The development of an Integrated Effectiveness Model for Aerial Targets

by

Leo D. Tomé

Thesis presented in partial fulfilment of the requirements for the degree Masters of Engineering Science at Stellenbosch University, South Africa

Supervisor: Dr GJF Smit
March 2007
Declaration

I, the undersigned, hereby declare that the work contained in this thesis is my own original work and that I have not previously in its entirety or in part submitted it at any university for a degree.

Signature: ______________________  Date: ______________________
Abstract

During the design or acquisition of missile systems the effectiveness of the system needs to be evaluated. Often actual testing is not possible and therefore mathematical models need to be constructed and solved with the aid of software. The current simulation model is investigated, verified, and a mathematical model to aid in the design of the detonic payload, developed. The problem is confined to the end-game scenario with the developed simulation model focusing on the last milliseconds before warhead detonation. The model, that makes use of the ray-tracing methodology, models the warhead explosion in the vicinity of a target and calculates the probability of kill for the specific warhead design against the target. Using the data generated by the simulation model, the warhead designer can make the necessary design changes to improve the design. A heuristic method was developed and is discussed which assists in this design process. There is, however, a large population of possible designs. Meta-heuristic methods may be employed in reducing this population and to help confine the manual search to a considerably smaller search area. A fuze detection model as well as the capability to generate truly random intercept scenarios was developed as to enable employment of meta-heuristic search methods. The simulation model, as well as design optimising technology, has successfully been incorporated into a Windows based software package known as EVA (The Effectiveness and Vulnerability Analyser).
Opsomming

Die doeltreffendheid van ’n missielstelsel moet geëvalueer word gedurende die ontwerp of aanskaffing van sodanige stelsel. Werklike toetse is meestal nie moontlik nie en gevolglik moet wiskundige modelle, wat met behulp van sagteware opgelos word, ontwikkeld word. Die huidige simulasiemodel word ondersoek en geverifieer, en ’n meer volledige en geïntegreerde model word ontwikkeld vir die ontwerp van vooraf gefragmenteerde plofkoppe. Die probleemarea word beperk tot die eind scenario, die simulasiemodel wat ontwikkeld is fokus dus op die finale millisekondes voor die plofkop gedetoneer word. Die model, wat gebruik maak van die straal volging (ray-tracing) metodologie, modelleer die ontploffing van die plofkop in die direkte omgewing van die teiken en bereken die waarskynlikheid dat die teiken geneutraliseer sal word deur die spesifieke plofkop. Die plofkopontwerper kan, deur gebruik te maak van die informasie wat gegenerer is deur die simulasiemodel, die nodige veranderinge aan die ontwerp aanbring ter verbetering daarvan. ’n Heuristiese metode, wat die plofkop ontwerpsproses ondersteun, was ontwikkeld, en word bespreek, as deel van die studie. Die populieruimte van moontlike plofkopontwerpe is egter groot. Meta-heuristiese soekalgoritme het die potensiaal om die ruimte te verklein. Die buis opsporingsmodel, sowel as die funksionaliteit om ewekansige intersep scenarios te genereer was ontwikkeld om die implementeerbaarheid van meta-heuristiese soek algoritmes moontlik te maak. Die simulasiemodel, sowel as optimeringsfunksionaliteite is suksesvol in ’n Windows gebaseerde sagtewarepakket, bekend as EVA (The Effectiveness and Vulnerability Analyser), geïnkorporeer.
Acknowledgements

I wish to express my sincere gratitude to the following people:
- My supervisor, Dr GJF Smit, for his patience, mentoring, and guidance;
- The NRF for financial assistance;
- Martina Wium for moral support and encouragement;
- My parents for their support, encouragement, and the sacrifices made to afford me the opportunities they never had.
# Table of Contents

List of Figures ix

List of Tables xiii

List of Symbols xv

1 Effectiveness and Vulnerability 1

1.1 Introduction ......................................................... 1

1.2 Methodologies ....................................................... 4

1.2.1 Phenomenological Model ........................................ 5

1.2.2 Encounter Models ................................................ 5

1.2.2.1 Vulnerable Area ............................................. 7

1.2.2.2 Ray-tracing .................................................... 7

1.2.2.3 Comparison of Methodologies ............................... 8

1.2.3 International capabilities and models .......................... 8

1.2.3.1 United States of America .................................. 9

1.2.3.2 Europe ........................................................ 10

1.2.3.3 South Africa .................................................. 10

2 EVA 15

2.1 Introduction ......................................................... 15

2.2 Overview .......................................................... 15

2.3 Target Model ........................................................ 16

2.4 Warhead Model ....................................................... 18

2.5 Intercept Path Model ............................................... 18

2.6 Fuzing System ....................................................... 19

2.7 Effectiveness Model ................................................ 20

2.8 Heuristic Models ................................................... 22
TABLE OF CONTENTS

2.9 Future Developments as a result of this study ........................................... 22

3 The Target Model ......................................................................................... 23

3.1 Introduction ............................................................................................... 23

3.2 Target analysis ......................................................................................... 23

3.2.1 Kill levels ............................................................................................ 23

3.2.2 Technical and Functional description of the target ............................. 24

3.2.3 Critical component analysis ................................................................. 25

3.2.3.1 Step 1: Essential functions .......................................................... 26

3.2.3.2 Step 2: System Essential functions Relationships ........................ 27

3.2.3.3 Step 3: Failure mode and Effect Analysis (FMEA) ....................... 27

3.2.3.4 Step 4: Damage Modes and Effect Analysis (DMEA) ................. 29

3.2.3.5 Step 5: Kill trees and Kill Expressions ......................................... 31

3.3 The Target Model ..................................................................................... 31

3.3.1 Target file format conversions ............................................................. 33

3.3.1.1 The conversion for Upshot ........................................................... 33

3.3.1.2 An internal point in the polyhedron ......................................... 33

3.3.1.3 The maximum distance from the internal point to a corner of the respective polyhedron ......................................................... 34

3.3.1.4 The normal vectors of each surface ........................................... 34

3.3.1.5 The conversion to view the target graphically in EVA ............... 36

3.3.1.6 Determine all the points of a surface ....................................... 37

3.3.1.7 Ordering the points of a surface ............................................. 37

3.3.2 EVA .................................................................................................... 39

3.3.2.1 Create a new target ................................................................. 39

3.3.2.2 Delete an existing target .......................................................... 40

3.3.2.3 View a target graphically ........................................................ 40

3.3.2.4 View Target Input file ............................................................ 41

3.3.2.5 Improvements on SWEAT ...................................................... 41

4 Intercept Paths ........................................................................................... 45

4.1 Introduction .............................................................................................. 45

4.2 The Propagator ......................................................................................... 45

4.2.1 Platforms ............................................................................................. 45

4.2.2 Propagators ......................................................................................... 46

4.2.3 Guidance Systems ................................................................................ 47
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.3</td>
<td>Flight paths</td>
<td>52</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Intercept terminology</td>
<td>54</td>
</tr>
<tr>
<td>4.4</td>
<td>Generating random intercept scenarios</td>
<td>56</td>
</tr>
<tr>
<td>4.4.1</td>
<td>Simulating a Continuous Random Variable</td>
<td>57</td>
</tr>
<tr>
<td>4.4.1.1</td>
<td>Techniques for simulating a Continuous random variables</td>
<td>57</td>
</tr>
<tr>
<td>4.4.1.2</td>
<td>Beta Distribution</td>
<td>58</td>
</tr>
<tr>
<td>4.4.1.3</td>
<td>Discrete Distribution</td>
<td>59</td>
</tr>
<tr>
<td>4.4.1.4</td>
<td>Exponential Distribution</td>
<td>59</td>
</tr>
<tr>
<td>4.4.1.5</td>
<td>Gamma Distribution</td>
<td>59</td>
</tr>
<tr>
<td>4.4.1.6</td>
<td>Normal Distribution</td>
<td>60</td>
</tr>
<tr>
<td>4.4.1.7</td>
<td>Poisson Distribution</td>
<td>60</td>
</tr>
<tr>
<td>4.4.1.8</td>
<td>Uniform Distribution</td>
<td>61</td>
</tr>
<tr>
<td>4.4.2</td>
<td>The intercept path file</td>
<td>61</td>
</tr>
<tr>
<td>4.4.2.1</td>
<td>Miss Distance</td>
<td>62</td>
</tr>
<tr>
<td>4.4.2.2</td>
<td>Aimpoint in target coordinates</td>
<td>62</td>
</tr>
<tr>
<td>4.4.2.3</td>
<td>Drag</td>
<td>62</td>
</tr>
<tr>
<td>4.4.2.4</td>
<td>Missile Speed</td>
<td>62</td>
</tr>
<tr>
<td>4.4.2.5</td>
<td>Missile travel direction</td>
<td>63</td>
</tr>
<tr>
<td>4.4.2.6</td>
<td>The direction of the missile system</td>
<td>63</td>
</tr>
<tr>
<td>4.4.2.7</td>
<td>Coordinates of burst points</td>
<td>64</td>
</tr>
<tr>
<td>4.4.2.8</td>
<td>Target information</td>
<td>64</td>
</tr>
<tr>
<td>4.5</td>
<td>EVA</td>
<td>64</td>
</tr>
<tr>
<td>4.5.1</td>
<td>Endgame Scenario Distributions</td>
<td>65</td>
</tr>
<tr>
<td>4.5.2</td>
<td>View Path File</td>
<td>65</td>
</tr>
<tr>
<td>4.5.3</td>
<td>View Scenario File</td>
<td>66</td>
</tr>
<tr>
<td>4.5.4</td>
<td>Deleting files associated with paths</td>
<td>66</td>
</tr>
<tr>
<td>4.5.5</td>
<td>Improvements on SWEAT</td>
<td>66</td>
</tr>
<tr>
<td>5</td>
<td>Warhead</td>
<td>69</td>
</tr>
<tr>
<td>5.1</td>
<td>Introduction</td>
<td>69</td>
</tr>
<tr>
<td>5.2</td>
<td>Damage Mechanisms</td>
<td>69</td>
</tr>
<tr>
<td>5.3</td>
<td>Fragment Acceleration</td>
<td>72</td>
</tr>
<tr>
<td>5.3.1</td>
<td>Introduction</td>
<td>72</td>
</tr>
<tr>
<td>5.3.2</td>
<td>Inclusion of end effects</td>
<td>72</td>
</tr>
<tr>
<td>5.4</td>
<td>Fragment Retardation</td>
<td>74</td>
</tr>
<tr>
<td>Section</td>
<td>Subsection</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>------------</td>
<td>------</td>
</tr>
<tr>
<td>5.5</td>
<td>5.5.1</td>
<td>75</td>
</tr>
<tr>
<td></td>
<td>5.5.2</td>
<td>78</td>
</tr>
<tr>
<td></td>
<td>5.5.3</td>
<td>79</td>
</tr>
<tr>
<td></td>
<td>5.5.4</td>
<td>82</td>
</tr>
<tr>
<td></td>
<td>5.5.5</td>
<td>82</td>
</tr>
<tr>
<td>6</td>
<td>6.1</td>
<td>85</td>
</tr>
<tr>
<td></td>
<td>6.2</td>
<td>85</td>
</tr>
<tr>
<td></td>
<td>6.2.1</td>
<td>86</td>
</tr>
<tr>
<td></td>
<td>6.2.2</td>
<td>86</td>
</tr>
<tr>
<td></td>
<td>6.3</td>
<td>88</td>
</tr>
<tr>
<td></td>
<td>6.3.1</td>
<td>88</td>
</tr>
<tr>
<td></td>
<td>6.3.2</td>
<td>90</td>
</tr>
<tr>
<td></td>
<td>6.3.2.1</td>
<td>90</td>
</tr>
<tr>
<td></td>
<td>6.3.2.2</td>
<td>91</td>
</tr>
<tr>
<td></td>
<td>6.3.2.3</td>
<td>92</td>
</tr>
<tr>
<td></td>
<td>6.3.3</td>
<td>92</td>
</tr>
<tr>
<td></td>
<td>6.4</td>
<td>97</td>
</tr>
<tr>
<td></td>
<td>6.4.1</td>
<td>98</td>
</tr>
<tr>
<td></td>
<td>6.4.2</td>
<td>99</td>
</tr>
<tr>
<td></td>
<td>6.4.3</td>
<td>99</td>
</tr>
<tr>
<td></td>
<td>6.4.4</td>
<td>99</td>
</tr>
<tr>
<td>7</td>
<td>7.1</td>
<td>101</td>
</tr>
<tr>
<td></td>
<td>7.2</td>
<td>101</td>
</tr>
<tr>
<td></td>
<td>7.2.1</td>
<td>103</td>
</tr>
<tr>
<td></td>
<td>7.2.1.1</td>
<td>104</td>
</tr>
<tr>
<td></td>
<td>7.2.1.2</td>
<td>104</td>
</tr>
<tr>
<td></td>
<td>7.2.1.3</td>
<td>105</td>
</tr>
<tr>
<td></td>
<td>7.2.1.4</td>
<td>107</td>
</tr>
<tr>
<td></td>
<td>7.2.1.5</td>
<td>108</td>
</tr>
<tr>
<td></td>
<td>7.2.2</td>
<td>108</td>
</tr>
<tr>
<td></td>
<td>7.2.3</td>
<td>109</td>
</tr>
</tbody>
</table>
# Table of Contents

