.2.3, that the SSHP only depends on the threat-WS stand-off distance. If the generated random number is indeed smaller than the SSHP value, the engagement is deemed

[5]Overkill, in a military context, refers to the use of excessive force that goes further than what is necessary to achieve a specific goal.

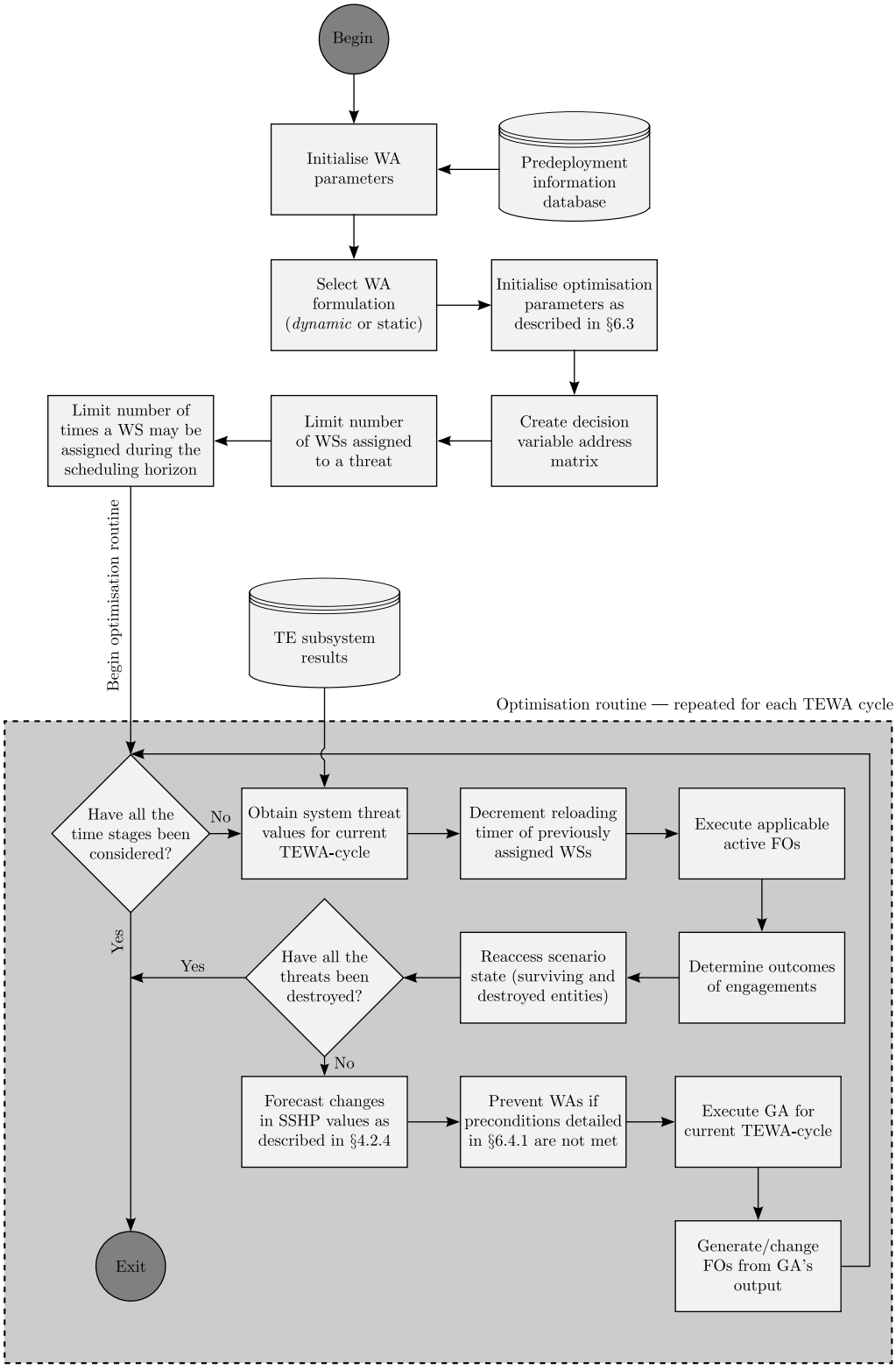


FIGURE 6.3: *Logic flow of the WA process in the simulation performance evaluation framework developed in this thesis.*

as having been successful and the threat destroyed; otherwise, the threat is considered to have survived.

Threat values are assumed to remain constant for the duration of the prediction time frame. The inclusion of a changing threat value will add more uncertainty to the problem and may, if incorrectly predicted, have a detrimental effect on the scenario's outcome. The increased complexity would therefore be difficult to justify operationally. For short prediction time frames, however, the threat values will not change significantly — the ranking of threats according to threat values should not change and, correspondingly, when using the threat values in a comparative manner, have a minimal impact on the WA output.

This process whereby the objective function is optimised is repeated until either all threats are destroyed, all DAs are destroyed (threats have reached reached their WRL) or the threat paths have ended.

6.4.2 WA Simulation Architecture

Because of the implemented approach, there are many feedback elements present in the WA sub-routine. An effective way to visualise these elements is through the use of a flow diagram. The logic-flow of the reactive WA modelling approach is visualised in Figure 6.3.

6.5 Chapter Summary

This chapter opened in §6.1 with an overview of the different WA problem formulations in the literature. More detail was provided in respect of the working of static WAM formulations (in §6.1.1) and dynamic WAM formulations (in §6.1.2).

Different methods for solving these WAMs were considered in §6.2. The chapter closed in §6.3 with a motivation and an overview of the genetic algorithm implemented in this thesis. After having gained an understanding the characteristics of the different WA formulations, the dynamic WA process implemented in this thesis was introduced in §6.4 with specific reference to the objective function, constraints and simulation framework adopted.

CHAPTER 7

Human-Machine Interface Design

A good solution applied with vigour now is better than a perfect solution applied ten minutes later.

— George Smith Patton

Contents

In a GBAD environment, the operators who are responsible for evaluating threats and assigning appropriate WSs to them need to act decisively within tight time schedules and often with incomplete and uncertain information. Within this domain, it is therefore crucial to ensure that the information provided to the operators do, indeed, aid them during their decision-making cycle and not cause a condition of "paralysis by analysis." The goal of this chapter is therefore to investigate the requirements that must be adhered to in order to effectively convey the analytical results from the TEWA system to the operators, thereby providing a high-level operational understanding to the human decision makers.

The chapter opens with an overview of the relationship between a TEWA DSS and data fusion, after which the notion of decision support in a GBAD environment is described. The complexities associated with providing germane decision support in a TEWA context is also explained. This is followed by a break-down of the TADMUS DSS, and several design guidelines — which forms the core of the chapter — are subsequently provided for the detailed design of a HMI. The chapter closes with an overview of MATLAB's HMI design capabilities and an HMI example is finally provided.

7.1 Data Fusion within the Decision Support System

With the volume and complexity of information brought upon by the information age (see §2.1), it is becoming increasingly important to be able to distinguish background noise from valuable insights [27]. Instead of enhancing an operator's ability to make accurate and timely decisions, this abundance of information threatens to diminish the decision maker's control over the situation. This is something that should be avoided at all costs, especially in a military environment where inaccurate decisions can cost lives. As such, one of the challenges of the information age is how to effectively integrate the human and the computer into an information-processing system which produces "good" decisions as outputs.

As mentioned, the tasks of evaluating aerial threats and assigning ground-based weapon resources to counter these threats in a GBAD environment, are the responsibility of a *Fire Control Officer* (FCO). In order to succeed in this highly dynamic, information-rich and complex environment, the FCO has to make effective use of a DSS which alleviates the strain on him by reducing the influx of information to manageable, meaningful levels. Generally, a TEWA DSS is employed by the FCO to aid him with the processes of TE and WA. DSSs are information systems responsible for assisting users in complex decision-making tasks [181]. Generally, this goal of assisting operators is achieved by identifying possible courses of action and by means of trade-off analysis, providing recommendations. A TEWA DSS is, in essence, typically a hybrid DSS and expert system, because it usually employs both computerised analytical methods (TE and WA algorithms) and heuristic rules[1] to aid the decision-making processes of the operator.

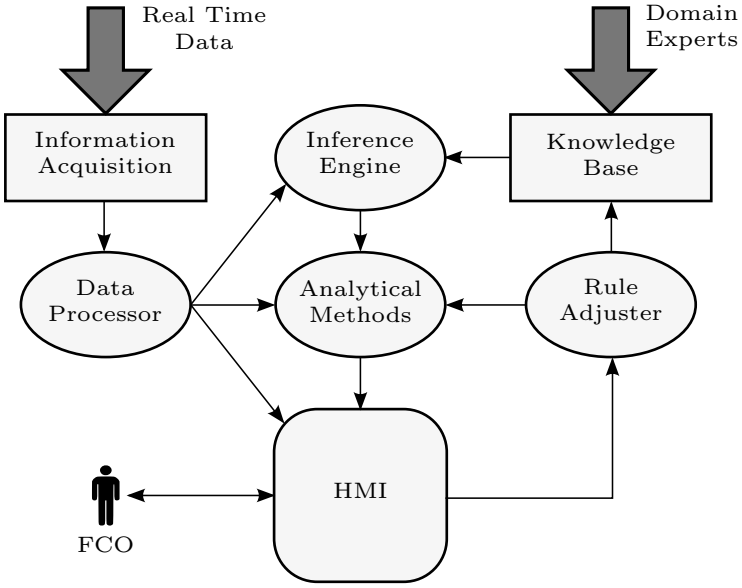


FIGURE 7.1: *Functional architecture of a typical TEWA DSS.*

The functional architecture of a typical TEWA DSS is depicted in Figure 7.1. At the heart of every DSS is multi-sensor data fusion. This figure is therefore interpreted in this section with reference to the various levels of information in the well-known (updated) *Joint Director of Laboratories* (JDL) II data fusion model [191]. According to the JDL model, data fusion may be defined as,

[1]Heuristic rules are typically developed through experience, intuition and judgement [90].

> *"... a process dealing with the association, correlation, and combination of data and information from single and multiple sources to achieve refined position and identity estimates for observed entities, and to achieve complete and timely assessments of situations and threats, and their significance. The process is characterized by continuous refinements of its estimates and assessments, and by evaluation of the need for additional sources, or modification of the process itself, to achieve improved results."* [191]

From this definition it is clear that the JDL model should form the foundation for the design of a TEWA DSS architecture. In Figure 7.1, it may be seen that the input information required by a TEWA DSS is a combination of real-time sensor data and pre-programmed domain expert knowledge. The real-time sensor data usually include the positional and kinematic data of the detected threats which normally have to be fused together from various sensor sources. This level 0 (source preprocessing) data fusion process, is generally performed within the sensors [191]. The knowledge base, on the other hand, typically includes pre-deployment data on enemy arsenals, threat types and electro-magnetic signatures of known threats.

The data processor unit is a level 1 (object assessment) data fusion process. This process is concerned with the estimation and prediction of relations between the entities (threats, DAs) and their relation to the environment, in order to develop the current situation picture from which further impact assessments can be performed. Typical functionality of this component may include triangulation and aircraft track extraction. The inference engine, on the other hand, is a level 2 (situation assessment) data fusion process. This process utilises a combination of the data from the data processor and pre-deployment data from the knowledge base to infer certain characteristics of the threats (threat type, weapon envelope, *etc.*) by using a combination of context-based reasoning, pattern recognition techniques and heuristic rules.

The analytical methods process is the focus area of this thesis. This process includes the TE and weapon assignment processes; both are level 3 (impact assessment) data fusion processes. During these processes, the levels of threat of the different aircraft are determined and counter attacks devised [70]. Finally, the results of the TE and WA processes are displayed on the *Human Machine Interface*[2] (HMI) and the FCO can interact with these solutions in order to select a suitable course of action.

The FCO can usually interact with the HMI through the rule adjuster component so as to configure certain analytical methods for the TE and WA processes, or update and modify the existing information in the knowledge base. This will ensure that the DSS complements the FCO's analysis style and enhances the FCO's confidence in the DSS.

7.2 Decision Support within a GBAD Environment

The context of decision support within a GBAD environment is different from that in other domains, because of the uniquely dynamic, stressful and complex environment. The focus in this section is specifically on decision support in the GBAD environment by virtue of the different classes of operators and the different types of information flow management strategies. Although the DSS is designed primarily for the FCO, it is nevertheless required to adopt a systems view of the DSS environment so as to better understand the role of the DSS in its overarching context.

[2] An HMI includes a graphical display and the fundamental software that is responsible for displaying the information from all relevant subsystems to the human operator, thereby assisting the operator during his or her decision-making processes.

7.2.1 A Hierarchy of Operators

Prior to considering decision support in a GBAD context, it is first required to understand the hierarchy of operators that exists within such an environment. There are essentially five different classes of operators within a GBAD environment — the *Opposing Force* (OPFOR), the sensor operator, the WS operator, the *Air Picture Manager* (APM) and the FCO [78]. A visual representation of how these operators interact within a GBAD environment is depicted in Figure 7.2.

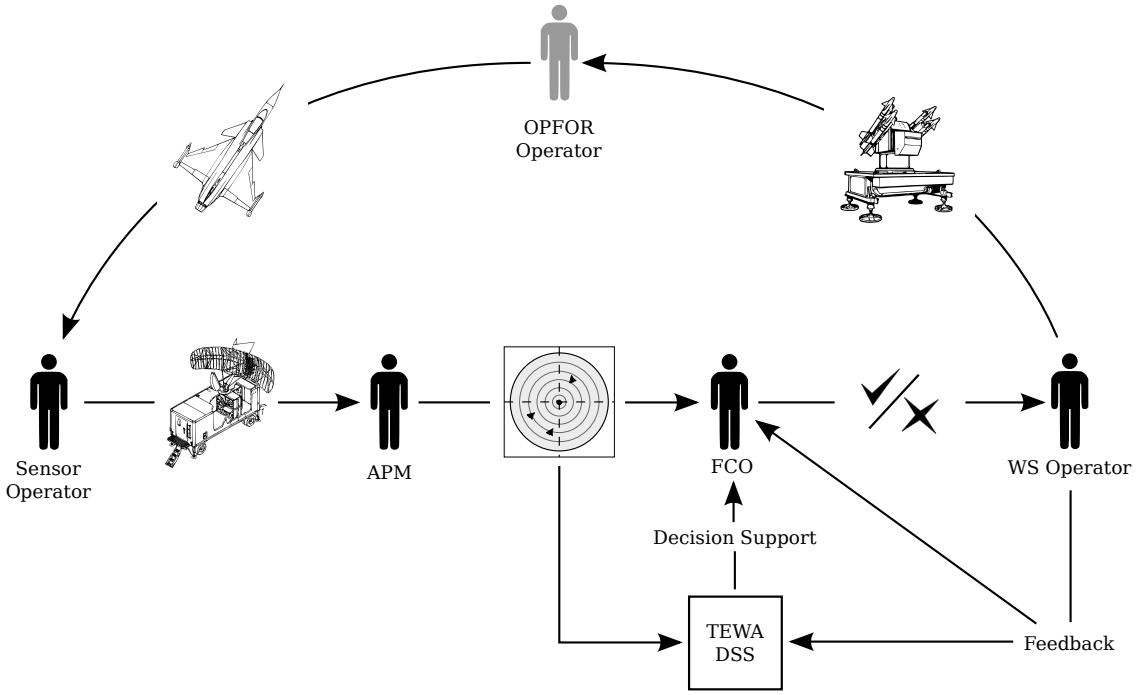


FIGURE 7.2: *The classes of different air-defense operators and their roles within an engagement cycle.*

Since a TEWA cycle is triggered in a GBAD context by the detection of an aerial threat (see §2.3), the OPFOR operators that pilot[3] these threats form a crucial part of the TEWA decision cycle, since their actions influence the response of the ground-based defenders.

The threats controlled by the OPFOR are, in turn, detected and tracked by an array of sensor systems. These sensor systems are operated by another important class of operators — the so-called sensor operators. These operators are responsible for maintaining the detection and tracking capabilities of the sensor systems, effectively assisting with the *observe* process during the OODA cycle. The data from the sensor systems are, subsequently, sent to a track management process after which the processes and filtered tracks are conveyed to the APM [78]. The responsibility of the APM is to manage the received tracks so as to form a comprehensive view of the current air picture, by employing various filtering techniques. The positions of own force elements (*e.g.* DAs and WSs) are generally also included in the air picture.

After the APM has generated an adequate view of the current situation, the air picture is conveyed to the FCO. During this stage, the TEWA system is employed in order to provide the crucial TE and WA information to the FCO. The FCO is typically authorised to override APM decisions or perform additional track-management tasks, if required. The FCO finally

[3]The threats can either be controlled remotely in the case of UAVs, or driven by human pilots in the case of manned fixed-wing aircraft.

utilises all available information presented, in conjunction with training and past experience, so as to *orient* himself. Subsequently, after the FCO has *decided* on a specific response, *Fire Orders* (FOs) are authorised for the engagement of threats.

The final class of operators — the WS operators — are responsible for triggering the actual engagement of threats (*i.e.* realising the *act* OODA stage) and ensuring that damage assessment results are returned to the TEWA system, thereby restarting the OODA cycle. Numerous WS operators may be responsible for operating various WSs; some WSs may also have multiple operators assigned to them [123].

For the purpose of this thesis, decision support is developed to aid the FCO who is mainly responsible for conducting the TE and WA processes in real time. For this purpose, it is important to pay careful attention to the level of complexity, flexibility and customization employed within the DSS. The operator's level of training and scope of experience, as well as the available timeframe, should therefore dictate the specific models employed and information shown to the operator. The aim should be to aid the operator in the decision-making process by executing complex computations, flagging suspicious behaviour and ultimately reaffirming the operator's initial intuition. This ideal operation will, however, not always be possible. If an operator does not understand why certain suggestions are made, he should be able to interact with the suggestions through the HMI to gain some understanding behind the algorithms' reasoning (*i.e.* have access to the raw data).

After reading this section, it should be clear to the reader that the slowest element in the TEWA decision-making process is the human operator. When applying the *Theory of Constraints*[4] (ToC) management methodology to improve the TEWA system, it is cardinal that the decision-making ability of the human decision maker should be enhanced by employing effective decision support aids.

Although this project's stakeholders wish to keep the operator-in-the-loop, there are some benefits to consider a fully-automated system. Adams [1] argued that if a human operator is part of the semi-automated system, it is most likely that he/she will be the most critical component of the system and the most difficult to replace (in terms of training time, cost and availability). It is reasonable to assume that the OPFOR will know this. Thus, when attacking the GBAD system, the OPFOR will exploit the situation by concentrating on attacking the human component, without which the system cannot function. This, in turn, places serious design restrictions on parts of the TEWA system in terms of more armour required, positioning of facilities and establishing life support services so as to improve the operators' survival probability. A possible response would be to locate the human operator far away from the TEWA system so that the system is operated remotely. This does, however, not eliminate the need for effective decision support aids (*i.e.* HMIs and algorithms that are specifically designed to aid the operators).

That said, even if the system is designed to be semi-automated, the change to a fully-automated system is a question of modifying aspects of the source-code (the core algorithms will stay the same), thereby allowing the system to directly act on the suggestions provided by the WA subsystem. In a GBAD environment, since civilian, friendly and hostile aerial threats co-exist in the airspace, the risks associated with implementing a fully-automated information processing system will be difficult to justify.

[4]ToC is a methodology for identifying the most limiting factor (*i.e.* constraint) that hinders a process from achieving a goal, and then systematically improving (reducing) the constraint until it is no longer the limiting factor [221].

7.2.2 Information Management Strategies

Information[5] is a term that includes many facets: What the ideal amount of information is that must be presented to the operators, and when the amount of information shown to the operator becomes excessive, resulting in an information overload, are hard questions to answer. As the quantity of data brought upon by the information age increases, the difficulty of analysing and interpreting it for decision-making purposes rises. Essentially, the system designers need to consider what information is relevant, reliable, value-adding and what information is deemed irrelevant. The operators should nevertheless also have the ability to interact with the system in order to clarify certain information or obtain additional information, if the situation requires this.

Typical examples of information in a TEWA system include, but are not limited to, the aircraft tracks, FECs of threats, threat values, WA solutions, terrain data as well as the positions of DAs and WSs. The information presented to the operators has one main goal — to enhance the situation awareness of the operators so as to improve decision-making. Effectively enhancing situation awareness is generally understood to be a crucial aspect in managing complex DSSs [35].

Information flow may be presented to the operator under two basic paradigms: A *demand-pull* paradigm and a *supply-push* paradigm [153]. The basic difference between these two information management paradigms is visualised in Figure 7.3. Demand-pull works according to the principle that information is inactive until a demand is made known for it. In a demand-pull system, information requirements are fully specified by the operator. This paradigm is more self-centred than the alternative, and if the requested information is readily available (*e.g.* in some local database) the information can typically be used quickly in an efficient manner within this paradigm. If, however, the information is not readily available, this paradigm can add additional stress to the operator. A demand-pull paradigm is therefore suited to optimising the use of limited resources by focussing on those tasks identified as critical by the operator. Such a tailored approach to information sharing may also introduce weaknesses in the OODA decision cycle of the operator. Important information may not be sent to the operator, thereby, limiting his ability to form a coherent situational picture. Furthermore, since an information search (through the databases) only commences as a result of an operator request, an undesirable time delay may be introduced, thereby decreasing the OPTEMPO of the TEWA system.

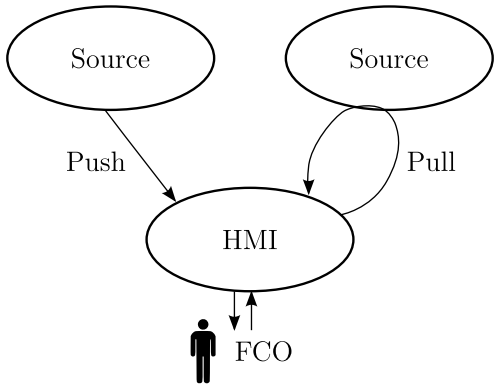


FIGURE 7.3: *Difference between demand-pull and supply-push information management.*

[5]There is often a misconception between the terms data and information. Data may be defined as raw material and organised facts regarding the operating environment that needs to be organised. Information, on the other hand, results when data are processed, organised and presented in a given context so as to be used for decision-making purposes.

In contrast, a supply-push information handling system pushes information from source to user. This push is either triggered as information becomes available or according to some pre-set schedule. By implementing this paradigm the unwanted time-delay is removed and information typically arrives in a timely fashion. The difficulty in this paradigm, however, is anticipating the operator's information requirements and preventing an information overload [169].

Since each of these paradigms have certain limitations and advantages, a combination of the two paradigms is more suitable for use in a TEWA DSS. Certain information, which may be seen as crucial for forming a holistic situation picture, such as the aircraft tracks, IFF responses and WSs positions, should be presented in a push fashion. Alternatively, the operator should also be able to request information from the rule-based system in order to perform further analyses on the reasoning behind a particular threat value or WA suggestion. It is essential that the operator should be supported by the DSS and that decisions should not be driven by the system.

7.3 Facilitating Germane Decision Support

It is ultimately a human operator who decides whether and how each aerial threat should be engaged — not a fully automated system — because decision-making in a GBAD environment can have severe (possibly catastrophic) consequences if inappropriate decisions are made, as described in §1.1. Hence, it is of utmost importance to ensure that the decision support information communicated to the operator is as clear and uncluttered as possible. By so doing, the operator is afforded the opportunity to effectively make use of the information for the purposes of analysis, interpretation and decision-making. There are, however, several complexities associated with designing an efficient DSS, some of which include operator stress and uncertainty. The extent and effect of these complexities on the system's performance are clarified in this section.

7.3.1 Complexities Surrounding a TEWA DSS

The form of decision-making explained above is a highly complex task and requires the integration of various data sources [113]. Military systems currently in their development stages, will be too fast and too numerous and will, resultantly, create an environment that is too complex for human operators to direct without assistance [1]. Furthermore, the proliferation of information-based systems will produce a data overload that will make it difficult or impossible for humans to directly intervene in decision-making.

A high level of tactical expertise and knowledge of the type of threats, prevailing legal frameworks and assessment heuristics from experience are crucial to succeed in this new environment [41]. Training and experience are, however, not enough to ensure tacit decision-making. According to Morrison *et al.* [133], the importance of ensuring that information is meaningful, timely and easily accessible cannot be underestimated. Operators are therefore required to become more reliant on machine assistance in order to succeed.

HMIs are, as a result, crucial in enhancing the situation awareness of the operators and, in turn, enable them to make good, germane decisions. The HMI should, however, not duplicate or dominate the decision maker's thinking, but rather complement it. To this end, the HMI should focus on enhancing the limiting capabilities of the human. Brown *et al.* [29] provided a comparison between the capabilities of man and machine, as visualised in Figure 7.4.

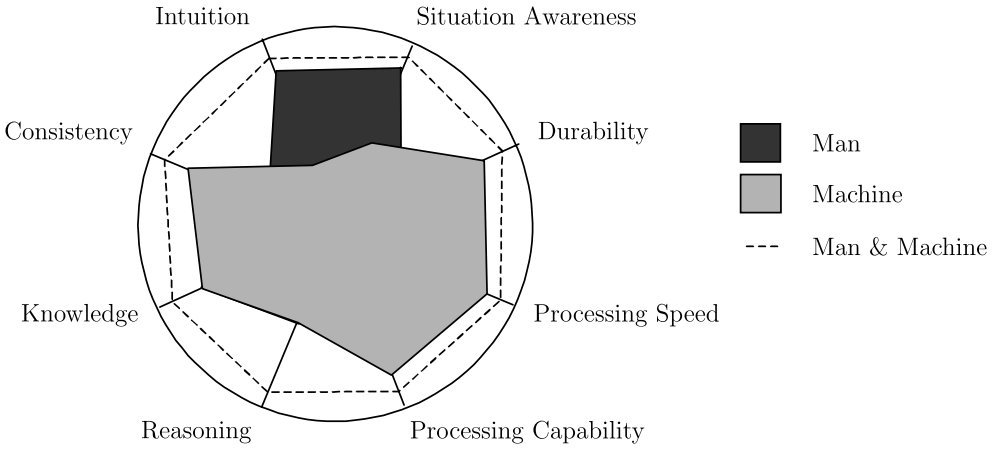


FIGURE 7.4: *Comparison between man and machine capabilities [29].*

In Figure 7.4 it may be seen that man excels in areas which require experience and intuition. Machines, on the other hand, outperform man in capabilities that require handling large amounts of data, obtaining results quickly and performing complex calculations. The HMI should complement the operator — especially with respect to addressing the limitations of the human operator — when performing the crucial C2 processes detailed in §2.2.

7.3.2 Effect of Operator Stress on System Performance

Research has found that the abundance of information — as is prevalent in a GBAD enviornment — can overwhelm the operator and lead to increased levels of stress for the decision maker [64]. In a military environment, stress can, in turn, severely affect the ability of human operators to make effective judgements. Stress may be defined as a non-specific response of the body to a certain event or stimulus, also known as a stressor [101]. In the past, several researchers have proposed that a negative-linear relationship exists between stress and performance [174]. Stated alternatively — increased levels of stress are associated with a decrease in performance. Salehi *et al.* [174], however, conducted an experiment to prove the existence of the so-called inverted-U-shape phenomenon, explaining an alternative relationship between stress and performance. This phenomenon states, informally, that an individual's performance will be lower at high and low levels of stress, but higher at an optimal, intermediate level of stress. At moderate levels of stress, performance should be improved because of the presence of enough stimulation to keep the operator vigilant and alert, as opposed to becoming bored and tired. The stress should, however, not divert the attention of the operator or absorb his energy. This relationship between stressors, stress and the performance of a system is depicted in Figure 7.5.

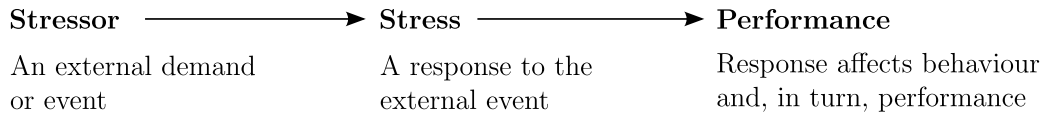


FIGURE 7.5: *How stress affects performance [101].*

As may be seen in this figure, a stressor (*e.g.* extreme temperature, lack of sleep or uncertain environment) induces a psychological response (*e.g.* increased blood pressure or elevated heart rate), commonly known as stress. This stress[6], in turn, influences the operator's decision-making ability and impacts on the performance of a TEWA system. Kavanagh [101] conducted

[6]Stress, in this context, refers to only the physiological response to stressors, as used in [101].

a literature review of the effects of stress on performance in the military domain and found the following common denominators explaining how excessive levels of stress affect the cognitive performance and decision-making ability of individuals[7] (operators):

- *Screen out peripheral stimuli.* Individuals pays less attention to perceptual cues and stimuli that could enhance their decision-making ability, also referred to as *perceptual narrowing.*
- *Make decisions solely based on heuristics*[8]*.* Individuals prefer heuristics and other simplified decision-making strategies when faced with increased levels of stress.
- *Suffer from performance rigidity or narrow thinking.* In other words, failing to consider the full range of alternatives and ignoring long-term consequences.
- *Loss of ability to analyse complicated scenarios and interpret information.* Individuals are more likely to obey orders (follow the results suggested by a TEWA DSS) without thinking for themselves — they ignore cues that imply the presence of something unusual.

In the stressor-stress-performance relationship illustrated by Figure 7.5, two methods may be employed to reduce the effects of stressors on system performance: (i) Reduce the psychological response of the operators to the stressors or (ii) mitigate the effect that stress can have on system performance. The first mitigation strategy may be achieved through stress-exposure training [58]. During such training, individuals are exposed to simulated stressors in a virtual or real simulation (see §4.1.1) and forced to perform targeted skills (evaluating threats and assigning WSs). Driskell and Johnston [58] suggested that stress-exposure training can gradually lessen an individual's physiological response to stressors by, effectively, reducing its novelty. The second mitigation strategy, on the other hand, may be implemented by limiting the system from executing certain responses (*e.g.* preventing execution of a FO from disobeying the established ROE).

7.3.3 Uncertainty Management

Uncertainty is a major concern when information is shown to human operators for the purpose of decision-making. When information is presented on a HMI it often appears absolute. In reality, however, data are rarely absolute [182]. The problem of uncertainty management is finding the truth from among uncertain observations in a reliable way. There are various ways in which uncertain observations can be modelled. Typical formal methods to model uncertainty include probability theory, fuzzy-set theory and random distribution [150]. This section contains a brief introduction to the types of uncertainties prevalent in a TEWA system — possible approaches are also suggested in order to manage each of these types of uncertainty.

Two general notions of uncertainty are *aleatory* and *epistemic* uncertainty [67, p.36]. Aleatory uncertainty is also referred to as stochastic uncertainty, variability or irreducible uncertainty and is generally the kind of uncertainty inherent in a process. The alternative, epistemic uncertainty, is the class of reducible uncertainties or state-of-knowledge uncertainties. This type of uncertainty can be mitigated or even eliminated if more information is obtained. The focus of the uncertainty management in the design of a TEWA DSS needs to address both types of uncertainties — aleatory uncertainty should be accounted for during the testing phases and fully

[7] People are referred to as "individuals" if the underlying research was not conducted in a military domain, whereas "operators" is used as the place-holder if the research conceded in a military domain.

[8] As mentioned in §7.1, heuristics essentially refer to the use of rule-of-thumb guidelines based on past experience to make decisions.

understood before implementation, whereas the epistemic uncertainty should be reduced during development. It is unlikely that the information shown to the operators are 100% correct and, as such, the level of uncertainty in the suggestions should be quantified in order to further aid the operators during their OODA cycles.

Skeets *et al.* [182] provided different classifications of uncertainty that should prove applicable to the typical uncertainties present in a TEWA DSS. The study by Skeets *et al.* recruited eighteen participants from various domains, several of the researchers and practitioners using a combination of machine learning, robotics, databases and computer graphics during their daily routine. Open-ended interviews were held with the different participants in order to elicit their understanding and management of uncertainty.

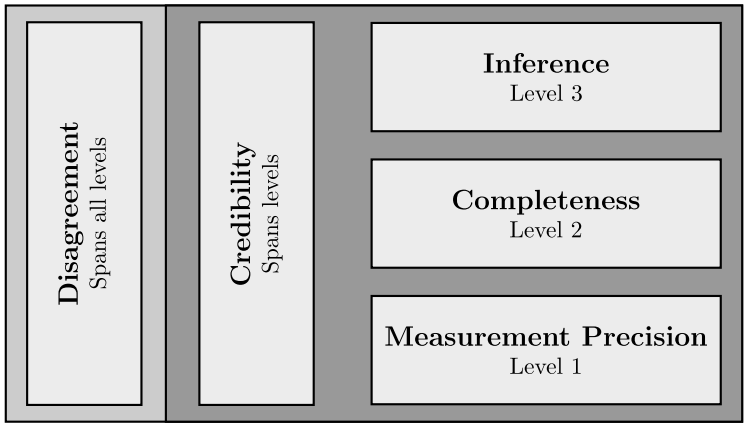


FIGURE 7.6: *Classification of uncertainties based on the explanations by [182].*

The lowest level of uncertainty, *measurement precision*, mainly results from imprecise measurements. This category covers any variation, tolerance or precision limitations in measurement techniques of systems that produce quantitative data. *Completeness* of data is seen as the middle level of importance and may be understood as the errors resulting from generalising results from a sample to the population. In a TEWA context, sensor sampling rates and unknown data regarding threats, are two factors contributing to this category.

According to Skeets *et. al.* [182], aggregating or summarising data in an irreversible way (as performed during the TE process) can also result in uncertainty, since some information are lost and the data are therefore no longer complete. The worst case of incomplete information is the so-called unidentified unknowns (*e.g.* undetected threats of incomplete FEC).

The highest level of uncertainty in Skeet's categorisation is *inference*. Inference spans all types of predictions and modelling assumptions. This type of uncertainty has a strong link with decision-making, as should be clear from the inference module present in the TEWA DSS in Figure 7.1.

The *credibility* of the information spans all these three levels, and is not easily quantified. The three levels of uncertainty all contribute to the credibility of the information that is used for decision-making. Different end-users also have different judgements in respect of what constitutes a credible source. The TEWA system should not be presented as a black-box system to the operators so as to allow them to critique the system's responses and, in turn, build confidence in the system. All the levels of uncertainty are spanned by *disagreement*. Disagreement is often associated with credibility and, correspondingly, the three levels of uncertainty. Untrustworthy information will generally result in a disagreement between the operator's intuition and the suggestions presented by the TEWA DSS which, in turn, will result in further confusion. In a

TEWA system, three of the main origins of uncertainty are,

- inaccuracies from sensor system measurements (measurement precision),
- differences between the actual SSHP of a WS and its stated theoretical SSHP (completeness),
- unknown accuracy of information obtained from domain experts (completeness),
- calculation of system threat value (inference), and
- quality of the WA suggestions (inference).

This list should not be seen as exhaustive, but serve as an introduction for further work into the uncertainties present in a TEWA system and possible uncertainty management strategies. The most commonly used method to visualise uncertainty is error bars [182]. Error bars can be effectively employed to visualise uncertainty, and mitigate its effects if the class of uncertainty lends itself to be visualised in this way. A commonly used method to reduce uncertainty is averaging operations [61]. Averaging operations in uncertain observation aggregation are justified when the set of sources can be viewed as a single random source producing different inputs (*i.e.* are independent sources). In that case, indeed, the set of data to be fused can be interpreted as standard statistics. For instance, several successive independent measurements from a single sensor can be viewed as the result of a random experiment, because of the required condition of independence. However, in the case of unique measurements issued from connected sensors, or in the case of expert knowledge databases, it is not clear whether averaging combination modes are tractable.