## 7.2.3.1 Initiate Fragment

## 7.2.3.2 Fragment Trajectory

## 7.2.3.3 Check which components are hit

## 7.2.4 Kill probabilities for each of the event that occur given a hit

### 7.2.4.1 Kill Category 1

### 7.2.4.2 Kill Category 2

### 7.2.4.3 Kill Category 3

### 7.2.4.4 Kill Category 4

## 7.2.5 The loop over all Systems

## 7.2.6 Calculate the probability of the various events

## 7.2.7 Calculate the mean probability for each of the mission criteria over all the intercept scenarios

## 7.2.8 The results

### 7.2.8.1 Mission criteria for each missile

### 7.2.8.2 Summed up mission criteria results

## 7.3 EVA

### 7.3.1 Standard Simulation Setup

### 7.3.2 Running the standard simulation

### 7.3.3 View Graph

### 7.3.4 Deleting a simulation run

### 7.3.5 View Upshot Output File (text)

### 7.3.6 Improvements on SWEAT

## 8 Optimisation

### 8.1 Introduction

### 8.2 The warhead search Algorithm

### 8.3 EVA

#### 8.3.1 Variable Simulation

#### 8.3.2 Variable simulation warhead generation

#### 8.3.3 Variable simulation run

#### 8.3.4 Variable Simulation Information

#### 8.3.5 Improvements on SWEAT

## 9 Conclusion

### 9.1 Introduction

### 9.2 Summary of the study
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.3 Contributions made</td>
<td>150</td>
</tr>
<tr>
<td>9.4 Future developments as a result of this study</td>
<td>152</td>
</tr>
<tr>
<td>References</td>
<td>155</td>
</tr>
<tr>
<td><strong>A Target data files</strong></td>
<td>157</td>
</tr>
<tr>
<td>A.1 Geometric Description File</td>
<td>157</td>
</tr>
<tr>
<td>A.1.1 File Format</td>
<td>157</td>
</tr>
<tr>
<td>A.1.2 Example of Geometric Description File</td>
<td>158</td>
</tr>
<tr>
<td>A.2 Critical Components and Subsystems File</td>
<td>161</td>
</tr>
<tr>
<td>A.2.1 File Format</td>
<td>161</td>
</tr>
<tr>
<td>A.2.2 Example of Critical File</td>
<td>161</td>
</tr>
<tr>
<td>A.3 Target Input File for Effect</td>
<td>163</td>
</tr>
<tr>
<td>A.3.1 File Format</td>
<td>163</td>
</tr>
<tr>
<td>A.3.2 Example of Target Input File for Effect</td>
<td>164</td>
</tr>
<tr>
<td>A.4 Graphical Output File</td>
<td>167</td>
</tr>
<tr>
<td>A.4.1 File format</td>
<td>167</td>
</tr>
<tr>
<td>A.4.2 Example of Graphical Output File for EVA</td>
<td>168</td>
</tr>
<tr>
<td><strong>B Proof of Equation 1.1</strong></td>
<td>171</td>
</tr>
<tr>
<td>B.1 Symbols</td>
<td>171</td>
</tr>
<tr>
<td>B.2 Assumptions</td>
<td>171</td>
</tr>
<tr>
<td>B.3 Proof</td>
<td>172</td>
</tr>
<tr>
<td><strong>C The Missile</strong></td>
<td>175</td>
</tr>
<tr>
<td><strong>D Intercept Scenarios data files</strong></td>
<td>177</td>
</tr>
<tr>
<td>D.1 Endgame Scenario Distributions Save File</td>
<td>177</td>
</tr>
<tr>
<td>D.1.1 File Format</td>
<td>177</td>
</tr>
<tr>
<td>D.1.2 Distribution Type/Parameters setups</td>
<td>178</td>
</tr>
<tr>
<td>D.1.3 Example of Endgame Scenario Distributions Save File</td>
<td>178</td>
</tr>
<tr>
<td>D.2 Path File</td>
<td>179</td>
</tr>
<tr>
<td>D.2.1 File Format</td>
<td>179</td>
</tr>
<tr>
<td>D.2.2 Example of Path File</td>
<td>180</td>
</tr>
<tr>
<td><strong>E Validation of distribution generation functions</strong></td>
<td>183</td>
</tr>
<tr>
<td>E.1 Beta</td>
<td>184</td>
</tr>
</tbody>
</table>
## Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>E.1.1</td>
<td>Input parameters for EVA: Beta(1, 6)</td>
<td>184</td>
</tr>
<tr>
<td>E.1.2</td>
<td>Input parameters for EVA: Beta(6, 15)</td>
<td>185</td>
</tr>
<tr>
<td>E.1.3</td>
<td>Input parameters for EVA: Beta(10, 10)</td>
<td>186</td>
</tr>
<tr>
<td>E.1.4</td>
<td>Input parameters for EVA: Beta(7, 2)</td>
<td>187</td>
</tr>
<tr>
<td>E.1.5</td>
<td>Input parameters for EVA: Beta(16, 4)</td>
<td>188</td>
</tr>
<tr>
<td>E.2</td>
<td>Exponential</td>
<td>189</td>
</tr>
<tr>
<td>E.2.1</td>
<td>Input parameters for EVA: Expo(1.25)</td>
<td>189</td>
</tr>
<tr>
<td>E.2.2</td>
<td>Input parameters for EVA: Expo(2.5)</td>
<td>190</td>
</tr>
<tr>
<td>E.2.3</td>
<td>Input parameters for EVA: Expo(11.11)</td>
<td>191</td>
</tr>
<tr>
<td>E.3</td>
<td>Gamma</td>
<td>192</td>
</tr>
<tr>
<td>E.3.1</td>
<td>Input parameters for EVA: Gamma(1, 3)</td>
<td>192</td>
</tr>
<tr>
<td>E.3.2</td>
<td>Input parameters for EVA: Gamma(0.2, 11)</td>
<td>193</td>
</tr>
<tr>
<td>E.4</td>
<td>Normal</td>
<td>194</td>
</tr>
<tr>
<td>E.4.1</td>
<td>Input parameters for EVA: Normal(10, 2.5)</td>
<td>194</td>
</tr>
<tr>
<td>E.4.2</td>
<td>Input parameters for EVA: Normal(7, 1)</td>
<td>195</td>
</tr>
<tr>
<td>E.5</td>
<td>Poisson</td>
<td>196</td>
</tr>
<tr>
<td>E.5.1</td>
<td>Input parameters for EVA: Poisson(10)</td>
<td>196</td>
</tr>
<tr>
<td>E.5.2</td>
<td>Input parameters for EVA: Poisson(20)</td>
<td>197</td>
</tr>
<tr>
<td>E.6</td>
<td>Uniform</td>
<td>198</td>
</tr>
<tr>
<td>E.6.1</td>
<td>Input parameters for EVA: Uniform(0, 10)</td>
<td>198</td>
</tr>
<tr>
<td>E.6.2</td>
<td>Input parameters for EVA: Uniform(2.33, 7.77)</td>
<td>199</td>
</tr>
<tr>
<td>F</td>
<td>Transformation Matrices</td>
<td>201</td>
</tr>
<tr>
<td>F.1</td>
<td>Rotation Operators</td>
<td>201</td>
</tr>
<tr>
<td>F.2</td>
<td>The Transformation Matrix</td>
<td>201</td>
</tr>
<tr>
<td>G</td>
<td>Drag Coefficients</td>
<td>205</td>
</tr>
<tr>
<td>G.1</td>
<td>Ball shaped fragments</td>
<td>205</td>
</tr>
<tr>
<td>G.2</td>
<td>Rectangularly (Cylindrical) shaped fragments</td>
<td>205</td>
</tr>
<tr>
<td>G.3</td>
<td>Cubed shaped fragments</td>
<td>206</td>
</tr>
</tbody>
</table>
# List of Figures

1.1 The engagement timeline ................................................. 3
1.2 Vulnerable Area Concept .................................................. 6

2.1 Modular Blocks EVA ....................................................... 15
2.2 Detailed Target ............................................................. 16
2.3 Target represented by polyhedrons and lines .......................... 16
2.4 Target Subsystem Model .................................................. 18
2.5 Prefragmented Warhead ................................................... 18
2.6 Intercept Path ............................................................... 19
2.7 Schematical representation of the cone used to model TDD ........... 20
2.8 Diagram of the events ...................................................... 21
2.9 Probability Graph .......................................................... 22

3.1 CCA methodology .......................................................... 26
3.2 Some essential functions and mission phases for an attack helicopter 27
3.3 Essential system-function relationships; systems compared with the same functions as shown in Figure 3.2 ................................. 28
3.4 Example of FMEA summary format ...................................... 29
3.5 Generic fault tree diagram ................................................ 29
3.6 DMEA Matrix ............................................................... 30
3.7 Example kill tree for a two-engine, two-pilot helicopter ................. 31
3.8 Target orientation in Cartesian coordinates ................................ 32
3.9 Polyhedron ................................................................. 33
3.10 Direction of normal vector: \( \cos \theta > 0 \) if \( \theta < 90^\circ \) .................... 35
3.11 Projecting normal component of vector ................................... 36
3.12 Target dropdown menu .................................................... 40
3.13 Create New Target .......................................................... 41
3.14 (a) Target deletion form, (b) Confirmation form ......................... 41
<table>
<thead>
<tr>
<th>Figure Number</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.15</td>
<td>View Target window</td>
<td>42</td>
</tr>
<tr>
<td>3.16</td>
<td>Original Input File Window</td>
<td>43</td>
</tr>
<tr>
<td>4.1</td>
<td>Threat Types</td>
<td>46</td>
</tr>
<tr>
<td>4.2</td>
<td>Missile guidance phases</td>
<td>47</td>
</tr>
<tr>
<td>4.3</td>
<td>Command guidance</td>
<td>48</td>
</tr>
<tr>
<td>4.4</td>
<td>Beam rider guidance</td>
<td>49</td>
</tr>
<tr>
<td>4.5</td>
<td>Homig guidance</td>
<td>50</td>
</tr>
<tr>
<td>4.6</td>
<td>Retransmission (TVM) guidance</td>
<td>51</td>
</tr>
<tr>
<td>4.7</td>
<td>The intercept geometry</td>
<td>54</td>
</tr>
<tr>
<td>4.8</td>
<td>Intercept angle</td>
<td>55</td>
</tr>
<tr>
<td>4.9</td>
<td>Cylinder angle</td>
<td>56</td>
</tr>
<tr>
<td>4.10</td>
<td>Target roll angle</td>
<td>57</td>
</tr>
<tr>
<td>4.11</td>
<td>The pitch, ( \theta ), and yaw, ( \psi ), angles relative to the Cartesian co-ordinate system</td>
<td>58</td>
</tr>
<tr>
<td>4.12</td>
<td>The pitch angle, ( \theta ), of the missile</td>
<td>59</td>
</tr>
<tr>
<td>4.13</td>
<td>The yaw angle, ( \psi ), of the missile</td>
<td>60</td>
</tr>
<tr>
<td>4.14</td>
<td>Direction vector of missile path</td>
<td>63</td>
</tr>
<tr>
<td>4.15</td>
<td>Paths dropdown menu</td>
<td>65</td>
</tr>
<tr>
<td>4.16</td>
<td>Endgame Scenario Distribution Form</td>
<td>66</td>
</tr>
<tr>
<td>4.17</td>
<td>Input forms for the Beta and Discrete Distributions</td>
<td>67</td>
</tr>
<tr>
<td>4.18</td>
<td>View the text file for a path</td>
<td>68</td>
</tr>
<tr>
<td>5.1</td>
<td>Damage on target plate illustrating mechanical cumulative effect</td>
<td>70</td>
</tr>
<tr>
<td>5.2</td>
<td>Additive and Cumulative Effects</td>
<td>71</td>
</tr>
<tr>
<td>5.3</td>
<td>Diagram illustrating relaxation areas according Randers-Pehrson</td>
<td>73</td>
</tr>
<tr>
<td>5.4</td>
<td>Diagram illustrating relaxation areas according Hennequin</td>
<td>74</td>
</tr>
<tr>
<td>5.5</td>
<td>Warhead dropdown menu</td>
<td>75</td>
</tr>
<tr>
<td>5.6</td>
<td>Warhead Variable Tab.</td>
<td>76</td>
</tr>
<tr>
<td>5.7</td>
<td>Warhead Blocks Tab.</td>
<td>77</td>
</tr>
<tr>
<td>5.8</td>
<td>Warhead Constants Tab.</td>
<td>78</td>
</tr>
<tr>
<td>5.9</td>
<td>Selection window of warhead for pre-processor</td>
<td>79</td>
</tr>
<tr>
<td>5.10</td>
<td>View Warhead output file for Upshot</td>
<td>80</td>
</tr>
<tr>
<td>5.11</td>
<td>Summative user output</td>
<td>81</td>
</tr>
<tr>
<td>5.12</td>
<td>Warhead deletion form</td>
<td>82</td>
</tr>
<tr>
<td>5.13</td>
<td>Warhead deletion confirmation form</td>
<td>83</td>
</tr>
</tbody>
</table>
List of Figures

6.1 Flow diagram - Fuze program.) .............................. 88
6.2 The \( \theta \) angle of the Fuze.) .............................. 89
6.3 The \( \alpha \) angle of the Fuze.) .............................. 90
6.4 Two dimensional side view of half the Fuze.) ........ 90
6.5 The vector from an internal point in the polyhedron, \( p_i \), to a point on the beam path, \( p_f \).) .................. 93
6.6 Calculating \( d'^2 \) using the theorem of Pythagoras.) .... 94
6.7 The perpendicular distance, \( d \), between the internal point, \( p_i \), and the side of the polyhedron. .................. 94
6.8 A point source and the relative direction of a beam from it to the normal vector of a plane. .................. 95
6.9 The beam moving in the general direction labelled 5 will intercept the plane in which side \( m \) lies first. ........ 98
6.10 Fuzing dropdown menu. .............................. 98
6.11 Fuze angle, step size, and reach distance. ........ 98
6.12 Fuze Window. .............................. 99
6.13 Fuze file. .............................. 100
6.14 Delete fuze file. .............................. 100
6.15 Flow diagram - Upshot program.) .................. 102
7.2 Possible intercept scenarios of the fragment with the cylinder.) .... 103
7.3 The cylinder around a polyhedron.) .................. 112
7.4 The cosines and sinus of the angle \( \theta \).) .................. 113
7.5 A diagrammatic illustration of the various variables.) ........ 115
7.6 A diagrammatic illustration of the various variables used to calculate the range, \( R \).) .................. 116
7.7 The distance the fragment needs to travel through the sides of the polyhedron.) . 120
7.8 Upshot dropdown menu. .............................. 129
7.9 Simulation setup window. .............................. 130
7.10 Select a standard simulation to be initiated. ........ 130
7.11 Select simulation results to view. ........ 130
7.12 Selecting graph of the simulation output. ........ 131
7.13 Select standard simulation run to delete. ........ 132
7.14 View the output file of Upshot in text format. ........ 132
7.15 The standard simulation graph. ........ 133
8.1 The warhead search algorithm flow diagram, part 1. ........ 139
8.2 The warhead search algorithm flow diagram, part 2. ............................ 140
8.3 Variable simulation setup window. .................................................. 141
8.4 Warhead generation for variable simulation. ................................... 142
8.5 Select a variable simulation to run. ............................................. 142
8.6 Select variable simulation information to be displayed. ................. 143
8.7 Variable Simulation Warhead Mass Graph. .................................... 144
8.8 Variable Simulation Warhead Fragment Graph. .............................. 145
8.9 View Variable Simulation Warhead Output file for user. .................. 146
8.10 Variable Simulation Graph. ........................................................ 147
B.1 ln(1 – p) vs –p where p << ....................................................... 173
C.1 A typical missile configuration (a). ............................................. 175
C.2 A typical missile configuration (b). ............................................. 176
E.1 Histogram of data and fit for Beta(0.966, 5.38212). ....................... 184
E.2 Histogram of data and fit for Beta(5.78, 13.2188). ......................... 185
E.3 Histogram of data and fit for Beta(9.26, 8.12886). ......................... 186
E.4 Histogram of data and fit for Beta(9.26, 8.12886). ......................... 187
E.5 Histogram of data and fit for Beta(12.6, 2.28084). ......................... 188
E.6 Histogram of data and fit for Expo(1.25). .................................... 189
E.7 Histogram of data and fit for Expo(2.52). .................................... 190
E.8 Histogram of data and fit for Expo(11.1). .................................... 191
E.9 Histogram of data and fit for Gamma(0.995, 3.01). ....................... 192
E.10 Histogram of data and fit for Gamma(0.199, 11). ......................... 193
E.11 Histogram of data and fit for Normal(9.95, 2.43). ......................... 194
E.12 Histogram of data and fit for Normal(6.98, 0.967). ....................... 195
E.13 Histogram of data and fit for Poisson(10). .................................. 196
E.14 Histogram of data and fit for Poisson(20). .................................. 197
E.15 Histogram of data and fit for Uniform(0, 10). ............................. 198
E.16 Histogram of data and fit for Uniform(2, 8). ............................. 199
## List of Tables

2.1 Component attributes ......................................................... 17
2.2 System breakdown probability due to component failure .................. 20
2.3 Target kill probability due to system breakdown. ........................ 21

4.1 Comparison of the midcourse and terminal guidance methods used on a variety of weapons. .................................................. 53

6.1 Cases for which the beam will potentially intercept the side first or not at all. 96
6.2 Beam characteristics .......................................................... 97

7.1 Formula for the average projected area \((m^2)\) and equivalent ball diameter \((m)\) of this area. ............................................................... 105
7.2 The constant values for the various fragment types as per Thor [27]. ........ 107
7.3 The data for crater calculations. .............................................. 107
7.4 The relation between the variables used in Upshot and Fuze. .............. 118
7.5 The IC Flag indicating the hit status. ....................................... 118
7.6 The metal type of the fragment. .............................................. 122
7.7 The number of hits on each vital part during intercept scenario \(i\). ......... 128
7.8 The mean number of fragments which penetrates the outer and internal protections for each vital part \(j\). .................................................. 128

E.1 Data analyses summary for Beta(1, 6). ...................................... 184
E.2 Data analyses summary for Beta(6, 15). .................................... 185
E.3 Data analyses summary for Beta(10, 10). ................................... 186
E.4 Data analyses summary for Beta(7, 2). .................................... 187
E.5 Data analyses summary for Beta(16, 4). ................................... 188
E.6 Data analyses summary for Expo(1.25). ................................... 189
E.7 Data analyses summary for Expo(2.5). .................................... 190
E.8 Data analyses summary for Expo(11.11). .................................. 191
E.9 Data analyses summary for Gamma(1, 3). ................................ 192
E.10 Data analyses summary for Gamma(0.2, 11)................................. 193
E.11 Data analyses summary for Normal(10, 2.5)............................... 194
E.12 Data analyses summary for Normal(7, 1).................................. 195
E.13 Data analyses summary for Poisson(10).................................... 196
E.14 Data analyses summary for Poisson(20).................................... 197
E.15 Data analyses summary for Uniform(0, 10)............................... 198
E.16 Data analyses summary for Uniform(2.33, 7.77)......................... 199

G.1 Ball shaped fragments......................................................... 205
G.2 Rectangularly (Cylindrical) shaped fragments............................. 205
G.3 Cubed shaped fragments..................................................... 206
## List of Symbols

The section in which symbols are defined or used for the first time in this thesis is given in brackets.