For the purposes of this thesis, the uncertainty in the TEWA system is accounted for by adding random noise around certain input information, such as the normal distribution around the input coordinates. The variability of success with which WSs are employed (hit or miss) may be managed by a utilising uniform distribution to determine the outcome of a WS engagement. The SSHP, on the other hand, are modelled as a function of stand-off distance as described in §4.2.3. Various methods that may be used for the visualisation of uncertainty are described in the following sections.

7.4 HMI Design Considerations

As stated in §3.3, there have been many studies on the design of HMIs [57, 113, 133, 139, 148], but detailed information on these studies is largely lacking. The majority of these studies within the military domain have, furthermore, been conducted in the context of a point-defense naval environment. For an area-based GBAD environment, different requirements exist for the presentation of information, although many similarities do exist. No detailed studies of a HMI for a GBAD environment could be found in the open-literature, but some of the available studies for a point-defense role are analysed in this section in order to derive meaningful considerations and guidelines for the design of a HMI for an area-based, GBAD environment.

7.4.1 Qualitative Evaluation of Existing HMIs

During the early 1990s, the *Tactical Decision Making Under Stress* (TADMUS) programme in the United States of America involved research in support of improved DSS design by integrating cognitive theory and human-machine interaction technology [133]. This programme

attempted, in essence, to design a DSS according to a "naturalistic" modelling approach toward decision-making in environments that are characteristically highly complex, short-fused and dynamic. The displays were developed to support the decision-making process by addressing the recognition-primed, explanation-based reasoning and cognitive limitations faced by the US Navy officers employed in a Combat Information Centre [148]. This research programme was most likely the foundation for a host of related research that followed.

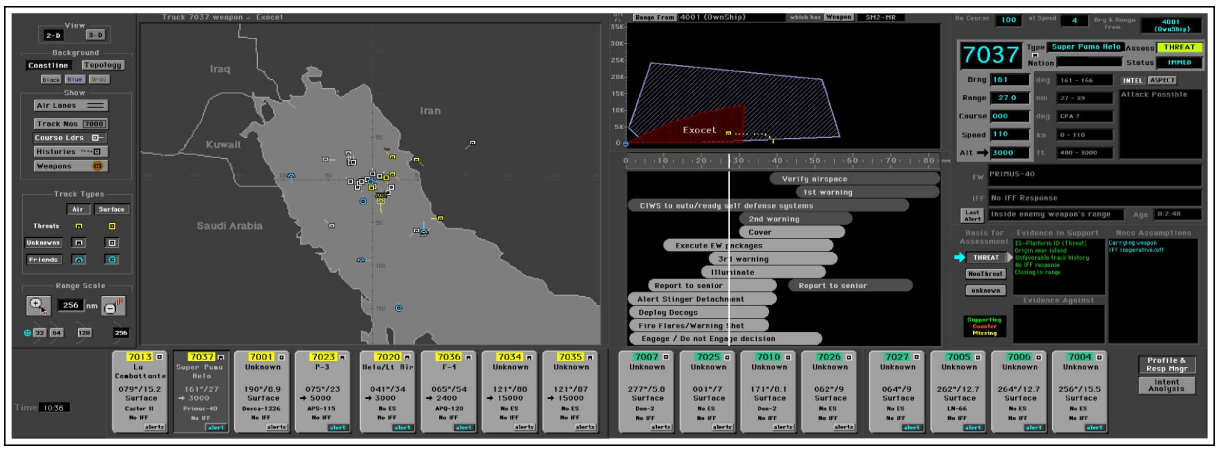


FIGURE 7.7: *Example of an effective HMI developed through the TADMUS programme [133].*

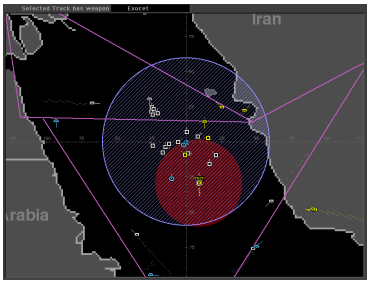
Some of the TADMUS programme's results have since been declassified, and include discussions on limitations that were present in a preliminary designed DSS [87]. The TADMUS HMI display is shown by Figure 7.7. As my be seen in this figure, the TADMUS DSS is organised into six modules, with different roles to aid the decision-making cycle of the operators:

Geo plot. This display presents the area surrounding the vessel. A desaturated version, with minimal clutter, is depicted by Figure 7.8(a). All detected surface, subsurface and air contacts are overlayed on this screen. The motivation for a 2D display comes from the need to be able to quickly and precisely locate tracks. This module is intended to be the primary focus of the operators and is designed in order to improve the operators' situation awareness. Symbols of all detected entities are colour and shape coded so as to indicate the status of track identification and TE. Different layers may be "switched" on/off so as to adapt to a particular situation.

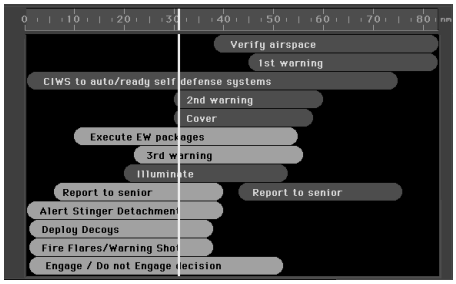
Track summary. This screen provides more detail on a selected track. Several flagging-related TE models, such as electronic warfare deployment status and IFF response information, are also displayed on this display. Threat-specific kinematic data are also shown on this display, as shown in Figure 7.8(f).

Track profile and aspect window. This module indicates the speed, altitude, course and range of a single track on a 2D display with altitude on the vertical axis and stand-off range on the horizontal axis, as shown in Figure 7.8(d). The most probable WRL of the threat and the WS ranges of the vessel are also indicated. This module has three main goals: (i) to locate the track's current position, (ii) to recognise whether the track can engage the vessel and (iii) to recognise whether the track is in range of the vessel's WSs.

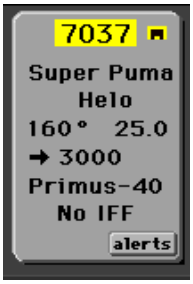
Response manager. This module is responsible for assisting the operator in delegating pre-planned responses against the selected threat. An example of this display is shown by Figure 7.8(b). Possible responses are graphically depicted in a Gantt chart type display



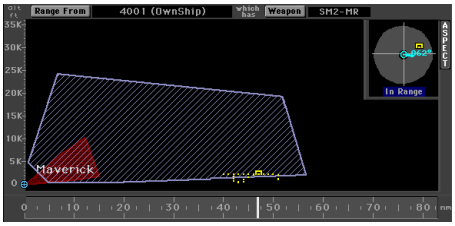
(a) *Integrated geo plot*



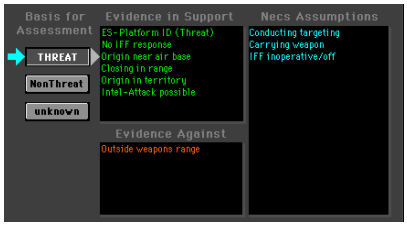
(b) *Response manager*



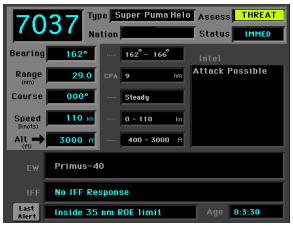
(c) *Mini-CRO for single track*



(d) *Track profile with engagement possibility*



(e) *Basis for assessment*



(f) *Track summary*

FIGURE 7.8: *Different TADMUS HMI modules in the large display of Figure 7.7.*

on the screen so as clarify their aims. A vertical line also indicates the threat's current position relative to the vessel. Furthermore, this display serves as a graphical embodiment of the current ROE and doctrine. This module, essentially, attempts to support a story telling approach to decision-making.

Basis for assessment. This module, shown in Figure 7.8(e), is an expert system which employs expert-based reasoning to add value to the current available information. The presented information provides the operator with a means to criticise his or her decision-making process with a view to improve the quality of the selected response.

Character read-outs. A series of *Character Read-Outs* (CROs) are displayed across the bottom of the display, as shown in Figure 7.7. These displays only indicate critical track information, such as track-number and aircraft type. Several contacts are focussed on simultaneously in this display. Contacts are prioritised according to the level of danger they pose to the vessel. Consequently, the order of the CROs change throughout the scenario, thereby allowing operators to prelude attention to the changes in priority.

These modules are, essentially, arranged in order of increasing information integration from the top to the bottom. The first three modules, namely the track summary, the track profile and geo-plot, which all analyse a specific threat, are located at the top of the display. The response manager and basis for assessment which, on the other hand, deal with the evaluation of possible responses, are located at the middle of the display. The CROs module, which present information on all the threats, is displayed at the bottom of the larger display.

According to the results of the TADMUS program [133], decision-makers (operators) seem to use a "stepping-stone" approach to reach a decision on which to act — available information is used in order to reach a consensus regarding the priority of a track, after which other information

sources are explored to ascertain further implications for that specific threat, given its assumed priority.

Better supporting such a decision-making process by, for example, by increasing the OPTEMPO of the OODA cycle, should allow the operators more time to coordinate responses. A possible means to achieve this is to a certain extent automate the first step of the "stepping-stone" strategy (prioritisation of threats). The TE subsystem described in Chapter 5 serves the purpose of automating this step. The flagging models may then be used in conjunction with other threat-specific information to decide on an appropriate course of action (*e.g.* engaging threats, sending warnings, employing electronic warfare capabilities or attempting to deter the threat).

Before determining whether this DSS is indeed fit for use in a GBAD environment for TEWA purposes, it is suggested that a *Work Domain Analysis* (WDA) is applied. A WDA was successfully applied by Burns *et al.* [31] to evaluate the suitability of the TADMUS DSS (which was designed for a US naval environment) to the C2 of the Canadian HALIFAX class frigate[9]. WDA is a generally applied technique for the design of new DSS; it has even been proposed to be used for evaluating possible acquisition projects. Nonetheless, some of the TADMUS HMI modules should prove to have some residual capability in the context of a GBAD environment.

7.4.2 Suggested HMI Design Guidelines

A major concern of data fusion is to ensure that the data collected from sensor systems are integrated and organised in such a way that it is useful when presented to an operator. In order for information to be useful, it must be meaningful, timely and easily accessible according to Morrison [133]. The design guidelines presented in this section were identified from studying the relevant literature [63, 113, 129]. These guidelines should at least be adhered to when designing an HMI for a GBAD environment, but the list should not be seen as exhaustive:

Graphically display the threat values of aircraft

The threat values should either be displayed in graphical format (bar plots) or threat-category classes (low, medium and high) in order to avoid a false sense of precision [129]. Because of the uncertainty inherent during the TE process and the fuzzy notion of threat values, numbers are not suitable for presenting the threat values to operators. Indicating the threats on a 0–1 interval may result in familiarity bias[10] where the operators start to rely solely on the threat value, disregarding other information, when determining a response. Lieberhaber [113] suggested the use of three verbal descriptions of threat levels — low, medium and high — to be used to assist the operator, rather than percentages or numbers. Graphic formats have also been shown to decrease the number of risky decisions taken by individuals [192]. Roux and Van Vuuren [169] also suggested the use of graphical formats in order to present data to operators since this type of visualisation complements the cognitive memory abilities of operators and are, therefore, better suited for value-based assessments.

Provide a prioritised threat list

Besides the detailed cue information related to specific threats, all the detected threats

[9]The HAFILAX class frigate, commissioned in 1992, forms the backbone of the Canadian Navy. These multi-purpose vessels were primarily designed for anti-submarine warfare and anti-surface warfare. With the current advent of asymmetrical threats, this frigate is undergoing modernisation programs to expand its capabilities, thereby enabling it to operate in an NCW environment [33].

[10]According to Huberman [85] familiarity bias is associated with the general sense of comfort with the known, and discomfort (or even fear) of the distant and alien.

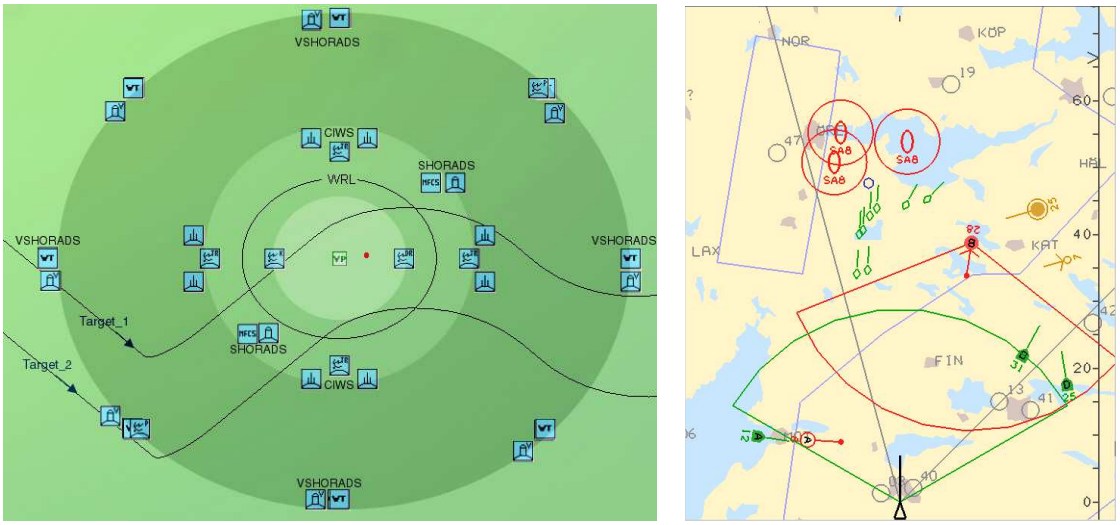
should also be arranged according to the danger they pose to the defended system. The purpose of such a list is to reduce the cognitive load of the operators by providing them with the opportunity to quickly grasp mission-critical information [113]. For the purposes of this list, the system threat values may be used in order to prioritise the threats. The resulting prioritised list should include information, such as the track number, bearing, altitude, identified aircraft type and threat value [172].

Show the threat value history

Most of the problems observed during testing of the preliminary TADMUS HMI were linked to basic cognitive limitations — memory and attention span [87]. Incorporation of a threat-value history display should greatly decrease the short-term memory requirements on the operators. In addition, it may also enhance the capability of the operators in terms of making sense of the data and, in turn, promote a story-telling approach to decision-making [129].

Display a threat assessment window on-screen when a specific threat is selected

This suggestion is similar to the CROs described in §7.4.1. This pop-up window should include information such as threat value, flagging models, threat value history and the status of several cues. According to [113] air-defense operators have indicated that they prefer this type of information to be displayed on the so-called geo-plot, instead of separately as currently implemented on the TADMUS display. Miller [129] also suggested that this information should be displayed on the geo-plot in order to assist detailed tracking of the selected threat. Two additional examples of geo-plots which are used in a military domain are depicted in Figure 7.9.



(a) *A typical geo-plot of a GBAD deployment [63]*

(b) *Part of the tactical display on a fighter aircraft [77]*

FIGURE 7.9: *Two examples of geo-plots used in the military domain.*

Access to raw data should be available

A major concern for the successful operation of the DSS is that the operators should trust the results displayed by the HMI. To this end, Rummel [172] suggested that raw data should be made available to the operators. The operators should be able to verify certain suggestions generated by the DSSs and, in so doing, build more confidence in the results. It is unlikely, however, that operators will analyse raw data during the highly dynamic and

time-compressed GBAD scenario. The raw data may nevertheless be used for training and evaluation purposes after the scenario in order to build operator-confidence in the DSS, as was suggested by Helldin *et al.* [77] for the design of a fighter aircraft HMI.

Provide a prioritised list of assessment cues

Ranking the corresponding cues will help facilitate building a coherent story and thereby assist germane decision-making [113]. Typical examples of assessment cues for an air-defense environment are listed in Table 7.1. These cues are ranking according to their relative importance and the six most important cues are in bold-font. The data values of these cues (altitude value or speed value, *etc.*) should also be displayed. Binary flagging models should consequently be accompanied by the reason for their activation, in order to assist in a naturalistic approach toward decision-making. Miller [129] suggested that by providing a comprehensive list of cues, several operator biases may be reduced and, correspondingly, enhance confidence. In addition, such a comprehensive cues list should prevent over-reliance on specific cues, which is typical behaviour if operators are placed under stress, as explained in §7.2.1.

TABLE 7.1: *Examples of cues used by air-defense operators in the US Navy for TE purposes [113].*

Attribute	Description
Country of origin	**Most likely country of origin**
IFF mode	**Signal from aircraft if it is friendly, or perhaps neutral**
Airlane	**Is the track following a commercial air route**
Altitude	**Approaching height of aircraft, or change in elevation**
Radar/ESM	**Type of radar employed by the aircraft**
Speed	**Approximate air speed of aircraft, or change in air speed**
Range from CPA	**Closest point of approach**
Feet Wet/Dry	Determines whether aircraft is flying over land (dry) or water (wet)
Number/Composition	Number of aircraft in formation
Coordinated activity	Communication, or proximity to other tracks
Manoeuvres	Past manoeuvres executed, or most probable current manoeuvre
Wings Clean/Dirty	Designates whether an aircraft is armed (dirty) or unarmed (clean)
Course	Heading angle relative to DA
Range	Slant range from DA
Visibility	Indication of atmospheric conditions
Weapon Envelope	A track's position relative to its predicted weapon release line

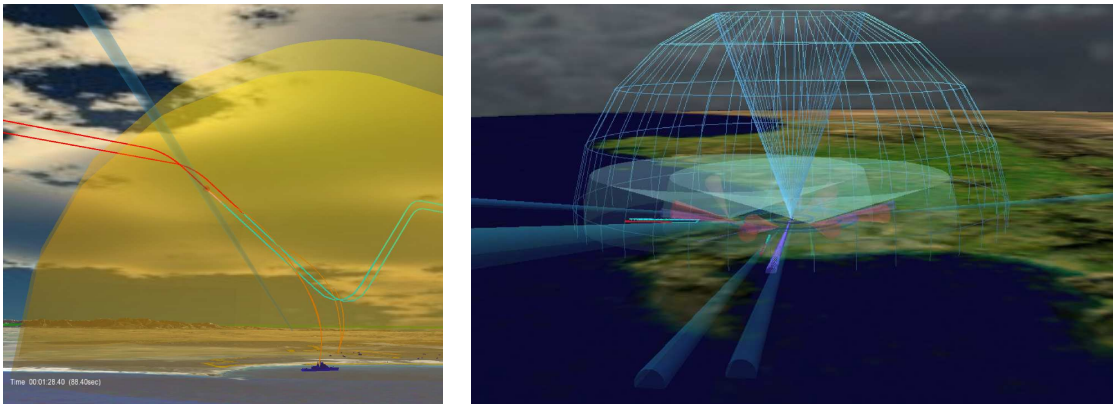
Show the impact of each cue or TE model on the system threat value

Since the determination of the level of threat of a specific aircraft entails the evaluation of uncertain and sometimes contradicting information, the extent to which a specific TE model contributes to the system threat value should be shown to the operators. This information will enable the operators to evaluate counter and supporting evidence and, consequently, support an explanation-based reasoning approach.

3D visualisation must be included

The South African GBAD program developed a 3D visual analyis tool to be used its the system of systems simulation capability [63]. In contrast to a 2D display, a 3D display should allow operators to assess general spacial relations. By so doing, the situation awareness of the operators will improve and enable better decisions to be made. According to Duvenhage *et al.* [63], 3D visualisation is particularly helpful to users who are not yet

experienced enough to combine multiple 2D displays and other information into a higher dimensional mental picture. The technological drive resulting from the gaming industry, especially in the realm of computer graphics, has made it possible to generate more realistic displays of the GBAD environment, while still allowing for sufficient rendering speed.



(a) *View of several simulation objects, including missile trails and radar domes*

(b) *Visualisation of the radar dome and virtual cones in the GBAD environment*

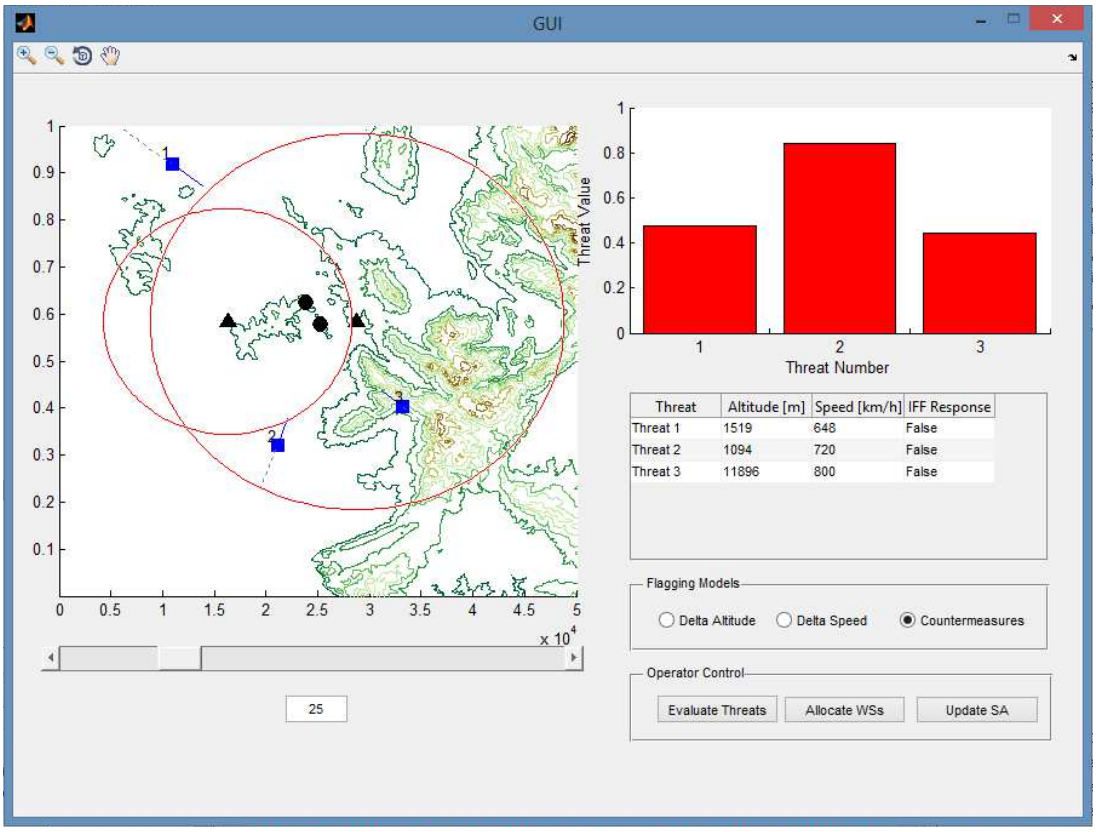
FIGURE 7.10: *South African 3D GBAD system visual analysis tool displays [63].*

7.5 HMI Designed in Matlab

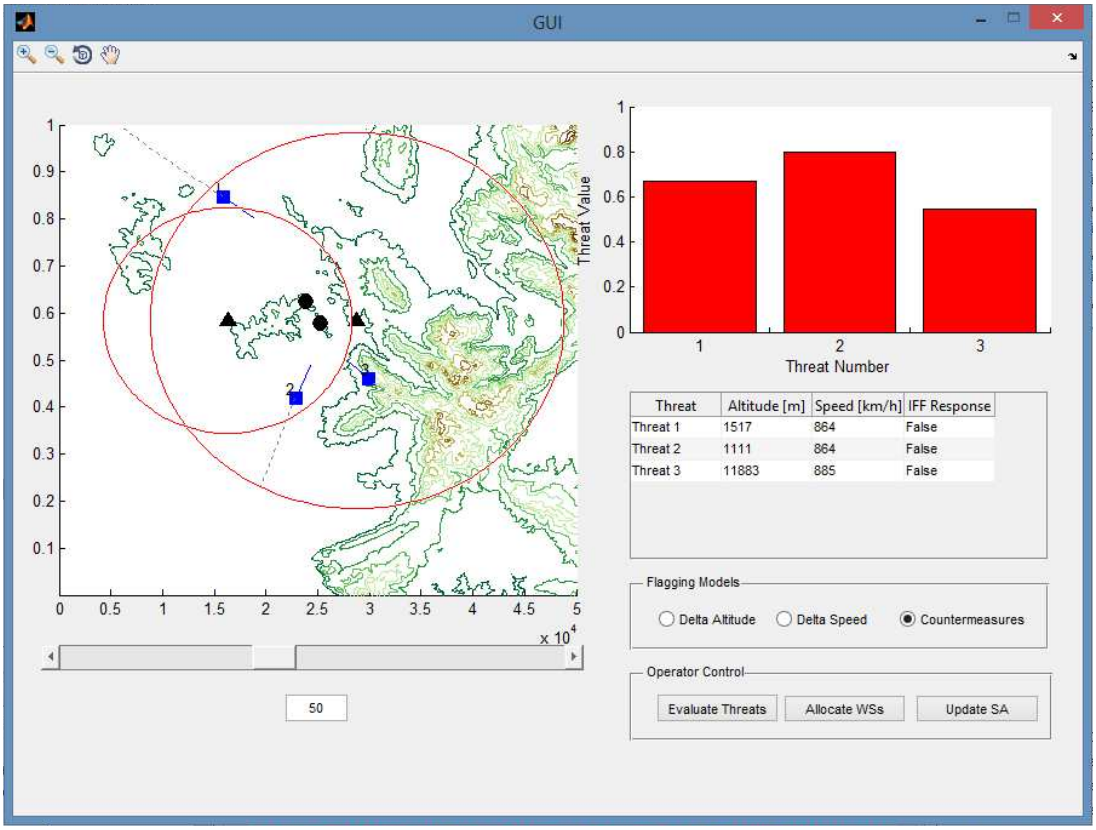
A preliminary HMI was designed using MATLAB'S graphic user interface tool. This design should not be seen as a final design, but rather as the foundation for a more detailed HMI that may be used to validate and evaluate the final TEWA DSS in conjunction with its intended end-users (the operators). Furthermore, this design also serves the purpose of demonstrating the capabilities of the MATLAB graphic user interface design environment.

MATLAB is viewed by many users as both a high-performance language for technical computing as well as a convenient environment for the development of HMIs [179]. The majority of military HMI designs continuously change during the project life-cycle, as the customer requirements of the intended end-users are reiterated [88]. It is therefore unavoidable that the system designers need to adopt an evolutionary development process. In addition, the preliminary design should account for these possible future changes in order to save time, cost and improve flexibility, thereby allowing the design to better adapt to changing customer requirements.

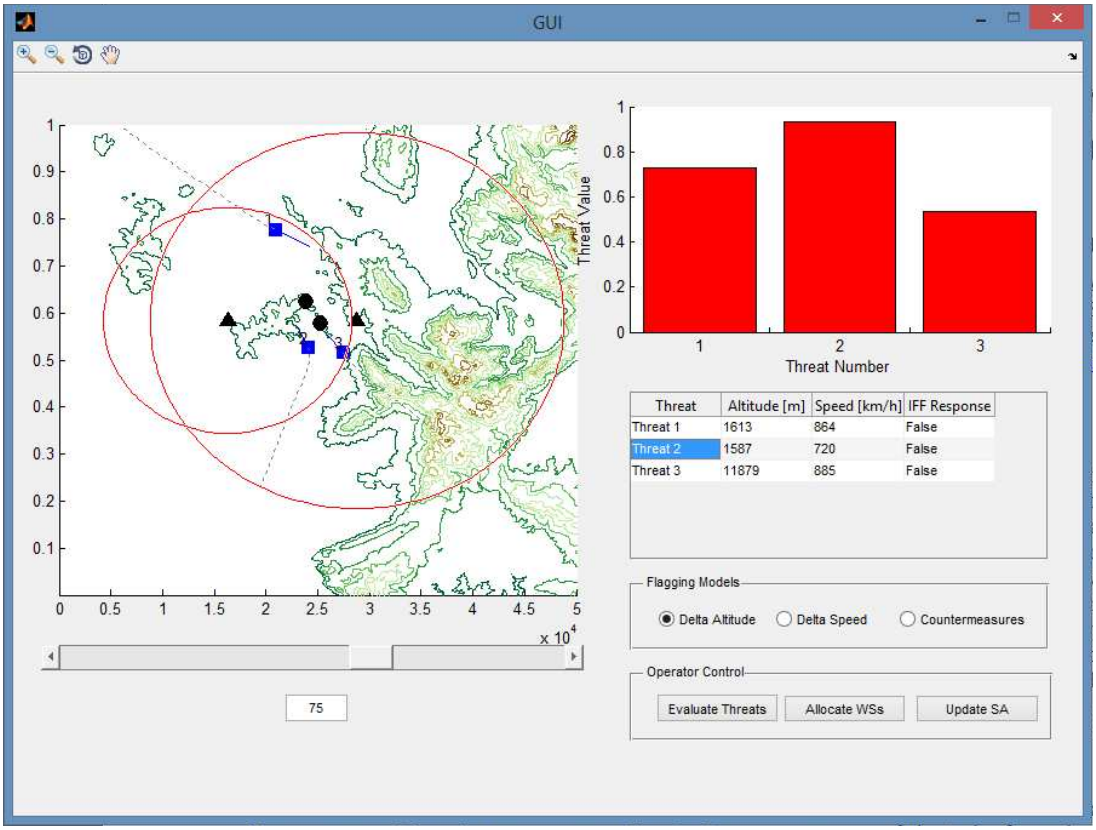
The layout of the preliminary designed HMI is shown in Figure 2.1(a)–(d). The demonstrated scenario is the hypothetical GBAD scenario introduced in §4.6. The four different figures correspond to a different TEWA-cycle (scenario time), and the four selected times correspond to the TEWA-cycle times of 25, 50, 75 and 100. The two DAs are depicted by black-filled circles, whereas the two WSs are depicted by the black-filled triangles. The maximum ranges of the two WSs are visualised by the red circles. The three threats, on the other hand, are depicted as squares, their past tracks are indicated by grey dotted lines and their current velocity vectors are depicted by lines protruding from the squares. Furthermore, the system threat values of these three threats are visualised by the bar-chart in the top right-hand corner and the results of a number of binary flagging models are indicated for the currently selected threat. In addition, the values of specific assessment cues are listed in the table. Some operator controls are also included to be used for detailed validation purposes in the future.



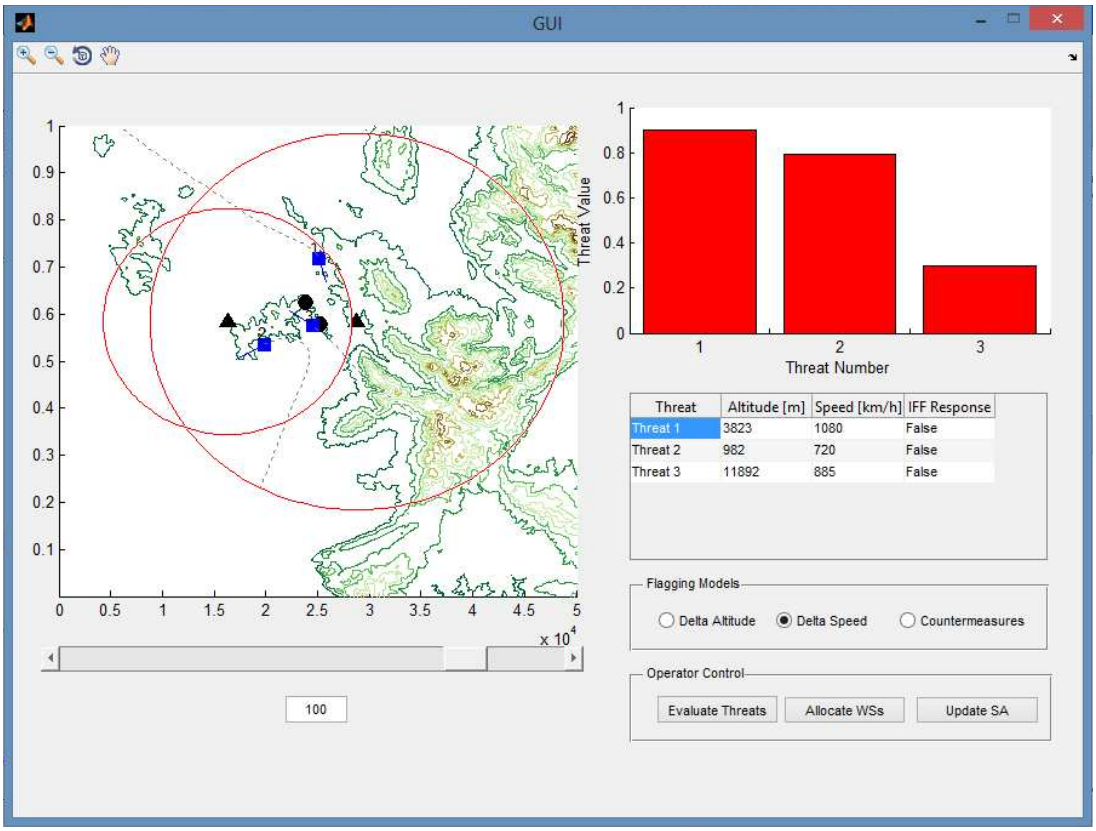
(a) *TEWA cycle 25*



(b) *TEWA cycle 50*



(c) *TEWA cycle 75*



(d) *TEWA cycle 100*

FIGURE 7.10: *Preliminary designed* MATLAB *HMI.*

7.6 Chapter Summary

A number of considerations were provided in this chapter for inclusion in the detailed design of the TEWA DSS HMI. The chapter opened with an overview of the required DSS functionality in §7.1. After understanding the requirements and functioning of the required DSS, decision support pertaining to a GBAD environment was described in general in §7.2. Special attention was afforded to the operators in a GBAD environment (§7.2.1) and the types of information management strategies (§7.2.2).

After understanding the context and requirements of the DSS, the importance of facilitating germane decision support for the operators was detailed in §7.3. Special attention was afforded to the complexities associated with the provision of germane decision support to the FCO. Furthermore, the effect of operator stress on system performance, and the uncertainties present in a TEWA DSS were described in §7.3.2 and §7.3.3, respectively. An existing HMI interface was subsequently described in §7.4.1 and a number of design suggestions were made for the detailed design of a HMI in §7.4.2. The chapter closed with an overview of a preliminary designed HMI and demonstrated the capabilities of the MATLAB graphic user interface design environment in §7.5.

PART III

PERFORMANCE EVALUATION

CHAPTER 8

Performance Evaluation Framework

True genius resides in the capacity for evaluation of uncertain, hazardous and conflicting information.

— Winston Churchill

Contents

As mentioned in Chapter 1, the aim in this project is to integrate the available TE and WA algorithms in order to develop a simulation of a working TEWA system. After developing the simulation, it is required to evaluate the validity and quality of the results returned by the system. Because of the many elements in the system, in practice, this can become a very complex and iterative process that intersects many disciplines.

This chapter opens by first providing an introduction to the evaluation of TEWA systems, after which the underlying concepts of a *System of Systems* (SoS) approach is explained. This is followed by a review of available performance evaluation approaches for TEWA systems as identified from literature and, finally, four metrics are proposed to be used within the developed simulation paradigm.

8.1 Performance Evaluation of TEWA Systems Overview

Before the 1980s, military systems were largely stand-alone, analogue and mechanically controlled [194]. Today, most military systems are highly-complex[1] in the sense that numerous on-board subsystems typically interact to provide the system capability in a collective fashion. As systems become unavoidably more complex, as is the case with NCW (see §2.1.2), it is important to recognize the complexity inherent in defining the common system capabilities as well as to understand the constraints and requirements of all the potential users, thoroughly. This understanding is crucial in order to ensure that the final commissioned system functions in such a way that the user requirements and constraints are met.

The only way to ensure that a system satisfies all the requirements and accommodates all the constraints of its end-users, is through rigorous testing of the system, both in a simulated-environment (as advocated and implemented in this thesis), and in a physical environment where the system is tested in concert with the constituent systems (namely the operators, WSs and meteorological conditions *etc.* in the case of a TEWA system) [65]. This *Testing and evaluation* is an iterative process of performance measurement, correction of deficiencies and remeasuring of the resulting performance [194]. This testing process should commence as early as possible during the design process of a system and should be conducted throughout system development, as advocated by the *International Council of Systems Engineering* [49]. Testing and evaluation are not separate from the design process, but rather an inherent, intrinsically critical part of it. Testing and evaluation should also not end when the system is commissioned, the performance should be verified throughout its operational phase until the system is retired [49].

The ability to influence a system's characteristics diminishes rapidly as the development of the TEWA system progresses through its life-cycle stages [187]. It is therefore of utmost importance to ensure that diligent attention is afforded to during the early life-stages when developing a new system. This concern is partly the reason for the adoption this preliminary performance evaluation through this study.

The main purpose of the preliminary testing of a TEWA system, as applied in this thesis, is to identify general design deficiencies and specific conflicts present in the internal algorithms of the system and, consequently, highlighting required corrective action. By following a bottom-up testing approach[2], it is possible to reduce the risks associated with the final commissioned system. According to Sparrius [187], the only way to demonstrate risk reduction in terms of reducing the number of predicted system failures — as a prerequisite for an increase in resource commitment (time and capital) — is through testing and evaluation.

In view of this, M&S may be seen as the key enabler for effectively focussing and executing the testing and evaluation of a complex system such as a TEWA DSS. An M&S approach may be used to predict system performance, identify risk areas, review technology suitability and support the evaluation of the system's effectiveness (*i.e.* suitability and survivability) [194]. Despite the lack of academic references for analysis methods applied within the defense-domain, the authors know that M&S are common tools used for the evaluation of TEWA systems.

[1] For an explanation of what a complex system entails, the reader is referred to §4.1.

[2] Bottom-up refers to an integrated testing approach in which the lowest level components (TE and WA) are tested first, before testing of the higher-level system components [188].

8.2 The Concept of System-of-Systems Analysis

As is the case with many modern systems, the processes of design and evaluation of a TEWA system is highly complex because of the magnitude of the system and the complexities of all the subsystems involved. In the case of a TEWA system, it should be relatively easy to determine certain system specifications, such as whether the system can operate in specific weather conditions or whether the range of the WSs are adequate. To determine whether the system capabilities do in fact assist commanders in performing their tasks — given the many types of possible scenarios and the variety of possible threats — constitutes a significant challenge to system designers. A novel system testing approach is therefore required for this purpose.

The performance of a TEWA system depends sensitively on the synergies between its subsystems which, in turn, gives rise to the emergent properties[3] of the system. These emergent properties cannot be accounted for by following a reductionistic approach only (*i.e.* individually testing the WA and TE subsystems). Such an approach will not enable one to assess whether the entire TEWA system will function as expected.

In order to test a TEWA system effectively, a *system-of-systems* (SoS) approach is required. Proper SoS essentially engineering entails the allocation of functionality to components as well as inter-component interactions [92, p.6–11], in contrast to the analytical approach in which system functionality is explained by combining the understanding of separate components [188]. SoS engineering is a very powerful tool in terms of exploiting synergies between subsystems and in identifying capabilities that no standalone system testing can provide [186].

Prior to identifying a possible SoS engineering evaluation approach, it is first required to understand the principal characteristics that define an SoS. The Army Research Laboratory of the United States of America [185] conducted a literature review on this topic after which the following characteristics were identified as common denominators in the majority of SoS definitions:

Operational independence of elements. The subsystems in an SoS should be independently operational. For the SoS to achieve its purpose, the normal operating modes of the subsystems may be subordinated to the desired operation of the system. Conversely, if subsystems are not directed to achieve a specific central goal, as required by the SoS, the performance of the system may be compromised.

Managerial independence of elements. It is recognised that the individual subsystems of an SoS may be acquired individually and operated independently. Consequently, the subsystems can maintain an operational existence independent of the SoS. The SoS functionality is therefore constrained by the evolution of subsystems in terms of operational capability and management of the SoS.

Evolutionary development. Since an SoS is composed of systems that are integrated to satisfy a higher-level, overall purpose or task, the existence of an SoS can be evolutionary as different purposes and tasks are added, or modified. An SoS may therefore not appear fully formed after its initial development; an evolutionary development process is often required. As the system evolves, it is likely that the overall system functionality can no longer be attributed to the sum of the subsystem functionalities, but that it instead provides a greater functionality which is a result of inter-component interactions.

[3] *Emergent properties* are those properties that derive from the interaction between the elements/components of the system, but cannot be reduced to them [49].

Emergent behaviour. The SoS behaviour and/or functionality cannot be attributed to any one subsystem, but is rather the collective behaviour and synergy between subsystems that define the behaviour of the SoS. The purpose or task of the SoS is therefore achieved by the collection of subsystems. It is possible, however, that as the components interact to produce the emergent properties of the system, some undesirable emergent behaviour may be observed.

Geographic distribution. In all SoSs there is a geographical dispersion of the subsystems, with no limit on the extent of this graphical distribution. The only constraint, however, is that the subsystems are readily able to exchange information, but not necessarily mass or energy. When systems are geographically dispersed, there is no common thread according to which the resulting emergent behaviour may be predicted as the system adapts to changing environments.

For the case of a TEWA system, the threats, WSs and sensors all likely evolve independently of one another and only collaborate through information exchange. The overarching system functionality — assisting the commander in performing his/her task — is also a difficult emergent property to predict. Finally, the subsystems (mainly the WSs and sensors) are distributed over a large geographical area and should be able to readily exchange information as required by NCW doctrine (see §2.1.1). All these characteristics are common denominators of a true SoS.

8.3 Adopted System of Systems Approach

There are two accepted schools-of-thought for analysing problems or, in this case, systems. They include the analytical approach and systems approach [92, 188]. The differences between these two approaches are elucidated with reference to Figure 8.1. In the *analytical approach*, the system is first decomposed into several parts, as illustrated in Figure 8.1 (i). After sufficient separation, as determined by the system designer, the functionalities and properties of the components are investigated separately, as illustrated in Figure 8.1 (ii), and the individual understandings are combined in order to derive an understanding of the system as a whole, as illustrated in Figure 8.1 (iii). In this method, the testing process does not take into account the environment as a potential functionality-driver. The *systems approach*, on the other hand, involves first identifying the environment[4] within which the system is to employed, as illustrated in Figure 8.1 (iv). The properties and functionalities of the containing whole — which comprises the environment together with the system — is investigated and explained, as illustrated in Figure 8.1 (v). Subsequently, the system is evaluated in terms of its function and role within the context of its environment, as illustrated in Figure 8.1 (vi). The result is an approach which accounts for the crucial emergent SoS functionality.

Adopting an analytical testing approach for an SoS, such as a TEWA system, is doomed to failure; both in terms of understanding the SoS functionality and identifying its limitations. The reason for this failure is that most TEWA systems are sensitively scenario-dependent and their constituent parts of the system are, as mentioned, driven by inter-related purposes and embedded concepts, all of which need to be studied in conjunction with a commander's intent and the relevant *Concept of Operation* (CONOPS)[5]. An analytical approach, in which the system

[4]The environment refers to all elements constituting the GBAD environment — that is, the physical environment as well as all the auxiliary systems, as described in §2.4.

[5]A CONOPS is a document that describes the overall system characteristics and required performance from a user's perspective. This document is also used to describe the organisation, objectives and mission from an integrated systems point of view and is, furthermore, used to communicate the system characteristics (quantitative

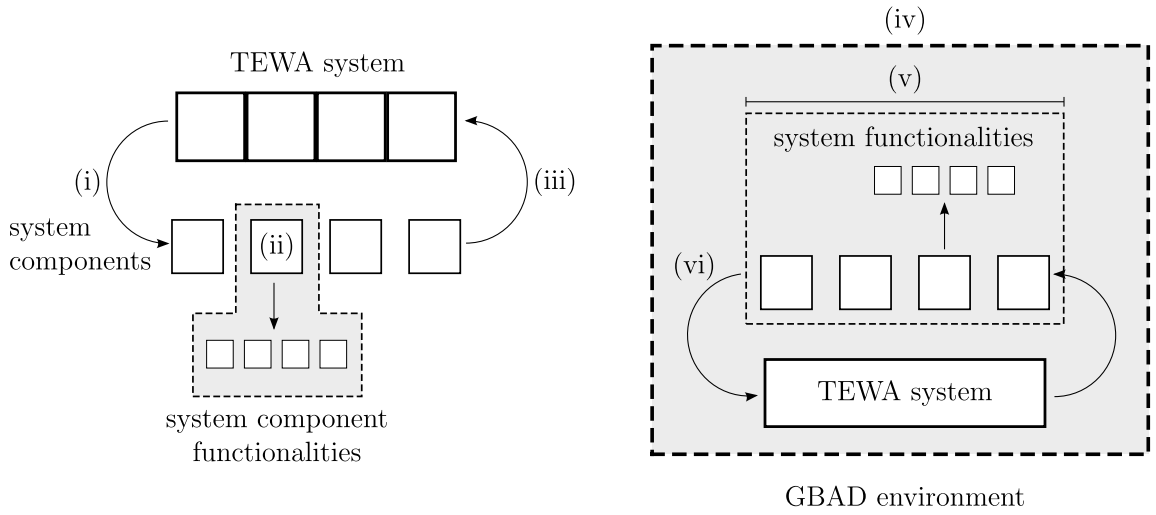


FIGURE 8.1: *Comparison between the analytical (left) and system (right) testing approaches.*

is decomposed for analysis purposes, will therefore result in unreliable and untraceable[6] results.

A more suitable alternative when addressing the challenges associated with an SoS, is to develop a scenario-dependent simulation model and adopt an SoS analysis approach, as illustrated in Figure 8.1. The simulation environment may then be used to evaluate the TEWA system in its totality. Through this approach, the understanding of the separate system components, together with the embedded concepts and inter-related purposes of the components, may be simulated in order to arrive at a composite and holistic view of the system's performance. To a certain extent, the evaluation will address the variety within — as well as the scale of — a GBAD scenario which are both important considerations that can significantly influence system performance. Because of the many stochastic elements in the simulation paradigm (SSHPs, heuristic algorithms and stochastic-related track generation), such a scenario-dependent performance evaluation paradigm can only state probabilities (or confidence levels) associated with certain possibilities (*e.g.* events), similar to the SoS approach developed by [185].

A good example of the application of the aforementioned considerations is the system testing approach applied by Sommerer *et al.* [186]. They applied diligent SoS engineering in the analysis of the Aegis BMD system. A high-fidelity simulation model was designed to aid in the testing processes. The "way-ahead" (*i.e.* future-plan) M&S process planned by Sommerer *et al.* is shown in Figure 8.2. This figure served as guidance for the understanding of the testing procedures briefly recounted here, and also as an understanding of the complexities and opportunities associated with adopting an SoS approach.

Sommerer *et al.* [186] adopted a testing process to assess the system performance as well as the effectiveness to which it contributes to mission success. The process starts at the element (*e.g.* WS and sensor) level after which the element functionalities are aggregated to form the system or mission level functionality. The process of achieving this aggregation of subcomponent functionalities, by accounting radio-frequency sensor data, kill-chain elements and sensor performance *etc.*, is illustrated in Figure 8.2. A major challenge throughout their M&S approach (it is still ongoing) is finding the right balance between modelling fidelity and simulation responsiveness in order to sufficiently capture the dynamic nature of the system. Such an integrated vision of

and qualitative) to stakeholders [89].

[6]It would be difficult to trace defects in the final system if the emergent properties were not accounted for during the testing phases.

an effective SoS testing approach should provide a clear understanding of the influence of the different key performance drivers on the system performance.

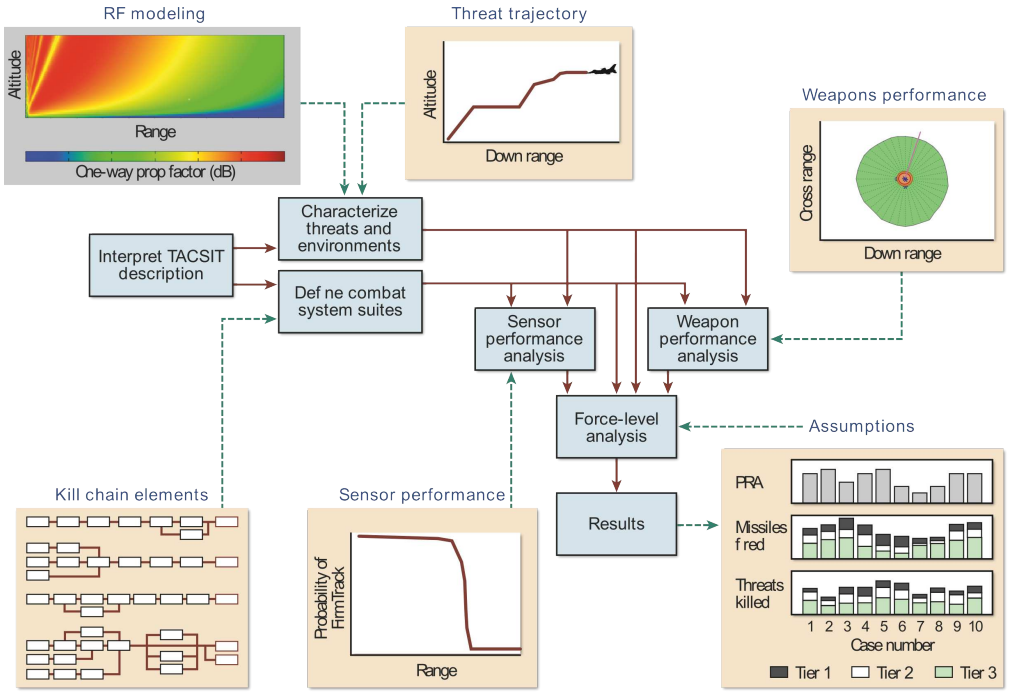


FIGURE 8.2: *The "way-ahead" M&S process for the Aegis BMD system [186].*

8.4 Performance Evaluation Approaches

Three common approaches adopted to evaluate the performance of TEWA-related systems were identified during the literature review conducted during this study. These approaches are prototype evaluation in conjunction with end-users, single scenario evaluation and batch-simulations. The limitations and advantages of each of these approaches are described in this section within the context of a TEWA system, referring specifically to how, and when, these methods should be applied during a system's life-cycle.

8.4.1 Prototype Evaluation in Conjunction with End-Users

The preferred method (in terms of building confidence) of testing whether a TEWA DSS adheres to the system specifications, is to conduct live flight tests [92]. The complex nature of a TEWA system together with the high cost of such an approach, however, makes it intractable to conduct flight tests merely for the purposes of system evaluation during the early stages of system development. System designers are therefore forced to rely on M&S to perform trade-off studies and sensitivity studies for the purposes of system evaluation [186].

Furthermore, flight tests alone do not provide insight into scenarios that were not actually tested [65]. It is not practical to test all the possible engagement scenarios with full-scale flight tests. Moreover, it would not be feasible to test enough replications of a specific engagement scenario live in order to determine the system's performance with statistical significance. Because of the confidential nature of this research area, historical data on flight tests, from which system

designers can gain insight into the performance characteristics of existing TEWA systems, are very rare.

Even so, in order to fully understand the success of a DSS, it is required to include the end-users into the performance evaluation stage. The insights provided by these end-users (operators of the system) are crucial in developing a highly efficient system. The prototype evaluation of a DSS is therefore an attractive evaluation technique during the final stages of system development. Testing a system in conjunction with the intended end-users in an environment similar to the actual operational environment should provide an accurate indication of whether the system is indeed useful, as well as identify weaknesses and strengths in a bid to further improve the system. As mentioned in §8.2, this evolutionary development is, indeed, a characteristic property of an SoS.

This prototype evaluation method has been applied successfully by Smith *et. al.* [184] as part of the TADMUS programme. The end-users were divided into two groups — the one group had access to the DSS which aided the TE process of the operators, while the other group performed without it. A total of 15 teams were created of which eight teams had access to the TADMUS DSS. The same TEWA scenario was presented to both groups. At critical times during the scenario, the teams were asked to provide a prioritised list of the most threatening targets and their decisions were evaluated against the "correct" answers as calculated by the TADMUS system algorithms. The results of the study indicated that the operators who had access to the DSS performed significantly better at detecting deceptive threats.

The problems associated with practical implementation is that this method requires many resources and significant time to generate useful results [184, 186]. The success of this approach also depends on the HMI implemented — even more so than on effective algorithms [67]. Furthermore, a suitable high-stress, dynamic scenario must be created, or otherwise the results will not be valid for real situations. System designers are therefore "forced" to utilise M&S tools in the evaluation of the performance of TEWA systems; especially during the early phases of a system life-cycle.

8.4.2 Single Scenario Evaluation

Another approach towards DSS performance evaluation involves the construction of a single scenario. The algorithms may then be implemented in the scenario and the results are evaluated in that context. Simulations need to be run for numerous iterations in order to derive probabilities estimates associated with certain events, as stated in §8.2. In this way, it is possible to achieve a rudimentary understanding of the algorithms' performance and identify opportunities for improvement of their working. The results of such an approach can be validated by showing the response to domain experts and determining whether the outputs correspond to their intuition [218].

When applying this approach, it is still required to determine if the outcomes of the simulation provides the expected and, to a certain extent, realistic results. One method to gain such confidence, is to generate scenarios for which the correct solutions are known (*e.g.* a historical scenario). Proving that the system functions as expected in known circumstances is helpful and, in fact, desirable, but it does not prove that the system will generate accurate, reliable outputs for all circumstances. Although a single simulation output may match a known solution, it does not necessarily prove the successful functioning of the system. To ascertain that the system will work for all situations, it is required to execute a large number of different simulations and validate their outcomes. The outcomes may be validated by comparing the results with expert

opinions (*i.e.* by presenting the same situational problem executed by the simulation to military experts and comparing their answers to that of the simulation).

This single scenario approach may also be used to test the performance of different algorithm combination designed to achieve the same goal (*e.g.* different combinations of TE and WA algorithms). For instance, numerous algorithms may be applied to the same scenario and the results compared. Johansson and Falkman [95] followed this approach by developing two algorithms and comparing their results in an identical scenario. Not only were the results analysed, but also the ability of the algorithms to adapt to changes, such as missing or incomplete information, and abrupt threat value changes [95].

The single scenario method should require less resources than the aforementioned prototype evaluation method. This approach is therefore more suitable during the earlier iterative design stages of a project, especially when the developed system does not yet exist. When applying this approach, it is important that the results are carefully considered and that the limitations of the specific test scenarios are understood, since the results only show how the system performs in the particular scenario tested. Therefore, instead of evaluating whether the system can be used in generic situations, this approach merely clarifies specific strengths and limitations of the models (and implemented algorithms) and highlights particular characteristics of the system. As such, careful consideration should be taken when selecting the scenario(s) for which the tests will be conducted.

8.4.3 Batch-simulations

The use of batch-simulations entails executing a large number of scenarios so as to be able to perform statistical analyses on the conditions for which the algorithms perform well. This method builds upon the single-scenario approach, since each tested scenario would need to be tested sufficiently in order to account for SoS effects. In order to apply this method, performance measures have to specified for the use in the ensuing statistical analysis, since it is typically not possible for a human analyst to analyse all the scenarios separately by considering the outcomes individually.

A problem experienced in many batch-simulation studies in the open-literature where TE or WA are evaluated[7], is that the majority disregard the complexities defining the CPM problem, as mentioned in §6.1. Most studies use a combination of straight-path threats with no reference to realistic tracks [135], homogeneous WSs (with constant SSHP and no different types of WSs) [125], simplified WA constraints set-ups and randomly generated scenarios [93, 205]. Consequently, it is both difficult to draw significant conclusions regarding the functioning of the TEWA system and to compare the results of different studies, as motivated in §6.2. Because of all the possible different formulations of this problem (TE and WA combined), the focus can only be on evaluating the effectiveness of the implemented set of algorithms with certain types of data-sets (scenario set-up, WS properties *etc.*).

Because of the scarcity of realistic scenario-related information, this method is difficult to apply in an academic setting. Batch-simulations should only be performed if a high-level of confidence has been reached with respect to the effectiveness of the algorithms. As such, a realistic single-scenario approach, where algorithms can be tested sufficiently, should be seen as a prerequisite before applying this approach.

[7]Studies where a TEWA system is evaluated, with interoperable TE and WA subsystems, could not be found. The focus is generally on one of the two subsystems. In the case where WA is evaluated, a rudimentary TE approach (a single TE measure, generally distance) is generally implemented and *vice versa* [135, 215].

8.5 Performance Evaluation Metrics

In this section the use of four metrics is proposed for evaluating the performance of a TEWA DSS within a simulation modelling paradigm. These metrics may serve either as *absolute* or *comparative* evaluation measures in the sense that the value of a metric may quantify the suitability of assignments proposed by a TEWA DSS in a specific scenario in absolute terms, but may also be used to identify limitations present in its constituent algorithms, by comparing the metric values for different scenarios in a relative manner.

These metrics may be employed in both the single-scenario evaluation and batch-simulation modes, as detailed in §8.4. For the single-scenario case these metrics may be used to quantify the extent of the influence of stochastic elements on the performance of the TEWA system while for batch-simulations it should be possible to determine the conditions under which the algorithms behave poorly by evaluating the metric values of different scenarios. For an comparative analysis, the metrics may be employed in two different ways — (i) different algorithm combinations (*i.e.* TE and WA) may be evaluated collectively in order to determine their performance for the tested scenario, thereby allowing limitations to be identified; (ii) by keeping the algorithms combination constant, different scenario setups may be tested to determine the conditions (GBAD scenario setup and/or threat numbers and attack manoeuvres) under which the algorithms behave adequately or unsatisfactorily. Finally, these metrics may be used to perform a sensitivity analysis with respect to the implemented algorithms, thereby providing valuable insight into the functioning and limitations of the TEWA system as a whole.

There is often a misconception about the term *performance metric*. A metric is a standard definition of any measurable quantity, while a performance metric goes further by gauging some aspect of a system's performance. For a performance metric to be successfully employed, it must adhere to certain requirements — evaluating a performance metric should be achievable in a reliable, repeatable and consistent manner, independently of the pressure to drive performance [122].

The performance metrics proposed in this section include *survivability*, *economy*, *engagement-effectiveness* and *adaptability* metrics. The metrics are explained and motivated in the following sub-sections.

8.5.1 Survivability Metric

The main role of the WSs in an air-defence scenario is to protect the DAs. Therefore, *survivability* of the DAs is an important criterion for measuring the performance of a TEWA system. Johansson [72] suggested the use of the survivability metric

$$S = \frac{\sum_{j=1}^{D} \omega_j u_j}{\sum_{j=1}^{D} \omega_j}, \tag{8.1}$$

where D is the number of DAs in a simulation performance evaluation environment, ω_j is the importance value associated with DA j, and u_j is a binary variable which assumes the value 1 if DA j survives, and the value 0 if it is destroyed by an aerial threat. Hence, the survivability S is the ratio between the protection value of surviving assets to the total protection value of all the assets.

8.5.2 Economy Metric

The metric in (8.1) does not penalise the engagement of superfluous aircraft as unnecessary engagements. The introduction of an *economy* metric may, however, account for the cost of engagement by each WS — thereby penalising unnecessary engagements. The economy metric

$$M = \sum_{i=1}^{W} \left(C_i \sum_{j=1}^{T} x_{ij} \right) \tag{8.2}$$

is proposed for this purpose, where C_i denotes the cost of one burst or round of ammunition for WS i, x_{ij} is the number of times WS i engages threat j, and W and T are respectively the number of WSs and threats. Hence, the economy metric M represents the total WS capital expenditure, based on ammunition used, associated with an engagement strategy.

This metric is not proposed to be a ratio, since it may also serve the purpose of indicating the scale of a specific scenario. A large value generally means that the tested scenario is on a larger scale than, for instance, a scenario associated with a smaller value.

8.5.3 Engagement Effectiveness Metric

It is preferable to destroy high-value aerial threats, especially in an ongoing conflict. In this context, the value of an aerial threat may be interpreted as its ability to cause considerable damage to important classes of DAs. The next metric, *engagement effectiveness*, is designed to reward the successful engagement of high-value threats. The value of a specific aerial threat may be determined during the pre-deployment phase of a mission and programmed into the TE subsystem. As stated above, it is desirable from an economic point of view not to engage superfluous targets. During an ongoing conflict, however, it may be beneficial to destroy these high-value threats even if they do not pose an imminent danger, in a bid to ensure that these threats do not return to attack DAs in the future.

Furthermore, a critical performance-related problem is the engagement of friendly and/or civilian aircraft (see §1.2). Modern wars have proved that both military and commercial aircraft can and do co-exist in the aerial environment. The engagement effectiveness metric may also be used to penalise friendly engagements by assigning a large negative importance value to friendly and commercial aircraft. The engagement effectiveness metric is given by

$$E = \frac{\sum_{j=1}^{T} \nu_j e_j}{\sum_{j=1}^{T} \nu_j}, \tag{8.3}$$

where ν_j denotes the importance value associated with destroying threat j and e_j is a binary variable assuming the value 1 if threat j is destroyed, or the value 0 if the threat survives. The value of ν_j may be interpreted as the perceived value that the enemy is most likely to assign to an aircraft — the more important the aircraft, the higher the value. The engagement effectiveness E is the ratio of the importance value of destroyed threats to the total importance value associated with all threats encountered throughout the engagement.

8.5.4 Adaptability Metric

The final metric attempts to quantify the *adaptability* of a specific engagement strategy. This metric is given by

$$A = \min_i \left\{ A_i - \sum_{j=1}^{T} x_{ij} \right\}, \tag{8.4}$$

where A_i denotes the initial amount of ammunition available to WS i and the remaining parameters have the same meanings as before.

The metric A is designed to measure the propensity of an engagement strategy to maximize the number of times that a WS is available for re-engagement after the proposed assignment, thereby ensuring that as many WSs as possible are reusable in future engagements. By using ammunition more effectively during an engagement, WSs on the ground will be more adaptable to changing conditions, such as responding to newly detected threats and performing follow-up engagements.

8.6 Practical Simulation Characteristics

In addition to the metrics introduced in §8.5, certain periphery characteristics also need to be considered in order to ensure a successful functioning TEWA system. These include the time required to generate WA allocation suggestions and the memory storage requirements of the internal algorithms of the system.

Scenario space saturation

An important consideration of a specific GBAD scenario setup and the constituent TEWA algorithms, are the number of threats that can be countered effectively. This characteristic is a function of both the TEWA algorithms and the specific setup (positions and numbers of DAs, WSs and threats). A simulation performance evaluation framework may be used to ascertain the scenario space saturation characteristics of a specific algorithms-scenario combination. On the contrary, however, the risk of mass air-assaults is decreasing, according to Hutchings and Street [86]. As such, more value should be obtained by shifting developmental efforts towards countering high-technology, jamming, stealth and long-range stand-off WSs.

Extent of WA switching

Drastic changes in threat values, as was identified in Chapter 5, may have a detrimental effect on the functioning of the WA subsystem. These threat value variations are undesirable from a C2 decision-making perspective, since these threat value variations will affect the suggestions provided by the WA subsystem. This detrimental effect may materialise as a switching of WA (WS-to-threat assignments) recommendations. This switching may become a problem if the WA recommendations changes rapidly and excessively over a small subset of consecutive time-stages [118]. This kind of behaviour may be ascribed to the changing threat values as well as the various WA stochastic elements (SSHP values and heuristic solution methodologies).

Consequently, this switching may cause confusion on the part of the FCO and thereby result in the FCO questioning the credibility of the TEWA system's results. This, in turn, may lead to disagreement uncertainty, as descried in §7.3.3. This confusion may, subsequently, result in the operator relying on his own judgement by resorting to heuristic

rules (the reader is referred to §7.3.2 for a more in-depth discussion) instead of relying on the TEWA DSS, thereby resulting in sub-optimal responses.

In order to mitigate this problem, the use of threshold values was suggested in §6.4.1. A new WA solution should only be accepted if the new solution is deemed significantly improving (*i.e.* if the new objective function value exceeds a specified threshold tolerance over and above the current objective function value). Allouche [9] suggested the use of smoothing techniques — specifically Kohonen's self-organizing maps[8] — to smooth the trajectory of a threat in an attempt to render the threat values more gradually changing. Allouche's focus was on missiles in a sea-based environment, but these methods may be tailored and applied to a GBAD environment.

Required computational resources

It is important to ensure that the FCO has enough time to utilise the results generated by the WA subsystem. Also, depending on the environment in which the TEWA system is implemented, there might be memory restrictions (*e.g.* micro-controller architectures or FPGA restrictions). There is often a trade-off between the time complexity and memory complexity of an internal TEWA algorithm — increased memory consumption generally corresponds to faster execution times, and *vice versa*. The increased need of advanced visualisation capabilities (see §7.4.2) also present further challenges to the system designers in terms of being able to ensure that information is updated continuously and visualisations are rendered quickly, so as to assist the operators with their decision-making cycle.

Although MATLAB is used in this thesis — which is a high-level programming language that is known to perform slower when performing certain operations — the execution times of a *Matlab* simulation may serve as an initial estimate by which to understand the scope of required computational resources. In practice, however, when the system is implemented using a lower-level programming language (C or assembly), the execution time may decrease by several orders of magnitude.

8.7 Chapter Summary

This chapter opened in §8.1 with an introduction to the evaluation of TEWA systems, after which the underlying concepts of an SoS approach was explained in §8.2. After understanding the notion of an SoS, a possible methodology to adopting an SoS approach was elucidated in §8.3. This approach, essentially, requires the adoption of a scenario-dependent simulation model in an SoS analysis context, as opposed to an analytical approach involving unrealistic hypothetical scenarios. Because the outputs of a TEWA system strongly depend on the specific GBAD setup as well as the working TEWA algorithms, such a systems approach to performance evaluation is expected to provide more significant results.

After a possible approach to performance evaluation had been elucidated, three possible scenario-dependent performance evaluation approaches were described and qualitatively evaluated in §8.4. These scenario-dependent approaches include prototype evaluation with end-users, single scenario evaluation and batch-simulations. Four TEWA-specific performance evaluation metrics were also proposed in §8.5. Each metric was designed so as to provide different insights into the results generated by a TEWA simulation. The metrics include a survivability metric, an economy metric, an engagement effectiveness metric and an adaptability metric. This chapter closed in §8.6 by clarifying three practical considerations that should be borne in mind when using a TEWA performance evaluation simulation framework.

[8] This is a special case of neural networks, *i.e.* a type of self-organizing map.

CHAPTER 9

Worked Example

Anomalies are things that either do not happen and should, or that do happen and should not.

— Herman Kahn

Contents

The purpose of this chapter is to demonstrate the workability of the concepts, strategies and algorithmic models reviewed and presented in this thesis. In addition, the demonstration of the constituent TE and WA models — applied to a new GBAD scenario — also serves as an additional verification step to ascertain the correct functioning of the algorithmic models. This demonstration is achieved by means of a comprehensive, near-realistic, but hypothetical scenario created by Potgieter [155] in cooperation with a military expert, Visser [219].

The chapter opens with an overview of the scenario setup considered, and this is followed by a description of the attack profiles of the attacking aircraft and the positioning of WSs in the scenario. The physical elements within the simulation are modelled as described in Chapter 4. After having gained an understanding of the functioning elements within the simulation environment, the outputs of the TE subsystem, as explained in Chapter 5, is provided and interpreted. This TE information is, in turn, used to generate an allocation suggestion of WSs to threats, as described in Chapter 6. The chapter closes with the calculation of the system performance evaluation metrics presented in Chapter 8.

9.1 Experimental Approach

As explained in §8.3, a constructive discrete-event simulation framework is more suitable for the performance evaluation of a complex TEWA DSS in its totality than a purely reductionistic (analytical) approach where TE and WA are tested in isolation. Such a discrete-event TEWA simulation has strong parallels with the notion of wargaming [104]. The steps in a simulation study, as detailed in §4.1.2, are therefore similar to the required steps of a wargaming exercise. These steps are paraphrased below:

1. *Define the problem and objectives of the experiment.* The purpose of this simulation is to demonstrate the workability of the concepts, strategies and algorithmic models reviewed in the previous chapters of this thesis.

2. *Prepare input data.* In the case of this experiment, the inputs include details on the GBAD scenario setup — positions of DAs, WSs as well as the input coordinates of the threats that are used for the flight path generation. The preparation of the inputs also includes detailing the properties of all the simulation model entities. The modelling approaches described in §4.2.1 are used for the representation of the simulation model entities (*e.g.* DAs, sensors, WSs and threats).

3. *Execute a preliminary experiment and execute production runs.* The preliminary experiment includes a performance appraisal of the TEWA system described in the previous chapters in the context of the illustrative example introduced in §4.6. Part of the preliminary experiment entailed a detailed analysis of the TE and WA models implemented so as to validate and verify their functioning. Different limitations of the WA and TE subsystems were also identified throughout the thesis and mitigation strategies were implemented. The final production runs are used for the performance evaluation of the system in this chapter.

4. *Analyse simulation outputs and system performance measures.* After completion of the production runs, it is possible to commence with the performance evaluation of the constituent algorithmic model combinations. As explained in §8.3, the outcome of a TEWA simulation performance evaluation study can only be certain possibilities associated with certain events (for example, the set of possible TEWA-cycles during which a threat was successfully engaged). From this evaluation, areas for further improvement may be identified.

9.2 GBAD Scenario Deployment

The first step in the simulation performance evaluation is clarifying how the physical GBAD system entities are portrayed in the simulation experiment, as was identified in §2.4. The placement and properties of the WSs, DAs and threats are described in this section. A top view of the hypothetical GBAD scenario is shown in Figure 9.1. In this figure the threat paths of four threats are indicated by the four splines, the positions of two DAs (A and B) are depicted by black squares and the WSs (1–11) are depicted as black triangles. Furthermore, the threat tracks correspond to an overall time frame of 126 seconds. The black dots on these aircraft tracks indicate the respective TEWA-cycle time stages (τ) and serve as reference points for the explanations to follow. A side-view of the scenario is also provided in Figure 9.2 in order to illustrate the altitudes reached during the execution of the threats' attack manoeuvres.