### Roman Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A vector lying in the plane of which the side of a polyhedron lies as well.</td>
<td>(3.3.1)</td>
</tr>
<tr>
<td>$\mathbf{a} = (x_a, y_a, z_a)$</td>
<td>A vector lying in the plane of which the side of a polyhedron lies as well.</td>
<td>(3.3.1)</td>
</tr>
<tr>
<td>$A$</td>
<td>A cross-sectional area of the body at right angles to the flight direction.</td>
<td>(5.4)</td>
</tr>
<tr>
<td>$\mathcal{A}$</td>
<td>The 2 equation system as per Cramer’s rule.</td>
<td>(7.2.3)</td>
</tr>
<tr>
<td>$A_1, A_2, A_3, A_4$</td>
<td>The value of a number of factors in the fragment velocity formula.</td>
<td>(7.2.3)</td>
</tr>
<tr>
<td>$A_f(i)$</td>
<td>Projected Area of fragment $i$.</td>
<td>(7.2.1)</td>
</tr>
<tr>
<td>$a_i$</td>
<td>The angle of point $i$.</td>
<td>(3.3.1)</td>
</tr>
<tr>
<td>$A_j$</td>
<td>The area of the opening in component $j$ that will cause a hundred percent (100%) failure of the component.</td>
<td>(7.2.4)</td>
</tr>
<tr>
<td>$A_{p_i}$</td>
<td>The presented area of the $i$th component.</td>
<td>(1.2.2)</td>
</tr>
<tr>
<td>$\mathcal{A}_p$</td>
<td>The average area on the effectiveness graphs under the probability of kill line and after the target detection for the various intercept scenarios.</td>
<td>(2.9)</td>
</tr>
<tr>
<td>$A_{v_i}$</td>
<td>The vulnerable area of the $i$th component.</td>
<td>(1.2.2)</td>
</tr>
<tr>
<td>$A_V$</td>
<td>The total vulnerable area.</td>
<td>(1.2.2)</td>
</tr>
<tr>
<td>$\mathbf{b} = (x_b, y_b, z_b)$</td>
<td>A vector lying in the plane of which the side of a polyhedron lies as well.</td>
<td>(3.3.1)</td>
</tr>
<tr>
<td>$B$</td>
<td>The resultant vector of the vectors $u_s$ and $d_c$.</td>
<td>(7.2.3)</td>
</tr>
<tr>
<td>$b_f(i)$</td>
<td>The width of fragment $i$.</td>
<td>(7.2.1)</td>
</tr>
<tr>
<td>$B_{frag}$</td>
<td>The width of the fragments in the warhead.</td>
<td>(8.2)</td>
</tr>
<tr>
<td>$b_i = (x_{bi}, y_{bi}, z_{bi})$</td>
<td>The coordinates for the $i^{th}$ burst point.</td>
<td>(4.4.2)</td>
</tr>
<tr>
<td>$c$</td>
<td>Constant.</td>
<td>(4.4.1)</td>
</tr>
<tr>
<td>$C$</td>
<td>Mass of the explosive.</td>
<td>(5.3.1)</td>
</tr>
<tr>
<td>$C_e$</td>
<td>The end coordinates of the cylinder defined around the polyhedron.</td>
<td>(7.2.3)</td>
</tr>
<tr>
<td>$\mathcal{g}_i$</td>
<td>The coordinates of the internal point of polyhedron $i$ in the earth</td>
<td>(7.2.3)</td>
</tr>
</tbody>
</table>
coordinate system.

$C_f(i)$ The coordinates of fragment $i$. (7.2.1)

$C_{fe}(i)$ The coordinates of fragments $i$ in earth coordinates. (7.2.3)

$c_i = (x_c, y_c, z_c)$ The coordinates of a point inside of the polyhedron. (3.3.1)

$C_{inner}$ The density of the inner casing. (8.2)

$Comp(k)$ The subsystem of which component $k$ is a part of. (7.2.1)

$C_{kill, crit}(i)$ Category for the kill criteria (1 - Personal, 2 - Ordinary Component, 3 - Fuel Tank, 4 - Wiring) for the critical component $i$. (7.2.1)

$C_{kill, line}(i)$ Category for the kill criteria (1 - Personal, 2 - Ordinary Component, 3 - Fuel Tank, 4 - Wiring) for line element $i$. (7.2.1)

$C_{ki}$ Constant $i$ for fragment type $k$ relevant to the penetration formula. (7.2.1)

$C_{layer}$ The number of layers of fragments the warhead has, given that the fragments are cubes. (8.2)

$C/M$ The charge-to-metal mass ratio. (8.2)

$C_{max}$ The maximum mass of explosives that can be fitted into the warhead. (8.2)

$C/M_{min}$ Minimum ratio of the mass of the explosive to the surrounding mass for the warhead search algorithm. (8.2)

$Comp_{i,j}$ The number of the $j^{th}$ component in system $i$. (7.2.1)

$C_{outer}$ The density of the outer casing. (8.2)

$Cov(1)$ The direction cosine. (6.3.3)

$Cov(2)$ The direction cosine. (6.3.3)

$C_p$ The coordinates of missile path $p$. (7.2.1)

$C_{pc}$ The coordinates of the missile in earth coordinates. (7.2.3)

$C_s$ The beginning coordinates of the cylinder defined around the polyhedron. (7.2.3)

$C_t$ The coordinates of the target. (7.2.1)

$c_w$ Drag coefficient. (5.4)

D

$d$ The distance from the internal point of a polyhedron to each of the sides. (3.3.1)

$D$ The diameter of the warhead. (5.3.1)

$d'$ The shortest distance that the beam can pass from the internal point. (6.3.3)

$d_c$ The distance between the coordinate of the fragment and the cylinder axes. (7.2.3)

$d_{cmax}(i)$ The maximum distance from the internal point to a corner of polyhedron $i$. (7.2.1)

$d_{cyl}$ The diameter of the cylinder around the polyhedron. (7.2.3)

$d_e$ The shortest distance from the fragment path to the point $C_e$. (7.2.3)

$D_{exp}$ The diameter of the explosive. (8.2)

$D_f(i)$ Ball Diameter of fragment $i$. (7.2.1)

$d_{int}$ The distance which the fragment is exposed to the internal protection. (7.2.3)

$d_j$ The distance from point $c$ to point $p_j$. (3.3.1)

$D_{line}(i)$ The diameter of line-element $i$. (7.2.1)

$d_{max}$ The maximum distance from the internal point to a corner of the polyhedron $i$. (3.3.1)

$D_{max}$ Maximum diameter for the warhead search algorithm. (8.2)

$d_{min}$ The minimum distance from the fragment path to a point on the axes. (7.2.3)
of the cylinder defined around the polyhedron.

\( d_{\text{max}} \) The maximum distance from the internal point, \( p_i \), to any point on the surface of the polyhedron.

\( d_{\text{pol}}(m, i) \) The perpendicular distance from the internal point, \( c_i \), to surface \( m \) of polyhedron \( i \).

\( d_r \) The reach distance of the sensors.

\( d_s \) The shortest distance from the fragment path to the point \( C_s \).

\( D_{\text{step}} \) Diameter step size for the warhead search algorithm.

\( d_{\text{temp}} \) The length of the direction vector, \( r_i \).

\( d_{\text{vpol}}(m, i) \) The perpendicular distance from the internal point, \( I_{pi} \), to surface \( m \) of polyhedron \( i \).

\( d_x \) The length of any vector in the \( xy \)-plane.

\( dx \) The small sub-divided width of a control area of the area of a component that is projected onto a plane.

\( dy \) The small sub-divided length of a control area of the area of a component that is projected onto a plane.

\( dz \) The length of the vector normal to the \( xy \)-plane.

\( \mathbf{E} \) The vector from the internal point to a point on the surface.

\( \mathbf{E} = (E_x, E_y, E_z) \) The vector from the internal point to a point on the surface.

\( E \) The Gurney energy of the explosive.

\( EM \) The transformation matrix from earth coordinates to missile coordinates.

\( ET \) The transformation matrix from earth coordinates to target coordinates.

\( E(X) \) The expected value (mean).

\( f \) Density function.

\( F \) Continues distribution function.

\( F_{\text{drag}} \) Total drag force on a fragment.

\( f_{\text{expos}} \) The fragment coordinates in earth coordinates.

\( f_{\text{fold}} \) A variable storing the previous \( f_{\text{time}} \) “time”.

\( f_{\text{pos}}(i) \) The starting coordinates of the fragment in the target coordinate system.

\( f_{\text{pos}}(i) \) The position of fragment \( i \) near the first polyhedron to be hit.

\( f_{\text{time}} \) Where the fragment is in “time”.

\( F_x \) The relaxation coefficient.

\( g \) Density function.

\( H \) A flag indicating the “type” of hit.

\( h_f(i) \) The height of fragment \( i \).

\( H_{\text{flag}} \)
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H_{\text{frag}}$</td>
<td>The height of the fragments in the warhead.</td>
<td>(8.2)</td>
</tr>
<tr>
<td>$\text{Hit}_{\text{Target}}$</td>
<td>A logical flag (a Boolean variable) which is true if the fragment will hit the target.</td>
<td>(7.2.3)</td>
</tr>
<tr>
<td>$H_{\text{min}}$</td>
<td>Minimum hole diameter for the warhead search algorithm.</td>
<td>(8.2)</td>
</tr>
<tr>
<td>$i$</td>
<td>The angle between the detonation front and the normal to the explosive/metal interface.</td>
<td>(5.3.1)</td>
</tr>
<tr>
<td>$ic$</td>
<td>The variable indicating the hit status.</td>
<td>(7.2.3)</td>
</tr>
<tr>
<td>$I_{\text{drag}}$</td>
<td>A boolean variable indicating whether drag is taken into account or not.</td>
<td>(5.3.1)</td>
</tr>
<tr>
<td>$\text{In}_{\text{Target}}$</td>
<td>A logical flag which indicates a direct hit, in other words the warhead explodes within the target.</td>
<td>(7.2.3)</td>
</tr>
<tr>
<td>$\text{Int}_{\text{prot}}(i)$</td>
<td>Internal protection in mm Al for vital part $i$.</td>
<td>(7.2.1)</td>
</tr>
<tr>
<td>$I_{p_i} = (x_i, y_i, z_i)$</td>
<td>The coordinates of the internal point of the polyhedron.</td>
<td>(7.2.1)</td>
</tr>
<tr>
<td>$k$</td>
<td>The direction vector of the cylinder defined around a polyhedron.</td>
<td>(7.2.3)</td>
</tr>
<tr>
<td>$k_1$ and $k_2$</td>
<td>Empirical constants.</td>
<td>(5.3.2)</td>
</tr>
<tr>
<td>$\vec{k}_u$</td>
<td>The normalised direction vector of the cylinder.</td>
<td>(7.2.3)</td>
</tr>
<tr>
<td>$L$</td>
<td>The length of the warhead.</td>
<td>(5.3.1)</td>
</tr>
<tr>
<td>$l_{\text{cyl}}$</td>
<td>The length of the cylinder defined around a polyhedron.</td>
<td>(7.2.3)</td>
</tr>
<tr>
<td>$L/D$</td>
<td>The length to diameter ratio.</td>
<td>(8.2)</td>
</tr>
<tr>
<td>$L/D_{\text{max}}$</td>
<td>Maximum length over diameter for the warhead search algorithm.</td>
<td>(8.2)</td>
</tr>
<tr>
<td>$L/D_{\text{min}}$</td>
<td>Minimum length over diameter for the warhead search algorithm.</td>
<td>(8.2)</td>
</tr>
<tr>
<td>$L/D_{\text{step}}$</td>
<td>Length over diameter step size for the warhead search algorithm.</td>
<td>(8.2)</td>
</tr>
<tr>
<td>$\text{Length}_s$</td>
<td>The distance from the sensor to the point on side that is intercepted.</td>
<td>(6.3.3)</td>
</tr>
<tr>
<td>$l_f(i)$</td>
<td>The length of fragment $i$.</td>
<td>(7.2.1)</td>
</tr>
<tr>
<td>$L_{\text{frag}}$</td>
<td>The length of the fragments in the warhead.</td>
<td>(8.2)</td>
</tr>
<tr>
<td>$L_{\text{max}}$</td>
<td>Maximum length for the warhead search algorithm.</td>
<td>(8.2)</td>
</tr>
<tr>
<td>$m$</td>
<td>Shape parameter for the Beta distribution.</td>
<td>(4.4.1)</td>
</tr>
<tr>
<td>$m$</td>
<td>The mass of the projectile.</td>
<td>(5.4)</td>
</tr>
<tr>
<td>$M$</td>
<td>The mass of the material surrounding the explosive.</td>
<td>(5.3.1)</td>
</tr>
<tr>
<td>$ME$</td>
<td>The transformation matrix from missile coordinates to earth coordinates.</td>
<td>(6.3.2)</td>
</tr>
<tr>
<td>$M_{\text{exp}}$</td>
<td>The maximum possible explosive mass for the warhead.</td>
<td>(8.2)</td>
</tr>
<tr>
<td>$m_f(i)$</td>
<td>The mass of fragment $i$.</td>
<td>(7.2.1)</td>
</tr>
<tr>
<td>$m_{\text{max}}$</td>
<td>Maximum fragment mass for the warhead search algorithm.</td>
<td>(8.2)</td>
</tr>
<tr>
<td>$M_{\text{max}}$</td>
<td>Maximum warhead mass for the warhead search algorithm.</td>
<td>(8.2)</td>
</tr>
<tr>
<td>$m_{\text{min}}$</td>
<td>Minimum fragment mass for the warhead search algorithm.</td>
<td>(8.2)</td>
</tr>
<tr>
<td>$m_{\text{step}}$</td>
<td>Fragment mass step size for the warhead search algorithm.</td>
<td>(8.2)</td>
</tr>
</tbody>
</table>
List of Symbols

**M**<sub>total</sub>  The total mass of the warhead.  (8.2)

\( n \)
Shape parameter for the Beta and Gamma distribution.  (4.4.1)

\( n \)
The number of possible values in the Discrete distribution.  (4.4.1)

\( \mathbf{n} = (x_n, y_n, z_n) \)
The normal vector of a surface of a polyhedron.  (3.3.1)

\( N_{\text{comp}}(i) \)
The number of components in system \( i \).  (7.2.1)

\( n_{\text{components}} \)
The total number of vital components for the target.  (7.2.1)

\( n_f \)
The number of fragments in the warhead.  (7.2.1)

\( N_f\text{frag} \)
The number of fragments.  (8.2)

\( n_{is} \)
The number of intercept scenarios.  (2.9)

\( \mathbf{n}_k \)
A vector which is normal to the direction vector of centerline, \( \mathbf{k}_u \), and the path, \( \mathbf{r}_f \).  (7.2.3)

\( n_{k,\text{abs}} \)
The absolute value of the normal vector \( \mathbf{n}_k \).  (7.2.3)

\( n_{ohit}(i) \)
The number of hits on the outer skin of the target during intercept scenario \( i \).  (7.2.2)

\( N_{ohit} \)
The number of hits on the outer skin of the target during all the intercept scenarios.  (2.9)

\( n_p \)
The number of polyhedrons hit by the fragment.  (7.2.3)

\( N_p \)
The number of high probability regions in the detonation vicinity.  (2.9)

\( n_{hit} \)
The total number of surfaces hit.  (7.2.3)

\( N_{\text{pol}} \)
The number of polyhedrons describing the outer geometry.  (7.2.1)

\( \mathbf{n}_{\text{pol}}(m, i) \)
The normal vector for surface \( m \) of polyhedron \( i \).  (7.2.1)

\( n_s \)
The number of sensors in the fuze.  (6.3.2)

\( N_{\text{pol}}(i) \)
The number of surfaces for polyhedron \( i \).  (7.2.1)

\( N_{\text{vpol}}(i) \)
The number of surfaces for the vital part polyhedron \( i \).  (7.2.1)

\( N_{\text{syst}} \)
The number of systems.  (7.2.1)

\( \mathbf{n}_{\text{vpol}}(m, i) \)
The normal vector for surface \( m \) of the vital part polyhedron \( i \).  (7.2.1)

\( N_{\text{vhit}}(j) \)
The number of hits on critical component \( j \) over all paths.  (7.2.2)

\( n_{vhit}(i, j) \)
The number of hits on vital component \( j \) of the target for intercept scenario \( i \).  (7.2.2)

\( N_{\text{eline}} \)
The number of lines describing the vital parts of the target.  (7.2.1)

\( N_{\text{vpol}} \)
The number of polyhedrons describing the vital parts of the target.  (7.2.1)

**P**

\( P(A) \)
The sum of the probability of event \( A \) occurring for all the evaluated burst points.  (7.2.2)

\( P_A(i) \)
The probability that the mission will be aborted and the target will be killed should system \( i \) fail.  (7.2.1)

\( P(B) \)
The sum of the probability of event \( B \) occurring for all the evaluated burst points.  (7.2.2)

\( P_B(i) \)
The probability that the mission will be aborted, but the target is not killed should system \( i \) fail.  (7.2.1)

\( P_{\text{breakdown}}(X_{js}) \)
The probability that system \( s \) will breakdown given a hit on component \( j \).  (2.7)

\( P(C) \)
The sum of the probability of event \( C \) occurring for all the evaluated burst points.  (7.2.2)
\( P_C(i) \) \( (7.2.1) \) \( P_C(i) \): The probability that the mission will be accomplished and the target will be killed should system \( i \) fail.

\( P_{comp}(i) \) \( (7.2.2) \) \( P_{comp}(i) \): The probability of a specific component failing for a specific intercept scenario.

\( P(D) \) \( (7.2.2) \) \( P(D) \): The sum of the probability of event D occurring for all the evaluated burst points.

\( P_D(i) \) \( (7.2.1) \) \( P_D(i) \): The probability that the mission will be accomplished, the target is not killed, but does need repairs should system \( i \) fail.

\( P_e(s) \) \( (2.7) \) \( P_e(s) \): The probability that event \( e \) will occur given that system \( s \) fails.

\( p_f \) \( (6.3.3) \) \( p_f \): A point on the beam path.

\( P_H \) \( (1.1) \) \( P_H \): Probability of a hit.

\( p_i \) \( (6.3.3) \) \( p_i \): An internal point in the polyhedron.

\( P_{mean}(X) \) \( (7.2.7) \) \( P_{mean}(X) \): The mean probability for mission criteria \( X \) over all the intercept scenarios.

\( P_{O\#} \) \( (7.2.3) \) \( P_{O\#} \): The actual hit out of the recorded outer skin hits.

\( P_{Ot} \) \( (7.2.3) \) \( P_{Ot} \): The last used polyhedron from the outer skin on the analyses of the fragment path.

\( P_{otime} \) \( (7.2.3) \) \( P_{otime} \): The time the outer skin was hit.

\( p_q = (x_q, y_q, z_q) \) \( (x_q, y_q, z_q) \): A point on the plane in which the polyhedron side lies.

\( p_r = (x_r, y_r, z_r) \) \( (x_r, y_r, z_r) \): The projection of point \( i \) on the plane.

\( (x_{pri}, y_{pri}, 0) \) \( (x_{pri}, y_{pri}, 0) \): The internal protection of the polyhedron going into it on the \( n_p^{th} \) hit.

\( Prot_{in} \) \( (7.2.3) \) \( Prot_{in} \): The internal protection of line-element \( i \) in millimeter aluminium.

\( Prot_{internal}(i) \) \( (7.2.1) \) \( Prot_{internal}(i) \): The expected value for the internal protection of polyhedron \( i \).

\( Prot_{out}(i) \) \( (7.2.1) \) \( Prot_{out}(i) \): The outer protection of line-element \( i \) in millimeter aluminium.

\( P_{s/k} \) \( (1.2.2) \) \( P_{s/k} \): The probability that the critical system will fail given that the component fails.
List of Symbols

\( P_s \)
Probability of survival. (1.1)

\( p_{t_f}(i,j) \)
The fraction, in percentage, for each value, \( p_{t_p}(i,j) \), of the structure, \( j = 1,2,3 \). (7.2.1)

\( P_{\text{total/s}} \)
The probability of kill of the target given a specific threat. (1.2.2)

\( P_{\text{total}}(X) \)
The probability that event \( X \) will occur at burst point \( i \) taking all of the systems into account. (7.2.6)

\( p_{t_p}(i,j) \)
The internal structure in mm aluminium per meter length (three (3) values, \( j = 1,2,3 \)). (7.2.1)

\( P_{t/s} \)
The probability that the target will fail given that the critical system fails. (1.2.2)

\( p(x,y) \)
The kill probability of a fragment hitting a target at the point \( (x,y) \). (1.2.2)

\( Q \)

\( Q_1 \)
The dot product of \( n \) and \( r_l \). (6.3.3)

\( Q_2 \)
The dot product of \( n \) and \( r_{pf} \), the direction of the point on the beam path. (6.3.3)

\( R \)
The distance from the original coordinate of the fragment to where the hit takes place, called the range. (7.2.3)

\( r_{f,\text{Abs}} \)
The fragment direction, \( r_f(i) \), in earth coordinates. (7.2.3)

\( r_l \)
The direction the point, passing the polyhedron, is moving in. (6.3.3)

\( r_{si} = (x_{si}, y_{si}, z_{si}) \)
The search direction of sensor \( i \) in the fuze, in missile coordinates. (6.3.2)

\( r_{s} = (x_{r}, y_{r}, z_{r}) \)
The direction vector, \( r_i = (x_{ri}, y_{ri}, z_{ri}) \), for the intercept point \( i \). (6.3.1)

\( r_{sf} = (x_{rf}, y_{rf}, z_{rf}) \)
The direction of fragment \( i \) in earth coordinates. (7.2.3)

\( R_{\text{max}} \)
The maximum possible radius of the explosive for the warhead. (8.2)

\( R_{\text{min}} \)
The miss distance of the intercept path. (2.5)

\( r_{te} \)
The absolute size of the direction of the target in earth coordinates. (7.2.3)

\( r_{te,\text{abs}} \)
The absolute size of the direction of the target in earth coordinates. (7.2.3)

\( S \)
The shape of the fragment type in the warhead. (8.2)

\( S_{\text{Description}}(i) \)
The description of system \( i \). (7.2.1)

\( S_f(i) \)
The shape of fragment \( i \). (7.2.1)

\( S_{\text{in}}(n_p) \)
The skin thickness on the side entered on the \( n_p^\text{th} \) hit. (7.2.1)

\( S_{\text{out}}(n_p) \)
The skin thickness of the exit side on the \( n_p^\text{th} \) hit. (7.2.1)
LIST OF SYMBOLS

$S(\theta, \psi, \phi)$ The transformation matrix for the missile. (4.4.2)
$S'(\theta, \psi, \phi)$ The transformation matrix. (4.4.2)

$T$

$t_c$ The time between the coordinate of the fragment and the cylinder axes. (7.2.3)
$t_e$ The shortest “time” from the fragment path to the point $C_e$. (7.2.3)
$TE$ The transformation matrix from target coordinates to earth coordinates. (6.3.2)
$t_f$ The shortest time to a point on the fragment path. (7.2.3)
$th_{pol}(m,i)$ The thickness of surface $m$ of polyhedron $i$. (7.2.1)
$th_{vpol}(m,i)$ The thickness of surface $m$ of the vital part polyhedron $i$. (7.2.1)
$t_i = (x_{ti}, y_{ti}, 0)$ The translated-projection of point $i$ on the $xy$-plane. (3.3.1)
$t_{in}(n_p)$ The time when the fragment hits the polyhedron on the $n_p^{th}$ hit. (7.2.3)
$T_{inner}$ The height of the inner casing of the warhead. (8.2)
$T_l$ The “time” that the beam takes to reach side $l$. (6.3.3)
$t_m$ The “time” that the beam takes to reach side $m$. (6.3.3)
$T_{max}$ The position where the polyhedron is intercepted second. (6.3.3)
$t_{min}$ The minimum “time” from the fragment path to a point on the axes of the cylinder defined around the polyhedron. (7.2.3)

$T_{min}$ The position where the polyhedron is intercepted first. (6.3.3)
$t_n$ The “time” that the beam takes to reach side $n$. (6.3.3)
$t_{out}(n_p)$ The time when the fragment leaves the polyhedron on the $n_p^{th}$ hit. (7.2.3)
$T_{outer}$ The height of the outer casing of the warhead. (8.2)
$t_s$ The shortest “time” from the fragment path to the point $C_s$. (7.2.3)
$T(\theta, \psi, \phi)$ The transformation matrix for the target. (4.4.2)

$U$

$u = (x_u, y_u, z_u)$ The unit normal vector of a surface of a polyhedron. (3.3.1)
$U$ An uniform(0,1) random variable. (4.4.1)
$u_e$ The vector from a point on the line passing the fragment path and $C_e$. (7.2.3)
$u_e$ The magnitude of vector $u_e$. (7.2.3)
$u_{path}$ The unit vector for the direction of the missile. (4.4.2)
$u_s$ The vector from a point on the line passing the fragment path and $C_s$. (7.2.3)
$u_s$ The magnitude of vector $u_s$. (7.2.3)

$V$

$v$ The speed of the projectile. (5.4)
$v_0$ The starting speed. (5.4)
$V_{Abs,m}$ The absolute velocity of the missile. (6.3.2)
$V_{Abs,t}$ The absolute velocity of the target. (6.3.2)
$v_{am}$ The absolute velocity of the missile. (7.2.1)
$v_{at}$ The absolute velocity of the target. (7.2.1)
$v_{avg}$ The average velocity, just prior to a hit on a part taking place, taking the effect of drag into account, relative to the earth. (7.2.3)
$V_{avg}(i)$ The average velocity, just prior to a hit on polyhedron $i$ taking place, taking the effect of drag into account, relative to the target. (7.2.3)
List of Symbols

- \( v_{\text{begin}}(i) = (x_{bi}, y_{bi}, z_{bi}) \): Coordinates for the starting point of the line-element. (7.2.1)
- \( v_c \): The fragment velocity. (5.3.1)
- \( v_{\text{Cat}}(n_{vp}) \): The category in which component number \( v_#(n_{vp}) \) falls. (7.2.3)
- \( v_D \): The detonation velocity. (5.3.1)
- \( v_{\text{end}}(i) = (x_{ei}, y_{ei}, z_{ei}) \): Coordinates for the end point of the line-element. (7.2.1)
- \( v_f(i) \): The velocity of fragment \( i \). (7.2.1)
- \( v_f(i) \): The absolute velocity of fragment \( i \). (7.2.1)
- \( V_f \): The fragment velocity, in the target coordinate system, taking drag into account and positioned near the first polyhedron to be hit by the fragment. (7.2.3)
- \( v_{\text{final}} \): The magnitude of the fragment velocity after all the hits. (7.2.3)
- \( V_{\text{frag}} \): Fragment velocity in the target coordinate system relative to the target. (7.2.3)
- \( z_{vf}(i) \): The velocity of fragment \( i \) relative to the earth, in earth coordinates. (7.2.3)
- \( v_{fme}(i) = (x_{vf}(i), y_{vf}(i), z_{vf}(i)) \): The velocity of fragment \( i \) relative to the missile in earth coordinates. (7.2.3)
- \( v_{fme,abs}(i) \): The absolute size of \( v_{fme}(i) \). (7.2.3)
- \( v_{in}(n_p) \): The magnitude of the velocity of the fragment when it is going into the polyhedron on the \( n^{th} \) hit. (7.2.3)
- \( V_i \): The \( i^{th} \) value for the Discrete distribution. (4.4.1)
- \( V_m \): The closing speed of the missile relative to the target. (4.4.2)
- \( V_M \): The velocity vector of the missile. (4.3.1)
- \( V_{\text{mach}} \): The Mach number. (7.2.3)
- \( v_{me} \): The velocity of the missile in earth coordinates. (6.3.2)
- \( v_#(n_{vp}) \): The number of the vital part that is hit by the \( n^{th} \) fragment hit. (7.2.3)
- \( v_{out}(n_p) \): The magnitude of the velocity of the fragment when it is going out of the polyhedron on the \( n^{th} \) hit. (7.2.3)
- \( VP_{\text{last}} \): The last used polyhedron of the vital parts. (7.2.3)
- \( V_R \): The velocity vector of the missile relative to the target. (4.3.1)
- \( V_T \): The velocity vector of the target. (4.3.1)
- \( V_t = (V_x, V_y, V_z) \): The target velocity in the earth coordinate system. (4.4.2)
- \( v_{val}(n_{vp}) \): The probability of kill of the part given a hit. (7.2.3)
- \( v_{te} \): The velocity of the target in earth coordinates. (6.3.2)
- \( v_x \): The fragment velocity just prior to a hit on a part taking place, relative to the earth. (7.2.3)
- \( V_x(i) \): The fragment velocity just prior to a hit on polyhedron \( i \) taking place, relative to the target. (7.2.3)
- \( V(X) \): The variance. (4.4.1)

W

- \( w_{1..3} \): The weights for the various terms in the objective function used in the meta-heuristic search.
**X**

\[ x = (l_x, m_x, n_x) \] Direction cosine. \hspace{1cm} (3.3.1)
\[ \overline{x} = (x_1, x_2) \] The unique solution for the 2 equation system solved making use of Cramer’s rule. \hspace{1cm} (7.2.3)

\[ X \] Random Variable. \hspace{1cm} (4.4.1)
\[ x_0 \] The x coordinate of the origin of the fuze detection sensors. \hspace{1cm} (6.3.1)
\[ x_{avg} \] The average x and y coordinates of the projected points on a plane. \hspace{1cm} (3.3.1)
\[ x_i \] The x coordinate of the maximum reach distance for sensor, \( i \). \hspace{1cm} (6.3.1)
\[ x_{ip} \] The x coordinate of the internal aim point. \hspace{1cm} (4.4.2)
\[ X_{js} \] The random variable indicating a hit on component \( j \) in system \( s \). \hspace{1cm} (2.7)
\[ x_{rel} \] The x coordinate of the zero burst point. \hspace{1cm} (2.5)
\[ X_s \] The distance from the original coordinate of the fragment to the perpendicular point where the fragment passes the axis of the cylinder. \hspace{1cm} (7.2.3)