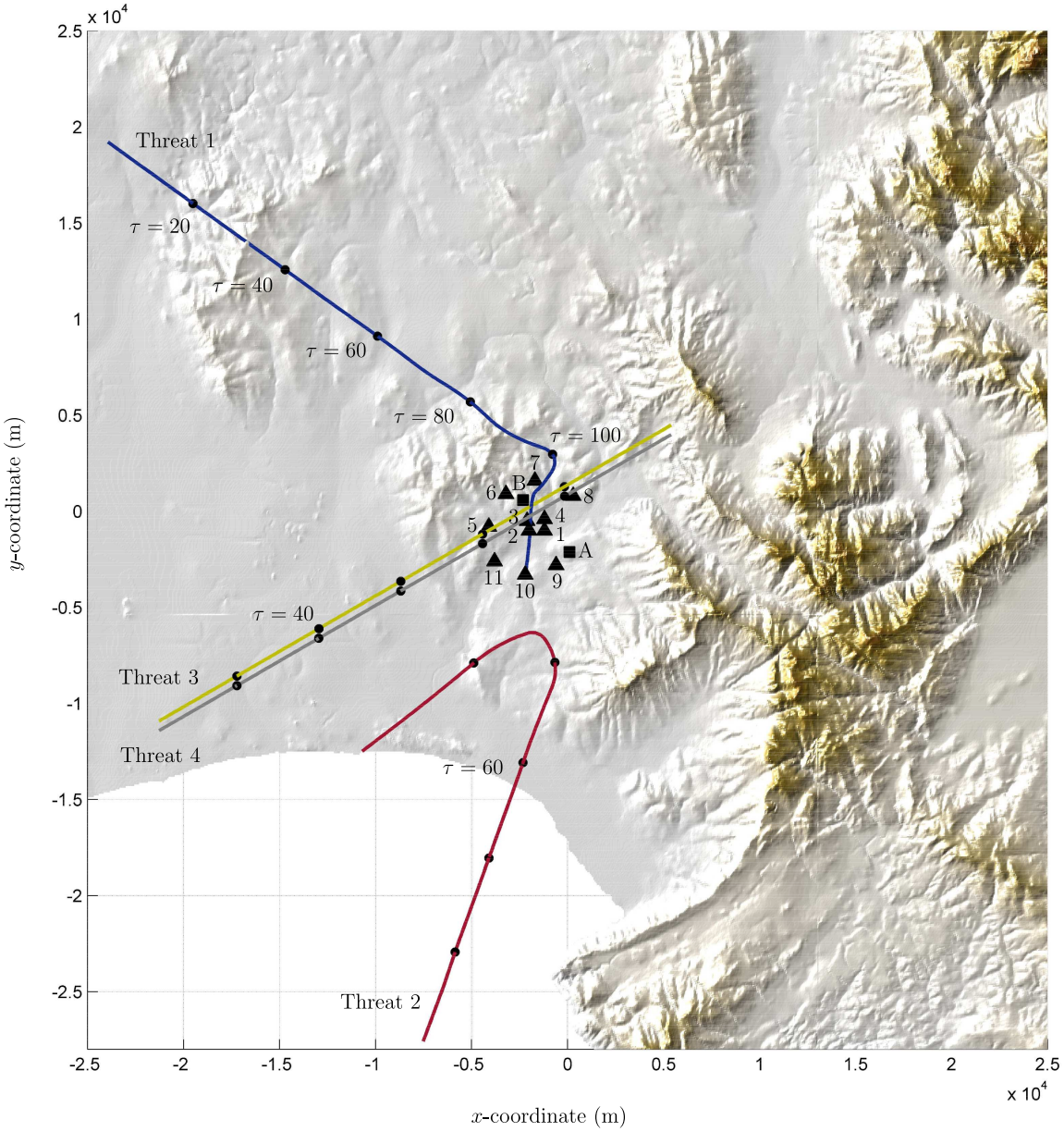


FIGURE 9.1: *Top view of the GBAD scenario considered as hypothetical worked example in this chapter.*

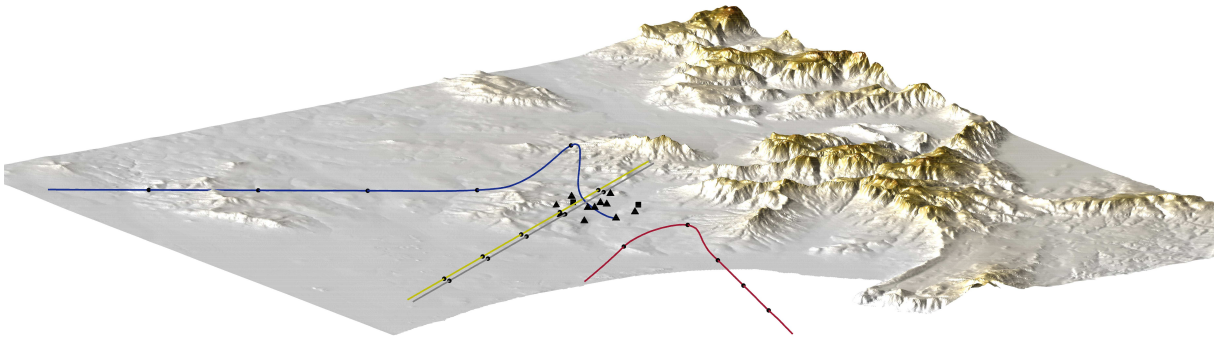


FIGURE 9.2: *Side view of the GBAD scenario considered as hypothetical worked example in this chapter.*

9.2.1 Defended Asset Placement

Consider two DAs — a command centre and a hangar — which are deemed very important to protect by the defending force. The priority values[1] of these DAs and their positions are listed in Table 9.1.

TABLE 9.1: *Properties of the DAs within the hypothetical GBAD scenario.*

	DA A	**DA B**
DA type	Command Centre	Aircraft Hangar
Priority Value	90	50
x-coordinate (m)	100	$-2\,315$
y-coordinate (m)	$-2\,115$	577
z-coordinate (m)	0	0

9.2.2 Weapon System Placement

The DAs indicated in Table 9.1 are afforded protection by a near-impenetrable layered arrangement of WSs, which is typical of a GBAD setup (see §2.4.4). The ground-based defenders have at their disposal, four *Close-in Weapon Systems* (CIWSs) and seven *Very Short-range Air Defense Systems* (VSHORADSs). These WSs and their properties are listed in Table 9.2. The SSHP volumes of these WSs are unique for each type of WS and are only a function of stand-off range to a threat, as described in §4.2.3.

9.2.3 Threats Attack Profiles

Suppose the *Opposing Force* (OPFOR) has one type of fixed-wing aircraft at its disposal (a Gripen[2]), capable of executing three types of attack manoeuvres (a pitch and dive manoeuvre,

[1] Recall from §4.2.2 that these priority values quantify the relative importance to the defending force of protecting the DAs and is typically a function of a DA's repairability, vulnerability and strategic importance.

[2] The SAAB Gripen is a multi-role fighter, in operation with the South African Air Force, with an extensive range of air-to-air, air-to-surface and reconnaissance mission capabilities. The Griphen is also equipped with an array of sensors and advanced HMIs so as to enable it to thrive in a NCW environment [173].

TABLE 9.2: *Properties of the WSs within the hypothetical GBAD scenario.*

	WS 1	**WS 2**	**WS 3**	**WS 4**
WS type	CIWS	CIWS	CIWS	CIWS
x-coordinate	−2 000	−2 100	−1 200	−1 200
y-coordinate	−1 000	−500	−400	−1 000
	WS 5	**WS 6**	**WS 7**	**WS 8**
WS type	VSHORAD	VSHORAD	VSHORAD	VSHORAD
x-coordinate	−4 100	−3 200	−1 700	300
y-coordinate	−800	900	1 600	800
	WS 9	**WS 10**	**WS 11**	
WS type	VSHORAD	VSHORAD	VSHORAD	
x-coordinate	−600	−2 300	−3 800	
y-coordinate	−2800	−3 300	−2 600	

a toss bombing manoeuvre or a low-level fly over) during which two types of ordnance can be launched (rockets or guided ballistic bombs). Suppose furthermore that all threats have been interrogated for IFF, but that no identifiable response was received. Consequently, the threats cannot be identified positively as friendly or hostile and are, as a result, flagged as unknown. The TE models are therefore solely responsible for predicting the opportunity, capability and intention of the various threats. The formative element combinations of the threats, together with their targets and *Weapon-release Points* (WRPs), are given in Table 9.3.

TABLE 9.3: *Formative element combinations and additional information on the threats.*

	Type	Attack Profile	Weapon Envelope	Target	WRP
Threat 1	Gripen	Pitch and Dive	Unguided Rockets	DA B	113
Threat 2	Gripen	Toss Bomb	Guided Bombs	DA A	82
Threat 3	Gripen	Low-level Fly Over	None	None	N/A
Threat 4	Gripen	Low-level Fly Over	None	None	N/A

The OPFOR is executing an attack on the GBAD system employing these four aircraft with intentions described below. Threat 1 is approaching from the north west and begins execution of a pitch when it is approximately 7 500 m from the hangar (DA B). It turns in towards DA B in order to execute its pitch and dive attack manoeuvre after which it delivers its rockets at an approximate stand-off range of 800 m (TEWA-cycle 113).

The aircraft approaching from the south west, Threat 2, travels along a trajectory that does not cross over the DAs. The threat, however, pitches and turns towards the command centre when at a range of 9 000 m from DA A. The guided ballistic munitions of Threat 2 are released when it is at an approximate stand-off range of 8 000 m (TEWA-cycle 82) from the command centre, thereby completing its toss bomb attack manoeuvre.

The last two threats — Threats 3 and 4 — both serve as decoys and execute low-level fly overs. These two threats approach from the south west and approach at a constant low altitude over the defended area.

9.3 Threat Evaluation Application

Prior to WA, the threat values of all the threats first have to be estimated. This threat value fusion process is crucial for effective TEWA functioning, since the subsequent engagement decisions are mainly based on the resulting threat values. If this subsystem generates inadequate outputs, the downstream system functionality will be adversely affected. The threat values of a single execution of the simulation model, associated with each threat-DA-DM triple, are shown in Figures 9.3.

For all the threats the distance-related threat value increases almost linearly as the threats approach the DAs and decreases linearly as the threats regress from the defended area. At time 100 Threat 1 is pitching before executing its pitch and dive manoeuvre.

The high-frequency noise superimposed on the threat values, specifically noticeable in the CPA threat values, is as a result of the randomly generated spheres around the input coordinates. Recall from §4.2.4 that the spheres are used to simulate measurement inaccuracies and pilot deviation from the intended standard manoeuvre track. Although the noise may seem significant for the threat-DA-DM threat triples, the resulting system threat values are more robust in terms of the fluctuations in threat-DA-DM threat values when following the hierarchical aggregation fusion approach, as may be seen from the system threat values shown in Figure 9.5.

As described in §5.3, the first step is to fuse the threat values in order to obtain the threat-DA threat list. These threat values are depicted in Figure 9.4. During this fusion step the KOB-scaling is enforced as described in §5.3.2. In the threat-DA threat lists four sudden rises are observed during the threat value calculation process for each threat. For the threat value of Threat 3 and DA 1, the four drastic threat value changes occur at approximately times 10, 60, 92 and 124. Similar changes may be observed for the other threat-DA pairs. The first of these increases occurs when the threat enters the AOR and is, as such, within range of the sensor systems and susceptible to TE. The second and fourth drastic threat value changes are ascribed to the KOB-scaling. The second increase is generally when a threat crosses the KOB, whereas the fourth decrease is when a threat exits the prohibited zone enclosed by the KOB. The third decreasing jump — between the two KOB-related scalings — is as a result of the "switching-off" of the passing distance-related DM. This DM is only active if the threat is heading towards a DA.

Rapid changes in the threat values may, however, be of concern to the effective functioning of the TEWA system, as described by Lötter and Van Vuuren [118]. The rapid changes in threat values, described above, may result in switching of weapon assignment recommendations. This type of switching is a typical emergent property of TEWA systems and is something that must be fully understood before implementing such a system.

The final step of the fusion process is the computation of the system threat values, as described in §5.3.3. Recall that these system threat values represent the danger that a specific threat poses to the entire defended system and are, in turn, used for WA purposes. The system threat values of the four threats are shown in Figure 9.5. From this figure it is clear that the threat values do, indeed, provide realistic treat estimates of the various threats. The weapon release for Threat 1 occurs at time 113, when the threat has a threat value of 0.987 and, for Threat 2, weapon release occurs at time 82, when the threat has an associated threat value of 0.917. It is therefore encouraging to note that the system threat values of these threats are at their highest levels during the weapon release stage.

A possible observation of concern, however, is that the two decoys have relatively high threat values considering that they are not intending to attack the DAs. Nonetheless, the decoys are, in

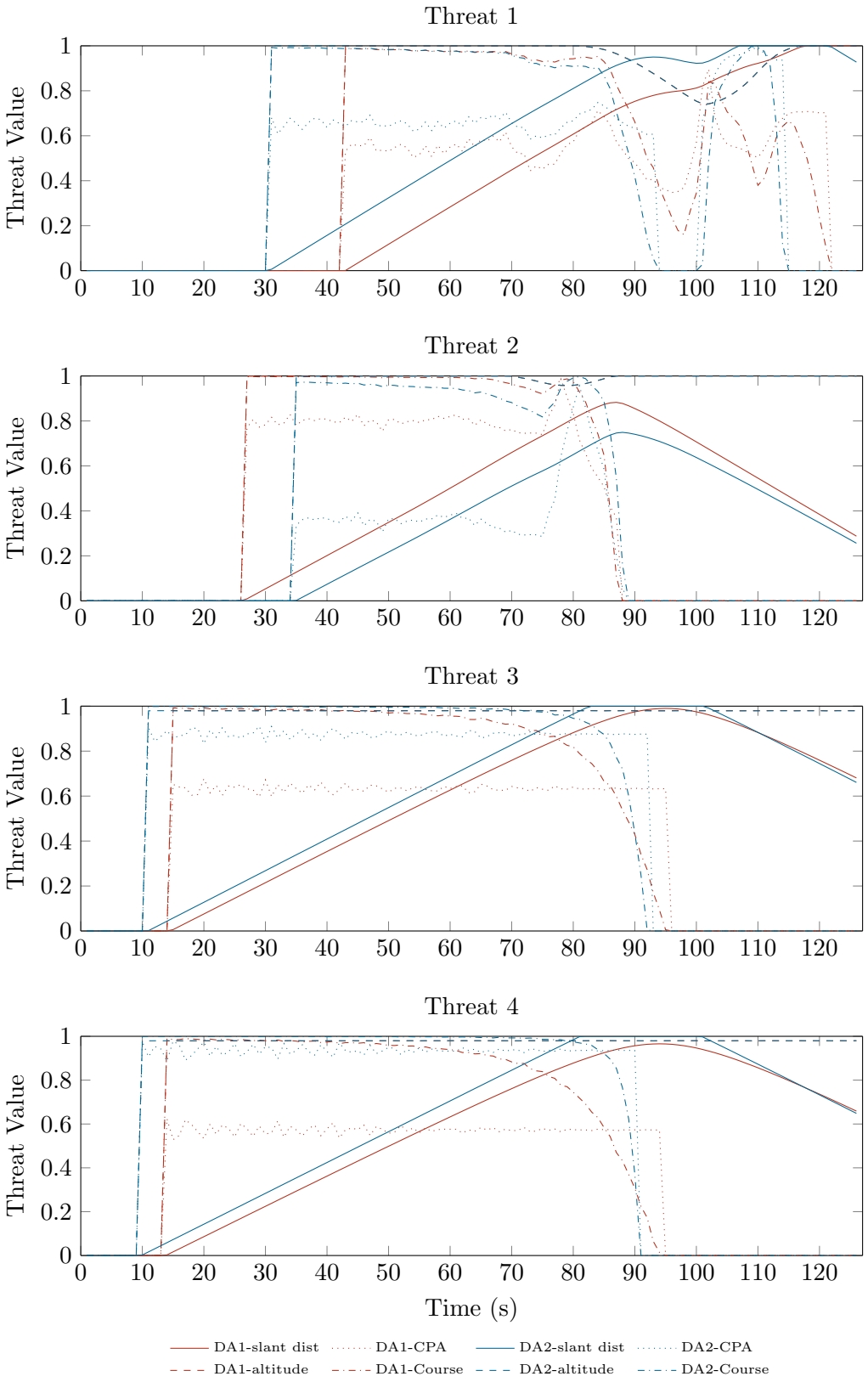


FIGURE 9.3: *Threat values, distinguished in terms of threat, DA and DM.*

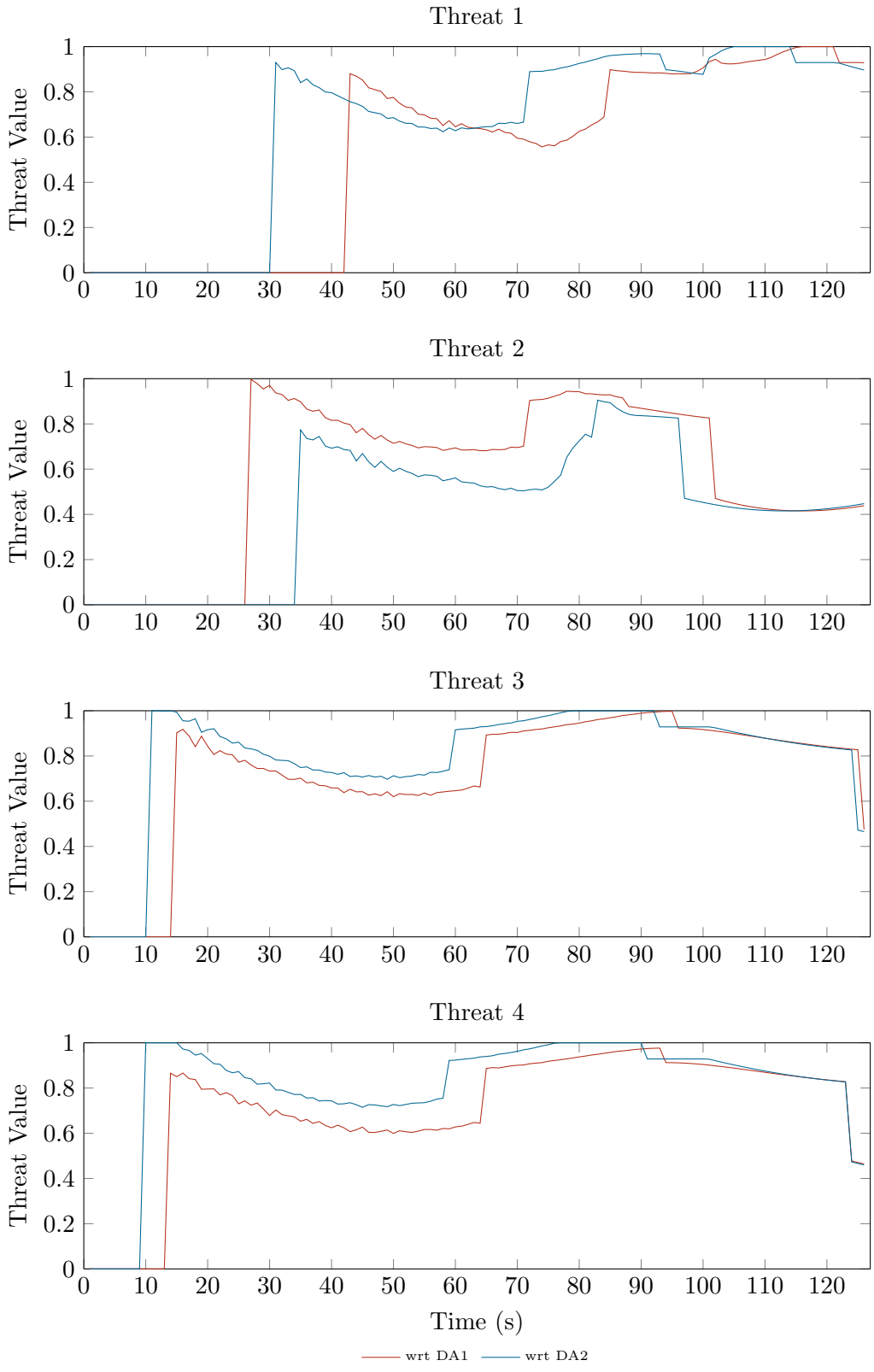


FIGURE 9.4: *Threat-DA pair threat values.*

fact, meant to do exactly that — deceive the TE models. If the DM threat values are evaluated in conjunction with flagging models, indicating that no drastic kinematic changes are observed, the FCO should be able to make a more informed decision. Other information contained in the FECs, such as detected weapon envelope and most probable origin should further aid the FCO to make more timely, accurate decisions than when only relying on the deterministic threat values. From solely a deterministic threat value perspective, without the aid of the complementary flagging models, it is not possible to realise that Threats 3 and 4 are, indeed, not posing a danger to the defended system.

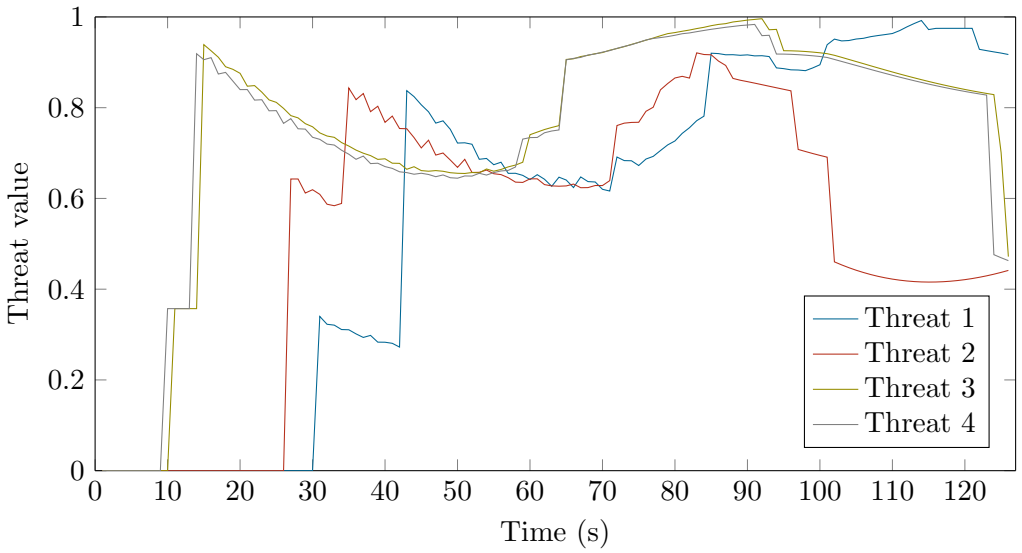


FIGURE 9.5: *System threat values as a function of time.*

9.4 Weapon Assignment Application

The next step in the discrete-event simulation model, is the assignment of WSs to engage aerial threats. As mentioned previously, this step uses the output threat values resulting from the TE subsystem as well as the SSHP values associated with WS-threat pairs during the different time-stages. The process of WA within a TEWA context is two-fold: (i) formulating the problem as a resource allocation problem and (ii) the process of solving this problem (approximately) by means of a genetic algorithm.

Since this is a discrete-event simulation with many stochastic elements, there is variability in the results. The spread of the TEWA-cycles during which the threats were actually destroyed are shown in Figure 9.6. From this figure it is clear that the non-attacking threats 3 and 4 are eliminated before the threats 1 and 2 that are, in fact, attacking the DAs. There are several reasons for this. Most importantly, at times 57–70, when threats 3 and 4 are generally eliminated, the system threat values of all the threats are very close together as may be seen from Figure 9.5. The deciding factor for engagement is therefore the SSHP values of the WS-threat pairs. At time 65 (approximately the middle of all the tracks), the fly over aircraft are significantly closer to the concentration of WSs than the attacking threats. Threats 1 and 2 only come within range of the WSs at time stages 77 and 79, respectively. The WA algorithms therefore prefer the engagement of Threats 1 and 2 (during TEWA-cycles 57–70) because of the larger SSHP values.

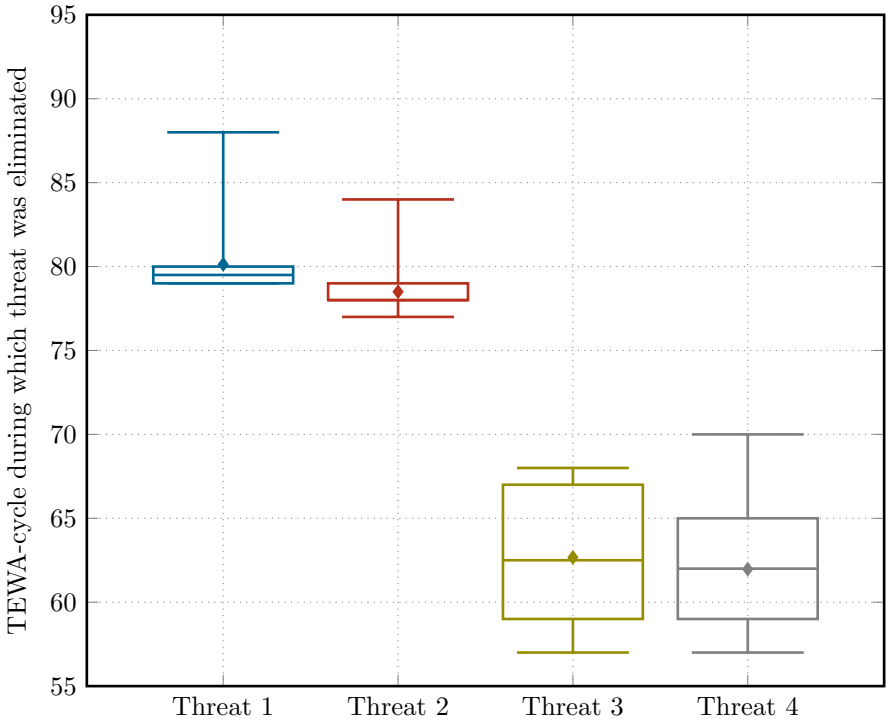


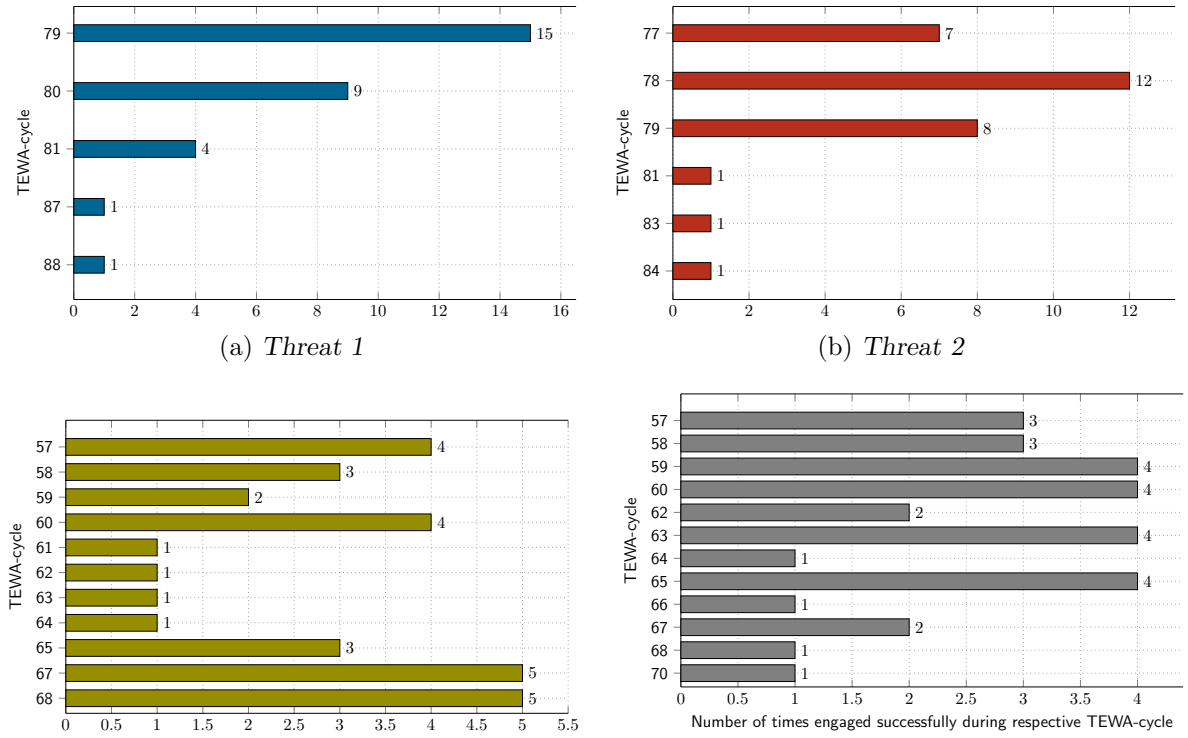
FIGURE 9.6: *TEWA-cycles during which threats were successfully engaged during all scenario runs.*

The number of times during a specific TEWA-cycle when a threat was successfully engaged, is visualised in Figure 9.7. This visualisation clarifies the shape of the engagement distribution per threat over the possible TEWA-cycles. Since all threats were eliminated for all the scenario runs, the total number of engagements per threat accumulates to thirty as may be seen in the figure. For Threats 1 and 2, it is clear that, considering all the successful engagements, that the majority of engagements favour earlier TEWA-cycles. For Threat 1 half of the successful engagements (15 out of 30) occurred during TEWA-cycle 79, whereas for Threat 2 the largest number of successful engagements realised during time-stage 78.

All engagements of threats distinguished by WSs, are depicted in Figure 9.8. It is encouraging to note, from a validation perspective, that the WS placed closest to a threat's avenue of approach is the most likely WS to engage a threat. Threat 1 was mainly engaged by WS 6, Threat 2 by WS 10, Threat 3 by WS 11 and Threat 4 by WS 5. Considering Figure 9.1, it may be seen that these WSs are, indeed, the first WSs that are within range of the various threats. The four CIWSs 1–4, did not engage any threats since they never came within range of these WSs.

As in Figure 9.8, the degree to which the different WS were utilised are visualised in Figure 9.9. The number of successful engagements, distinguished by WS and threat, are shown in Figure 9.9(a), whereas the number of engagements (hits and misses) are shown in Figure 9.9(b). Unsurprisingly, the overall form and distribution of these two graphs are very similar, with small exceptions. The graphs indicate that WS 11 is both the most utilised WS (largest number of engagements) and the most successful WS (largest number of successful engagements). For such a GBAD attack scenario it is therefore advised that more ammunition is allocated to WS 11 in an attempt to avoid a condition where its ammunition supply becomes depleted, thereby preventing the WS from engaging threats.

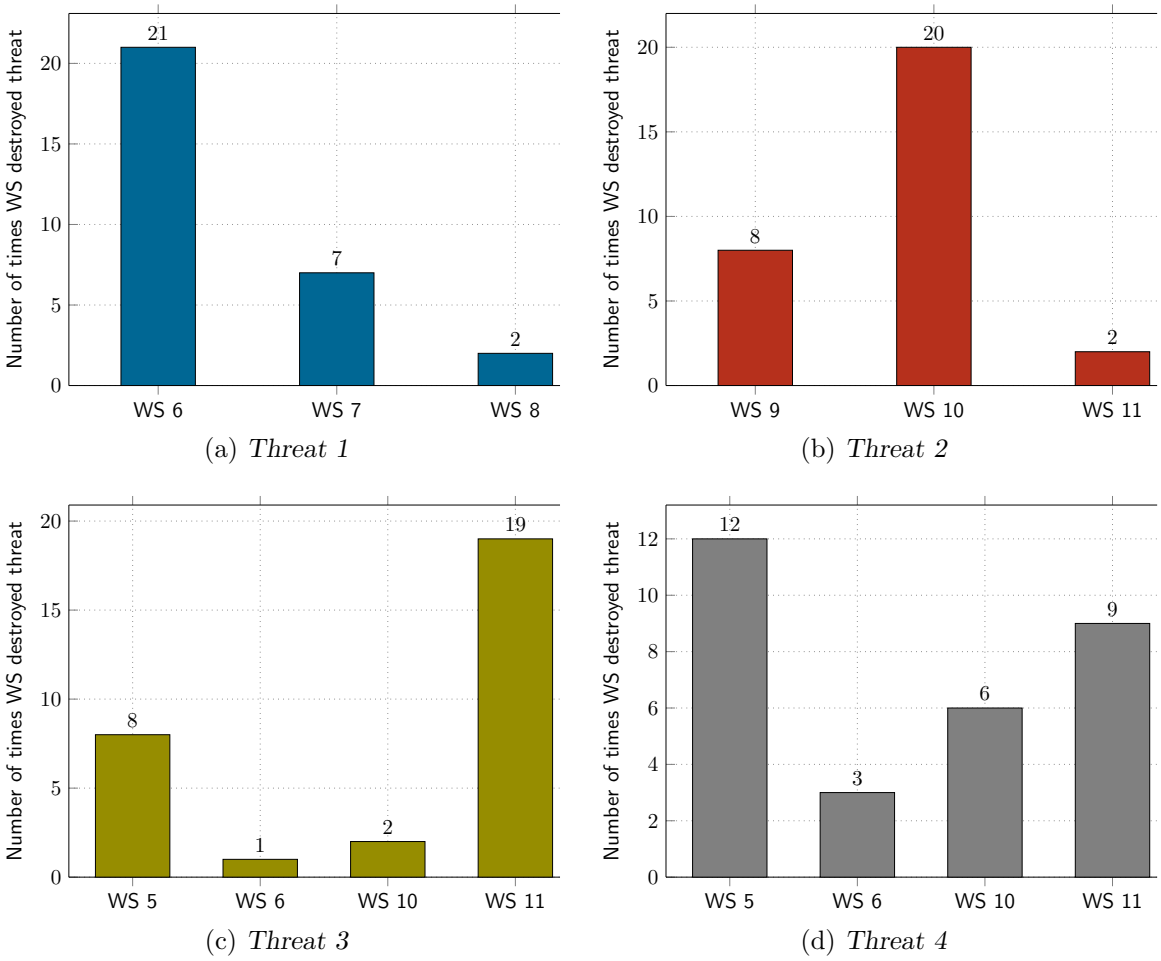
Recall from §6.4.1 that a stage-utility function is used to provide utility to earlier engagements and to prevent the WA subsystem from always selecting later engagement stages. This notion



(a) *Threat 1* (b) *Threat 2*

(c) *Threat 3* (d) *Threat 4*

FIGURE 9.7: *The number of engagements by different WSs, distinguished in terms of the threat that was engaged.*



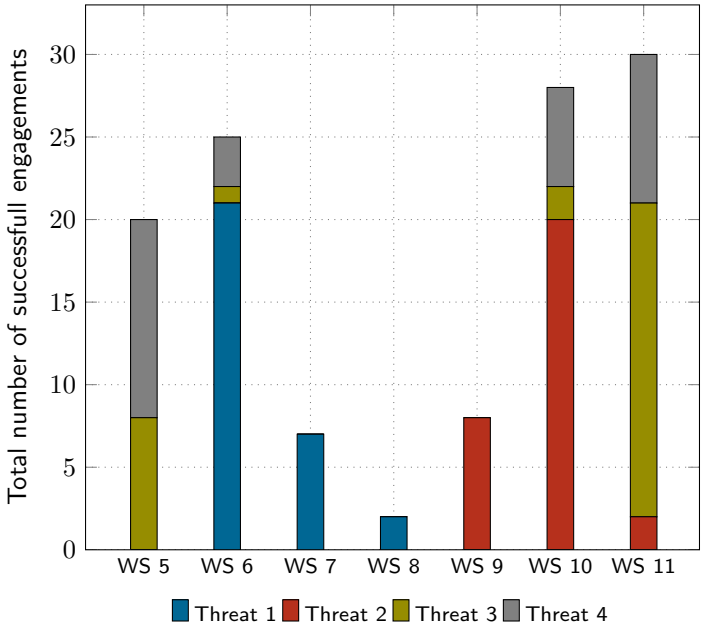
(a) *Threat 1*

(b) *Threat 2*

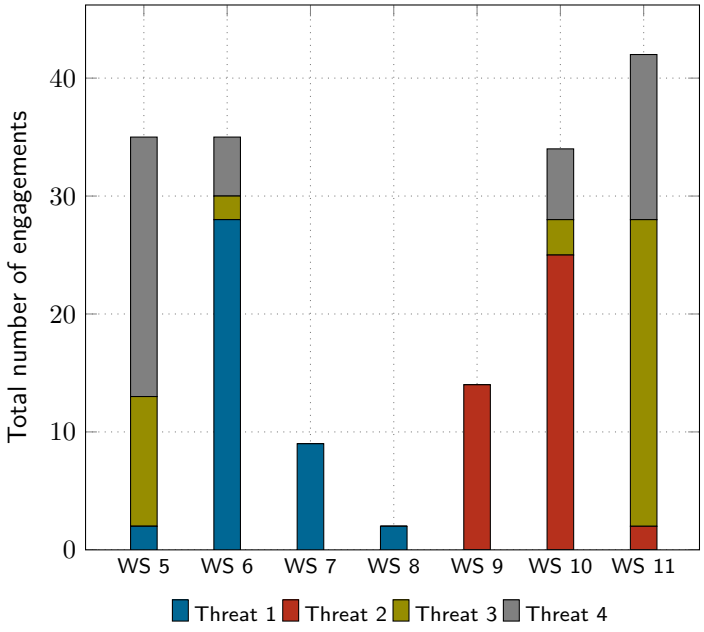
(c) *Threat 3*

(d) *Threat 4*

FIGURE 9.8: *The number of engagements by different WSs, distinguished in terms of the threat that was engaged.*



(a) *Successfull engagements by WSs*



(b) *All engagements (successful and unsuccessful) by WSs*

FIGURE 9.9: *The number of engagements by different WSs, distinguished in terms of the threat that was engaged.*

of utility is fundamental to effectively performing the trade-off analysis when selecting earlier or later stages for engagement. This utility is, however, difficult or even impossible to measure directly. In order to circumvent this difficulty, the utility function should be calibrated by eliciting heuristic decision criteria from domain experts. The methodology of how exactly this may be achieved is left for future work.