**Y**

\[ y = (l_y, m_y, n_y) \] Direction cosine. \hspace{1cm} (3.3.1)
\[ y_0 \] The y coordinate of the origin of the fuze detection sensors. \hspace{1cm} (6.3.1)
\[ y_{avg} \] The average y coordinates of the projected points on a plane. \hspace{1cm} (3.3.1)
\[ y_i \] The y coordinate of the maximum reach distance for sensor, \( i \). \hspace{1cm} (6.3.1)
\[ y_{ip} \] The y coordinate of the internal aim point. \hspace{1cm} (4.4.2)
\[ y_{rel} \] The y coordinate of the zero burst point. \hspace{1cm} (2.5)

**Z**

\[ z = (l_z, m_z, n_z) \] Direction cosine. \hspace{1cm} (3.3.1)
\[ z_0 \] The z coordinate of the origin of the fuze detection sensors. \hspace{1cm} (6.3.1)
\[ z_i \] The z coordinate of the maximum reach distance for sensor, \( i \). \hspace{1cm} (6.3.1)
\[ z_{ip} \] The z coordinate of the internal aim point. \hspace{1cm} (4.4.2)
\[ z_{rel} \] The z coordinate of the zero burst point. \hspace{1cm} (2.5)

**Greek Symbols**

\[ \alpha \] The fragment ejection angle. \hspace{1cm} (5.3.1)
\[ \alpha \] The number of degrees two sensors are apart (the density of the sensors). \hspace{1cm} (6.3.1)
\[ \alpha \] The value of a number of factors in the drag formula. \hspace{1cm} (7.2.3)
\[ \beta \] Cylinder angle in degrees. \hspace{1cm} (4.3.1)
\[ \chi \] Velocity Elevation, relative to the target. \hspace{1cm} (4.4.2)
\[ \epsilon \] A small value. \hspace{1cm} (6.3.3)
\[ \phi_f \] The flux of fragments through the vulnerable area. \hspace{1cm} (1.2.2)
\[ \phi \] Missile roll angle, in the missile system. \hspace{1cm} (4.4.2)
\[ \phi_{target} \] The roll angle of the target, in radians. \hspace{1cm} (4.4.2)
\[ \phi_{deg} \] The roll angle in degrees. \hspace{1cm} (4.4.2)
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\gamma)</td>
<td>Velocity Heading, relative to the target.</td>
<td>(4.4.2)</td>
</tr>
<tr>
<td>(\lambda)</td>
<td>The scale parameter for the Gamma distribution.</td>
<td>(4.4.1)</td>
</tr>
<tr>
<td>(\lambda)</td>
<td>The mean parameter for the Poisson distribution.</td>
<td>(4.4.1)</td>
</tr>
<tr>
<td>(\theta)</td>
<td>Missile pitch angle, in the missile system.</td>
<td>(4.3.1)</td>
</tr>
<tr>
<td>(\theta)</td>
<td>Half of the angle that the detection sensors of the fuze span.</td>
<td>(6.3.1)</td>
</tr>
<tr>
<td>(\theta_s)</td>
<td>The angle between (\mathbf{u}_s) and the fragment path.</td>
<td>(7.2.3)</td>
</tr>
<tr>
<td>(\theta_e)</td>
<td>The angle between (\mathbf{u}_e) and the fragment path.</td>
<td>(7.2.3)</td>
</tr>
<tr>
<td>(\theta_{target})</td>
<td>The pitch angle of the target, in radians.</td>
<td>(4.4.2)</td>
</tr>
<tr>
<td>(\rho_f(i))</td>
<td>The density of fragment (i).</td>
<td>(7.2.1)</td>
</tr>
<tr>
<td>(\rho)</td>
<td>The air density.</td>
<td>(7.2.3)</td>
</tr>
<tr>
<td>(\rho_{frag})</td>
<td>The density of the fragment type in the warhead.</td>
<td>(8.2)</td>
</tr>
<tr>
<td>(\rho_{exp})</td>
<td>The density of the explosive.</td>
<td>(8.2)</td>
</tr>
<tr>
<td>(\sigma)</td>
<td>The standard deviation.</td>
<td>(4.4.1)</td>
</tr>
<tr>
<td>(\psi)</td>
<td>Missile yaw angle, in the missile system.</td>
<td>(4.3.1)</td>
</tr>
<tr>
<td>(\psi_{target})</td>
<td>The yaw angle of the target, in radians.</td>
<td>(4.4.2)</td>
</tr>
</tbody>
</table>
Chapter 1

Effectiveness and Vulnerability

1.1 Introduction

In October 1943 the US 8th Air Force suffered a 24% attrition rate during unescorted daylight raids against the ball bearing factories in Schweinfurt, Germany. The US Air force’s daytime deep penetration flights were suspended as a result [1].

Mitsusa Kofukuda, Commander of the 6th Japanese Air Force during World War II, stated that the ability of the US B-17 and B-24 to complete their missions despite fighter opposition was the deciding factor in the final outcome of the war between Japan and the United States [1].

These two very contrasting scenarios were brought about by the ability and inability of the enemy to affect sufficient damage to the bombers. In the case of the US 8th Air Force, the effectiveness of the air defences led to complete mission suspension in the long-term. Contrary to this, the inefficiency of the Japanese defences resulted in eventual defeat.

On the first day of the Yom Kippur War, 1973, between Israel and Syria, Israeli tank units received accurate ground support from Israeli aircraft. These aircraft, although sufficiently equipped against the Syrian SA-2 surface-to-air (SA) missiles, were vulnerable against the semi active homing SA-6, as well as the radar-directed 23-mm cannon of the ZSU-23-4, built by the Soviets. During the first afternoon of fighting, the losses suffered by the Israelis were so severe that all subsequent air strikes over the Golan Heights were cancelled. This cancellation of flights led to the rapid deterioration of the ground situation, which in turn caused the resumption of the air strikes, but with different tactics [1].

The efficiency of the Syrian weapons against its specific target made the Israelis change their tactics and initially had the Israelis overpowered.

The ability or inability to effectively neutralise the threat of the enemy through a defence system can turn the tide of a campaign. Having effective weapons that fully exploit vulnerability of the enemy threats is of utmost importance. From this need the Missile Efficiency Discipline (MED) developed. The aim of MED is to identify the vulnerable features in a specific target that may be exploited to increase the effectiveness of the missile as a weapon system.

The field of study that has many similarities to MED, but aims to achieve the direct oppo-
site, is the *Weapon Combat Survivability Discipline* (WCSD). It aims to minimise susceptibility and vulnerability of one weapon system to another. Stated differently, the WCSD aims to identify those specific survivability features that increase the effectiveness of the aircraft as a weapon system.

**Susceptibility** measures the inability of a target to avoid a man-made hostile environment. This can be shown as the probability, $P_H$, that the target is hit by a damage-causing mechanism. The higher the susceptibility of a target the better the chances of a target being detected and being hit by a damage-causing mechanism [1]. **Vulnerability**, on the other hand, is the inability of a target to withstand a man-made hostile environment. Each component in a target has a degree of vulnerability, and individually contributes to the total vulnerability of the target. The larger the conditional probability of a target being “killed”, given a hit by a damage-causing mechanism, $P_{K|H}$, the larger the vulnerability of the target [2].

In general, the word vulnerable is used to refer to a quality of something that can be injured, damaged or killed given a certain situation. For example, a region might be seen to be vulnerable to natural threats, e.g. earthquakes, floods, hurricanes etc. In the defence industry the term vulnerability has more specific definitions. If the example of the flood is extended to explain the terminology as used in the defence environment one would say that the region is very susceptible to floods, but due to the building methods and lifestyle it is not so vulnerable to it. Vulnerability, to a certain extent, also has a very loose definition in the sense that it is also used differently amongst the various communities in the defence environment. Should the question be posed: “What is the vulnerability of the target?” This question can be interpreted in more than one way. Vulnerability can be seen as referring to the target in a more holistic manner, looking at all its aspects. Should the target be easily detected by the enemy it would then be seen to be more vulnerable. Vulnerability in this case refers to the ability to avoid a man-made hostile environment, to avoid being hit should the aircraft be in the hostile environment, and how lethal the effect of a hit or multiple hits will be. At the other end of the spectrum, vulnerability might refer to the likelihood that a certain target is killed, given a hit [3].

**Endgame** refers to the last few milliseconds of an engagement between a missile and a target, as illustrated in Figure 1.1. It is the critical part of the ‘life’ of a missile, i.e. when the guidance system has become redundant and the final intercept path of the missile is known. At the end of this path the warhead is detonated, either by impact or due to a proximity fuze [1] [2] [4].

For the purposes of this study vulnerability will refer to the likelihood that a target will be killed given that a certain warhead of a certain missile will detonate within the vicinity of the target and the target is hit. In other words, by confining the environment to the endgame scenario, how probable is the kill of a target given that there was a hit for a certain warhead and missile design.

Endgame evaluation supports both the analyses of the survivability of a weapon system against anti-air missile threats, as well as the design of a missile system to be used against enemy targets [5].

When the survivability of a weapon system against anti-air missile threats is analysed, the actual or conceptual design of the missile and weapon system are evaluated in various operational scenarios. The probability of survival, $P_S$, can be determined by taking the complement
of the probability of kill, given that the weapon system was hit,

\[ P_S = 1 - P_H P_K | H. \]

This probability can then be used in higher level models for the evaluation of survivability trade-offs [5].

When an endgame evaluation is done in the design of a missile system to be used against enemy targets, it includes the analyses of fuze design, detonation control logic, and time delay functions. Terminal missile guidance system design trade-offs can be evaluated as well as the effectiveness of a warhead design concept [5]. The warhead effectiveness, \( P_K \), is also described as the probability of kill, given a hit.

Endgame analysis aims, on the one hand, to maximise the effectiveness of a weapon system against specific targets, whilst on the other hand, it aims to minimise the vulnerability of a weapon system, e.g. an airplane [5]. Whilst vulnerability is a characteristic of a weapon system or target quantifying the effect of various damage mechanisms on the vulnerable components of the system or target and the possible malfunction thereof as a result, effectiveness refers to the ability to inflict damage on a target and is given as a statistical estimate.

MED and WCSD both look at the complete engagement process, seen in Figure 1.1, but from two completely different perspectives. MED aims to prevent an enemy weapon system from completing a mission, whilst WCSD aims to achieve mission success. Vulnerability and warhead effectiveness have many similarities, but also have definite and important differences. In much of the literature these two concepts are grouped and often seen as one. Vulnerability, together with susceptibility, forms part of WCSD, whilst warhead effectiveness is a subset of the MED. In both cases the value is given as \( P_K \).

Warhead effectiveness is used to evaluate different warhead designs against a specific target.
for a specific missile. In the case of warhead effectiveness only a single $P_K$ value is given. This value is then used to compare the various warhead designs against one another.

One of the uses of vulnerability analyses is to determine where design changes are needed and to evaluate the success of these changes in improving the vulnerability of the weapon system. In the case of vulnerability a much more detailed weapons system or target description is needed than for the warhead effectiveness analyses. Vulnerability has multiple values that describe the probability of kill given a hit. These are e.g. the probability of killing the component given a hit on the component itself, killing the target given a hit on the target, and killing the component given a hit on the target [1]. These different conditional kill probabilities may then be used to determine which components need more protection, or as part of higher level survivability studies [6].

1.2 Methodologies

Modelling is an important tool in the field of effectiveness and vulnerability analysis. This is due to the costly nature of live fire testing. Apart from the production of a warhead being an expensive exercise in itself, the destruction of a target, e.g. an aircraft, is an even more expensive exercise. There are also other expenses to take into account, e.g. a test range and personnel to conduct the tests. As with any experiment there also exist the possibility that little information can be gained after the test has been done. For this reason models with predictive capabilities, in conjunction with data produced from tests, are useful as well as important tools in effectiveness assessment environments. For accurate effectiveness assessments a good balance between data from tests and modelling is required. Modelling is also important because it is often the only way that a large number of threat-target interactions can be examined [7].

Currently effectiveness modelling is, at best, an art of estimation. The models that are used depend on approximations made from empirical observations. If incorrect models are used to replicate test results, their predictions will be incorrect. As a result, verification, validation and authentication (VV&A) of these models is an extremely important (albeit an often difficult) step in the effectiveness assessment process.

Modelling extends the limited number of experiments and tests that are viable to cover the basically infinite possible threat-target interaction scenarios. In doing an effectiveness study it is also important to understand the process involved, modelling helps to do this and also points out misunderstandings.

A model is a mathematical construction that describes a physical process or complex sequence of processes. Below is a list of various types of models applicable to effectiveness modelling, with an integration of these models typically used in constructing an effectiveness model [7].

*Closed-Form Model:* Is a mathematical equation describing a phenomenon (e.g. $F = ma$).

*Deterministic Model:* Gives a definite result for a specific scenario instead of a probabilistic estimate (e.g. Ohm’s law).

*Empirical Model:* Relates a complex physical event to an equation through
1.2. Methodologies

There is little attempt to describe the actual physical mechanisms.

**Encounter Model:** Produces an estimate of the probability of kill for encounters between a target and ammunition.

**Numerical Model:** This model derives a result through extensive calculations (e.g. finite element model).

**Phenomenological Model:** Models a physical process that is subordinate to a complete effectiveness analyses (e.g. a model of the penetration capabilities of a bullet).

**Probabilistic Model:** Uses parameters such as probability of occurrence, mean value, and variance to describe a process.

**Stochastic Model:** Uses repetitive calculations with random sampling in its probabilistic model (also known as a Monte Carlo simulation.)

1.2.1 Phenomenological Model

The quality of the phenomenological models used in any encounter model determines the validity of the latter model. The phenomenological models may be closed-form, numerical-analysis, or empirical models. The processes described by these models can become complicated physically and the closed-form models are many times an empirical fit to experimental data, and often expressed probabilistically. For these models extensive experimentation is required, both to cover the range of parameter values required as well as to develop a statistically valid representation (e.g. experimentation has been the main approach for obtaining penetration and component-damage data).

1.2.2 Encounter Models

With missile/target interaction, where the initial conditions are variable and the results of the model are expressed as probabilities, it follows that encounter models are probabilistic. Even though the events are probabilistic, these models more often than not use a deterministic way in which to calculate the probabilities. Stochastic models are often used to obtain a better interface with the results of live fire tests.

There are currently two main encounter models being used globally with respect to warhead effectiveness, namely the *vulnerable area* concept and the *ray-tracing* methodology.
Figure 1.2: Vulnerable Area Concept.
1.2. Methodologies

1.2.2.1 Vulnerable Area

A vulnerable area is connected to both the target and its critical components. The vulnerable area of the \(i\)th component, \(A_{vi}\), is the product of the presented area, \(A_{pi}\), of the component in the plane normal to the direction of the approaching fragment and the kill probability of the \(i\)th component given that it was hit, \(P_{k/h_i}\). Thus, from Ball [1],

\[
A_{vi} = A_{pi} \cdot P_{k/h_i}.
\]

In other words, if the area of a component that is projected onto a plane is divided into small control areas with a length \(dy\) and width \(dx\), and \(p(x, y)\) is the kill probability of a fragment hitting a target at the point \((x, y)\), then

\[
A_{vi} = \int \int_{A_{pi}} p(x, y) dx dy.
\]

It is assumed that the location on the presented target area for a single hit and multiple hits has a random distribution and that each damage mechanism, e.g. fragment, has the same attack direction. In other words, they travel in parallel shotlines to one another and are all normal to the same plane. This follows from the assumption that the fragments are not directed to a specific component, subsystem, or part of the target.

Since the kill probability of a fragment is part of the function to generate the vulnerable area of the component, it is clear that the vulnerable area is also a function of fragment parameters, e.g. velocity, size, shape.

The model, as a result, makes use of the flux, \(\phi_f\), of fragments through the vulnerable area, with the total vulnerable area denoted as \(A_V\) and flux denoting the number of fragments/m². There is, however, only a limited number of plane orientations used in the generation of the vulnerable area data and therefore if the flux is not normal to any one of these, the vulnerable areas need to be extrapolated to a new orientation that is normal to the flux [9]. The overall kill probability, \(P_K\), of the target, to an externally detonated high explosive (HE) warhead due to fragments, is,

\[
P_K = 1 - e^{-A_V \phi}.
\] (1.1)

Equation (1.1) is explained in more detail in Appendix B.

1.2.2.2 Ray-tracing

This model traces each fragment individually through the target and assesses the damage on each component hit by the fragment. In the target description the ‘protection’ of each component is defined in terms of material thickness, minimum penetration required, etc., as well as the kill criterion which is the probability of kill, \(P_{k/h}\), for the component. Critical components are grouped into critical systems, i.e. systems that are crucial for the functioning of the target. Also in the description of each component is a kill probability, \(P_{s/k}\), that the critical system will fail given that the component fails. Each of the critical systems also has a kill probability, \(P_{t/s}\), that the target will fail if the critical system fails. By combining all of the above kill probabilities for each fragment the model, assigns a kill probability, \(P_{total/s}\), to the target failing for a specific threat.
It is clear that the target model used in the ray-tracing methodology is quite detailed. Because of this it is possible for the methodology to handle an extensive range of fragment parameters, i.e. fragment types, sizes, shapes and velocities, using the same target model.

The total kill probability can be described mathematically as,

$$P_{total/s} = \sum_s P_{t/s}(\sum_k P_{s/k}(\sum_h P_{k/h})).$$

1.2.2.3 Comparison of Methodologies

A vulnerable areas database (VADB) must be generated for each fragment size and velocity when using the vulnerable area methodology. These calculations are, however, only made for fragments that hit the target from predefined (selected) directions with interpolation being used to generate the VADB for other directions. Thus, the time-consuming process of generating a VADB must be undertaken for every range of fragment parameter values with extrapolation needed for different velocities and attack angles [10].

The target generating process for the ray-tracing process can be time-consuming. For this methodology the target model needs to be generated once, and because the path of each fragment is traced individually, no extrapolation is needed for different angles as well as different fragment velocities, and mass. This, however, causes the model on execution to be slower than the vulnerable area model for complex targets. However, as computing speed has increased, processor speed has become less of a factor. The fact that each fragment is traced individually also makes the model valid for a wide range of fragments [8].

The ray-tracing model is much more sensitive to the warhead position. This is because a single fragment has a chance of missing all the critical components. For this reason a smoothing (averaging) process is normally used on the final data. When using the vulnerable area concept, the kill probabilities of the various components are distributed over an area and the fragments are described in terms of flux. As a result the model is relatively insensitive to the warhead position, due to the kill probabilities and fragment distribution being averaged out [8].

1.2.3 International capabilities and models

Information on models (and simulation software) used in the United States of America (USA) is much more readily available than on those used in Europe. It was, however, clear that the models of the European countries are primarily based on the Ray-tracing methodology. The earlier USA models, on the other hand, are mostly based on the Vulnerable Area concept. The current trend in the USA also seems to follow the Ray-tracing methodology.

The various models that information could be obtained for will now be discussed.
1.2. Methodologies

1.2.3.1 United States of America

AJEM

AJEM is an acronym for Advanced Joint Effectiveness Model. The model is incorporated into a software package through which it is possible to analyze one or more damage mechanisms attacking a single rotary-wing or fixed-wing aircraft, or ground-mobile system [9].

AJEM is the most recent model of the USA DoD and was designed to be the standard computer simulation for evaluating the lethality and terminal effectiveness of munitions and the vulnerability of aircraft, missiles, and ground-mobile systems, including battle damage assessment and repair (BDAR) [9].

It combines elements of target model viewing, threat modelling, encounter kinematics, generation of weapon burst points, propagation of damage mechanisms to the target, damage mechanism/target interaction (penetration, fire, blast, etc.), target system relationships (functionality, redundancies, etc.), and target remaining capability or loss of function. This is accomplished by, i.e. combining the capabilities of a number of models previously used. These include COVART vulnerability/lethality model and the JSEM endgame model which the USA DoD aims to replace with AJEM [9].

AJEM is the first USA model that is based on the Ray-tracing methodology [9].

COVART

COVART is an acronym for Computation of Vulnerable Repair Time. The model predicts the ballistic vulnerability of vehicles, such as fixed- and rotary wing aircraft, given ballistic penetrator impact. COVART is based on the vulnerable area concept and evaluates the vulnerable areas of components, sets of components, systems, and total vehicle.

FASTGEN

The aim of the model is to trace the path of the shotline for a projectile through a target and develop line-of-sight data to be used by other vulnerable area models, i.e. COVART.

JSEM

The terminal effectiveness of a fragmenting munition against a target (usually airborne) is evaluated by this computer simulation program. The model is confined to the endgame scenario with the initial conditions being used, i.e. the dynamic missile orientations to the target (velocities, angles and miss distances) and fuzing time, being calculated with other simulation models.

FATEPEN

FATEPEN (Fast Air target Encounter Penetration) is a fragment penetration program used by the USA Navy, Army, and Air force for vulnerability/lethality assessments and warhead calculations.
1.2.3.2 Europe

*United Kingdom*

No detailed information could be obtained on the model used in the UK. From a lecture by Andrew Button, DSTL, on *Visualisation and Optimisation of Warhead-Target Interaction* at the European Air Defence Symposium, 2002, it could be deduced that DSTL has the capability to simulate the endgame scenario.

*Netherlands*

Using the simulation model TARVAC, TNO Prins Maurits laboratory, can model the effect of blast, fragmentation and radiation of a warhead and thus the effectiveness of a weapon as well as the vulnerability of a platform [11].

*Sweden*

No recent information could be obtained. From a 1980 comparative study between the Swedish and USA methodologies it can be assumed that the Swedish still make use of the ray-tracing methodology and have the capability to do warhead effectiveness and target vulnerability assessments [10].

1.2.3.3 South Africa

*SWEAT*

Denel Land System, Western Cape, has a fragmentation warhead, effectiveness simulation package SWEAT (Simulation of Warhead Effectiveness against Arial Targets). This model has the ability to evaluate the effectiveness of fragmentation warheads against soft-targets, i.e. including rotary-wing or fixed-wing aircraft, ships, and radar; excluding heavily armoured vehicles such as tanks.

SWEAT consists of the following computer programs:

- **TargetModeller:** This module is used to generate a target model to be used by TargetDetectPos and Upshot.
- **Warhead:** Simulates fragmentation warheads.
- **TargetDetectPos:** Determines the burst position(s) relative to the target.
- **Upshot:** Makes use of a deterministic model to calculate the kill probability of a warhead against a specific target, at a specific burst point.
- **Intercept Mattack:** This module, in conjunction with TargetDetectPos, defines the intercept path and the positions of the burst points relative to the target for a given intercept scenario.
1.2. Methodologies

SWEAT is a DOS based package requiring an in-depth knowledge of it and is relatively user unfriendly. Therefore, based on the following, a need for a single package that incorporates all the capabilities of SWEAT and provides a much more user friendly, Windows based, user interface, developed.

(1) Most modern users, especially in an industrial environment, are only familiar with a Graphics User Interface (GUI) environment.

(2) The programs in the SWEAT suite have been mutated a number of times to accommodate various needs. This causes confusion for the user as to what version to use.

(3) It is the responsibility of the user to manage the use and interaction of the various programs and the input and output files generated and required by these programs.

(4) The results are written to text files leaving the user with a cumbersome process of converting it to a more sensible and user friendly format.

This thesis documents how the above mentioned shortcomings with regard to the warhead effectiveness analysis capabilities within the South African weapons industry have been addressed.

The first phase of the study comprised of an in-depth study of the source code of the modules of which SWEAT is comprised. The theory behind the model is well documented in this thesis, a first for South Africa, reducing the vulnerability to the loss of expertise through contingency planning, this is one of the major objectives of this study. During this phase the FORTRAN source code was also modified to accommodate the development of one package that incorporates the capabilities of all of the modules of SWEAT.

This package is known as EVA (The Effectiveness and Vulnerability Analyser) and,

(1) Operates in a Windows environment.

(2) Handles the interaction of the various programs and the input and output files generated and required by these programs.

(3) Extracts the information from the result files and represent it in a user friendly format, e.g. graphs.

In the second chapter the model being employed is explained on a high and more holistic level. The reader is orientated on how all of the sub-models fit together as to generate an effectiveness estimate for a specific warhead and target. This enables the user to study each of the sub-models in context as each model is discussed in detail in the remaining chapters.

Background information on target analysis is provided in Section 3.2 in which concepts such as kill levels, technical and functional descriptions of a target and critical component analysis are discussed. Chapter 3 further expands the concepts surrounding the target model discussed in Section 2.3 and details the mathematical model employed in the TargetModeller module so as to convert the data file describing a target to the format required as input by Upshot. New
functionalities were also developed with regard to the target as a part of this study. Within the environment of EVA it is now possible to view the target graphically. A mathematical model needed to be developed so as to convert the target data provided into suitable information for EVA to do so. This enables users to visualise targets and efficiently verify the integrity of the data describing a target. Previously this was difficult and cumbersome since the user only had a data file, containing coordinates. The model is detailed in Section 3.3.1.5.

Flight paths, which describe the intercept scenarios, is covered in Chapter 4. As a foundational background propagators and guidance systems applicable to guided missiles, to which this study is confined, are discussed in Section 4.2. These are some of the factors that influence the variable flight paths of a missile. In this chapter variable flight paths and the terminology used, are also explained. Previously, flight paths needed to be generated manually. This is time consuming and limits the number as well as randomness of the intercept scenarios which are used to evaluate the effectiveness of a warhead. A new capability developed as a part of this study is the facility to generate random paths from distributions. The formulas used to generate the random variables are given in Section 4.4.1. This newly developed capability was made possible by advances in the South African weapons industry over the last decade, which has made it possible to generate distributions for the various parameters describing an intercept scenario. Other optimisation techniques, such as Monte Carlo simulations, combined with heuristic search algorithms can now be employed due to this new technology.

As mentioned earlier, this study is confined to prefragmented warheads. The damage caused by these fragments, or damage mechanisms, is due to a number of factors and more specifically the cumulative effect of these factors. The factors that influence the extent of the damage caused by a warhead are discussed in Section 5.2. Also briefly discussed in this section are some of the fragment types and their characteristics and the factors determining the choice of fragment. The generation of the output by Warhead, namely the ejection angle and the velocity for each fragment, are discussed in Section 5.3.1. The two values are respectively calculated using the Taylor equation [12] and Gurney formula [13]. The model used divides the warhead into relatively short discs, each having the length of a fragment. This is contrary to the fact that the Gurney formula describes the fragment velocities for long cylindrical warheads. To compensate, the Gurney formula used, is adapted to accommodate the end effects of cylindrical charges. This modification is briefly discussed in Section 5.3.2.

Chapter 6 discusses a completely new functionality, namely to calculate the detection position of a specific fuze design. The definition of a fuze, the purpose, and classification of various systems are discussed in Section 6.2. The mathematical model that was built to model a fuze and calculate the position on a specific path at which a target is detected is explained in Section 6.3. This new functionality supports the heuristic search algorithms used to confine the search space. Knowing the detection position is important in that any possible hits prior to detection, unless it is a direct hit, has no impact on the effectiveness analysis. As a result it forms part of the objective function used in the search algorithm.

The effectiveness model incorporates all of the above models and is discussed in Chapter 7. The method followed to calculate which vital parts are hit by which fragment and the effect this has on the probability of kill are calculated. The number of hits and probability of an event occurring at each burst point for a specific intercept scenario is thus calculated, but also averaged out to calculate the mean probability of an event occurring, the mean number of hits on the target, and the mean number of hits on each vital part the specific intercept scenario.
1.2. Methodologies

A heuristic model was developed as a part of this study to search for a warhead that has the largest possible diameter and total mass for which the momentum of the fragment will be larger than a specified minimum, given user defined constraints. This model establishes trends and supports the designer in the process of optimising the warhead design. The model and functionalities are discussed in Chapter 8.

The platform for future development with regard to the implementation of meta-heuristic search algorithms has been built as a part of this study. A summary of the contributions made to this extent as well as the possible future developments are given in Chapter 9.

To summarise the above, the aim of this study is:

- The easier transferability of this knowledge, as a result of it being comprehensively documented, making it more accessible to users and thus protecting the technology, ensuring contingency planning;

- Defining the scope of the warhead effectiveness analyses discipline;

- Verifying the methodology, and source code through which it is employed, of the model used in South Africa;

- The development of an integrated and more user friendly model through the implementation of a GUI application, and;

- The automation of the warhead design process through e.g. developing the ability to implement Monte Carlo simulations and meta-heuristic search algorithms.
Chapter 2

EVA

2.1 Introduction

Chapter 2 gives an overview of the model used for the effectiveness and vulnerability analyses. The various facets of the model are discussed on a high level to enhance understanding of the simulation model EVA. More detailed discussions on the model will follow in the subsequent chapters.

2.2 Overview

The model consists of three sub-models, namely the target, attack paths (intercept scenarios) and warhead models. These models are used as input for the fuzing and effectiveness models which form the endgame model. The results of the endgame model are used as input for the heuristic method used in the solution space confinement process. A diagram of this process can be seen in Figure 2.1.

![Modular Blocks of EVA](image)

Figure 2.1: Modular Blocks of EVA.
2.3 Target Model

EVA is only applicable to soft targets, e.g. aircraft, helicopters, missiles, etc., as opposed to hard or armoured targets such as tanks. As can be seen in Figure 2.2 a target is built up by various components, with some of these components being vital to the functioning of the target. The target with its vital or critical components is modelled by using polyhedrons and line elements. An example of a target being modelled in this fashion is illustrated in Figure 2.3.

Components are grouped into vital components and non-vital components. The vital components contribute to the possibility of the target being killed, given a hit and also “protect” the other vital components. The non-vital components only “protect” the vital components. In modelling this, each component is given certain attributes; these are listed in Table 2.1. The thickness of a surface emulates, for example, the outer skin of the component and in effect reduces the speed of the penetrating fragment.
2.3. Target Model

Table 2.1: Component attributes.

<table>
<thead>
<tr>
<th>Non-vital component</th>
<th>Vital Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>• A thickness for each surface</td>
<td>• A thickness for each surface</td>
</tr>
<tr>
<td>• Internal protection</td>
<td>• Internal protection</td>
</tr>
<tr>
<td></td>
<td>• Category of kill</td>
</tr>
<tr>
<td></td>
<td>• Kill probability</td>
</tr>
</tbody>
</table>

In the case of the non-vital components the internal protection is described by three probabilities. Each of these probabilities is associated with a grouping of non-vital components of a certain thickness. Given that a fragment hits a non-vital component, the probability that it will hit one of the non-vital components in a certain grouping, with the associated thickness that needs to be perforated, is considered. Non-vital components are thus described as polyhedrons in the case of the larger shapes, e.g. the outer body of the target, as well as an internal protection of these non-vital and vital components. The main reason for including these is to model the degradation of the velocity of a fragment penetrating a target.