For the time length of the forecasting window, a duration of eight seconds was implemented. The rationale for this decision is as follows: Within eight seconds, if an aircraft travels at 250 m/s, the threat will have travelled a distance of two kilometres. Considering that the effective ranges of some WSs are four kilometres, two kilometres will result in a significant change in SSHP volumes during the forecasting. A longer forecasting time frame is also associated with more uncertainty and a longer computational time is required to solve the WA problem. For this proof-of-concept demonstration, a forecasting time length of eight seconds is therefore deemed sufficient.

The stage utility function may be analysed by determining the most used forecasting time-frame. The percentages of times that various forecasting time lengths were sufficient, taking into account all successful assignments, are shown in Figure 9.10. This clarifies how the stage utility function influences the selection of a forecasting time. For instance, in this scenario, a large number of assignments (63%) are scheduled at the end of the forecasting time-frame (8 seconds). The eighth and seventh second future time-stages contribute 82% to all the successful fire-order executions. It is therefore possible that the stage utility function is not providing enough utility to earlier engagements. This is a typical emergent property of the TEWA system, mainly resulting from the interaction between the stage utility function, flight path prediction model and genetic algorithm.

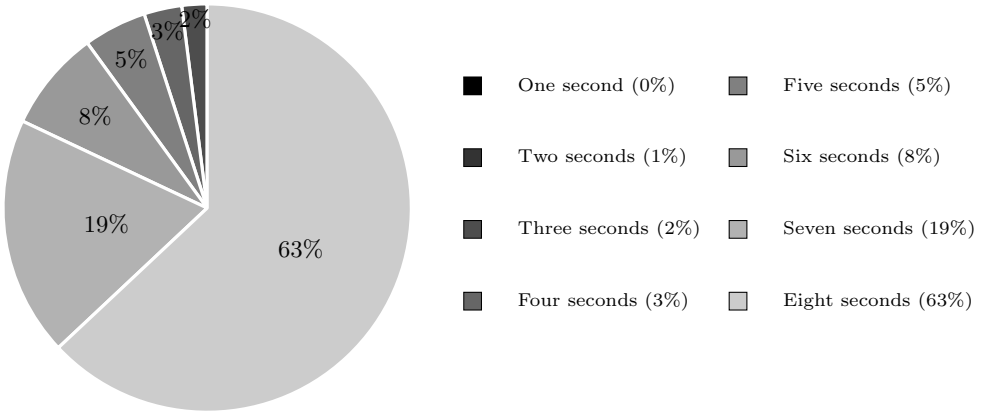


FIGURE 9.10: *Selected forecasting time length duration for all scenario runs and threat-WS combinations.*

9.5 Performance Metrics Calculation

Although it was stated in §8.5 that the true value of the performance metrics lies in comparing the performance of the algorithmic combination for different GBAD scenarios or testing different combinations of TE and WA algorithms for a single scenario, the metrics are nevertheless calculated and analysed here in an absolute manner. This is done in order to clarify their purpose within a larger evaluation study.

Prior to calculating the performance metrics, certain information on the simulation entities need to be known, such as the DA priority values, WS ammunition cost (interceptor or burst of rounds) and the value of the aerial threats from the perspective of the OPFOR. The DAs have the same importance values as introduced in the preceding sections. With regard to the WS engagement costs[3], the Umkhonto WSs 5, 7 and 10 are associated with an engagement cost of \$250 000 whereas the Marlin WSs 6, 8, 9 and 11 are associated with an engagement cost of \$600 000.

TABLE 9.4: *Performance metrics for all scenario runs.*

Scenario Run	Survivability	Economy	Adaptability	Effectiveness
1	1.00	4 000 000	1	1.00
2	1.00	1 350 000	2	1.00
3	1.00	3 150 000	1	1.00
4	1.00	2 900 000	1	1.00
5	1.00	3 150 000	1	1.00
6	1.00	2 550 000	1	1.00
7	1.00	1 950 000	1	1.00
8	1.00	2 300 000	2	1.00
9	1.00	3 150 000	1	1.00
10	1.00	2 900 000	1	1.00
11	1.00	3 150 000	1	1.00
12	1.00	2 300 000	1	1.00
13	1.00	1 700 000	2	1.00
14	1.00	2 550 000	1	1.00
15	1.00	3 150 000	1	1.00
16	1.00	3 400 000	1	1.00
17	1.00	3 650 000	0	1.00
18	0.36	2 900 000	1	1.00
19	1.00	1 950 000	1	1.00
20	1.00	1 700 000	2	1.00
21	1.00	2 550 000	1	1.00
22	1.00	2 050 000	2	1.00
23	1.00	1 700 000	2	1.00
24	1.00	1 950 000	2	1.00
25	0.36	4 500 000	0	1.00
26	1.00	3 150 000	1	1.00
27	1.00	2 900 000	1	1.00
28	1.00	1 950 000	1	1.00
29	1.00	2 300 000	2	1.00
30	1.00	1 700 000	2	1.00
Average	**0.96**	**2 620 000**	**1.23**	**1.00**
Standard deviation	**0.16**	**755 394**	**0.57**	**0.00**

The WS failed during two scenario runs (18 and 25) to protect a DA, which is apparent from the low survivability metric values in Table 9.4. For both these scenario runs, Threat 2 successfully engaged DA A (Command Centre). It is common practice to assume the worst case when testing

[3]These costs are solely used for illustrative purposes. The costs were derived from the compiled list of typical WSs costs [131]. Two different WSs are introduced in order to add some variety to the calculation of the metrics and, thereby, better clarifying the value of the performance metrics.

performance. Consequently, a DA is assumed to be destroyed when a threat reaches its weapon-release point. There is, as such, no notion of a DA vulnerability index[4]; a hit is assumed to be a kill and the threats have a hit probability of 100% when engaging a DA.

During scenario run 18, unsuccessful engagements resulted in the WSs being unavailable for reassignment because of the reloading delay, thus forcing the TEWA system to select less effective WSs during critical times close to weapon-release. During scenario run 25, on the other hand, the WS engaged unsuccessfully numerous times. As may be seen in Table 9.4, scenario run 25 is associated with the largest ammunition expenditure ($4 500 000) as well as sharing the lowest adaptability metric value of zero with simulation run 17. This provides evidence that the scenario run had an abnormal number of unsuccessful WS engagements when compared to other simulation runs, which may be ascribed to the stochastic nature of determining the outcomes of engagements. In order to prevent the recurrence of such an undesirable event, an additional WS should be placed closer to the avenue of approach of Threat 2 — this threat attempts to stay out of range of the WSs and launch its ordnance from far away before regressing. A more layered arrangement of WSs with more engagement opportunities far from the DAs, may be valuable. Furthermore, more ammunition should be allocated to WS 10, which depleted its resources during simulation runs 17 and 25. Additional ammunition will enable it to engage more often and better react to follow-up attacks.

In retrospect, since this GBAD scenario has a high WS to threats ratio (11/4) there is a minimal probability of the threats surviving the engagement. The only concern, therefore, is whether the threats can be eliminated before weapon-release. Such a high ratio is advantageous to the GBAD defenders and thereby encumbers the objective evaluation of the algorithms by inhibiting the conditions during which the algorithms perform well. Since a large number of GBAD resources are available, the allocation of resources are not constrained enough to identify the limitations of the algorithms; most of the allocations seem to be intuitive and should be able to be enforced by an operator. In order to assess the effective utilisation of limited resources, it is advised that a lower WS to threats ratio scenario is tested. Such a GBAD scenario will better clarify the advantages and limitations of the TEWA algorithms.

9.6 Chapter Summary

The chapter opened in §9.1 with a brief overview of the approach followed when performing the experimental study of this chapter. The steps of the study was clarified with reference to the earlier chapters.

After the processes entrenched in the simulation performance evaluation framework was understood, the deployment of the GBAD scenario was clarified in §9.2. Special attention was afforded to the placement of the DAs, the placement of the different types of WSs and the attack manoeuvres executed by the threats were also elucidated.

The threat evaluation process was demonstrated in §9.3. The different types of threat values were clarified and the results described and validated. The next step in the simulation entails the allocation of WSs to threats; the results of this process were presented in §9.4. These results clarified the utilisation of the different WSs and from the results it is clear that the modelling approach detailed in the preceding chapters seems to be appropriate for the application.

In §9.5, the chapter concluded by providing some perspective regarding the developed perfor-

[4] A vulnerability index quantifies the probability of survival of a DA which, in turn, depends on the ordnance launched at the DA and the air-to-surface hit probability of the engaging threat.

mance metrics. The metrics for all thirty simulation runs were calculated and analysed. The values of these metrics, when applied to an in-depth performance evaluation study, was also mentioned.

In addition to the performance evaluation advantages, this simulation may in the future also be used for training operators. The insights provided by the simulation may be used to clarify to the operators why certain WS assignments are better than the alternatives. Operators may even, possibly, compete against the simulation in order to access their tactical expertise and readiness.

CHAPTER 10

Conclusion

The outcome of any serious research can only be to make two questions grow where only one grew before.

— Thorstein Veblen

Contents

In the first section of this final chapter, the work contained in this thesis is briefly recounted. This is followed by an appraisal of the contributions made by this thesis. The chapter closes with a brief reflection on avenues of possible future work following on the work reported in this thesis.

10.1 Thesis Summary

This thesis opened with a description of two historical incidents in order to elucidate the critical, dynamic nature of TEWA systems. Furthermore, the importance of evaluating the performance of TEWA systems was also introduced in Chapter 1. The main aim of the thesis and the objectives that were pursued, were also detailed in the first chapter.

In Chapter 2, context related to the predominant theories underlying a TEWA DSS were provided, in partial fulfilment of Objective I of §1.4. The notion of NCW was explained in §2.1, and the importance of implementing NCW principles in military systems that are designed to operate within the current knowledge-era, was motivated. The concept of military C2 was also described in §2.2 with reference to the processes executed during a typical TEWA cycle. A possible decision process model — the OODA loop — was also reviewed in §2.2.2 for use in representing the C2 processes within a TEWA environment. The physical elements — environment, sensors, threats, WSs, DAs and HMI — of a GBAD system were finally described in §2.4.

The literature related to TEWA systems was summarised in Chapter 3 from a domestic (§3.2) as well as an international (§3.3) perspective. This chapter opened with an overview of the current state of the South African GBAD programme in §3.1. Subsequently, the domestic part mainly reviewed all the work done at the Stellenbosch TEWA Centre of Expertise — this forms

the foundation for the work contained in this thesis. The international part, on the other hand, provided a brief overview of the current state of global TEWA knowledge. In this part it was mentioned that it is difficult to obtain detailed information related to the algorithms that form the core of internal TEWA systems. The chapter further contributed to the fulfilment of research Objective I.

The development of the simulation performance evaluation framework was detailed in Chapter 4. This chapter opened with a literature review of simulation in a military context and motivated the selection of the simulation paradigm adopted in §4.1. The core of the chapter (§4.2) detailed the modelling of all the physical elements, as alluded to briefly in §2.4. The selection of MATLAB as the simulation environment is motivated in §4.3, after which the high-level design of the simulation performance evaluation framework was described in §4.4. The verification and validation carried out throughout the simulation development were described in §4.5. The chapter finally closed in §4.6 with a description of a hypothetical GBAD scenario which was used to demonstrate the working of the TEWA algorithms and simulation processes throughout the remainder of the thesis. The material of the chapter was presented in fulfilment of Objective IV.

In Chapter 5, the process of evaluating threats was described. The chapter opened in §5.1 with an overview of TE in general, after which the TE models implemented in this thesis were described in §5.2. The implemented models include a slant distance-related model (§5.2.1), a course-related model (§5.2.2), a CPA model (§5.2.3) and a newly proposed altitude-related model (§5.2.4). Two data fusion processes were described in §5.3 for use in calculating the single system threat value of a specific threat. The chapter closed with a presentation of the logic flow of the TE simulation architecture in §5.4. This chapter contributed to the fulfilment of Objective III, since model limitations were identified throughout the chapter, especially in §5.3.

The WA sub-process, which utilises the threat values resulting from the TE sub-process of Chapter 5, was described in Chapter 6. The chapter opened with an overview of the WA problem formulation in §6.1 together with a description of the complexity of formulating the WA problem as an instance of the typical assignment problem in §6.1. Thereafter, existing static and dynamic WAM formulations which formed the foundation of the WAMs adopted in this thesis were described in §6.1.1 and §6.1.2, respectively. Different solution approaches toward solving the WA problem were described in §6.2 and the selection of a genetic algorithm for solving the WAMs in this thesis was motivated. The implemented genetic algorithm was briefly described in §6.3 after which the WAMs adopted in this thesis were described in §6.4. This chapter, together with Chapter 5, stand in fulfilment of Objective IV. The frameworks presented in §5.4 and §6.4, as well as the results later presented in Chapter 10 serve as evidence in support of this claim.

Since the human operator forms a crucial part of the TEWA DSS, it is of utmost importance to ensure that salient decision support is provided to the operator. A literature review of the current research into HMIs was therefore provided in Chapter 7. The notion of decision support within a GBAD environment was first clarified in §7.2, after which requirements for the provision of germane decision support were detailed in §7.3. The nature of germane decision support was elucidated with reference to the effect of operator stress on system performance and the uncertainty inherent in a typical TEWA system. Consequently, several design considerations for an efficient HMI were provided in §7.4, taking into account the complexities of providing sound decision support. A preliminary designed HMI was demonstrated in §7.5, thereby forming a foundation for future work and ascertaining the feasibility of the MATLAB simulation environment in terms of facilitating a detailed design of a TEWA HMI. The display contributions of this chapter, together with the literature review on decision support in a military context, satisfies research Objective II.

The performance evaluation of complex systems, such as TEWA systems, was considered in Chapter 8. The notion of following an SoS approach for evaluating the performance of TEWA systems was described and motivated in §8.2. The only tractable way of evaluating the performance of TEWA systems, is where the scenario setup as well as the employed algorithms are taken into consideration when analysing results. Three different approaches to evaluating the performance of TEWA systems, adhering to the requirements provided in §8.2 and the methodology described in §8.3, were summarised in §8.4. In addition, four performance evaluation metrics to be used within a scenario-dependent simulation environment were explained in §8.5. Some practical considerations that may be derived from employing the developed performance evaluation framework were documented in §8.6. The considerations provided in this chapter, together with the developed simulation performance evaluation framework, achieves Objective V.

All the algorithmic models, simulation entities and processes were applied to a near-realistic, but hypothetical GBAD scenario in Chapter 9. The setup of the scenario was detailed in §9.2, after which the TE process and WA process were subsequently applied to the scenario. The results were presented and analysed in §9.3 and §9.4, respectively. The results indicated that the simulation does, indeed, provide realistic results. The results furthermore indicated that the performance evaluational framework is, in fact, suitable for use in further detailed performance evaluation studies. The performance metrics of §8.5 were finally calculated in §9.5. The demonstration contained in this last chapter, as well as the illustrative example used throughout the thesis, were presented in fulfilment of Objective VI.

10.2 Appraisal of the Work Contained in this Thesis

The main contributions of this thesis are fourfold. The first contribution is the design and demonstration of a value-based threat value fusion approach in order to calculate a single system threat value from a multitude of initial threat values. The TE models implemented are extensions of the work of Roux [167], who designed the architecture for a TE subsystem and suggested several specific TE models, and Heyns [78] who implemented MAUT methods to obtain an ordinal list of threats, as opposed to a cardinal ranking of threats as was done in this thesis. This contribution was published in [207].

The second contribution is the development and implementation of a novel WA formulation. This formulation is, in essence, an extension of the work by Du Toit [59] and incorporates some of the considerations suggested by Van der Merwe [215]. The implemented WA model is unique in the sense that different constraints are implemented and the WA problem is solved reactively, in the sense that historically observed events are taken into account when suggesting future WS engagements. Because of the complex nature of this formulation, a dynamic programming approach was followed in order to model it and a genetic algorithm was used to solve it, as may be seen in Appendix C.

The third contribution is the preliminary design of a TEWA HMI. No previous work related to the design of a TEWA HMI for a GBAD system could be found. The majority of literature related to HMI design is for point-defence, as opposed to the area-based nature of a GBAD environment. The available point-defense articles were therefore studied for residual capacity in order to derive the display guidelines as provided in Chapter 7.

The final contribution is the development and demonstration of the simulation performance evaluation framework for TEWA systems. This simulation framework was developed in MATLAB and include the majority of research done on TEWA related algorithms at Stellenbosch University over the period 2006–2014. All simulation model entities were modelled with as much detail

as possible, subject to the scarcity of detailed TEWA-related information. This development essentially provides a simulation framework which ties together a series of previously distinct, isolated algorithmic models into an aggregated system that allows for the detailed analysis of a SoS. The algorithms were tested within the context of a comprehensive, near-realistic example in Chapter 9 and the results indicated that the algorithms implemented do, indeed, provide meaningful results. There is, as such, potential for real-world application. Several considerations for evaluating the performance of TEWA systems were also provided in Chapter 8 and some of these considerations were published in [206].

10.3 Suggestions for Further Work

The suggestions put forward in this section provide possible avenues for future follow-up work in order to continue the development of a fully-fledged TEWA performance evaluation study, thereby satisfying the last research objective of §1.4 — Objective VII.

Proposal 1: *Perform the detailed design of a HMI for a TEWA DSS employed in a GBAD environment.*

Chapter 7 may form the foundation for the detailed design of a TEWA HMI. As may be expected, for decision making in a military context, it is crucial to ensure that such a HMI should be designed with utmost diligence. Some considerations were provided in Chapter 7, but only a preliminary design was presented. A detailed design should, however, include all known information (*e.g.* threat types, flagging models and possible responses) which should be presented to the operators. This information should, in turn, be presented in such a way that it enhances the operators' decision-making cycle and not overwhelm them with information. The detailed design should therefore follow an evolutionary development approach by validating the design in conjunction with the intended end-users and incorporate their concerns into the designed HMI, similar to the study executed as part of the TADMUS programme [43].

Proposal 2: *Standardise the representation of the TEWA DSS's architecture.*

Since the architecture of a typical TEWA system is highly complex, it should add great value to the current work if a standard framework is used to document the system architecture. Doing this will enable future students as well as the stakeholders to better understand the complex TEWA processes and sub-system functionalities in a smaller timeframe. A possible approach for such an object-oriented design, is the *Unified Modeling Language* (UML). UML is seen as the standardised methodology for object-oriented system analysis [103]. It offers a framework that provides logical, thorough and rapid methods for creating new systems adaptable to changing environments. Adopting this object-oriented paradigm will offer a new SoS requirement and design methodology which provides for the minimisation of accidental complexity, the control of essential complexity through the use of decentralised control flow, minimal messaging between classes, implicit case analysis and information hiding mechanisms [30]. UML diagrams have also shown to enhance understanding for the operators as well as for the end-users of the system [103].

Proposal 3: *Implement terrain and, accordingly, line-of-sight considerations into the simulation framework.*

In support of military applications, the *National Imagery and Mapping Agency* has developed standard digital data sets (*Digital Terrain Elevation Data* (DTED)) which are uniform matrices of terrain elevation values which provides basic quantitative data for systems and applications that require terrain elevation, slope, and/or surface roughness information. Incorporating a more complicated surface grid by including different terrains and "obstacles", such as mountains and valleys, should provide for more detailed performance evaluation. Line-of-sight calculations

may, for example, be implemented in order to assess how the terrain influences the outcome of a specific GBAD scenario. Different WS and DA placement configurations may be assessed for terrain robustness when a specific type of attack is launched against the GBAD setup considered.

Proposal 4: *Develop vulnerability indices for use within the simulation.*

With the introduction of more precise, advanced and expensive WSs (*e.g.* new generation precision guided munitions) into the battlefield, questions related to their operation effectiveness become more prevalent. In the simulation performance evaluation framework of this thesis a hit was modelled as a kill, but this is not necessarily true in practice. The notion of a vulnerability index was introduced in §4.2.3. Such an inclusion should provide more confidence in the simulation results and allow for more detailed analysis. For an academic simulation, however, the information required to sufficiently represent vulnerability indices may be difficult to obtain.

Proposal 5: *Investigate possible sensor-management strategies and the effect of incomplete damage assessment on system performance.*

In this thesis, the assumption was made that sensor systems are able to perfectly assess the damage after a threat has been engaged by a WS. In reality, however, this is not usually the case. In order to ensure that the TEWA system is effective, both in terms of ammunition expenditure and time efficiency, it is important that the system is able to properly assess the damage that was inflicted by previously fired shots. Different methods for evaluating the performance of TEWA-related systems in the presence of incomplete damage assessment is provided in [15][1]. Probabilities may also be assigned to the probability of detection for specific threats. Lee and Mason [110] developed a MATLAB simulation for computing the probability of radar detection. The effects of different environmental conditions on the probability of detection was investigated in [26]. Numerous other articles related to sensor-management have also recently been published, as stated in §3.3. Currently, there has not been a significant focus on the management of sensor systems — the focus has been more on the TE algorithms and management of WSs. The incorporation of sensor-related considerations should, however, allow for more meaningful conclusions related to the functioning of a TEWA system as a whole.

Proposal 6: *Perform detailed performance evaluation on different algorithm-scenario combinations, in order to find the best algorithms to be used in a GBAD context, by employing the performance evaluation framework proposed in this thesis.*

A performance evaluation framework was put forward in this thesis which may be used to evaluate different combinations of algorithms, different scenario setups and/or different simulation model entity properties. As such, if different DMs, such as time-related DMs, are included, their effectiveness may be investigated by employing the developed simulation framework. The effects of different parameters on the emergent properties of the TEWA system is also an important aspect to investigate. Some emergent properties of the TEWA system modelled in this thesis were identified, but a detailed sensitivity analysis was not performed due to time constraints. By understanding how different parameters affect the system outcomes, more possibilities for future work can be identified.

Finally, in order to gain a better understanding of the functioning of the system model entities, *what-if* analyses may be performed. The outcomes may be analysed if alternative constraints are added, more WSs are deployed, *etc.* An important aspect of the WA objective function is its stage utility function. The effect of different stage-utility functions may also be analysed and a suitable one selected, calibrated and motivated.

[1] Although the focus was only on shoot-look-shoot tactics for a single shooter, this article should nevertheless still provide value insights for incorporating incomplete damage assessment constraints in a GBAD TEWA environment.

Some of these suggestions for future work, specifically (3)–(6), require a high-fidelity simulation model. These future work suggestions should therefore only be attempted once the TE subroutine, WA processes and WAM solution methodology are deemed sufficient. As stated in Chapter 8, the development of such a complex simulation is, in fact, an iterative process. It is therefore possible that the "ideal" solution methodology will change as more high-fidelity models and concepts are introduced within the simulation environment. For an academic TEWA simulation it would, however, always be difficult to obtain the required information to effectively model all the physical elements in a TEWA system. Future work may therefore expand upon the work contained in this thesis and, thereby, prevent the resolving of problems that have already been solved.

References

[1] Adams TK, 2001, *Future warfare and the decline of human decision making*, Parameters, **31(4)**, pp. 57–71.

[2] Ahuja RK, Kumar A, Jha KC & Orlin JB, 2007, *Exact and heuristic algorithms for the weapon-target assignment problem*, Operations Research, **55(6)**, pp. 1136–1146.

[3] Ahuja RK, Kumar A, Jha K & Orlin JB, 2003, *Exact and heuristic methods for the weapon target assignment problem*, (Unpublished) Technical Report 4464-03, MIT Sloan School of Management, Cambridge (MA).

[4] Alberts DS, 2002, *Information age transformation: Getting to a 21st century military*, (Unpublished) Technical Report ADA457904, Office of the Assistant Secretary of Defense, Washington (DC).

[5] Alberts DS & Hayes RE, 2003, *Power to the edge: Command and control in the information age*, (Unpublished) Technical Report ADA457861, Office of the Assistant Secretary of Defense, Washington (DC).

[6] Alberts DS, Garstka JJ & Stein FP, 1999, *Network centric warfare: Developing and leveraging information superiority*, (Unpublished) Technical Report 20020905-096, Command and Control Research Program Publication Series, Washington (DC).

[7] Alexander M, 2012, *Decision-making using the AHP and SAS/IML*, (Unpublished) Technical Report SD-04, Social Security Administration, Baltimore (MD).

[8] Alghamdi AS, 2009, *Evaluating defense architecture frameworks for C4I system using analytic hierarchy process*, Journal of Computer Science, **5(12)**, pp. 1075–1081.

[9] Allouche MK, 2005, *Real-time use of Kohonen's self-organizing maps for threat stabilization*, Information Fusion, **6(2)**, pp. 153–163.

[10] Anderson J & Hong L, 2008, *Sensor resource management driven by threat projection and priorities*, Information Sciences, **178(8)**, pp. 2007–2021.

[11] Angerman WS, 2003, *Coming full circle with Boyd's OODA loop ideas: An analysis of innovation diffusion and evolution*, Masters Thesis, Air Force Institute of Technology, Dayton (OH).

[12] Arevalo MR & Bland G, 2012, *Determination of weapons fratricide probability*, US Patent 8 176 834, URL: `https://www.google.com/patents/US8176834`.

[13] Army Technology, 2015, *S-350E Vityaz (50R6) surface-to air defence missile system*, [Online], [Cited March 10th, 2015], Available from `http://www.army-technology.com/projects/s-350e-vityaz-50r6-surface-to-air-defence-missile-system/`.

[14] Association for Diplomatic Studies and Training, 2015, *Moments in US diplomatic history: USS Vincennes shoots down Iran Air Flight 655*, [Online], [Cited September 28th, 2015], Available from `http://adst.org/2014/07/uss-vincennes-shoots-down-iran-air-flight-655/`.

[15] Aviv Y & Kress M, 1997, *Evaluating the effectiveness of shoot-look-shoot tactics in the presence of incomplete damage information*, Military Operations Research, **3(1)**, pp. 79–89.

[16] Ball RE, 1965, *The fundamentals of aircraft combat survivability: Analysis and design*, 2nd Edition, American Institute of Aeronautics and Astronautics, New York (NY).

[17] Banks J, 1998, *Handbook of simulation: Principles, methodology, advances, applications and practice*, 1st Edition, John Wiley & Sons, Hoboken (NJ).

[18] Barbera RS, 1994, *The Patriot missile system: A review and analysis of its acquisition process*, PhD Dissertation, Naval Postgraduate School, Monterey (CA).

[19] Baylis J, Wirtz JJ & Gray CS, 2009, *Strategy in the contemporary world*, 3rd Edition, Oxford University Press, New York (NY).

[20] Behrens M, 2014, CEO at *Cybicom Technologies (Pty) Ltd*, [Personal Communication], Contactable at `malcolm@cybicom.com`.

[21] Benaskeur AR, Kabanza F & Beaudry E, 2010, *CORALS: A real-time planner for anti-air defense operations*, Transactions on Intelligent Systems and Technology, **1(2)**, pp. 13–30.

[22] Benaskeur AR, Kabanza F, Beaudry E & Beaudoin M, 2008, *A probabilistic planner for the combat power management problem*, Proceedings of the International Conference on Autonomous Planning and Scheduling, pp. 12–19.

[23] Benaskeur A, Bossé É & Blodgett D, 2007, *Combat resource allocation planning in naval engagements*, (Unpublished) Technical Report TR 2005-486, Defence R&D Canada, Valcartier.

[24] Blasch E, Valin P & Bosse E, 2010, *Measures of effectiveness for high-level fusion*, Proceedings of the 13th IEEE Conference on Information Fusion, pp. 1–8.

[25] Bo Z, Feng-Xing Z & Jia-hua W, 2011, *A novel approach to solving weapon-target assignment problem based on hybrid particle swarm optimization algorithm*, Proceedings of the 13th Conference on Electronic and Mechanical Engineering and Information Technology, pp. 1385–1387.

[26] Bocquet S, 2011, *Calculation of radar probability of detection in K-distributed sea clutter and noise*, (Unpublished) Technical Report DSTO-TN-1000, Australian Department of Defence, Joint Operations Division, Canberra.

[27] Bonczek RH, Holsapple CW & Whinston AB, 2014, *Foundations of decision support systems*, 1st Edition, Academic Press, New York (NY).

[28] Braybrook R, 2002, *Land-based VSHORAD and SHORAD systems*, Armada International, **26(2)**, pp. 39–53.

[29] Brown C, Fagan P, Hepplewhite A, Irving B, Lane D & Squire E, 2001, *Real-time decision support for the anti-air warfare commander*, Proceedings of the 6th International Command and Control Research and Technology Symposium.

[30] Brown WD, 2006, *Analysis and design of a cooperative weapon assignment module for advanced battle manager of a ballistic missile defense system*, MSc Thesis, Naval Postgraduate School, Monterey (CA).

[31] Burns CM, Bryant D & Chalmers B, 2002, *Assessment of the TADMUS DSS with work domain analysis*, Proceedings of the 3rd Human Factors and Ergonomics Society Annual Meeting, pp. 453–457.

[32] *C Programming and C++ Programming*, 2015, [Online], [Cited November 2nd, 2015], Available from `http://www.cprogramming.com/`.

[33] Canadian Armed Forces, 2015, *Halifax-class modernisation and life extension*, [Online], [Cited October 7th, 2010], Available from `http://www.forces.gc.ca/en/news/article.page?doc=halifax-class-modernization-and-life-extension`.

[34] Carling RL, 1993, *A knowledge-base system for the threat evaluation and weapon assignment process*, Naval Engineers Journal, **105(1)**, pp. 31–41.

[35] Carver L & Turoff M, 2007, *Human-computer interaction: The human and computer as a team in emergency management information systems*, Computer Machinery, **50(3)**, pp. 33–38.

[36] Castro CA & Adler AB, 1999, *OPTEMPO: Effects on soldier and unit readiness*, Parameters, **29(3)**, pp. 86–95.

[37] Cebrowski AK, 2005, *The implementation of network-centric warfare*, [Online], [Cited November 12th, 2014], Available from `http://www.au.af.mil/au/awc/awcgate/transformation/oft_implementation_ncw.pdf`.

[38] Chen CC, Chen J & Lin PC, 2009, *Identification of significant threats and errors affecting aviation safety in Taiwan using the AHP*, Journal of Air Transport Management, **15(5)**, pp. 261–263.

[39] Chen H, Liu Z, Sun Y & Li Y, 2012, *Particle swarm optimization based on genetic operators for sensor-weapon-target assignment*, Proceedings of the 5th Symposium on Computational Intelligence and Design, pp. 170–173.

[40] Chen S, He J & Liu H, 2012, *Realization and simulation of parallel ant colony algorithm to solve WTA problem*, Proceedings of the International Conference on Systems and Informatics, pp. 2458–2461.

[41] Cohen MS, Freeman JT & Thompson BB, 1997, *Integrated critical thinking training and decision support for tactical anti-air warfare*, Proceedings of the 1997 Command and Control Research and Technology Symposium.

[42] Cole A, Drew P & Mandsager D, 2009, *Sanremo handbook on Rules of engagement*, 1st Edition, International Institute of Humanitarian Law, Sanremo.

[43] Collyer SC & Malecki GS, 1998, *Tactical decision making under stress: History and overview*, American Psychological Association, Washington (DC).

[44] Council for Scientific and Industrial Research, 2014, *An edge in radar & electronic warfare*, [Online], [Cited November 28th, 2014], Available from `http://www.csir.co.za/dpss/conference/docs/Sensors_&_Electronic_Warfare.PDF`.

[45] Creveld Mv, Canby SL & Brower KS, 1994, *Air power and maneuver warfare*, (Unpublished) Technical Report 0704-0188, School of Advanced Airpower Studies, Maxwell (AL).

[46] Cutler PR & Nguyen XT, 2003, *Description of a rule-based model for the automatic allocation of airborne assets*, Proceedings of the 6th International Symposium on Information Fusion, pp. 979–985.

[47] Dahl EJ, 2002, *Network centric warfare and the death of operational art*, Defence Studies, **2(1)**, pp. 1–24.

[48] Dahlstrom MP & Peeler DL, 2013, *Network centric warfare: Advantages and disadvantages*, Impact Strategic, **2013(3)**, pp. 94–100.

[49] Dantzig B, 2010, *Systems engineering handbook: A guide for system life cycle processes and activities*, 3rd Edition, International Council on Systems Engineering, San Diego (CA).

[50] Deep K, Singh KP, Kansal ML & Mohan C, 2009, *A real coded genetic algorithm for solving integer and mixed integer optimization problems*, Applied Mathematics and Computation, **212(2)**, pp. 505–518.

[51] *Defence Science Board*, 2014, [Online], [Cited August 10th, 2015], Available from `http://www.acq.osd.mil/dsb/charter.htm`.

[52] Defence Web, 2015, *Reutech radar systems develops naval air and sea surveillance radar with model-based design*, [Online], [Cited April 12th, 2014], Available from `http://tinyurl.com/pqvhapb`.

[53] Denel Dynamics, 2014, *The SABLE air defence system*, [Online], [Cited March 10th, 2015], Available from `http://www.deneldynamics.co.za/products/integrated-systems-solutions/sable`.