For vital components the internal protection is simply described by a thickness. This thickness is used to calculate the decrease in the velocity of a penetrating fragment after perforating the outer casing of the component.

Vital components are grouped into four categories of kill. These are:

1. Soft targets (Personnel)
2. Ordinary components (e.g. hydraulics)
3. Fuel tanks
4. Wiring.

The kill probabilities for various critical components are calculated according to the category of kill that the specific component resides in; these will be discussed in Chapter 7, which covers the effectiveness model in more detail.

The critical components form the building blocks of the target. These components are used to build up the various subsystems that the target consists of; a diagram of this model can be seen in Figure 2.4.

The target model is discussed in detail in Chapter 3.
2.4 Warhead Model

The aim of the Warhead model is to predict the velocity and ejection angle of each fragment in a prefragmented warhead. Figure 2.5 shows an example of a prefragmented warhead made up from copper and tungsten fragments. These fragments are made in a certain shape with a specific weight. The warhead model is discussed in detail in Chapter 5.

2.5 Intercept Path Model

The effectiveness of the warhead is evaluated at various burst points along an intercept path. The intercept path is defined by a miss distance ($R_{min}$), cylinder angle, and direction. The
Intercept path is a representation of a specific intercept scenario between a missile and target. For each scenario an aim point on the target axis system is defined. The zero burst point \((x_{rel}, y_{rel}, z_{rel})\) on the intercept path is defined as the coordinate for which the perpendicular distance from the intercept path to the aim point is equal to the miss distance. There are a number of coordinates that satisfy this criterion. Using a cylinder angle that is taken from the point in the positive quadrant of the \(xy\)-plane the specific zero burst point is determined. Taking this point as origin, the other burst points on the path are calculated by taking equally sized steps in the positive and negative direction of the path. Figure 2.6 shows a graphical representation of an intercept path. The intercept path and statistical generation thereof are discussed further in Chapter 5.

![Intercept Path](image)

Figure 2.6: Intercept Path.

### 2.6 Fuzing System

Detonation of a warhead can be obtained in two ways. It can either be triggered by the contact fuze or by the proximity fuze. The proximity fuze detonates the warhead, if there has not been a direct hit, after some delay following detection by the target-detecting device (TDD), which will be referred to as the fuze in this study. It is important to synchronize the delay, the design of the fuze, and the warhead design. As a result, the position at which the target is detected, on a given path by a specific TDD design, needs to be determined.

A simple model was developed for EVA to simulate the fuze and determine the detection position. A cone, seen in Figure 2.7, with a specific slant height, which is the reach distance of the TDD, and top angle, is generated. The position on the path where the side of the cone first intersects one of the components describing the outer geometry of the target is then calculated. Knowing this position a time delay can be set to attempt to detonate the warhead in an effective region. The fuze model is discussed in Chapter 6.
2.7 Effectiveness Model

The effectiveness model (Upshot) evaluates the effectiveness of a warhead against a target at a specific position relative to the target. The model makes use of the ray-tracing methodology, as discussed in Section 1.2.2.2.

Upshot first calculates the probability, $P_{\text{kill}}(j)$, that the critical component $j$ will be killed as a result of the detonation of the warhead. Each of the component-system combinations in a target has a probability, $P_{\text{breakdown}}(X_{js})$, that system $s$ will breakdown, given a hit on component $j$, $X_{js}$ (Table 2.2).

![Figure 2.7: Schematical representation of the cone used to model TDD.](image)

<table>
<thead>
<tr>
<th>Vital part No</th>
<th>System</th>
<th>$P_{\text{breakdown}}(X_{js})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Pilot</td>
<td>1.0</td>
</tr>
<tr>
<td>2</td>
<td>Radar</td>
<td>0.8</td>
</tr>
<tr>
<td>3</td>
<td>Radar</td>
<td>0.5</td>
</tr>
<tr>
<td>4</td>
<td>Engine</td>
<td>0.3</td>
</tr>
</tbody>
</table>

The final result of a hit is one of four events. These are:

A. Mission aborted. Aircraft lost.
B. Mission aborted. Aircraft returns to base.
C. Mission accomplished. Aircraft lost.

The intersect of events (A and B) is Mission aborted and the intersect of events (A and C) Aircraft lost. The sample space for these events can be seen graphically in the Event diagram in Figure 2.8.
A probability, $P_e(s)$, that event $e \in \{A, B, C, D\}$ will occur, given that system $s$ fails, must be assigned to each system for the calculation of the probability of an event occurring. Table 2.3 shows these probabilities being associated with the various relationships.

**Table 2.3: Target kill probability due to system breakdown.**

<table>
<thead>
<tr>
<th>System No</th>
<th>Event</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.0</td>
<td>Pilot</td>
</tr>
<tr>
<td>2</td>
<td>–</td>
<td>Fuel Tanks</td>
</tr>
<tr>
<td>3</td>
<td>0.5</td>
<td>Radar</td>
</tr>
</tbody>
</table>

By now using Bayes theorem and all of the above probabilities, the probability that event A, B, C, or D will occur can then be calculated as follows:

$$P(event) = \sum_s \sum_j P_e(s)P_{breakdown}(X_{js})P_{kill}(j).$$

Figure 2.9 is a typical graph that shows the probability that the event of the aircraft being lost will occur for a specific intercept path and warhead design. Also shown on the graph is the number of hits and the detection position of the fuze. The left vertical axes show the kill probability, the right vertical axis the number of hits, and the horizontal axis the position on the path relative to the zero burst point.
2.8 Heuristic Models

A heuristic search algorithm was designed that functions in conjunction with the warhead module, discussed in Section 2.4. The object of this model is to find:

The warhead that has the largest possible diameter and total mass for which the momentum of the fragment will be larger than a specified minimum for a given fragment size, given the constraints.

Using the model the designer can analyse certain warhead trends related to the fragment size. This model and the functionalities associated with it are discussed in more detail in Chapter 8.

2.9 Future Developments as a result of this study

Due to various improvements and additions to the model during this study, it is now possible to make use of Monte Carlo simulations, and subsequently meta-heuristic methods to search for a good solution.
Chapter 3

The Target Model

3.1 Introduction

This chapter refers to Section 2.3 and discusses all aspects of the target modelling with regard to EVA. EVA makes use of two pre-processors to convert the input file into a suitable format for Upshot and the graphical display of the target. Upshot additionally makes use of a file describing the critical components of the target. The target analysis and the setting up of the original input and the critical file, the conversion of the input file, as well as how all the functionalities are accessed in EVA, are discussed in this chapter.

3.2 Target analysis

The first step in the modelling of a target for an effectiveness or vulnerability assessment is to identify the vital components, the components that, if lost, could lead to the target being killed. Vital components are also referred to as critical components, with the process of identifying them known as the Critical Component Analysis (CCA). According to Ball [1], whose work this section follows, components can be classified as vital either because they perform some essential function, or the manner in which the components fail causes some other critical component to fail.

A general procedure for determining the critical components has been developed. This procedure consists of the following three steps:

1. a selection of the kill levels or categories to be considered,
2. an assembly of the technical and functional description of the target,
3. determining the critical components of the target and their damage-caused failure modes for the selected kill levels.

3.2.1 Kill levels

A hit on a critical component can result in various degrees of performance degradation. There are a number of different categorisation systems that are used by the various development
communities in the world to differentiate between these degrees of performance degradation. Before the vulnerability of a target can be assessed it is essential to define these kill categories. One common categorisation is,

- attrition kill,
- mission abort kill,
- forced landing kill.

Attrition kill refers to the target being lost from inventory, and in many cases this category has sub-categories.

Mission abort kill measures the degree by which the damage inflicted on the target impedes it from completing the mission. In this case, however, damage is not enough to cause an attrition kill.

A forced landing kill occurs any time a target is forced to land due to damages before all onboard fuel is lost.

For the purposes of this study a combination of the kill levels define attrition kill and mission abort kill. These were discussed as the four events in Section 3.7, with the intersection of event types A and B being mission aboard kill and the intersection of the A and C events being attrition kill.

It is thus possible to redefine the notion of a critical component to:

“A critical component is any component which, if damaged or destroyed, would yield a definable aircraft kill level.”

### 3.2.2 Technical and Functional description of the target

When modelling a target, as much technical and functional information as possible on the target need to be obtained. As was mentioned in Section 2.3, a target is divided into various major systems, with each of these being subdivided into subsystems and components. The technical description should consist of information specifying the location, size, material, construction, and operation of all of these, in effect describing the major system. Information required by a functional description includes the functionalities of each component as well as the redundancies for the component.

In compiling a thorough assessment from which to model the target; perspective drawings, schematics, scaled three-dimensional drawings, detailed inboard drawings, and various cross-section drawings are required. Much of this information can be obtained from the manufactures of the target in the form of, e.g. maintenance manuals. Information such as descriptions of systems and components, and their functioning and relationship to the overall functioning of the target can be obtained from this source. Another useful information source is the operators of the weapon system, as well as the input of design personnel.
3.2.3 Critical component analysis

Critical components can be grouped into two categories; these are:

1. Redundant,
2. Non-redundant.

A critical component is redundant if the weapon system can still function after the component has been killed, but will fail should the component be killed in conjunction with another component. Non-redundant components will cause a system failure should the component fail. The following example explains the distinction between the two categories.

If a helicopter with two pilots is considered, and a scenario in which one of the pilots is lost is sketched, it will not necessarily result in the loss of the other pilot. If the assumption is made that the helicopter will still function with only one pilot, it follows, from the definition of a critical component, that a pilot is not a critical component. An assumption that only one hit can be expected is, however, not valid. Given that one pilot is hit, there is also a possibility that both pilots can be hit and lost. This will result in an aircraft kill. A pilot can thus be considered as a redundant component, since the loss of one pilot will not result in an aircraft kill, whilst in the second instance, with both lost, it will be considered as a critical component. A distinction can thus be made between redundant and non-redundant critical components.

Also, only a hit on one of the pilots will result in mission abort. This makes the critical component redundant in the one category and non-redundant in the other.

A similar analogy can be sketched by making use of the engine of a twin engine aircraft.

A CCA comprises five basic steps; these are:

1. Identify the essential functions to be performed by the target to achieve its goal.
2. Identify the major systems and subsystems performing the critical functions.
3. Identify the relationship between each type of individual component or subsystem failure mode and the performance of essential functions. This is known as Failure Mode and Effect Analysis (FMEA.)
4. Relate component or subsystem failure modes to damage caused in combat, known as Damage Modes and Effect Analysis (DMEA.)
5. Identify the redundant and non-redundant critical components for the selected kill level. This is done through developing a visual presentation of the list of critical components, a kill tree, and/or a logical expression, a kill expression.

A Fault Tree Analyses (FTA) is a useful tool for gaining additional insight for the identification of critical components. The CCA methodology is illustrated in Figure 3.1 and will now be discussed in more detail. A helicopter aircraft will be used as an example target in the supporting explanations of some of the concepts.
3.2.3.1 Step 1: Essential functions

An essential function can be defined as a system or subsystem needed by the target to maintain its designed capabilities. The essential function analysis should consider each phase of the mission of a target. Examples of mission phases are takeoff, climb to cruise altitude, cruise to attack area, descend to attack area, target location, ordnance delivery, exit the target area, climb to cruise altitude, return cruise, descend, and landing. Each essential function should be assigned a level of operation, e.g., lift and thrust may not be completely lost should one of the engines of a twin engine helicopter be lost, but it will most certainly lead to the reduction of the performance capabilities. This may cause the helicopter to be more vulnerable in a hostile environment and increase the possibility of an attrition kill. The form in Figure 3.2 can be used in the identification process of the essential functions and some of the mission phases of e.g., an attack helicopter. Lift and thrust will be an essential function through all of the phases, whilst the ability of locating and identifying targets will only be an essential function during the preliminary cruising phases.
3.2. Target analysis

3.2.3.2 Step 2: System Essential functions Relationships

The continued operation of the systems and subsystems that perform the essential functions is a crucial element in the ability of any target to perform the designated mission. Damage to the target may impair or stop the functioning of some subsystem components which may result in the loss of some essential functions. How rapidly these essential functions are lost will determine the designated kill level.

Each system and subsystem of the aircraft must be examined to establish the contribution made to the essential functions that had been identified. The form in Figure 3.3 relates the systems and subsystems to the essential functions to which they contribute.

3.2.3.3 Step 3: Failure mode and Effect Analysis (FMEA)

The Failure mode and Effect Analysis is a procedure that:

(1) Identifies and documents all possible failure modes of a component or subsystem.

<table>
<thead>
<tr>
<th>ITEM</th>
<th>ESSENTIAL FUNCTIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>FLIGHT:</td>
</tr>
<tr>
<td>2</td>
<td>Provide lift and thrust</td>
</tr>
<tr>
<td>3</td>
<td>MISSION:Communications</td>
</tr>
<tr>
<td>4</td>
<td>Start systems</td>
</tr>
<tr>
<td>5</td>
<td>Monitor systems</td>
</tr>
<tr>
<td>6</td>
<td>Provide air data intelligence</td>
</tr>
<tr>
<td>7</td>
<td>Maintain terrain clearance</td>
</tr>
<tr>
<td>8</td>
<td>Employ IFF/ECM</td>
</tr>
<tr>
<td>9</td>
<td>Navigate</td>
</tr>
<tr>
<td>10</td>
<td>Locate/Identify targets</td>
</tr>
<tr>
<td>11</td>
<td>Employ weapons</td>
</tr>
</tbody>
</table>

Figure 3.2: Some essential functions and mission phases for an attack helicopter [1].
CHAPTER 3. THE TARGET MODEL

Figure 3.3: Essential system-function relationships; systems compared with the same functions as shown in Figure 3.2 [1].

(2) Determines the effect of each failure mode upon the capability of the system and/or subsystem to perform its essential functions.

Some of the types of component failure considered in FMEA are:

- Premature operation,
- Failure to operate,
- Failure during operation,
- Failure to cease operation,
- Degraded or out-of-tolerance operation.

Figure 3.4 shows an example of a summary format of the FMEA for two flight control rod failure modes. Interesting to note is that the rod is a critical component in the event of it jamming, but not when it is severed. This is because in the latter scenario it can be remedied using other control surfaces.

FMEA is not only applicable to single component failures, but also to multiple. Given that combat damage causes failure of a component, there is a large probability that other components were also damaged. The consideration of multiple component failures is thus extremely important.

In the FMEA the failure of a component is assumed and the resultant effect identified. The FMEA is thus a bottom-up approach enabling the analyst to classify the component as critical or not, and if critical, as non-redundant or redundant. Contrary to this the Fault Tree Analyses (FTA), another procedure for identifying the critical components, is a top-down approach. The
3.2. Target analysis

Figure 3.4: Example of FMEA summary format [1].

The procedure starts with an undesired event, and then determines the event or combination of events that can cause the undesired event (or fault.) Figure 3.5 shows a generic example of a fault tree diagram.

FTA makes use of Boolean logic and the diagram in Figure 3.5 should be interpreted as follows. The undesired event $X$ can only occur if both events $A$ and $B$ occur, with event $A$ only occurring if event $C$ and/or event $D$ occurs, and event $B$ only occurring if event $E$ and/or event $F$ occurs.

![Fault Tree Diagram](image)

Figure 3.5: Generic fault tree diagram [1].