[54] Denney SH, 1970, *A review of literature on the theory of hit and kill probabilities*, MSc Thesis, Naval Postgraduate School, Monterey (CA).

[55] *Dictionary.com*, 2014, [Online], [Cited October 28th, 2014], Available from `http://dictionary.reference.com/browse/black%20box`.

[56] Dillon SM, 1998, *Descriptive decision making: Comparing theory with practice*, Unpublished manuscript, Department of Management Systems, University of Waikato, Hamilton.

[57] Drillings M & Serfaty D, 1997, *Naturalistic decision making in command and control*, 1st Edition, Lawrence Erlbaum Associates, Hoboken (NJ).

[58] Driskell JE & Johnston JH, 1998, *Making decisions under stress: Implications for individual and team training*, Stress Exposure Training Journal of the American Psychological Association, **13(447)**, pp. 191–217.

[59] Du Toit FJ, 2009, *The dynamic weapon target assignment problem in a ground based air-defence environment*, MSc Thesis, Stellenbosch University, Stellenbosch.

[60] Du Toit J, 2009, *Modelling rigid body kinematics with splines*, MSc Thesis, Stellenbosch University, Stellenbosch.

[61] Dubois D & Prade H, 2004, *On the use of aggregation operations in information fusion processes*, Fuzzy Sets and Systems, **142(1)**, pp. 143–161.

[62] Duvenhage A, 2011, *A software framework to support distributed command and control applications*, MEng Thesis, University of Pretoria, Pretoria.

[63] Duvenhage B, Delport J & Louis A, 2007, *A 3D visual analysis tool in support of the SANDF's growing ground based air defence simulation capability*, Proceedings of the 5th International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa, pp. 39–46.

[64] Edmunds A & Morris A, 2000, *The problem of information overload in business organisations: A review of the literature*, International Journal of Information Management, **20(1)**, pp. 17–28.

[65] Ender T, Leurck RF, Weaver B, Miceli P, Blair WD, West P & Mavris D, 2010, *System-of-systems analysis of ballistic missile defense architecture effectiveness through surrogate modelling and simulation*, Systems Journal, **4(2)**, pp. 156–166.

[66] Endsley MR, 2000, *Theoretical underpinnings of situation awareness: A critical review*, CRC Press, Boca Raton (FL), pp. 3–32.

[67] Erlandsson T, 2011, *Situation analysis for fighter aircraft combat survivability*, Licentiate Thesis, Örebro Universitet, Kållered.

[68] Fadok DS, 1995, *John Boyd and John Warden: Air power's quest for strategic paralysis*, Masters Thesis, School of Advanced Airpower Studies, Maxwell (AL).

[69] Falkman G, 2015, *University of Skövde*, [Online], [Cited February 12th, 2014], Available from `http://www.his.se/en/about-us/Facts-and-figures/staff/goran_falkman/`.

[70] Falzon L, 2006, *Using Bayesian network analysis to support centre of gravity analysis in military planning*, European Journal of Operational Research, **170(2)**, pp. 629–643.

[71] Fein G, 2015, *Aegis BMD demonstrates engagement coordination*, Jane's Defence Weekly, **52(9)**, pp. 10–11.

[72] Fredrik J, 2010, *Evaluating the performance of TEWA systems*, PhD Thesis, Orebro Universitet, Kållered.

[73] Garcia J, 2013, *Sensors: Types and characteristics*, Illinois Institute of Technology, Aerospace Laboratory II, Chicago (IL).

[74] Grant T & Kooter B, 2005, *Comparing OODA & other models as operational view C2 architecture topic: C4ISR/C2 architecture*, Proceedings of the 10th International Command and Control Research and Technology Symposium.

[75] Grottke M & Trivedi KS, 2007, *Fighting bugs: Remove, retry, replicate, and rejuvenate*, Computer, **40(2)**, pp. 107–109.

[76] Guha M, 2010, *Reimagining war in the 21st century: From Clausewitz to network-centric warfare*, 1st Edition, Routledge Critical Security Studies, Abingdon.

[77] Helldin T, Falkman G, Alfredson J & Holmberg J, 2011, *The applicability of human-centred automation guidelines in the fighter aircraft domain*, Proceedings of the 29th Annual European Conference on Cognitive Ergonomics, pp. 67–74.

[78] Heyns AM, 2008, *Measuring the threat value of fixed wing aircraft in a ground-based air defence environment*, MSc Thesis, Stellenbosch University, Stellenbosch.

[79] Hildreth SA, 2007, *Ballistic missile defense: Historical overview*, (Unpublished) Technical Report RS22120, Defence and Trade Division, Washington (DC).

[80] Hosein PA, 1989, *A class of dynamic nonlinear resource allocation problems*, (Unpublished) Technical Report LIDS-TH-1922, Defence Technical Information Center Document, Cambridge (MA).

[81] Hosein PA & Athans M, 1990, *Some analytical results for the dynamic weapon-target allocation problem*, (Unpublished) Technical Report LIDS-P-1944, Massachusetts Institute of Technology, Cambridge (MA).

[82] *How Israel's Iron Dome works*, 2015, [Online], [Cited February 12th, 2014], Available from `http://abcnews.go.com/International/israels-iron-dome-works/`.

[83] *How the Iron Dome anti-RAM system works*, 2015, [Online], [Cited March 21st, 2014], Available from `http://www.businessinsider.com/how-israels-iron-dome-anti-missile-system-works-2014-7`.

[84] Howard M & Payton D, 2011, *System and method for distributed engagement*, US Patent 7 912 631, URL: http://www.google.com/patents/US7912631.

[85] Huberman G, 2001, *Familiarity breeds investment*, Review of Financial Studies, **14(3)**, pp. 659–680.

[86] Hutchings PJ & Street NJ, 2001, *Future short range ground-based air defence: System drivers, characteristics and architectures*, (Unpublished) Technical Report RTO MP-063, Airspace Management Systems Department, Malvern.

[87] Hutchins SG & Kowalski JT, 1993, *Tactical decision making under stress: Preliminary results and lessons learned*, (Unpublished) Technical Report AD-A270-618, Naval Command, Control and Ocean Surveillance Centre, San Diego (CA).

[88] Hwang SM & Jo JY, 2008, *A development method of GUI in military system software*, Proceedings of the Conference on Advanced Software Engineering and its Applications, pp. 249–251.

[89] IEEE Computer Society, 1998, *IEEE guide for information technology: Concept of operations document*, IEEE Computer Society, New York (NY).

[90] Ignizio JP, 1991, *An introduction to expert systems*, 1st Edition, McGraw-Hill, New York (NY).

[91] Irandoust H, Benaskeur A & Bellefeuille P, 2011, *Distributed threat evaluation in naval tactical battle management*, Proceedings of the 16th International Command and Control Research Symposium, pp. 1–44.

[92] Jamshidi M, 2008, *Systems of systems engineering: Principles and applications*, 1st Edition, CRC press, Boca Raton (FL).

[93] Johansson F, 2015, *Air defense: Greedy algorithms best for multiple targets*, [Online], [Cited August 19th, 2010], Available from http://www.sciencedaily.com/releases/2010/12/101209074158.htm.

[94] Johansson F & Falkman G, 2008, *A Bayesian network approach to threat evaluation with application to an air defense scenario*, Proceedings of the 11th International Conference on Information Fusion, pp. 1–7.

[95] Johansson F & Falkman G, 2008, *A comparison between two approaches to threat evaluation in an air defense scenario*, 1st Edition, Modeling Decisions for Artificial Intelligence, Springer, New York (NY).

[96] Johansson F & Falkman G, 2011, *Real-time allocation of firing units to hostile targets*, Journal of Advances in Information Fusion, **6(2)**, pp. 187–199.

[97] Johnson MVR, McKeon MF & Szanto TR, 1998, *Simulation based acquisition: A new approach*, 1st Edition, Defence Systems Management College Press, Belvor (VA).

[98] Jordan SE, Snell MK, Madsen MM, Smith JS & Peters BA, 1998, *Discrete-event simulation for the design and evaluation of physical protection systems*, Proceedings of the 30th conference on Winter simulation, pp. 899–906.

[99] Judd KB & McLain TW, 2001, *Spline based path planning for unmanned air vehicles*, Proceedings of the Aeronautics and Astronautics International Conference on Guidance, Navigation, and Control, pp. 6–9.

[100] Kauderer HT, 2000, *Air-directed surface-to-air missile study methodology*, Johns Hopkins APL Technical Digest, **21(2)**, pp. 244–250.

[101] Kavanagh J, 2005, *Stress and performance: A review of the literature and its applicability to the military domain*, (Unpublished) Technical Report 20051026-116, RAND Corporation, Arlington (VA).

[102] Keeney RL & Raiffa H, 1993, *Decisions with multiple objectives: Preferences and value trade-offs*, 1st Edition, Cambridge University Press, Cambridge.

[103] Kendall K & Kendall J, 2011, *Systems analysis and design*, Pearson, Upper Saddle River (NJ).

[104] Kok BJ, 2010, *Evaluation of a threat evaluation and weapon assignment system in a ground-based air defence environment*, MSc Thesis, Stellenbosch University, Stellenbosch.

[105] Kumar S & Dixit AM, 2012, *Threat evaluation modelling for dynamic targets using fuzzy logic approach*, Proceedings of the International Conference on Computer Science and Engineering.

[106] Labuschagne PH, 2004, *ADA Doctrinal Note*, (Unpublished) Technical Report ADA-04/0028H Rev B, Air Defence Artillery, Kimberly.

[107] Lambert DA, 2001, *Situations for situation awareness*, Proceedings of the 4th International Conference on Information Fusion.

[108] Law AM & Kelton WD, 2000, *Simulation modeling and analysis*, 3rd Edition, McGraw-Hill, New York (NY).

[109] Lawler EL & Wood DE, 1966, *Branch-and-bound methods: A survey*, Operations Research, **14(4)**, pp. 699–719.

[110] Lee A & Mason M, 2002, *MATLAB simulation for computing probability of detection*, Proceedings of the 2002 IEEE Radar Conference, pp. 478–483.

[111] Lee ZJ, Su SF & Lee CY, 2003, *Efficiently solving general weapon-target assignment problem by genetic algorithms with greedy eugenics*, Transactions on Systems, Man and Cybernetics, **33(1)**, pp. 113–121.

[112] Lewis H & Papadimitriou CH, 1997, *Elements of the theory of computation*, 2nd Edition, Prentice-Hall, Cambridge (MA).

[113] Liebhaber MJ & Feher B, 2002, *Air threat assessment: Research, model, and display guidelines*, (Unpublished) Technical Report 0704-0188, Defence Technical Information Center Document, San Diego (CA).

[114] Liebhaber MJ & Smith C, 2000, *Naval air defense threat assessment: Cognitive factors and model*, (Unpublished) Technical Report 0704-0188, Pacific Science & Engineering Group, San Diego (CA).

[115] Lloyd SP & Witsenhausen HS, 1986, *Weapons allocation is NP-complete*, Proceedings of the 1986 Summer Computer Simulation Conference, pp. 1054–1058.

[116] Lötter DP, 2012, *Modelling weapon assignment as a multi-objective decision problem*, MComm Thesis, Stellenbosch University, Stellenbosch.

[117] Lötter DP & Van Vuuren JH, *A tri-objective, dynamic weapon assignment model for surface-based air defence*, ORiON, To appear.

[118] Lötter DP & Van Vuuren JH, 2014, *Implementation challenges associated with a threat evaluation and weapon assignment system*, Proceedings of the 43rd Annual Conference of the Operations Research Society of South Africa, pp. 27–35.

[119] Lötter DP & Van Vuuren JH, *Weapon assignment decision support in a surface-based air defence environment*, Military Operations Research, [Submitted].

[120] Lötter DP & Van Vuuren JH, 2013, *Weapon assignment decision support in a surface-based air defence environment*, ORiON, **29(1)**, pp. 31–54.

[121] Macfadzean RH, 1992, *Surface-based air defense system analysis*, 1st Edition, Artech House, Norwood (MA).

[122] Madhavan R & Messina E, 2006, *Performance metrics for intelligent systems (PerMIS) workshop: Summary and review*, (Unpublished) Technical Report 20899-8230, Intelligent Systems Division, National Institution of Standards and Technology, Gaithersburg (MD).

[123] Malcolm WP, 2004, *On the character and complexity of certain defensive resource allocation problems*, (Unpublished) Technical Report DSTO-TR-1570, Systems Sciences Laboratory, Weapon Systems Division, Edinburgh.

[124] Malhotra A & Jain R, 2002, *Genetic algorithm for optimal weapon allocation in multilayer defence scenario*, Defence Science Journal, **51(3)**, pp. 285–293.

[125] Manne AS, 1958, *A target-assignment problem*, Operations Research, **6(3)**, pp. 346–351.

[126] MathWorks, 2014, *User manual: Global optimisation toolbox*, [Online], [Cited October 10th, 2015], Available from `http://www.mathworks.com/help/pdf_doc/gads/gads_tb.pdf`.

[127] *MATLAB — The language of technical computing*, 2015, [Online], [Cited November 2nd, 2015], Available from `http://www.mathworks.com/products/matlab/`.

[128] Metler W & Preston F, 1989, *Solutions to a probabilistic resource allocation problem*, Proceedings of the 28th IEEE Conference on Decision and Control, pp. 1606–1611.

[129] Miller TE, Wolf SP, Thordsen ML & Klein G, 1992, *A decision-centered approach to storyboarding anti-air warfare interfaces*, (Unpublished) Technical Report 66001, Klein Associates Inc, Fairborn (OH).

[130] Ministry of Defence, 2004, *Aircraft accident to Royal Air Force Tornado*, (Unpublished) Technical Report GR MK4A ZG710, Directorate of Air Staff, London.

[131] *Modern day military pricing list*, 2015, [Online], [Cited October 2nd, 2015], Available from `http://nation-creation.wikia.com/wiki/Modern_Day_Military_Pricing_List`.

[132] Morecroft J & Robinson S, 2005, *Explaining puzzling dynamics: Comparing the use of system dynamics and discrete-event simulation*, Proceedings of the 23rd International Conference of the System Dynamics, pp. 17–21.

[133] Morrison JG, Kelly RT, Moore RA & Hutchins SG, 1997, *Tactical decision making under stress (TADMUS) decision support system*, Proceedings of the 1997 National Symposium on Sensor and Data Fusion, pp. 17–28.

[134] Murphey RA, 2000, *An approximate algorithm for a weapon target assignment stochastic program*, Approximation and Complexity in Numerical Optimization, Springer, Valparaiso (FL), pp. 406–421.

[135] Naeem H, Masood A, Hussain M & Shoab K, 2009, *A novel two-staged decision support based threat evaluation and weapon assignment algorithm*, International Journal of Computer Science and Information Security, **2(1)**, pp. 132–138.

[136] Naidoo S, 2009, *Modelling and simulation-based acquisition decision support: Present and future*, Proceedings of the South African Joint Air Defence Symposium, pp. 1–7.

[137] *NBS Mantis air defence protection system*, 2015, [Online], [Cited March 3rd, 2014], Available from `http://www.army-technology.com/projects/mantis/`.

[138] Ng GW & Ng KH, 2000, *Sensor management — What, why and how*, Information Fusion, **1(2)**, pp. 67–75.

[139] Nguyen XT, 2002, *Threat assessment in tactical airborne environments*, Proceedings of the 5th International Conference on Information Fusion, pp. 1300–1307.

[140] Nilsson M, 2008, *Mind the gap: Human decision making and information fusion*, Licentiate Thesis, University of Skövde, Örebro.

[141] North Atlantic Treaty Organization, 2012, *Glossary of terms and definitions*, (Unpublished) Technical Report AAP-6, Standarization Agency.

[142] North Atlantic Treaty Organization, 2015, *What is NATO?*, [Online], [Cited September 28th, 2015], Available from `http://www.nato.int/nato-welcome/`.

[143] Okello N & Thoms G, 2003, *Threat assessment using Bayesian networks*, Proceedings of the 6th International Conference on Information fusion, pp. 1102–1109.

[144] Olwell D & Washburn A, 2002, *Internetting of fires*, (Unpublished) Technical Report NPS-OR-02-003-PR, Naval Postgraduate School, Monterey (CA).

[145] Oracle, 2015, *Java Programming Language*, [Online], [Cited November 2nd, 2015], Available from `http://www.oracle.com/technetwork/java/index-138747.html`.

[146] Ottino JM, 2004, *Engineering complex systems*, Nature: International Weekly Journal of Science, **427(399)**.

[147] Oxenham M, 2000, *Automatic air target to airlane association*, Proceedings of the 3rd International Conference on Information Fusion, pp. 14–19.

[148] Oxenham MG, 2003, *Enhancing situation awareness for air defence via automated threat analysis*, Proceedings of the 6th International Symposium on Information Fusion, pp. 1086–1093.

[149] Paradis S, Benaskeur A, Oxenham M & Cutler P, 2005, *Threat evaluation and weapons allocation in network-centric warfare*, Proceedings of the 8th International Conference on Information Fusion, pp. 8–50.

[150] Patrick JD, 2001, *Electronic checklists on multi-purpose displays: A better way for fighter pilots to manage information and situational awareness during Periods of high workload*, (Unpublished) Technical Report AY 00-01, School of Advanced Military Studies, Topeka (KS).

[151] *Patriot battalion and battery operations field manual*, (Unpublished) Technical Report FM 3-01.85, 2003, Department of the Army, Washington (DC).

[152] *Patriot & PAC-3*, 2015, [Online], [Cited March 21st, 2014], Available from `http://www.bga-aeroweb.com/Defense/Patriot-PAC-3.html`.

[153] Perkins DG, 2013, *The importance of understanding mission command*, Proceedings of the AUSA Mission Command Symposium, Cansas (MO).

[154] Peterson C, 1990, *Parallel distributed approaches to combinatorial optimization: Benchmark studies on the travelling salesman problem*, Neural Computation, **2(3)**, pp. 261–269.

[155] Potgieter JJ, 2008, *Real-time weapon assignment in a ground based air defence environment*, MEngSc Thesis, Stellenbosch University, Stellenbosch.

[156] Presagis, 2015, *Top 5 trends in simulation-based training*, (Unpublished) Technical Report V5.1, Lockheed Martin.

[157] Qinga V, 2014, *Major milestone reached on GBADS*, Denel Integrated Systems Solutions Newsletter, Centurion.

[158] Rafael Advanced Defence Systems, 2014, *Iron Dome: Defense system against short range artillery rockets*, [Online], [Cited November 18th, 2014], Available from `http://www.rafael.co.il/marketing/SIP_STORAGE/FILES/6/946.pdf`.

[159] Raytheon, 2014, *Company profile*, [Online], [Cited November 18th, 2014], Available from `http://www.raytheon.com/ourcompany`.

[160] *Reutech radar systems*, 2012, [Online], [Cited February 24th, 2015], Available from `http://www.rrs.co.za/products/current-products/3d-radar`.

[161] *Rheinmetall Defence*, 2015, [Online], [Cited March 23rd, 2015], Available from `http://www.rheinmetall-defence.com/en/rheinmetall_defence/`.

[162] Rheinmetall Defence, 2014, *Rheinmetall's advanced air defence systems: Countering threats from above*, [Online], [Cited April 4th, 2014], Available from `http://www.rheinmetall-defence.com/media/editor_media/rm_defence/publicrelations/pressemitteilungen/2014_1/aad/2014_09_17_AAD_04_Air_Defence.pdf`.

[163] Richardson GP, 1999, *Feedback thought in social science and systems theory*, 2nd Edition, Pegasus Communications Inc, Waltham (MA).

[164] Robinson S, 2002, *Modes of simulation practice: Approaches to business and military simulation*, Simulation Modelling Practice and Theory, **10(8)**, pp. 513–523.

[165] Roeva O, Fidanova S & Paprzycki M, 2013, *Influence of the population size on the genetic algorithm performance in case of cultivation process modelling*, Proceedings of the IEEE Conference on Computer Science and Information Systems, pp. 371–376.

[166] Rosenberger JM & Yucel A, 2005, *The generalized weapon target assignment problem*, Proceedings of the 10th International Command and Control Research and Technology Symposium, McLean (VA), pp. 1–12.

[167] Roux JN, 2010, *Design of a threat evaluation subsystem in a ground-based air defence environment*, PhD Thesis, Stellenbosch University, Stellenbosch.

[168] Roux JN & Van Vuuren JH, 2008, *Real-time threat evaluation in a ground based air defence environment*, ORiON, **24(1)**, pp. 75–101.

[169] Roux JN & Van Vuuren JH, 2007, *Threat evaluation and weapon assignment decision support: A review of the state of the art*, ORiON, **23(2)**, pp. 151–187.

[170] Roy J, 2001, *From data fusion to situation analysis*, Proceedings of the 4th International Conference on Information Fusion.

[171] Roy J, Paradis S & Allouche M, 2002, *Threat evaluation for impact assessment in situation analysis systems*, Proceedings of the International Conference on Signal Processing, Sensor Fusion and Target Recognition, pp. 329–341.

[172] Rummel BK, 1998, *Subjective evaluation of human-computer interface options for a tactical decision support system*, (Unpublished) Technical Report 92152-5001, Defence Technical Information Center, San Diego (CA).

[173] SAAB, 2015, *Gripen fighter system*, [Online], [Cited September 21st, 2015], Available from `http://saab.com/air/gripen-fighter-system/gripen/gripen/the-fighter/gripen-cd/`.

[174] Salehi B, Cordero MI & Sandi C, 2010, *Learning under stress: The inverted-U-shape function revisited*, Learning & Memory, **17(10)**, pp. 522–530.

[175] Salerno J, Hinman M, Boulware D & Bello P, 2003, *Information fusion for situational awareness*, (Unpublished) Technical Report ADA442007, Defence Technical Information Center, Rome (NY).

[176] Salmon PM, Stanton NA, Walker GH, Jenkins D, Ladva D, Rafferty L & Young M, 2009, *Measuring situation awareness in complex systems: Comparison of measures study*, International Journal of Industrial Ergonomics, **39(3)**, pp. 490–500.

[177] Şandru V & Rădukescu M, 2013, *Requirements for ground-based air defense integrated systems*, RECENT Journal, **14(3)**, pp. 186–190.

[178] Sargent RG, 2005, *Verification and validation of simulation models*, Proceedings of the 37th Conference on Winter Simulation, pp. 130–143.

[179] Sergey S, 2005, *Introduction to Matlab graphical user interfaces*, (Unpublished) Technical Report DSTO-GD-0442, Australian Department of Defence, Defence Science and Technology Division, Edinburgh.

[180] Shahbazian E, Blodgett DE & Labbé P, 2001, *The extended OODA model for data fusion systems*, Proceedings of the International Conference on Information Fusion, pp. 19–25.

[181] Shim JP, Warkentin M, Courtney JF, Power DJ, Sharda R & Carlsson C, 2002, *Past, present, and future of decision support technology*, Decision Support Systems, **33(2)**, pp. 111–126.

[182] Skeels M, Lee B, Smith G & Robertson GG, 2010, *Revealing uncertainty for information visualization*, Information Visualization, **9(1)**, pp. 70–81.

[183] Smith Jr EA, 2001, *Network-centric Warfare: What's the point?*, Naval War College Review, **54(1)**, pp. 59–75.

[184] Smith C, Johnston J & Paris C, 2004, *Decision support for air warfare: Detection of deceptive threats*, Group Decision and Negotiation, **13(2)**, pp. 129–148.

[185] Smith JA, Harikumar J & Ruth BG, 2011, *An army-centric system-of-systems analysis definition*, (Unpublished) Technical Report ARL-TR-5446, Army Research Laboratory, Albuquerque (NM).

[186] Sommerer S, Guevara MD, Landis MA, Rizzuto JM, Sheppard JM & Grant CJ, 2012, *System-of-systems engineering in air and missile defence*, Johns Hopkins APL Technical Digest, **31(1)**, pp. 5–20.

[187] Sparrius A, 2014, Private Owner at *Ad Sparrius System Engineering and Management (Pty) Ltd*, [Personal Communication], Contactable at `ad_sparr@iafrica.com`.

[188] Sparrius A, 2014, *Foundations of Systems Engineering*, Lecture Notes, Stellenbosch University, Stellenbosch.

[189] *STAGE — A complete simulation development environment*, 2015, [Online], [Cited November 2nd, 2015], Available from `http://www.presagis.com/products_services/products/modeling-simulation/simulation/stage/`.

[190] Steinberg AN, 2009, *An approach to threat assessment*, Harbour Protection Through Data Fusion Technologies, Springer, Buffalo (NY).

[191] Steinberg AN & Bowman CL, 2004, *Rethinking the JDL data fusion levels*, Proceedings of the International Symposium on Sensor and Data Fusion, pp. 39–56.

[192] Stone ER, Yates JF & Parker AM, 1997, *Effects of numerical and graphical displays on professed risk-taking behavior*, Journal of Experimental Psychology, **3(4)**, pp. 243–257.

[193] Storr J, 2003, *A command philosophy for the information age: The continuing relevance of mission command*, Defence Studies, **3(3)**, pp. 119–129.

[194] Streilein JJ, 2009, *Test and evaluation of highly complex systems*, International Test and Evaluation Association, **30(2)**, pp. 3–6.

[195] Sullivan JM, 2005, *Revolution or evolution: The rise of the UAVs*, Proceedings of the IEEE Symposium on Prevention and Safety in a Time of Fear, pp. 94–101.

[196] Tako AA & Robinson S, 2009, *Comparing discrete-event simulation and system dynamics: Users' perceptions*, Journal of the Operational Research Society, **60(3)**, pp. 296–312.

[197] Telford B, 2012, *Verification, validation and accreditation recommended practices guide*, (Unpublished) Technical Report 03-57, Marine Corps M&S Management Office, Alexandria (VA).

[198] TERMA, 2015, *Threat evaluation: A component in the T-core complex*, [Online], [Cited March 20th, 2012], Available from `http://www.terma.com/defense/joint-and-land-systems/air-defense/`.

[199] Thales, 2014, *Advanced air defence*, [Online], [Cited November 18th, 2014], Available from `https://www.thalesgroup.com/sites/default/files/asset/document/AIR_DEFENCE_Brochure.pdf`.

[200] Thales, 2014, *South African Air Defence Formation proves there is no hiding place from the Starstreak day or night*, [Online], [Cited November 17th, 2014], Available from `https://www.thalesgroup.com/sites/default/files/asset/document/RSA%20firing%20camp%20-%20FINAL%20UK%20-%2026%20Nov.pdf`.

[201] The R Foundation, 2015, *What is R?*, [Online], [Cited November 2nd, 2015], Available from `https://www.r-project.org/about.html`.

[202] *The SAGE air defence system*, 2014, [Online], [Cited October 28th, 2014], Available from `https://www.ll.mit.edu/about/History/SAGEairdefensesystem.html`.

[203] The Staff of the Lincoln Laboratory, 1954, *The ground environment problem in air defense: An appraisal of the Lincoln Transition System*, (Unpublished) Technical Report, Lincoln Laboratory, Lexington (MA).

[204] Tin G & Cutler P, 2006, *Accomodating obstacle avoidance in the weapon allocation problem for tactical air defense*, Proceedings of the 9th International Conference on Information Fusion, pp. 1–8.

[205] Tokgöz A & Bulkan S, 2013, *Weapon target assignment with combinatorial optimization techniques*, International Journal of Advanced Research in Artificial Intelligence, **2(7)**, pp. 39–50.

[206] Truter ML & Van Vuuren JH, 2014, *Prerequisites for the design of a threat evaluation and weapon assignment system evaluator*, Proceedings of the 43rd Annual Conference of the Operations Research Society of South Africa, pp. 54–61.

[207] Truter ML & Van Vuuren JH, 2015, *Value-based methods for the threat value fusion process within a ground-based air defense environment*, Proceedings of the 44th Annual Conference of the Operational Research Society of South Africa, pp. 123–132.

[208] Ullman DG, 2003, *The mechanical design process*, 2nd Edition, McGraw-Hill, New York (NY).

[209] United States Army, 2003, *Mission command: Command and control of army forces*, (Unpublished) Technical Report FM 6-0, Headquarters, Department of the Army, Washington (DC).

[210] United States Department of Defence, 2015, *Missile Defence Agency*, [Online], [Cited March 18th, 2014], Available from `http://www.mda.mil/index.html`.

[211] United States Department of Defence, 1998, *Modeling and simulation glossary*, Modelling and Simulation Coordination Office, Alexandria (VA).

[212] United States Department of Defence, 2003, *Modeling and simulation verification, validation and accreditation*, (Unpublished) Technical Report 5000-61, Department of Defense Procedures for Management of Information Requirements, Washington (DC).

[213] United States Department of Defence, 2014, *Program aquisition cost by weapon system*, [Online], [Cited November 9th, 2014], Available from `http://comptroller.defense.gov/Portals/45/documents/defbudget/fy2015/fy2015_Weapons.pdf`.

[214] Vaidya OS & Kumar S, 2006, *Analytic hierarchy process: An overview of applications*, European Journal of Operational Research, **169(1)**, pp. 1–29.

[215] Van der Merwe M, 2013, *The weapon assignment scheduling problem in a ground-based air defence environment*, MSc Thesis, Stellenbosch University, Stellenbosch.

[216] Van Staden HE, 2013, *Attack technique classification in a ground-based air defence environment*, MComm Thesis, Stellenbosch University, Stellenbosch.

[217] Varshney L, 2002, *Human machine interface for radar systems*, (Unpublished) Technical Report Revision 3, Syracuse Research Corporation, North Syracuse (NY).

[218] Virtanen K, Hamalainen RP & Mattila V, 2006, *Team optimal signaling strategies in air combat*, Systems, Man and Cybernetics, **36(4)**, pp. 643–660.

[219] Visser B, 2006–2007, at *Military expert: Reutech Radar Systems*, [Personal Communication], Contactable at `bvisser@rrs.co.za`.

[220] Von Alan RH, March ST, Park J & Ram S, 2004, *Design science in information systems research*, Management Information Systems Quarterly, **28(1)**, pp. 75–105.

[221] Vorne Industries, 2013, *Lean production: Theory of constraints*, [Online], [Cited March 19th, 2015], Available from `http://www.leanproduction.com/theory-of-constraints.html`.

[222] Warren L, 2006, *Structural uncertainties in numerical induction models*, (Unpublished) Technical Report DSTO-TR-1895, Defence Science and Technology Organisation, Edinburgh.

[223] Wilson C, 2004, *Network centric warfare: Background and oversight issues for congress*, (Unpublished) Technical Report RL32238, Congressional Research Service, Washington (DC).

[224] Wohl JG, 1981, *Force management decision requirements for air force tactical command and control*, Systems, Man and Cybernetics, **11(9)**, pp. 618–639.

[225] Wohl JG, Entin EE & Eterno JS, 1983, *Modeling human decision processes in command and control*, (Unpublished) Technical Report 460-001, Alphatech Inc, Burlington (MA).

[226] Xiong N & Svensson P, 2002, *Multi-sensor management for information fusion: Issues and approaches*, Information Fusion, **3(2)**, pp. 163–186.

APPENDIX A

Multiple-attribute Utility Function Data

Data related to the construction of the multi-attribute utility function (5.7) in §5.3.1 are presented in Table A.1. The *Input* column refers to the information elicited from experts which was used to construct the multi-attribute utility function. The *Output* column, on the other hand, refers to the fused threat value calculated by the utility function. The last column is the difference between the output and input values which was used for verification purposes.

TABLE A.1: *Data points for the construction of the multiple-attribute utility function (5.7).*

V_d	V_p	V_a	Input	Output	Residual
L	L	L	0.16	0.23	0.07
L	L	M	0.29	0.34	0.05
L	L	H	0.42	0.46	0.04
L	M	L	0.36	0.36	0.00
L	M	M	0.50	0.48	−0.02
L	M	H	0.62	0.59	−0.03
L	H	L	0.56	0.49	−0.07
L	H	M	0.69	0.60	−0.09
L	H	H	0.82	0.72	−0.10
M	L	L	0.29	0.29	0.00
M	L	M	0.37	0.41	0.04
M	L	H	0.44	0.52	0.08
M	M	L	0.43	0.42	−0.01
M	M	M	0.50	0.54	0.04
M	M	H	0.57	0.65	0.08
M	H	L	0.56	0.55	−0.01
M	H	M	0.63	0.67	0.04
M	H	H	0.70	0.78	0.08
H	L	L	0.39	0.35	−0.04
H	L	M	0.50	0.47	−0.03
H	L	H	0.60	0.58	−0.02
H	M	L	0.51	0.48	−0.03
H	M	M	0.62	0.60	−0.02
H	M	H	0.72	0.72	0.00
H	H	L	0.62	0.61	−0.01
H	H	M	0.73	0.73	0.00
H	H	H	0.83	0.84	0.01

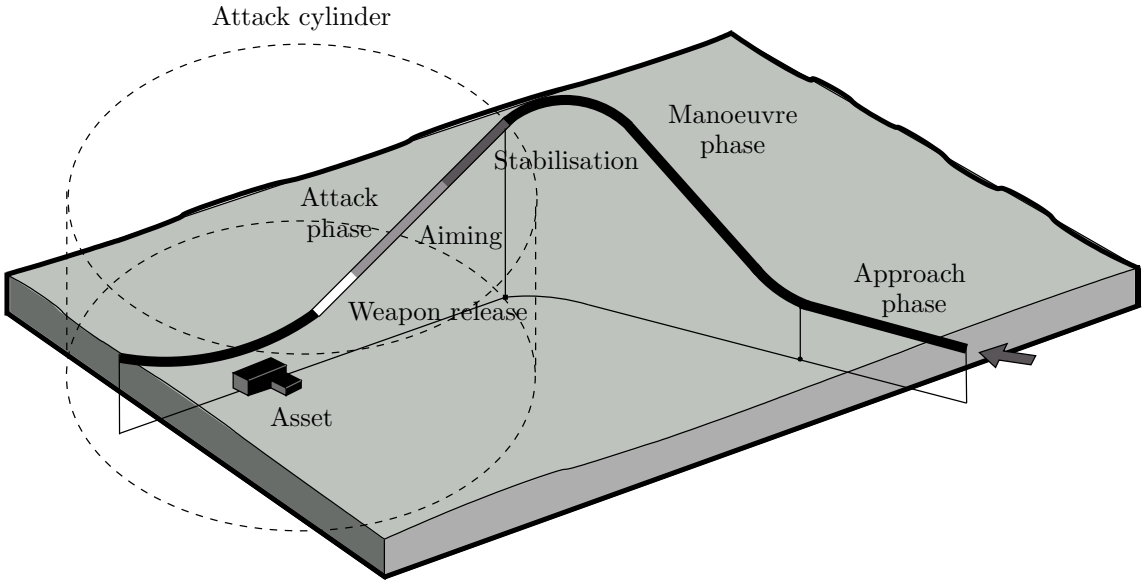
APPENDIX B

Aircraft Aerial Attack Manoeuvres

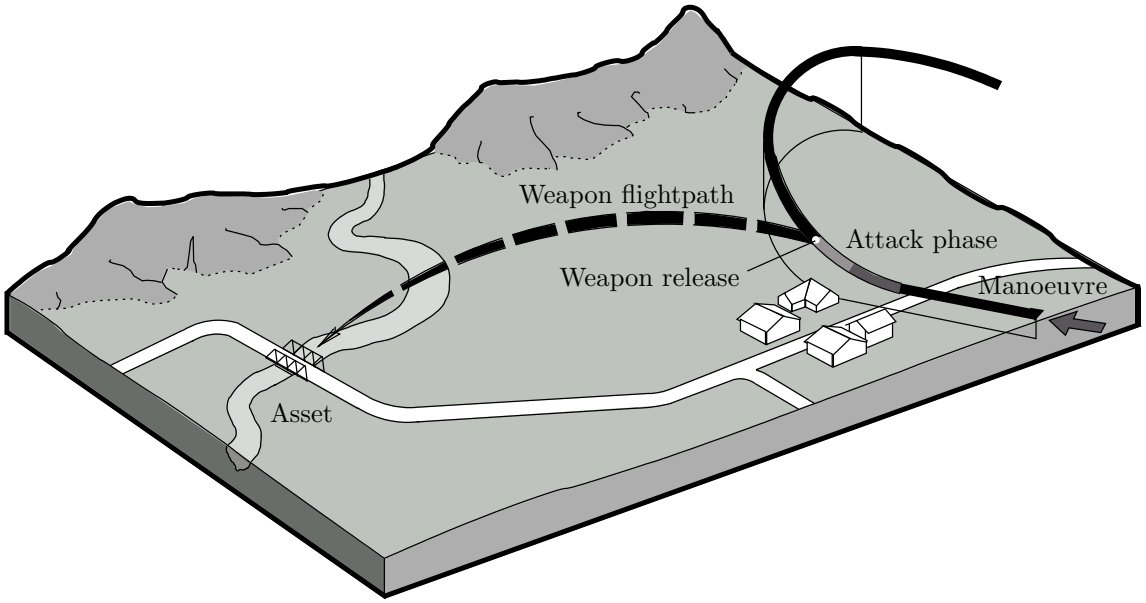
The profiles of the two types of air-to-surface aircraft attack techniques used in the examples presented in this thesis are graphically illustrated in Figure B.1. They include the pitch and dive attack technique (Figure B.1(a)) and the toss bomb attack technique (Figure B.1(b)).

The so-called *pitch and dive technique* — also sometimes referred to as the *combat hump dive* — is performed against static ground-based targets (*e.g.* DAs) and requires thorough planning before the mission, making it susceptible to known attack manoeuvre identification techniques [216]. This technique consists of a low-level approach, tangent to the so-called attack cylinder. The approach is followed by an rapid increase in altitude (pitch), a turn-in towards the DA (yaw and roll) and a stabilising dive before releasing its ordnance on the target.

The toss bomb technique, on the other hand, is generally performed against heavily defended, static DAs and also requires thorough planning beforehand [216]. The aircraft approaches at a low altitude which depends on the terrain and weather conditions. The point at which the aircraft executes its pull-up (pitch) is determined beforehand and calculated by the aircraft's navigational systems. The target acquisition, aiming and tracking are performed during this stage. The ordnance is released during the pull-up phase after which the aircraft regresses by turning away from the defended system.



(a) *Pitch and dive attack manoeuvre*



(b) *Toss bomb attack manoeuvre*

FIGURE B.1: *Typical fixed-wing aircraft air-to-surface attack manoeuvres [167].*

APPENDIX C

MATLAB Source Code

The source code of the simulation performance evaluation framework developed in Matlab® for this thesis is provided in this appendix. The main program code is provided in the first section, after which the source code for TE and WA sub-routines are given. The final section contains the source code of the periphery functions, such as the calculation of the SSHP value and visualisation of the spheres. A high-level overview of the roles of these source code sub-routines within the simulation paradigm is provided in §4.4. This section is also best read in conjunction with Figure 4.7.

C.1 Main Program

The `main.m` file is the heart of the simulation model. All the simulation entities are initialised and visualised in this program. The inputs with respect to the threats, WSs and DAs are programmed in this routine and the threat tracks are generated using B-splines. The tracks are also discretized into intervals that correspond to the duration of the TEWA cycle time. Each of the sub-routines performs operations on data structures within the simulation environment which are declared as global variables in order to increase computation speed and allow for better traceability.

```
1  c% Clear console, variables and close all figure windows
2  clc
3  clear all
4  close all
5
6  format long % Increase precision
7
8  % Change to current directory
9   currentDir = pwd;
10  cd(currentDir);
11
12 %% Constants
13 global N T D W defAirspace threat DA WS KOB
14
15 N = length(2:127);          % Number of TE intervals per threat path
16 defAirspace = 20e3;         % Area of Responsibility (AOR): Defended airspace ...
       sphere radius around each DA [m]
```

```
17 KOB = 8e3;                   % Keep Out Boundary - if threat is closer than this ...
       to DA, threat value should be scaled
18 T = 4;                       % Number of threats (max 4 for main program)
19 D = 2;                       % Number of defended DAs
20 W = 11;                      % Number of WSs
21 P = length(2:127);           % Number of input coordinates for threats
22
23 %% Input Parameters
24
25 load scenariodata            % Data file recieved from Andries Heyns
26 load terrain                 % Terrain data for visualisation
27
28 % Scale terrain data to fit threat tracks
29 scaleX = 1.883e+04+0.8e4+1200+2000;
30 scaleY = 3.396e+04-0.6e4+2000;
31 xmap = linspace(-25000, 25000, 785);
32 ymap = linspace(-28000, 25000, 651);
33
34 % Visualise terrain data
35 mapData = surf(xmap, ymap, AORarea-100,'FaceColor','interp','EdgeColor','none');
36 camlight left
37 demcmap(AORarea, 64)
38 % NOTE: Manual scaling of the colormap is required in order to obtain realistic ...
       terrain colors
39
40 % Input/load threats coordinates [m]
41 threat(1).WDP = 'Pitch and Dive';
42 threat(1).x = c813(2:127,2)'-scaleX;
43 threat(1).y = c813(2:127,3)'-scaleY;
44 threat(1).z = c813(2:127,4)';
45 threat(1).speed = 240;        % Speed m/s
46 threat(1).WRL = 113;          % Weapon Release Line
47 threat(1).target = 2;         % DA target
48 threat(1).success = 0;        % Binary variable: 1 destroyed, 0 if not
49 threat(1).numEngaged = 0;     % Time stage threat was destroyed
50
51 threat(2).WDP = 'Toss Bomb';
52 threat(2).x = tb(2:127,2)'-scaleX;
53 threat(2).y = tb(2:127,3)'-scaleY;
54 threat(2).z = tb(2:127,4)';
55 threat(2).speed = 240;
56 threat(2).WRL = 82;
57 threat(2).target = 1;
58 threat(2).success = 0;
59 threat(2).numEngaged = 0;
60
61 threat(3).WDP = 'Low Level Fly-Over';
62 threat(3).x = fover1(2:127,2)'-scaleX;
63 threat(3).y = fover1(2:127,3)'-scaleY;
64 threat(3).z = fover1(2:127,4)';
65 threat(3).speed = 250;
66 threat(3).numEngaged = 0;
67
68 threat(4).WDP = 'Low Level Fly-Over';
69 threat(4).x = fover2(2:127,2)'-scaleX;
70 threat(4).y = fover2(2:127,3)'-scaleY;
71 threat(4).z = fover2(2:127,4)';
72 threat(4).speed = 250;
73 threat(4).numEngaged = 0;
74
75 % Defended DA details - all coordinates in metres
```

```
76 scaleWSx = -1700;
77 scaleWSy = 0;
78
79 DA(1).type = 'Command Centre';
80 DA(1).x = 100;
81 DA(1).y = -2115;
82 DA(1).z = 312;
83 DA(1).val = 90;
84
85 DA(2).type = 'Hanger';
86 DA(2).x = -2315;
87 DA(2).y = 577;
88 DA(2).z = 0;
89 DA(2).val = 50;
90
91 % WSs
92 WS(1).type = 'CIWS';
93 WS(1).x = -300;
94 WS(1).y = -1000;
95 WS(1).z = 0;
96 WS(1).reloadTime = 4;
97 WS(1).ammo = 11;
98 WS(1).cost = 100000;
99
100 WS(2).type = 'CIWS';
101 WS(2).x = -400;
102 WS(2).y = -500;
103 WS(2).z = 0;
104 WS(2).reloadTime = 4;
105 WS(2).ammo = 5;
106 WS(2).cost = 100000;
107
108 WS(3).type = 'CIWS';
109 WS(3).x = 500;
110 WS(3).y = -400;
111 WS(3).z = 0;
112 WS(3).reloadTime = 4;
113 WS(3).ammo = 3;
114 WS(3).cost = 500000;
115
116 WS(4).type = 'CIWS';
117 WS(4).x = 500;
118 WS(4).y = -1000;
119 WS(4).z = 0;
120 WS(4).reloadTime = 4;
121 WS(4).ammo = 3;
122 WS(4).cost = 500000;
123
124 WS(5).type = 'VSHORAD';
125 WS(5).x = -2400;
126 WS(5).y = -800;
127 WS(5).z = 0;
128 WS(5).reloadTime = 4;
129 WS(5).ammo = 3;
130 WS(5).cost = 500000;
131
132 WS(6).type = 'VSHORAD';
133 WS(6).x = -1500;
134 WS(6).y = 900;
135 WS(6).z = 0;
136 WS(6).reloadTime = 4;
```

```
137 WS(6).ammo = 3;
138 WS(6).cost = 500000;
139
140 WS(7).type = 'VSHORAD';
141 WS(7).x = -0;
142 WS(7).y = 1600;
143 WS(7).z = 0;
144 WS(7).reloadTime = 4;
145 WS(7).ammo = 3;
146 WS(7).cost = 500000;
147
148 WS(8).type = 'VSHORAD';
149 WS(8).x = 2000;
150 WS(8).y = 800;
151 WS(8).z = 0;
152 WS(8).reloadTime = 4;
153 WS(8).ammo = 3;
154 WS(8).cost = 500000;
155
156 WS(9).type = 'VSHORAD';
157 WS(9).x = 1100;
158 WS(9).y = -2800;
159 WS(9).z = 0;
160 WS(9).reloadTime = 4;
161 WS(9).ammo = 15;
162 WS(9).cost = 500000;
163
164 WS(10).type = 'VSHORAD';
165 WS(10).x = -500;
166 WS(10).y = -3300;
167 WS(10).z = 0;
168 WS(10).reloadTime = 4;
169 WS(10).ammo = 15;
170 WS(10).cost = 500000;
171
172 WS(11).type = 'VSHORAD';
173 WS(11).x = -2100;
174 WS(11).y = -2600;
175 WS(11).z = 0;
176 WS(11).reloadTime = 4;
177 WS(11).ammo = 15;
178 WS(11).cost = 500000;
179
180 % Scale WSs
181 for w = 1:W
182     WS(w).x =  WS(w).x + scaleWSx;
183     WS(w).y =  WS(w).y + scaleWSy;
184 end
185
186 % Find required altitude to place DAs and WSs so that they are on terrain surface
187 for d = 1:D
188     [c_temp_x, index_x] = min(abs(xmap - DA(d).x));
189     [c_temp_y, index_y] = min(abs(ymap - DA(d).y));
190     DA(d).z = AORarea(index_x, index_y);
191 end
192
193 for w = 1:W
194     [c_temp_x, index_x] = min(abs(xmap - WS(w).x));
195     [c_temp_y, index_y] = min(abs(ymap - WS(w).y));
196     WS(w).z = AORarea(index_x, index_y);
197 end
```

```
198
199 % Figure environment for scenario
200 figure(1)
201 hold on
202
203 %% Add random noise around input coordinates
204 SphereRadius = 5;   % [m] The randomised sphere around input coordinates
205
206 %Plot random spheres
207 [x_sphere, y_sphere, z_sphere] = sphere;
208 x_sphere =  SphereRadius.*2*x_sphere;
209 y_sphere =  SphereRadius.*2*y_sphere;
210 z_sphere =  SphereRadius.*2*z_sphere;
211
212 % Draw random spheres
213 for t = 1:T % For all threats
214     for k = 1:P % For all input coordinates
215         sphere_properties = surf(x_sphere + threat(t).x(k), y_sphere + ...
              threat(t).y(k), z_sphere + threat(t).z(k));
216         set(sphere_properties,'facealpha',0)
217     end
218 end
219
220 for t = 1:T % For all threats
221     for z = 1:3 % Counter for x-y-z points, ensure that an 'unique'random number ...
          is assigned to each point.
222         for k = 1:P
223             % Random noise added in the form of a sphere round the input ...
                coordinate with radius defined above, either positive or neg ...
                (last term)
224             switch z
225                 case 1
226                     threat(t).x(k) = threat(t).x(k) + (-SphereRadius + ...
                          (SphereRadius*2)*rand(1,1));
227                 case 2
228                     threat(t).y(k) = threat(t).y(k) + (-SphereRadius + ...
                          (SphereRadius*2)*rand(1,1));
229                 case 3
230                     threat(t).z(k) = threat(t).z(k) + (-SphereRadius + ...
                          (SphereRadius*2)*rand(1,1));
231                 otherwise
232                     break
233             end
234         end
235     end
236 end
237
238 %% Setup Scenario
239
240 % Plot input coordinates
241 graphProperties = 0;
242 for t=1:T
243     threat(t).combinedCoor = [threat(t).x; threat(t).y; threat(t).z];
244
245     switch graphProperties
246         case 0
247             properties = 'bo';
248         case 1
249             properties = 'ro';
250         case 2
251             properties = 'go';
```

```
252         case 3
253             properties = 'ko';
254         case 4
255             properties = 'yo';
256         otherwise
257             disp('ERROR: T exceeded') % If more that four threats are present
258     end
259
260     plot3(threat(t).x, threat(t).y, threat(t).z, properties,'LineWidth', 3);
261     graphProperties = graphProperties + 1;
262 end
263
264 %plot DAs positions
265 for d = 1:D
266     plot3(DA(d).x, DA(d).y, DA(d).z,'ks','MarkerSize', 8, 'MarkerFaceColor', 'k');
267 end
268
269 %plot WSs positions
270 for w = 1:W
271     plot3(WS(w).x, WS(w).y, WS(w).z,'k^','MarkerSize', 8, 'MarkerFaceColor', 'k');
272 end
273
274 % Label DAs and WSs
275 labelsDAs = cellstr(num2str([1:D]'));
276 labelsWSs = cellstr(num2str([1:W]'));
277
278 % Plot DA number on DA
279 for d = 1:D
280     text(DA(d).x, DA(d).y, DA(d).z, labelsDAs(d), 'HorizontalAlignment', 'right')
281 end
282
283 % Plot WS number on WS
284 for w = 1:W
285     text(WS(w).x, WS(w).y, WS(w).z, labelsWSs(w), 'HorizontalAlignment', 'right')
286 end
287
288  %Determine speed at each of the TE points
289  for t = 1:T % For all threats
290      if (length(threat(t).speed) > 1) % Only if more than one speed coordinate ...
            is provided
291
292          for k = 1:(P-1)
293              if k == 1
294                  threat(t).speedCompl(1 : k*(N/(P - 1))) = threat(t).speed(k);
295              else
296                  threat(t).speedCompl((k-1)*(N/(P - 1)) : k*(N/(P - 1))) = ...
                        threat(t).speed(k);
297              end
298
299          end
300      end
301  end
302
303 % Generate and visualise the Splines representing threat tracks
304 graphProperties = 0;
305 for t=1:T
306     % Create spline: use csvn, instead of spline(X,Y), because the coordinates ...
          are not necessarily unique or monotonic
307     threat(t).spline = cscvn(threat(t).combinedCoor);
308     % Divide break points into the desired number of intervals
```

```
309     threat(t).breakIntervals = linspace(min(threat(t).spline.breaks), ...
            max(threat(t).spline.breaks), N);
310     % Determine coordinates of points where TE needs to be conducted from the ...
            underlying piecewise-polynomials
311     threat(t).TEpoints = fnval(threat(t).spline, threat(t).breakIntervals);
312 
313     % Determine colour of spline
314         switch graphProperties
315         case 0
316             properties = 'b';
317         case 1
318             properties = 'r';
319         case 2
320             properties = 'g';
321         case 3
322             properties = 'k';
323         case 4
324             properties = 'y';
325         otherwise
326             disp('ERROR')
327         end
328 
329     % Plot splines with special function
330     fnplt(threat(t).spline, properties, 2)
331     graphProperties = graphProperties + 1;
332 end
333 
334 % Plot points to be used for verification purposes
335 TEWAcycle = 20;
336 for p = 1:5
337     for t = 1:T
338         plot3(threat(t).TEpoints(1,TEWAcycle), threat(t).TEpoints(2,TEWAcycle), ...
                threat(t).TEpoints(3,TEWAcycle),'.k', 'MarkerSize', 16)
339     end
340     TEWAcycle = TEWAcycle + 20;
341 end
342 
343 %% Visualize airspace
344 h1 = drawDome(DA(1).x, DA(1).y, DA(1).z, defAirspace); % DA1 defended airspace
345 h2 = drawDome(DA(2).x, DA(2).y, DA(2).z, defAirspace); % DA2 defended airspace
346 [h1 h2 h3 h4 h5] = drawSSHPvolume(WS(1).x, WS(1).y, WS(1).z, WS(1).maxRange); % ...
    WS1 SSHP volume
347 [g1 g2 g3 g4 g5] = drawSSHPvolume(WS(2).x, WS(2).y, WS(2).z, WS(2).maxRange); % ...
    WS2 SSHP volume
348 
349 shading(gca, 'interp')
350 transparency = 0.1;
351  % Set colors of SSHP volumes
352  set(h1, 'FaceColor', [0 0 0.5], 'FaceAlpha', transparency);
353  set(h2, 'FaceColor', [0 0.5 0.5], 'FaceAlpha', transparency);
354  set(h3, 'FaceColor', [0.5 0.5 0.5], 'FaceAlpha', transparency);
355  set(h4, 'FaceColor', [0.5 0.5 0], 'FaceAlpha', transparency);
356  set(h5, 'FaceColor', [0.5 0 0], 'FaceAlpha', transparency);
357 
358  set(g1, 'FaceColor', [0 0 0.5], 'FaceAlpha', transparency);
359  set(g2, 'FaceColor', [0 0.5 0.5], 'FaceAlpha', transparency);
360  set(g3, 'FaceColor', [0.5 0.5 0.5], 'FaceAlpha', transparency);
361  set(g4, 'FaceColor', [0.5 0.5 0], 'FaceAlpha', transparency);
362  set(g5, 'FaceColor', [0.5 0 0], 'FaceAlpha', transparency);
363 
364 xlabel('x coordinate (m)')
```

```
365 ylabel('y coordinate (m)')
366 zlabel('z coordinate (m)')
367
368 grid on
369 axis('equal');
370 view([-30, 30])
371
372 hold off
373
374 % Obtaining inputs for passenger plane - only used if required
375 view([0, 90])
376 [x_civ, y_civ] = ginput(P);
377
378 %% Execute Threat evaluation Routine
379 threatEvaluation()
380
381 %% Execute Threat Fusion Routine
382 ThreatFusion()
383
384 %% Execute Weapon Assignment Routine
385
386  weaponAssignment('Dynamic')
```

C.2 Threat Evaluation

The `threatEvaluation.m` file is responsible for calculating the deterministic model threat values, as detailed in §5.2, and performing all required underlying calculations. The deterministic threat values for all threats are also visualised by executing this program.

```
1  function [] = threatEvaluation()
2  % x: x positions, y: y positions, current: current position in vector
3  % Recieves all current and historical trajectory data.
4  % Utilize this data to calculate heading, identify attack manuevers etc.
5
6  global N T D defAirspace threat DA KOB
7
8  Pd = KOB*0.9;     % Action distance
9  WRL = 2.5e3;      % Weapon release line - destroy before this slant distance
10
11 %% Threat Values Calculations
12
13 for t = 1:T      % Determine TV for each threat
14     for d = 1:D  % for each DA
15         for n = 1:N % For all points N
16             currentPos = [threat(t).TEpoints(1,n), threat(t).TEpoints(2,n), ...
                   threat(t).TEpoints(3,n); DA(d).x, DA(d).y, DA(d).z];
17
18             % Only execute TE if a threat is within the defended airspace
19             if pdist(currentPos, 'euclidean') < defAirspace
20
21                 % Calculate threat value: normalised distance
22                 threat(t).da(d).dist(n) = (pdist(currentPos, 'euclidean') - ...
                       defAirspace)/(WRL-defAirspace);
23                 if threat(t).da(d).dist(n) > 1; threat(t).da(d).dist(n) = 1; end
24
25                 % Calculate threat value: course
```

```
26                  currentDirVector = createVector(t,n,3); % Current threat ...
                        directional vector - using current position and previous ...
                        position
27                  currentDaApproachVector = [DA(d).x - threat(t).TEpoints(1,n), ...
                        DA(d).y - threat(t).TEpoints(2,n), DA(d).z - ...
                        threat(t).TEpoints(3,n)]; % Current superimposed da approach ...
                        vector - using current position and previous position x,y ...
                        and z coordinates
28                  attackAngle = atan2(norm(cross(currentDirVector, ...
                        currentDaApproachVector)), dot(currentDirVector, ...
                        currentDaApproachVector)); % Smallest angle between ...
                        superimposed da approach vector and threat direction vector. ...
                        atan = (||a X b||/a dot b) - more robust, use sin and cos to ...
                        calculate angle
29                  if attackAngle < pi/2 % Check if angle is smaller than 90deg ...
                        (approaching da), otherwise 0
30                      threat(t).da(d).course(n) = cos(attackAngle);
31                  else
32                      threat(t).da(d).course(n) = 0;
33                  end
34
35                  % Calculate threat value: Altitude
36                  currentAlt = threat(t).TEpoints(3,n); % Current Alt in m
37                  if currentAlt <= 200
38                      threat(t).da(d).altitude(n) = 1;
39                  elseif currentAlt <= 10000
40                      threat(t).da(d).altitude(n) = -currentAlt/9800 + 100/98;
41                  else
42                      threat(t).da(d).altitude(n) = 0;
43                  end
44
45                  % Calculate threat value: CPA
46                  % Only x-y plane
47                  currentPos = [threat(t).TEpoints(1,n), threat(t).TEpoints(2,n); ...
                        DA(d).x, DA(d).y];
48                  currentDirVector = createVector(t,n,2);
49                  currentDaApproachVector = [DA(d).x - threat(t).TEpoints(1,n), ...
                        DA(d).y - threat(t).TEpoints(2,n)];
50                  attackAngle = acos(dot(currentDirVector, ...
                        currentDaApproachVector)/(norm(currentDirVector)* ...
                        norm(currentDaApproachVector))); % Angle between threat's ...
                        approach (direction vector) and superimposed straight line ...
                        to DA
51                  straightDist2Da = pdist(currentPos, 'euclidean');   % Calculate ...
                        euclidean (straight-line) distance from threat to the ...
                        defended asset
52                  pd = sin(attackAngle)*straightDist2Da;  % Current ortagonal ...
                        passing distance to defended asset
53                  if ((pd <= Pd) && (threat(t).da(d).course(n) ~= 0))  % check if ...
                        threat is passing at a threatening distance and if it is ...
                        approaching the DA, also if it is within a bombing altitude
54                      threat(t).da(d).passing(n) = 1 - pd/Pd;
55                  else
56                      threat(t).da(d).passing(n) = 0;
57                  end
58
59              else
60                  threat(t).da(d).dist(n) = 0;
61                  threat(t).da(d).passing(n) = 0;
62                  threat(t).da(d).course(n) = 0;
63              end
```

```
64          end
65      end
66  end
67
68  %% Plot Threat Values
69  xx = 1:N;
70
71  figure(2)
72  p1 = plot(xx, threat(1).da(1).dist(:), 'r', xx, threat(1).da(1).altitude(:), ...
        'r--', xx, threat(1).da(1).passing(:), 'r:', xx, threat(1).da(1).course(:), ...
        'r-.', xx, threat(1).da(2).dist(:), 'b', xx, threat(1).da(2).altitude(:), ...
        'b--', xx, threat(1).da(2).passing(:), 'b:', xx, threat(1).da(2).course(:), ...
        'b-.');
73  title('Threat 1')
74  ylabel('Threat value')
75
76  figure(3)
77  p2 = plot(xx, threat(2).da(1).dist(:), 'r', xx, threat(2).da(1).altitude(:), ...
        'r--', xx, threat(2).da(1).passing(:), 'r:', xx, threat(2).da(1).course(:), ...
        'r-.', xx, threat(2).da(2).dist(:), 'b', xx, threat(2).da(2).altitude(:), ...
        'b--', xx, threat(2).da(2).passing(:), 'b:', xx, threat(2).da(2).course(:), ...
        'b-.');
78  title('Threat 2')
79  ylabel('Threat value')
80
81  figure(5)
82  p4 = plot(xx, threat(4).da(1).dist(:), 'r', xx, threat(4).da(1).altitude(:), ...
        'r--', xx, threat(4).da(1).passing(:), 'r:', xx, threat(4).da(1).course(:), ...
        'r-.', xx, threat(4).da(2).dist(:), 'b', xx, threat(4).da(2).altitude(:), ...
        'b--', xx, threat(4).da(2).passing(:), 'b:', xx, threat(4).da(2).course(:), ...
        'b-.');
83  title('Threat 4')
84  ylabel('Threat value')
85  xlabel('Time (s)')
86  legend('DA1-slant dist', 'DA1-altitude', 'DA1-CPA','DA1-Course' ,'DA2-slant ...
        dist', 'DA2-altitude','DA2-CPA', 'DA2-Course', 'Location', 'SouthOutside')
87
88  figure(4)
89  p3 = plot(xx, threat(3).da(1).dist(:), 'r', xx, threat(3).da(1).altitude(:), ...
        'r--', xx, threat(3).da(1).passing(:), 'r:', xx, threat(3).da(1).course(:), ...
        'r-.', xx, threat(3).da(2).dist(:), 'b', xx, threat(3).da(2).altitude(:), ...
        'b--', xx, threat(3).da(2).passing(:), 'b:', xx, threat(3).da(2).course(:), ...
        'b-.');
90  title('Threat 3')
91  ylabel('Threat value')
92
93  %% Sub-functions
94  function vector = createVector(threatNum, pointNum, dimensions)
95      %   threatNum: number of threat under consideration
96      %   pointNum: number of current point for which the vector is reqire
97      %   dimensions: Number of dimensions to calculate vector (2 or 3)
98
99      % Avoid function trying to access an invalid matrix(0) entry - matlab ...
            matrixes start at 1
100     if pointNum == 1
101         pointNum = 2;
102     end
103
104     switch dimensions
105         case 3
```

```
106             dx = threat(threatNum).TEpoints(1,pointNum) - ...
                    threat(threatNum).TEpoints(1,pointNum - 1);
107             dy = threat(threatNum).TEpoints(2,pointNum) - ...
                    threat(threatNum).TEpoints(2,pointNum - 1);
108             dz = threat(threatNum).TEpoints(3,pointNum) - ...
                    threat(threatNum).TEpoints(3,pointNum - 1);
109             vector = [dx, dy, dz];
110         case 2
111             dx = threat(threatNum).TEpoints(1,pointNum) - ...
                    threat(threatNum).TEpoints(1,pointNum - 1);
112             dy = threat(threatNum).TEpoints(2,pointNum) - ...
                    threat(threatNum).TEpoints(2,pointNum - 1);
113             vector = [dx, dy];
114         otherwise
115             disp('ERROR: During vector creation');
116     end
117 end
118
119 end
```

C.3 Threat Value Fusion

The threat fusion sub-routine serves the purpose of executing the threat value fusion processes described in §5.3.

```
1  function [] = ThreatFusion(~)
2  %function [totPreferenceValue] = ThreatFusion(~) print out values
3  %   Fuse all the threat-DA threat values to obtain one system threat per threat ...
       value.
4  %
5
6  global N T D threat DA fusedTVs4Threat defAirspace KOB
7
8  M = 4;  % Number of Different Threat Evaluation Models
9  SCALING_KOB = 2/3; % Scaling factor for system threat value if within KOB
10
11 % Initialise variables
12 tempDist = zeros(1,N);
13 tempCourse = zeros(1,N);
14 tempPassing = zeros(1,N);
15 tempAltitude = zeros(1,N);
16
17 for t = 1:T
18     threat(t).system_tv = zeros(1,N);
19     threat(t).system_tv_new = zeros(1,N);
20 end
21
22 %% Normalised DA Importance Values
23 % Concatenates the DA importance values, calculates the sum and normalise the ...
       importance values - row index correspond to DA number
24 daNormValue = cat(1,DA.val)/sum(cat(1,DA.val));
25
26 %% Fusion first part, all DMs together, then fuse with respect to the different DAs
27
28 % The weighting functions for the model fusion is a function of distance from a DA
29 % These functions were used for the initial calibration purposes
30 a1 = -0.000957;
```

```
a2 = 1.245e4;
weightDist = @(x) 1/(1 + exp(-a1*(x-a2)));

b1 =  -7.281e-06;
b2 =   0.0005337;
b3 =           1;
b4 =  -1.024e-05;
weightPassing = @(x) b1.*exp(b2.*x) + b3.*exp(b4.*x);

c1 = -2.925e-09;
c2 = 1.41e-05;
c3 = 0.974;
weightCourse = @(x) c1*x^2 + c2*x + c3;

p1 =   4.583e-13;
p2 =   -9.25e-09;
p3 =  -3.333e-06;
p4 =           1;
weightAltitude = @(x) p1*x^3 + p2*x^2 + p3*x + p4;
figure(10)
hold on
fplot(weightDist, [0 20e3],'k');
fplot(weightPassing, [0 20e3],'k:');
fplot(weightCourse, [0 20e3],'k--');
fplot(weightAltitude, [0 20e3],'k*');
title('Models weighting functions')
ylabel('Function relative importance weight')
xlabel('Threat distance from defended DA')
legend('Slant Distance', 'CPA', 'Approach Angle (Course)', 'Altitude')
hold off

for t = 1:T % Do for all threats
    for d = 1:D % Do for all DMs
        for n = 1:N
            currentPos = [threat(t).TEpoints(1,n), threat(t).TEpoints(2,n), ...
                threat(t).TEpoints(3,n); DA(d).x, DA(d).y, DA(d).z];
            currentDist = pdist(currentPos, 'euclidean');
            threat(t).da(d).distanceFrom(n) = currentDist; % Current distance ...
                from DA

            % Calculate fused threat value
            if(currentDist < defAirspace) && (currentDist > KOB) % Only start ...
                fusion if threats are within defended airspace and outside KOB
                threat(t).da(d).fused(n) = ...
                    (weightDist(currentDist)*threat(t).da(d).dist(n) + ...
                    weightPassing(currentDist)*threat(t).da(d).passing(n) + ...
                    weightCourse(currentDist)* threat(t).da(d).course(n) + ...
                    weightAltitude(currentDist)* threat(t).da(d).altitude(n))/ ...
                    (weightDist(currentDist) + weightPassing(currentDist) + ...
                    weightCourse(currentDist) + weightAltitude(currentDist));

                threat(t).da(d).fusedFunc(n) = ...
                    valueFuse(threat(t).da(d).dist(n), ...
                    threat(t).da(d).passing(n), threat(t).da(d).altitude(n)); % ...
                    Different method
                if  threat(t).da(d).fusedFunc(n) > 1;  ...
                    threat(t).da(d).fusedFunc(n) = 1; end % Limit threat value

            elseif(currentDist < KOB)
                threat(t).da(d).fused(n) = ...
                    (weightDist(currentDist)*threat(t).da(d).dist(n) + ...
```

```
                    weightPassing(currentDist)*threat(t).da(d).passing(n) + ...
                    weightCourse(currentDist)* threat(t).da(d).course(n)+ ...
                    weightAltitude(currentDist)* threat(t).da(d).altitude(n))/ ...
                    (weightDist(currentDist) + weightPassing(currentDist) + ...
                    weightCourse(currentDist) + weightAltitude(currentDist));
78                  threat(t).da(d).fused(n) = threat(t).da(d).fused(n) + (1 - ...
                    threat(t).da(d).fused(n))*SCALING_KOB; % If within KOB, ...
                    scale threat value
79
80                  threat(t).da(d).fusedFunc(n) = ...
                    valueFuse(threat(t).da(d).dist(n), ...
                    threat(t).da(d).passing(n), threat(t).da(d).altitude(n)); % ...
                    Different method
81                  if  threat(t).da(d).fusedFunc(n) > 1;  ...
                    threat(t).da(d).fusedFunc(n) = 1; end % Limit threat value
82                  threat(t).da(d).fusedFunc(n) = threat(t).da(d).fusedFunc(n) + (1 ...
                    - threat(t).da(d).fusedFunc(n))*SCALING_KOB; % If within ...
                    KOB, scale threat value
83
84              else
85                  threat(t).da(d).fused(n) = 0;
86              end
87
88          end
89      end
90  end
91
92  % Calculate system threat values for each threat
93  for t = 1:T
94      for d = 1:D
95          threat(t).system_tv = threat(t).system_tv + ...
                threat(t).da(d).fused*daNormValue(d);
96          threat(t).system_tv_new = threat(t).system_tv_new + ...
                threat(t).da(d).fusedFunc*daNormValue(d);
97      end
98  end
99
100 %% Threat Value Fusion
101 % Fused threat values for different threat evaluation models, using additive ...
        fusion with DA importance value as weight
102 for t = 1:T
103     for d = 1:D
104         tempDist = tempDist + threat(t).da(d).dist*daNormValue(d);
105         tempCourse = tempCourse + threat(t).da(d).course*daNormValue(d);
106         tempPassing = tempPassing + threat(t).da(d).passing*daNormValue(d);
107         tempAltitude = tempAltitude + threat(t).da(d).altitude*daNormValue(d);
108     end
109
110     fusedTVs4Threat(t).dist = tempDist;
111     fusedTVs4Threat(t).course = tempCourse;
112     fusedTVs4Threat(t).passing = tempPassing;
113     fusedTVs4Threat(t).altitude = tempAltitude;
114
115     tempPassing = zeros(1,N);
116     tempCourse = zeros(1,N);
117     tempDist = zeros(1,N);
118     tempAltitude = zeros(1,N);
119 end
120
121 %% Plot Fused threat Values
122 x = 1:N;
```

```
123
124 % Fused threat values per DA
125 figure(12)
126 plot(x, threat(1).da(1).fusedFunc(:), 'r', x, threat(1).da(2).fusedFunc(:), 'b')
127 title('Threat 1')
128 ylabel('Threat Value')
129 xlabel('Time(s)')
130 axis([0 127 0 1])
131
132 figure(13)
133 plot(x, threat(2).da(1).fusedFunc(:), 'r', x, threat(2).da(2).fusedFunc(:), 'b')
134 title('Threat 2')
135 ylabel('Threat Value')
136 xlabel('Time (s)')
137 axis([0 127 0 1])
138 legend('wrt DA1', 'wrt DA2')
139
140 figure(14)
141 plot(x, threat(3).da(1).fusedFunc(:), 'r', x, threat(3).da(2).fusedFunc(:), 'b')
142 title('Threat 3')
143 ylabel('Threat Value')
144 xlabel('Time (s)')
145 axis([0 127 0 1])
146 legend('wrt DA1', 'wrt DA2')
147
148 figure(16)
149 plot(x, threat(4).da(1).fusedFunc(:), 'r', x, threat(4).da(2).fusedFunc(:), 'b')
150 title('Threat 4')
151 ylabel('Threat Value')
152 xlabel('Time (s)')
153 axis([0 127 0 1])
154 legend('wrt DA1', 'wrt DA2')
155
156
157 figure(15)
158
159 plot(x, threat(1).system_tv_new(:), 'b', x, threat(2).system_tv_new(:), 'r', x, ...
        threat(3).system_tv_new(:), 'g', x, threat(4).system_tv_new(:), 'k')
160 title('System threat values')
161 ylabel('Threat value')
162 xlabel('Time (s)')
163 legend('Threat 1', 'Threat 2', 'Threat 3', 'Threat 4')
164
165 % Routine to extract value function values for different TEM combinations ...
        (Low(0,16), Medium(0.5) and High(0.83))
166 for d = 1:D
167     for t = 1:T
168         for n = 1:N
169
170             % 1 = Low
171             % 2 = Medium
172             % 3 = High
173
174             if ((threat(t).da(d).dist(n) <= 0.33) && (threat(t).da(d).dist(n) > ...
                    0.05))
175                 threat(t).da(d).dist_class(n) = 1;
176             elseif (threat(t).da(d).dist(n) <= 0.66 && threat(t).da(d).dist(n) > ...
                    0.33)
177                 threat(t).da(d).dist_class(n) = 2;
178             elseif (threat(t).da(d).dist(n) < 1 && threat(t).da(d).dist(n) > 0.66)
179                 threat(t).da(d).dist_class(n) = 3;
```

```
180             else
181                 threat(t).da(d).dist_class(n) = 0;
182             end
183
184             if ((threat(t).da(d).passing(n) <= 0.33) && ...
                    (threat(t).da(d).passing(n) > 0.05))
185                 threat(t).da(d).passing_class(n) = 1;
186             elseif (threat(t).da(d).passing(n) <= 0.66 && ...
                    threat(t).da(d).passing(n) > 0.33)
187                 threat(t).da(d).passing_class(n) = 2;
188             elseif (threat(t).da(d).passing(n) < 1 && threat(t).da(d).passing(n) ...
                    > 0.66)
189                 threat(t).da(d).passing_class(n) = 3;
190             else
191                 threat(t).da(d).passing_class(n) = 0;
192             end
193
194             if ((threat(t).da(d).course(n) <= 0.33) && ...
                    (threat(t).da(d).course(n) > 0.05))
195                 threat(t).da(d).course_class(n) = 1;
196             elseif (threat(t).da(d).course(n) <= 0.66 && ...
                    threat(t).da(d).course(n) > 0.33)
197                 threat(t).da(d).course_class(n) = 2;
198             elseif (threat(t).da(d).course(n) < 1 && threat(t).da(d).course(n) > ...
                    0.66)
199                 threat(t).da(d).course_class(n) = 3;
200             else
201                 threat(t).da(d).course_class(n) = 0;
202             end
203
204             if ((threat(t).da(d).altitude(n) <= 0.33) && ...
                    (threat(t).da(d).altitude(n) > 0.05))
205                 threat(t).da(d).altitude_class(n) = 1;
206             elseif (threat(t).da(d).altitude(n) <= 0.66 && ...
                    threat(t).da(d).altitude(n) > 0.33)
207                 threat(t).da(d).altitude_class(n) = 2;
208             elseif (threat(t).da(d).altitude(n) < 1 && ...
                    threat(t).da(d).altitude(n) > 0.66)
209                 threat(t).da(d).altitude_class(n) = 3;
210             else
211                 threat(t).da(d).altitude_class(n) = 0;
212             end
213         end
214     end
215 end
216
217 flag = 0;
218 for t = 1:T
219     for d = 1:D
220         if flag == 0
221             class = [threat(t).da(d).dist_class; threat(t).da(d).passing_class; ...
                        threat(t).da(d).course_class; threat(t).da(d).altitude_class];
222             flag = 1;
223         else
224             class_temp = [threat(t).da(d).dist_class; ...
                        threat(t).da(d).passing_class; threat(t).da(d).course_class; ...
                        threat(t).da(d).altitude_class];
225             class = [class, class_temp];
226         end
227     end
228 end
```

```
229
230 class = class';
231 [class_unique, ia, ic] = unique(class, 'rows');
232 perm = permn([0.16 0.5 0.83], 3);
233 % columns: dist, passing, altitude, course
234 distance_permutation = 20e3 - 20e3.*perm(:, 1);
235
236 for p = 1:3^3
237     fused_tv(p) = (weightDist(distance_permutation(p))*perm(p,1) + ...
        weightPassing(distance_permutation(p))*perm(p,2) + ...
        weightAltitude(distance_permutation(p))*perm(p,3))/ ...
        (weightDist(distance_permutation(p)) + ...
        weightPassing(distance_permutation(p)) + ...
        weightAltitude(distance_permutation(p)));
238 end
239
240 L = 0.16;
241 M = 0.5;
242 H = 0.83;
243
244 xx = perm(:,1)';
245 yy = perm(:,2)';
246 zz = perm(:,3)';
247
248
249
250     function fusedTv = valueFuse(X1, X2, X3)
251         % X1 = distance TV
252         % X2 = passing dist TV
253         % X3 = altitude TV
254         fusedTv = (8719586279680049*X1^3)/18014398509481984 + ...
            (2358885918266645*X1^2*X2)/4503599627370496 + ...
            (3469810145740593*X1^2*X3)/2251799813685248 - ...
            (2772487191723963*X1^2)/2251799813685248 + ...
            (6476448612504021*X1*X2^2)/126765060022822940149670320537 6 - ...
            (5506627710609449*X1*X2*X3)/25353012004564588029934064107 52 - ...
            (6681694837972113*X1*X2)/9007199254740992 - ...
            (4725562727753645*X1*X3^2)/6338253001141147007483516026 88 - ...
            (4108499169635541*X1*X3)/2251799813685248 + ...
            (2826108733096399*X1)/2251799813685248 - ...
            (945702239520477*X2^3)/4503599627370496 - ...
            (4549002198991053*X2^2*X3)/2535301200456458802993406410752 + ...
            (2818192673770989*X2^2)/9007199254740992 - ...
            (5211824585907669*X2*X3^2)/6338253001141147007483516026 88 + ...
            (3603967500478037*X2*X3)/3169126500570573503741758013 44 + ...
            (4097853247384673*X2)/9007199254740992 - ...
            (4825177698429179*X3^3)/9007199254740992 + ...
            (1797378692664885*X3^2)/2251799813685248 + ...
            (7601046628211705*X3)/18014398509481984 - ...
            4313403330223459/36028797018963968;
255
256     end
257
258 end
```

C.4 Weapon Assignment

The `weaponAssignment.m` sub-routine contains an implementation of the WAM formulations described in §6.4, as well as the an implementation of the genetic algorithm described in §6.3. The WA sub-routine continues running until either all threats are destroyed, all DAs are destroyed or the threat tracks have ended. The outcomes of the scenario are also saved in this subroutine.

```
1  function [] = weaponAssignment(waModel)
2  % This function contains the routines for modelling the WA sub-routine.
3  %   The only provided input is the model type ('Static' or 'Dynamic)
4  % WaModel =  'Static', 'Dynamic'
5
6  global N T W threat WS numSSHPvolumes  assignments consTest threatData WSdata
7
8  numSSHPvolumes = 5;          % Number of domes required to represent the SSHP volumes
9  minPk4Engagement = 0.55;     % Min hit probability before WS is allowed to engage ...
       threat, between 50% - 70% as used by UK
10 numShotsFired = 0;           % Counter for number of shots fired from a WS
11 setupTime = 0;               % Counter for time between engagements
12 setFlag = 0;                 % Flag which gets set if a shot has been fired
13 NumEngagements = 1;          % Maximum times a WS may be assigned to a threat in ...
       a specific time stage
14 FL = 10;                     % Forecast length (intervals) used to forecast in ...
       the future
15 FW_minLeng = 4;              % Minimum FW length
16 H = 2;                       % Maximum number of WSs which may engage a threat ...
       during a scheduling horizon
17 AMMO = 3;                    % Starting number of ammunition for all WSs
18
19 % Replenish WS ammo
20 for w = 1:W
21     WS(w).ammo = AMMO;
22 end
23
24 %% WS range properties [m]
25 % CIWS
26 RminCIWS = 200;
27 RmaxCIWS = 4000;
28
29 % VSHORAD
30 RminVSHORAD = 900;
31 RmaxVSHORAD = 6000;
32
33 %SHORAD
34 RminSHORAD = 1200;
35 RmaxSHORAD = 20000;
36
37 % Set min and max ranges of WSs
38 for w = 1:W
39     tempWStype = WS(w).type;
40
41     switch tempWStype
42         case 'CIWS'
43             WS(w).minRange = RminCIWS;
44             WS(w).maxRange = RmaxCIWS;
45
46         case 'VSHORAD'
47             WS(w).minRange = RminVSHORAD;
48             WS(w).maxRange = RmaxVSHORAD;
```

```
49
50          case 'SHORAD'
51              WS(w).minRange = RminSHORAD;
52              WS(w).maxRange = RmaxSHORAD;
53      end
54  end
55
56  %% WA Problem Formulation
57  switch waModel % Class of WA model to solve
58
59      case 'Static'
60  %%%%%%%%%%%%%%%%%%%%%%%%% Constraint setup %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
61          % Constraint lower and upper bound of decision variables
62          lb = zeros(T*W,1);
63          ub = ones(T*W,1);
64          nvars = T*W; % Number of variables
65
66          % Preventing a WS from being assigned more than once during an time stage
67          avoid_sim_engagements = zeros(W, T*W);
68          for w = 1:W
69              for t = 1:T
70                  avoid_sim_engagements(w, W*t + (w - W)) = 1;
71              end
72          end
73          b_sim_enga = ones(W,1);
74
75          % Limite number of times a threat may be engaged by ws per time stage
76          prevent_overkill = zeros(T, T*W);
77          for t = 1:T
78              for w = 1:W
79                  prevent_overkill(t, (W*t - (W - 1)):((W*t - (W - 1))+(W-1))) = 1;
80              end
81          end
82          b_overkill = ones(T,1)*NumEngagements;
83
84          % Create matrix with vector positions of decision variables to ease ...
                handling of the reload, sshp and ammunition costraints
85          decisionVarPos = zeros(t,w);
86          pos = 1;
87          for t = 1:T
88              for w = 1:W
89                  decisionVarPos(t,w) = pos; % the position of a decision variable ...
                        can be obtained if the ws and threat num is known
90                  pos = pos + 1;
91              end
92          end
93
94          options = gaoptimset('StallGenLimit', 10, 'PlotFcns', {@gaplotbestf, ...
                @gaplotbestindiv}, 'Generations', 100); % Setting for optimisation ...
                function
95          optionsRetur = gaoptimset(@ga)
96          assignments = zeros(n,T*W);
97          consTest = zeros(n,T*W);
98
99          % Initialise variables
100         for t = 1:T
101             threat(t).hitStage = 0;     % Initialise the hitStage of each threat ...
                    to zer i.e it has not been destroyed
102             threat(t).engagedWs = 0;    % WS that destroyed threat
103         end
104
```

```
105         % Create a seperate variable for each WS or threat
106         numShotsFired = zeros(1,T); % Shots fired at specific threat
107         reloadFlag = zeros(1,W);
108         setupTime = zeros(1,W);
109
110
111
112   %%%%%%%%%%%%%%%%%%%%%%%% Optimisation routine %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
113
114         for n = 1:N % for each TEWA-cycle
115             % Change coefficients of objective function
116             V = [threat(1).system_tv_new(n), threat(2).system_tv_new(n), ...
                    threat(3).system_tv_new(n)];
117             Q = [threat(1).ws(1).hitProb(n), threat(2).ws(1).hitProb(n), ...
                    threat(3).ws(1).hitProb(n);
118                 threat(1).ws(2).hitProb(n), threat(2).ws(2).hitProb(n), ...
                        threat(3).ws(2).hitProb(n)];
119             Q = 1-Q;     % Survival probability of threats
120
121             % Recalculate reloading times
122             for w = 1:W
123                 if(reloadFlag(w) == 1) % Busy reloading
124                     setupTime(w) = setupTime(w) + 1; % Increment busy-time
125                     if(setupTime(w) >= WS(w).reloadTime) % check if enough time ...
                            has passed
126                         setupTime(w) = 0;
127                         reloadFlag(w) = 0;
128                     end
129                 end
130             end
131
132             % Dynamic constraints
133             conditions_const = zeros(1,T*W);
134             for t = 1:T
135                 for w = 1:W
136                     % Determine if all pre-conditions for engagement are met, if ...
                            not, prevent engagement
137                     if((threat(t).ws(w).hitProb(n) < minPk4Engagement) || ...
                            (threat(t).hitStage > 0) || (WS(w).ammo <= 0) || ...
                            (reloadFlag(w) == 1)) % All must be false for engagement
138                         conditions_const(decisionVarPos(t,w)) = 1; % ...
                                conditions_const*x <= 0
139                     else
140                         break
141                     end
142                 end
143             end
144
145             A = [avoid_sim_engagements; prevent_overkill; conditions_const];
146             b = [b_sim_enga; b_overkill; 0];
147
148             [assignments(n,:), fval, exitflag, output] = ga(@objectiveFunc, ...
                    nvars, A, b, [], [], lb, ub, [], 1:nvars, options) %xij i:WS ...
                    j:threat, order: x11 x21 x12 x22 x13 x23
149
150             % Determine new threat, WS conditions - hits, ammo etc
151             pos = find(assignments(n,:)); % Find index numbers of non-zero elements
152
153             for v = pos % for each of the non-zero decision variables
154                 [threatNum, WsNum] = find(v == decisionVarPos); % determine ...
                        threat number and WS number from the changed decision variable
```

```
155
156                 if((threatNum ~= 0) && (WsNum ~= 0)) % If a position was found ...
                        in above line
157                     WS(WsNum).ammo = WS(WsNum).ammo - 1;
158                     reloadFlag(WsNum) = 1; % Flag set - WS fired, needs to reload
159
160                     if(rand(1,1) < threat(threatNum).ws(WsNum).hitProb(n)) % ...
                            Successfull hit
161                         numShotsFired(threatNum) = numShotsFired(threatNum) + 1;
162                         threat(threatNum).engagedWS = WsNum;
163                         threat(threatNum).hitStage = n;
164                         threat(threatNum).numShotsFired = numShotsFired(threatNum);
165                     else % No hit
166                         numShotsFired(threatNum) = numShotsFired(threatNum) + 1;
167                     end
168
169                 end
170                 threatNum = 0;
171                 WsNum = 0;
172             end
173             pos = 0;
174
175             % Determine if all threats have been destroyed, if so, exit for loop
176             for t = 1:T
177                 killConfirmation(t) = threat(t).hitStage > 0; % 1: Threat t is ...
                        killed, 0: Still surviving
178             end
179
180             if killConfirmation(:) > 1
181                 break
182             end
183
184         end
185
186     case 'Dynamic'
187 %%%%%%%%%%%%%%%%%%%%%%%%%% Constraint setup %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
188         % Constraint lower and upper bound of decision variables
189         lb = zeros(T*W*FL,1);
190         ub = ones(T*W*FL,1);
191         nvarsDyn = T*W*FL; % Number of decision variables (assignments)
192
193         x_temp = zeros(1,nvarsDyn); % Vector with number of decision variable ...
                elements - to be used for constraint setup
194         A = x_temp;
195
196         assignmentsDyn = zeros(N, nvarsDyn);
197         populationSize = ceil(1.1*nvarsDyn);
198
199         % Construct decision variable matrix: The position of a decision ...
                variable can be obtained if the ws and threat num is known - numbers ...
                correspond to the order of decision variables in the decision vector ...
                used in the optimisation toolbox.
200         decisionVarPosDyn = zeros(T,W,FL); % Initialise matrix
201         pos = 1;
202         for ww = 1:W
203             for tt = 1:T
204                 for ff = 1:FL
205                     decisionVarPosDyn(tt,ww,ff) = pos;
206                     pos = pos + 1;
207                 end
208             end
```

```
209         end
210         assignin('base', 'DecVar', decisionVarPosDyn()) % Debuging
211
212         %% Create stage utility function - must be monotonically increasing.
213         x_su  = [1 FL/2 FL];
214         y_su  = [1 0.2 0.1];
215         fStageUtil = fit(x_su', y_su', 'exp1');
216         stageUtilWeights = fStageUtil(1:FL)./sum(fStageUtil(1:FL));
217
218         % Limit numer of times a WS may engage during a scheduling horizon
219         for w_c1 = 1:W
220             for t_c1 = 1:T
221                 for f_c1 = 1:FL
222                     x_temp(decisionVarPosDyn(t_c1, w_c1, f_c1)) = 1;
223                 end
224             end
225             A = [A; x_temp];
226             x_temp = zeros(1, nvarsDyn); % reset x_temp
227         end
228
229         A(1,:) = []; % remove dummy first row of constraint - only after first ...
                constraint
230         b_simEng = ones(W,1);
231
232         % Limit number of WSs assigned to a specific threat during a time horizon
233         for t_c1 = 1:T
234             for w_c1 = 1:W
235                 for f_c1 = 1:FL
236                     x_temp(decisionVarPosDyn(t_c1, w_c1, f_c1)) = 1;
237                 end
238             end
239             A = [A; x_temp];
240             x_temp = zeros(1, nvarsDyn);
241         end
242
243         b_overKill = ones(T,1)*H;
244         b_perm = [b_simEng; b_overKill];
245         A_perm = A;
246
247         % Variable for WS fire orders
248         for w = 1:W
249             WSfireOrder(w).timeStage = 0;
250             WSfireOrder(w).threatNum = 0;
251             WSfireOrder(w).assignmentStatus = 0;
252             WSfireOrder(w).reloadTimer = 0;
253         end
254
255         numShotsFired(1:T) = 0;
256         killConfirmation(1:T) = 0;
257
258         % Initialise optimisation variables
259         for t = 1:T
260             threat(t).hitStage = 0;
261             threat(t).engagedWS = 0;
262             threat(t).numShotsFired = 0;
263         end
264
265
266 %%%%%%%%%%%%%%%%%%%%%%%%% Optimisation routine %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
267
268         % Options for genetic algorithm
```

```
269         eliteCount = ceil(0.10*populationSize);
270         crossoverPer = 0.80;
271         options = gaoptimset('StallGenLimit', 8, 'InitialPenalty', 150, ...
                'PenaltyFactor', 100,  'Generations', 200, 'PopulationSize', ...
                populationSize, 'EliteCount', eliteCount, 'CrossoverFraction', ...
                crossoverPer, 'PlotFcns', {@gaplotbestf, @gaplotbestindiv}); % ...
                Settings for ga optimisation function
272
273         for n = 2:N % Only start from second time interval, because first time ...
                interval has no previous data for vector calculation
274
275             % Change coefficients of objective function to represent current ...
                    time stage
276             for t = 1:T
277                 V(t) = threat(t).system_tv_new(n);
278             end
279
280             % Decrement all WSs relaoding timers
281             for w = 1:W
282                 if(WSfireOrder(w).reloadTimer > 0)
283                     WSfireOrder(w).reloadTimer =  WSfireOrder(w).reloadTimer - ...
                            1; % Decrement by 1 TEWA cycle
284                 end
285             end
286
287             % Execute all active fire orders if possible
288             for w = 1:W
289                 if((WSfireOrder(w).assignmentStatus == 1) && ...
                        (WSfireOrder(w).timeStage <= n) && (WS(w).ammo > 0) && ...
                        (WSfireOrder(w).reloadTimer <= 0)) % If a WS has an active ...
                        FO then this is his time to shine
290
291                     if(threat(WSfireOrder(w).threatNum).hitStage > 0) % ...
                            Determine if threat has been destroyed already but still ...
                            has an active fire order assigned to it. If so, cancel FO
292                         WSfireOrder(w).assignmentStatus = 0;
293                     else
294                         WS(w).ammo = WS(w).ammo - 1; % Fire round - decrement ammo
295                         WSfireOrder(w).reloadTimer = WS(w).reloadTime; % ...
                                Reloading timer set - WS fired, needs to reload
296                         WSfireOrder(w).assignmentStatus = 0; % WS available for ...
                                reassignment - fire order has been executed
297
298                         % Log engagement history
299                         WSdata(w).engagements(iter, ...
                                numShotsFired(WSfireOrder(w).threatNum) + 1) =  ...
                                WSfireOrder(w).threatNum; % Log threats engaged by ...
                                specific WS
300                         threatData(WSfireOrder(w).threatNum).engagementHistory( ...
                                threat(WSfireOrder(w).threatNum).numEngaged, 1) = w; ...
                                % Time stage during which a WS engaged a threat, ...
                                successfully or not
301                         threatData(WSfireOrder(w).threatNum).engagementHistory( ...
                                threat(WSfireOrder(w).threatNum).numEngaged, 2) = n;
302
303                         % Determine outcome
304                         currentPos = ...
                                [threat(WSfireOrder(w).threatNum).TEpoints(1,n), ...
                                threat(WSfireOrder(w).threatNum).TEpoints(2,n), ...
                                threat(WSfireOrder(w).threatNum).TEpoints(3,n); ...
                                WS(w).x, WS(w).y, WS(w).z];
```

```
305                         threat2WSdist_temp = pdist(currentPos, 'euclidean');
306                         currentSSHP = killProb(w, threat2WSdist_temp);
307
308                         if(rand(1,1) < currentSSHP) % successfull hit
309                             numShotsFired(WSfireOrder(w).threatNum) = ...
                                    numShotsFired(WSfireOrder(w).threatNum) + 1;
310                             threat(WSfireOrder(w).threatNum).engagedWS = w;
311                             threat(WSfireOrder(w).threatNum).hitStage = n;
312                             threat(WSfireOrder(w).threatNum).forecastTimeIndex = ...
                                    WSfireOrder(w).forecastTime;
313                             threat(WSfireOrder(w).threatNum).numShotsFired = ...
                                    numShotsFired(WSfireOrder(w).threatNum);
314                         else % No hit
315                             numShotsFired(WSfireOrder(w).threatNum) = ...
                                    numShotsFired(WSfireOrder(w).threatNum) + 1;
316                         end
317
318                     end
319                 end
320             end
321
322             % Determine if all threats have been destroyed, if so, exit for loop
323             for t = 1:T
324                 killConfirmation(t) = threat(t).hitStage > 0; % 1: Threat t is ...
                        killed, 0: Still surviving
325             end
326
327             if killConfirmation(:) == 1 % If all threats have been destroyed
328                 break
329             end
330
331             % Predict changes in SSHP values for each forecast time stage
332             for t = 1:T
333                 currentSpeed = threat(t).speed;
334                 currentPosVec = ...
                        [threat(t).TEpoints(1,n)-threat(t).TEpoints(1,n-1), ...
                        threat(t).TEpoints(2,n)-threat(t).TEpoints(2,n-1), ...
                        threat(t).TEpoints(3,n)-threat(t).TEpoints(3,n-1)];
335                 currentPosUnitVec = currentPosVec./sqrt(sum(currentPosVec.^2));
336                 velocityVec = currentPosUnitVec.*currentSpeed;
337                 for w = 1:W
338                     for f = 1:FL %+1
339                         currentPos2WS = [threat(t).TEpoints(1:3,n)' + ...
                                velocityVec.*(f-1); WS(w).x, WS(w).y, WS(w).z];
340                         threat2WSdist = pdist(currentPos2WS, 'euclidean');
341                         threat(t).ws(w).forecastSSHP(n,f) = killProb(w, ...
                                threat2WSdist);
342                         Qdyn(t,w,f) = threat(t).ws(w).forecastSSHP(n,f);
343                     end
344                 end
345             end
346             Qdyn = 1-Qdyn;    % Survival probability of threats
347
348             A_conditions = zeros(1, nvarsDyn);
349             for t = 1:T
350                 for w = 1:W
351                     for f = 1:FL
352                         % Determine if all pre-conditions for engagement are ...
                                met, if not, prevent engagement
353                         if((threat(t).ws(w).forecastSSHP(n,f) <= ...
                                minPk4Engagement) || (WSfireOrder(w).reloadTimer > ...
```

```
                            0) || (WSfireOrder(w).assignmentStatus == 1) || ...
                            (threat(t).hitStage > 0) || (WS(w).ammo <= 0))
354                         A_conditions(decisionVarPosDyn(t,w,f)) = 1; % ...
                                conditions_const*x <= 0
355                     end
356                 end
357             end
358         end
359         b_temp = 0;
360         A  = [A_perm; A_conditions];
361         b = [b_perm; b_temp];
362
363         % Execute genetic algorithm
364         [assignmentsDyn(n,:), fval, exitflag, output] = ...
                ga(@objectiveFuncDyn, nvarsDyn, A, b, [], [], lb, ub, [], ...
                1:nvarsDyn, options);
365
366         % Generate fire orders from results returned by genetic algorithm
367         for i = 1:nvarsDyn
368             if(assignmentsDyn(n,i) == 1)
369                 [threatNum, WsNum, TiNum] = ind2sub(size(decisionVarPosDyn), ...
                        find(i == decisionVarPosDyn));
370
371                 if((WSfireOrder(WsNum).assignmentStatus ~= 1) && ...
                        (threat(threatNum).hitStage == 0) && ...
                        (threat(threatNum).ws(WsNum).forecastSSHP(n,TiNum) >= ...
                        minPk4Engagement)) % Determine if the WS has already ...
                        been scheduled and the threat has not been destroyed ...
                        before sending fire order
372                     WSfireOrder(WsNum).threatNum = threatNum;
373                     WSfireOrder(WsNum).timeStage = TiNum + n - 1; % Time ...
                            stage during which WS must engage (use current n to ...
                            determine)
374                     WSfireOrder(WsNum).assignmentStatus = 1; % Fire order ...
                            Active - must be active to engage threats 1: WS has ...
                            been scheduled to engage 0:Otherwise
375                     WSfireOrder(WsNum).forecastTime = TiNum; % For debugging ...
                            purposes
376                 end
377
378             end
379         end
380         threatNum = 0;
381         WsNum = 0;
382         TiNum = 0;
383     end
384
385     % Determine state (survived or destroyed) of DAs
386     for t = 1:T
387         if(threat(t).WRL <= threat(t).hitStage)
388             threat(t).success = 1;
389         else
390             threat(t).success = 0;
391         end
392     end
393
394     % Log data
395     for t = 1:T
396         threatData(t).engagedWS(iter) = threat(t).engagedWS;
397         threatData(t).hitStage(iter) = threat(t).hitStage;
398         threatData(t).success(iter) = threat(t).success;
```

```
399             threatData(t).numShotsFired(iter) = threat(t).numShotsFired;
400             threatData(t).forecastTimeIndex(iter) = threat(t).forecastTimeIndex;
401         end
402
403         for w = 1:W
404             WSdata(w).numberFired(iter) = AMMO - WS(w).ammo;     % Number of ...
                    rounds fired
405         end
406
407     case 'Test'
408         % Test routine employing traces for verification purposes
409         n = 120;
410         for t = 1:T
411             for w = 1:W
412                 currentPos = [threat(t).TEpoints(1,n), threat(t).TEpoints(2,n), ...
                        threat(t).TEpoints(3,n); WS(w).x, WS(w).y, WS(w).z]; % 2 ...
                        points for TE
413                 threat2WSdist_temp = pdist(currentPos, 'euclidean');
414                 currentSSHP = killProb(w, threat2WSdist_temp);
415                 temp = ['WS ', num2str(w), ' to Threat ', num2str(t), '; SSHP = ...
                        ', num2str(currentSSHP)];
416                 disp(temp)
417             end
418         end
419 end
420
421     function y = objectiveFunc(x)
422         % y = V(1)*(Q(1,1)^x(1) + Q(2,1)^x(2)) + V(2)*(Q(1,2)^x(3) + ...
            Q(2,2)^x(4)) + V(3)*(Q(1,3)^x(5) + Q(2,3)^x(6));
423
424         y = 0;
425         for t = 1:T
426             Qtemp = 0;
427             for w = 1:W
428                 Qtemp = Qtemp + Q(w,t)^x(decisionVarPos(t,w));
429             end
430             y = y + V(t)*Qtemp;
431         end
432
433     end
434
435     function y = objectiveFuncDyn(x)
436
437         y = 0;
438         for t = 1:T
439             Qtemp = 0;
440             for w = 1:W
441                 for f = 1:FL
442                     Qtemp = Qtemp + stageUtilWeights(f).* ...
                            Qdyn(t,w,f)^x(decisionVarPosDyn(t,w,f));
443                 end
444             end
445             y = y + V(t)*Qtemp;
446         end
447
448     end
449
450 end
```

C.5 Sub-functions

In order to determine the outcome of an engagement and for WA forecasting purposes, it is required to be able to determine the SSHP of a WS and threat combination if the distance to the threat is known. The `killProb.m` sub-function is responsible for returning the SSHP for the WS-threat pair under consideration.

```
function [ sshp ] = killProb(wsNo, distance2threat)
% Function that returns the SSHP value for the distance that a threat is from a ...
    specific WS
% wsNo: Number of WS under consideration, distance2threat in metres

switch WS(wsNo).type % SSHP value is dependent on the type of WS

    case 'CIWS'

        % Test if threat is within range of WS
        if ((distance2threat > RmaxCIWS) || (distance2threat < RminCIWS))
            sshp = 0;
        else % Calculate SSHP value as described in thesis
            z_temp = 1 - exp(-110/(0.05*distance2threat)^2);
            sshp = 1 - exp(30*log(1-z_temp));
        end

    case 'VSHORAD'
        if ((distance2threat > RmaxVSHORAD) || (distance2threat < RminVSHORAD))
            sshp = 0;
        else
            x_vshorad = [RminVSHORAD, ((RmaxVSHORAD - RminVSHORAD)*3/5 + ...
                RminVSHORAD), RmaxVSHORAD, ((RmaxVSHORAD - RminVSHORAD)*1/5 + ...
                RmaxVSHORAD)];
            y_vshorad = [0.5 0.85 0.5 0];
            pVSHORAD = polyfit(x_vshorad, y_vshorad, 3);
            sshp = polyval(pVSHORAD, distance2threat);
        end

    case 'SHORAD'
        if ((distance2threat > RmaxSHORAD) || (distance2threat < RminSHORAD))
            sshp = 0;
        else
            x_shorad = [RminSHORAD, ((RmaxSHORAD - RminSHORAD)*3/5 + ...
                RminSHORAD), RmaxSHORAD, ((RmaxSHORAD - RminSHORAD)*1/5 + ...
                RmaxSHORAD)];
            y_shorad = [0.5 0.85 0.5 0];
            pSHORAD = polyfit(x_shorad, y_shorad, 3);
            sshp = polyval(pSHORAD, distance2threat);
        end
end

end
```

This last two sub-functions — `drawSSHPvolume.m` and `drawDome.m` — are used in order to draw the different SSHP volumes and the visualisation of a dome which represents the AOR.

```
function [h1, h2, h3, h4, h5] = drawSSHPvolume(xc, yc, zc, Rmax)
% Draw a dome (half sphere) with centre specified by xc, yc, zc and radius ...
    specified by Rmax. All values in metres
% Dome is mainly used to display the defended airspace and aid verification. ...
    There is a separate function for calculating the SSHP values.

global numSSHPvolumes

numSSHPvolumes = 5; % Number of domes required to represent the SSHP volume
spacingSSHP = Rmax/numSSHPvolumes; % Distance between the different spheres

[x,y,z] = sphere;       % Makes a 21-by-21 point unit sphere
x = x(11:end,:);        % Keep top 11 x points
y = y(11:end,:);        % Keep top 11 y points
z = z(11:end,:);        % Keep top 11 z points

% Pre-allocating size for faster execution
rx = zeros(1,numSSHPvolumes);
ry = zeros(1,numSSHPvolumes);
rz = zeros(1,numSSHPvolumes);

% Define Rx, Ry and Rz for each sphere
for n = 1:numSSHPvolumes
    rx(n) = Rmax - spacingSSHP*(n-1); % Calculate the radius of each dome - ...
        sphere 1 is the outer most dome
    ry(n) = Rmax - spacingSSHP*(n-1);
    rz(n) = Rmax - spacingSSHP*(n-1);
end

% Scale x, y and z coordinates and move center to desired position
for n = 1:numSSHPvolumes
    xNew(:,:,n) = x.*rx(n) + xc;
    yNew(:,:,n) = y.*ry(n) + yc;
    zNew(:,:,n) = z.*rz(n) + zc;
end

% Draw Different SSHP volumes
% Plot the surface, multiplying unit coordinates with radii
h1 = surfl(xNew(:,:,1),yNew(:,:,1),zNew(:,:,1));
h2 = surfl(xNew(:,:,2),yNew(:,:,2),zNew(:,:,2));
h3 = surfl(xNew(:,:,3),yNew(:,:,3),zNew(:,:,3));
h4 = surfl(xNew(:,:,4),yNew(:,:,4),zNew(:,:,4));
h5 = surfl(xNew(:,:,5),yNew(:,:,5),zNew(:,:,5));
%transparency -  needs to be set after all domes have been drawn
```

```
function [h] = drawDome(xc, yc, zc, R)
%Draw a dome (half sphere) with centre specified by xc, yc, zc and radius by
%R.
%   Dome is mainly used to display the defended airspace. There is a separate ...
    function for displaying single shot hit probability spheres.

[x,y,z] = sphere;       % Makes a 21-by-21 point unit sphere
x = x(11:end,:);        % Keep top 11 x points
y = y(11:end,:);        % Keep top 11 y points
z = z(11:end,:);        % Keep top 11 z points
rx = R;ry = R;rz = R;  % Define rx, ry, rz
```

```
11
12 % Scale x, y and z coordinates and move center to desired position
13 x = x*rx + xc;
14 y = y*ry + yc;
15 z = z*rz + zc;
16
17 h = surfl(x,y,z);  % Plot the surface, multiplying unit coordinates with radii
18 %shading(gca, 'interp')
19 %set(h, 'FaceColor', [0.5 0 0], 'FaceAlpha', 0.2); % change color and
20 %transparency -  needs to be set after all domes have been drawn
21 %axis equal;            % Make the scaling on the x, y, and z axes equal
22
23 end
```

APPENDIX D

Content of the Accompanying Compact Disc

The scope of the content included with the CD is clarified in this appendix. The *bold-font* headings refer to the names of the folders contained in the root directory. The content of each of these folders is described below.

Deliverables. Reports and articles are contained within this folder. This includes the ARMSCOR progress reports, departmental progress reports as well as the two ORSSA articles.

Presentations. Includes all the ORSSA, EURO, LEDGER and Muses presentations that were presented throughout the duration of this research project.

References. All references that were consulted during the completion of the project are included in this folder. The references are first organised according to articles, theses, videos, patents, presentations and books, after which the references for the respective media types are categorised according to their domains — TE, WA, whole TEWA system, simulation development, South African GBAD system, decision making, HMI, genetic algorithm, system evaluation, missile defence, NCW and general air threats.

Programming. Includes the source code for the MATLAB® R2014a simulation with all the required plugins and data sets. Several backups are also included with their respective dates so as to clarify the progression of the project.

Pictures. All pictures used within the thesis may be found in this folder. INKSCAPE v0.48 was used to draw many of visual illustrations, this software would therefore be required in order to edit the illustrations. The source code for the Tikz-PGF plots are also included.

Videos. Some videos that may be used to understand some basic principles related to the functioning of a TEWA system are also included.

Thesis. The LaTeX source code of this thesis is contained in this folder.

In the future, if additional information or explanation regarding the author's work is required, the author may be contacted at `louw.truter@gmail.com`. The author will try his best to assist future work where possible. This topic has considerable potential for further work. The author wishes all the best to future students of the Stellenbosch TEWA Centre of Expertise.