3.2.3.4 Step 4: Damage Modes and Effect Analysis (DMEA)

At this point in the process the critical components have been identified. The cause(s) for the failure of each critical component, however, still needs to be identified. The failure may or may not be related to combat damage. DMEA identifies the component failures that are a result of damage caused in combat, e.g. mechanical damage of components caused by fragment penetration. DMEA thus relates the potential component or subsystem failures, and other possible damage-caused failures, which have been identified with specific damage mechanisms and their damage process. These failures are then brought into relation to the selected kill levels. DMEA also looks at the second level where the possibility of any secondary hazards caused by the initial damage process is identified. An example of a DMEA matrix is given in Figure 3.6.
<table>
<thead>
<tr>
<th>Component Name</th>
<th>Component Number</th>
<th>Disablement Diag. No.</th>
<th>Damage Mode</th>
<th>&quot;Kil&quot; category</th>
<th>Non Redundant</th>
<th>Redundant</th>
<th>Remarks</th>
<th>FKh Func. No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stick</td>
<td></td>
<td></td>
<td>Break or disable</td>
<td></td>
<td></td>
<td></td>
<td>Degraded</td>
<td></td>
</tr>
<tr>
<td>Assembly</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Flight Control</td>
<td>32</td>
</tr>
<tr>
<td>(Grp) 3001</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gas Sensor</td>
<td>3002</td>
<td>2</td>
<td>Loss of electrical</td>
<td>Loss of cas, pitch and roll control</td>
<td>X, X</td>
<td></td>
<td>Control through det</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Loss of cas</td>
<td></td>
<td></td>
<td></td>
<td>Reversion to Mech. (if fuel is lost) (Det = Direct)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Loss of electrical and Mechanical</td>
<td></td>
<td>X, X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>linkages</td>
<td>Electrical Link</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rudder Pedals</td>
<td>3006</td>
<td></td>
<td>Break or disable</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Arms</td>
<td>3007</td>
<td>3</td>
<td>one arm</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>32</td>
</tr>
<tr>
<td>Support</td>
<td>3008</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>32</td>
</tr>
<tr>
<td>Feel Spring Support</td>
<td>3301</td>
<td></td>
<td>Break or disable</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>32</td>
</tr>
<tr>
<td>Spring</td>
<td>3302</td>
<td></td>
<td>Break or disable</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>32</td>
</tr>
<tr>
<td>Transducer</td>
<td>3303</td>
<td></td>
<td>Spring essy, or</td>
<td></td>
<td>X, X</td>
<td></td>
<td>inputs to rudders</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>transducer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 3.6: DMEA Matrix [1].
3.2.3.5 Step 5: Kill trees and Kill Expressions

The result of the analysis of a particular target, at this point, is a list of critical components for a specific operational mode and selected kill level. Each of these critical components either makes a singular contribution to an essential function, or is one of two or more redundant components possibly making the necessary contribution to the function.

The kill tree provides the analyst with a visual illustration of the critical component and the component redundancy. Furthering the example explaining the concept of redundancy in Section 3.2.3, Figure 3.7 shows a kill tree for a two-engine, two-pilot helicopter. From the kill tree it is clear that a complete cut through the tree trunk is required to inflict a kill, e.g. both the pilot and co-pilot controls need to be lost to cause a target kill.

![Figure 3.7: Example kill tree for a two-engine, two-pilot helicopter [1].](image)

The relationship between a target kill and the loss of a component can also be expressed making use of a logical expression, known as a kill expression. An example, using the kill tree in Figure 3.7, would be: 

\[
\text{[(Pilot OR Pilot Controls) AND (Co-Pilot OR Co-Pilot Controls)] OR (Engine 1 AND Engine 2) OR (Drive Train) OR, etc [1].}
\]

3.3 The Target Model

The model used to summarise the data gathered in the above analysis and as input for the simulation model will depend on the specific simulation methodology to be used. Since this study focuses on the ray-tracing methodology (Section 1.2.2.2), only the target model used in this methodology will be discussed.

Ray-tracing determines the combined effect of the individual fragment hits on the critical components. This is done by evaluating the penetration depth, crater diameter, etc. Given this, it is clear that factors, such as the momentum and direction of each fragment, become important in
the calculation of the effectiveness. These factors, in turn, are affected by non-vital components which might possibly change the velocity of the fragment by decreasing the speed or affecting the direction. Information on the non-vital components must thus also be included in the target model. Table 2.1 lists the attributes of the vital and non-vital components, with each of these being explained in Section 2.3.

The information for the attributes, together with the coordinates describing the shape of the various components and their relative positioning with regard to the other components, are summarised in a text file. The format of this file can be seen in Appendix A.1.1, which is supported by an example of a modelled target in Appendix A.1.2. The reader should note that each surface is described by the coordinates of only three points. This follows from the fact that one and only one plane can pass through the same three points. Since the methodology only requires the various surfaces to be described as planes, this information is more than sufficient to calculate the detection position of a certain fuze design (Chapter 6), or whether or not a fragment hits a specific component (Chapter 7).

Another important aspect to note is that the “front” of a target is always found at the origin of the Cartesian axes, (0, 0, 0), with the positive x-axis forming the centre line of the geometric body as a whole. Figure 3.8 illustrates this concept more clearly.

Figure 3.8: Target orientation in Cartesian coordinates.

As was discussed in the target analysis, the relationship between the various critical components, the system(s) affected by a hit on a critical component, and the associated kill level of this hit, is of extreme importance. These relationships were briefly illustrated in Table 2.2 and Table 2.3. These tables also illustrate the concept of each component being associated with one or more system(s) and a probability that the system(s) will fail given a hit on the component. Furthermore, each system has a probability that a certain event (see Section 2.7) will occur, should the system fail. This information is required by the effectiveness model and summarised in a text file. The format of this file can be seen in Appendix A.2.1, which is supported by an example in Appendix A.2.2. The reader must note that the probabilities that a certain event will occur, given system failure, is listed from left to right, with the left referring to event A and the right-most to event D.
3.3.1 Target file format conversions

3.3.1.1 The conversion for Upshot

The simulation model, Upshot, needs the information contained in the geometric description file in a different format. The additional information that needs to be extracted from the file is the following:

- An internal point in each polyhedron.
- The maximum distance from this internal point to any corner of the polyhedron. The radius for each polyhedron.
- The corners in each plane are replaced by the normal vector to the plane. This vector describes the inclination of the plane.

The coordinates of the corners of each polyhedron are not required by Upshot and as a result are discarded from the Upshot input file, see Appendix A.3.1 for the file format and Appendix A.3.2 for an example. The coordinate values are, however, important in calculating the internal points, radii, and normal vectors. This will be discussed in the following sections.

3.3.1.2 An internal point in the polyhedron

By taking the average $x$, $y$, and $z$ coordinates of the corners of the polyhedron as the internal point it ensures that the $x$, $y$, and $z$ coordinates of the internal point will be larger than...
or equal to the smallest, and less than or equal to the largest $x$, $y$, and $z$ coordinates respectively.

The calculations for this are as follows:

If 

$$c_i = (x_c, y_c, z_c),$$

is a point inside the polyhedron, and

$$p_1 = (x_1, y_1, z_1), p_2 = (x_2, y_2, z_2), ..., p_{n-1} = (x_{n-1}, y_{n-1}, z_{n-1}), p_n = (x_n, y_n, z_n)$$

are the points that form the corners of the polyhedron, then

$$x_c = \left( \sum_{j=1}^{n} x_j \right)/n,$$

$$y_c = \left( \sum_{j=1}^{n} y_j \right)/n,$$

$$z_c = \left( \sum_{j=1}^{n} z_j \right)/n.$$

### 3.3.1.3 The maximum distance from the internal point to a corner of the respective polyhedron

If the distance from point $c$ to point $p_j$ is

$$d_j = \sqrt{(x_j - x_c)^2 + (y_j - y_c)^2 + (z_j - z_c)^2},$$

then the maximum distance from the internal point to a corner is

$$d_{\text{max}} = \max(d_1, d_2, ..., d_{n-1}, d_n).$$

### 3.3.1.4 The normal vectors of each surface

The vector between any two points that lies within a plane, will also lie in the plane. Since three points within each side of the polyhedron are known, two independent vectors lying within the plane can be calculated. Taking the cross product of these two vectors a vector that is normal to the plane is calculated. Upshot requires the direction of this vector to point to the outside of the polyhedron. The direction of the normal vector must thus be determined, and changed if necessary. The dot product of any two vectors that have an angle of less than $90^\circ$ between them will always be positive, see Figure 3.10. By taking the dot product of the normal vector to the plane and a vector that reaches from the internal point of the polyhedron to any one of the known points in the plane, the general direction of normal vector can be determined. Should the dot product be greater than zero, it follows that $\theta$ is less than $90^\circ$, the direction of the normal vector can be changed by multiplying it by $-1$.

The above is calculated as follows:
3.3. The Target Model

Figure 3.10: Direction of normal vector: \( \cos \theta > 0 \) if \( \theta < 90^\circ \).

If \( \mathbf{n} = (x_n, y_n, z_n) \) is the normal vector of a surface of the polyhedron and it is given that \( p_q = (x_q, y_q, z_q) \), \( p_r = (x_r, y_r, z_r) \) and \( p_s = (x_s, y_s, z_s) \) are three points in the plane, then

\[
\mathbf{a} = (x_a, y_a, z_a) = ((x_q - x_r), (y_q - y_r), (z_q - z_r))
\]

and

\[
\mathbf{b} = (x_b, y_b, z_b) = ((x_s - x_r), (y_s - y_r), (z_s - z_r))
\]

are two vectors lying in this surface.

Since the cross product of any two vectors yields a vector that is orthogonal to both of these vectors, if follows that a normal vector of the surface is

\[
\mathbf{n} = \mathbf{a} \times \mathbf{b} = (y_a z_b - z_a y_b, z_a x_b - x_a z_b, x_a y_b - y_a x_b),
\]

with a length of

\[
||\mathbf{n}|| = \sqrt{x_n^2 + y_n^2 + z_n^2}.
\]

Thus the surface has a unit normal vector

\[
\mathbf{u} = (x_u, y_u, z_u) = \left( \frac{x_n}{||\mathbf{n}||}, \frac{y_n}{||\mathbf{n}||}, \frac{z_n}{||\mathbf{n}||} \right).
\]

The unit normal vector, \( \mathbf{u} \), might possibly be directed into the polyhedron. The first step in testing whether or not this is the case is to calculate a vector \( \mathbf{E} \), which is a vector from the internal point to a corner of the surface. Let \( p_q \), one of the points in the plane, be a corner of the surface, then

\[
\mathbf{E} = (E_x, E_y, E_z) = (x_q - x_c, y_q - y_c, z_q - z_c).
\]

If

\[
x_u \times E_x + y_u \times E_y + z_u \times E_z \geq 0,
\]

then the direction of the normal unit vector is to the outside of the polyhedron. If this is not the case, then the direction must be changed by \( 180^\circ \) to be in the right format for Upshot. This is done by multiplying the unit normal vector by \(-1\)

\[
x_u \times (-1) = -x_u
\]
\[ y_u \times (-1) = -y_u \]
\[ z_u \times (-1) = -z_u. \]

### The distance from the internal point to each surface

The distance, \( d \), can be obtained by taking the dot product of a vector from the internal point to the surface and the normal unit vector of the surface, see Figure 3.11. This will give the orthogonal distance from the internal point to the surface. Thus,

\[ d = |E \cdot u| = |E_x \times x_u + E_y \times y_u + E_z \times z_u|. \]

![Figure 3.11: Projecting normal component of vector.](image)

#### 3.3.1.5 The conversion to view the target graphically in EVA

Some of the original target information, e.g. kill level, are unnecessary to be able to display the target graphically. Other information, e.g. all the corner points of each surface and the order of these points, that is imbedded in the geometric description file, need to be extracted. The procedure followed by EVA, when displaying the target graphically, is to draw a line from the first coordinate to the second, from the second to the third, \ldots, from the second to last to the last, and finally from the last coordinate to the first. This is done for each surface of each polyhedron.
3.3. The Target Model

The following information contained in the geometric description file is disregarded in the graphical display file:

- The internal structure for each of the non-vital components.
- The category for the kill criteria of each vital component and line element.
- The kill probability of each vital component and line element.
- The internal protection of each vital component and line element.
- The diameter of each line element.
- The outer protection of each line element.
- The thickness of all surfaces.

The format of the graphical input file for EVA is given in Appendix A.4.1 and supported by an example in Appendix A.4.2.

3.3.1.6 Determine all the points of a surface

The determinant of any three points in a plane will always equal zero. The original target description file contains the coordinates of all the corner points of each polyhedron and also associates three of these points to each of the surfaces. By selecting any two of these points and calculating the determinant with each of the remaining points, it follows that all of the remaining corner points that result in a determinant of zero also lie in the plane. Mathematically this is described as follows.

Given that the plane,

\[ c_1 x + c_2 y + c_3 z + c_4 = 0, \]

in 3-space passes through three noncollinear points \((x_1, y_1, z_1), (x_2, y_2, z_2)\) and \((x_3, y_3, z_3)\), the determinant equation will be

\[
\begin{vmatrix}
  x_1 & y_1 & z_1 \\
  x_2 & y_2 & z_2 \\
  x_3 & y_3 & z_3 \\
\end{vmatrix} = 0.
\]

3.3.1.7 Ordering the points of a surface

The points in the plane can now be projected onto the \(xy\)-plane. This is done by calculating the direction cosines and transforming the coordinates of the points by making use of rotation. The angle of each of these corner points can then be calculated by making use of the inverse tan function. Before this function can, however, be applied, the corner points first need to be distributed around the origin of the \(xy\)-plane. This distribution need to be as symmetric as possible and thus the points are moved so that the average \((x, y)\) coordinate coincide with the origin of the \(xy\)-plane. By ordering the corner sizes in ascending order, the corner points in a surface will be in the order prescribed in the file format.
Mathematically the process looks as follows. Given that three of the corner points in a plane are \((x_1, y_1, z_1), (x_2, y_2, z_2)\) and \((x_3, y_3, z_3)\), and that the \(x\), \(y\), and \(z\) direction cosines are \((l_x, m_x, n_x)\), \((l_y, m_y, n_y)\) and \((l_z, m_z, n_z)\) respectively, the direction cosines \((l_z, m_z, n_z)\) can be calculated by taking the cross product of two vectors in the plane, yielding

\[
\begin{align*}
x &= (y_2 - y_3) \times (z_1 - z_3) - (y_1 - y_3) \times (z_2 - z_3), \\
y &= (z_2 - z_3) \times (x_1 - x_3) - (x_2 - x_3) \times (z_1 - z_3), \\
z &= (x_2 - x_3) \times (y_1 - y_3) - (y_2 - y_3) \times (x_1 - x_3).
\end{align*}
\]

Then calculating the length of this vector that is normal to the \(xy\)-plane,

\[d_z = \sqrt{x^2 + y^2 + z^2},\]

and dividing the vector by its length,

\[
\begin{align*}
l_z &= \frac{x}{d_z}, \\
m_z &= \frac{y}{d_z}, \\
n_z &= \frac{z}{d_z}.
\end{align*}
\]

The \((l_x, m_x, n_x)\) direction cosines can now be calculated by taking any vector in the plane and dividing the vector by its length. Thus,

\[
\begin{align*}
l_x &= \frac{x_1 - x_2}{d_x}, \\
m_x &= \frac{y_1 - y_2}{d_x}, \\
n_x &= \frac{z_2 - z_1}{d_x},
\end{align*}
\]

where,

\[d_x = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}.\]

Finally the \((l_y, m_y, n_y)\) direction cosines are determined by taking the cross-product of the \(x\)- and the \(y\)-direction cosines, yielding

\[
\begin{align*}
l_y &= (m_z \times n_x) - (n_z \times m_x), \\
m_y &= (n_z \times l_x) - (l_z \times n_x), \\
n_y &= (l_z \times m_x) - (m_z \times l_x).
\end{align*}
\]

The coordinates, \((x_1, y_1, z_1), \ldots, (x_i, y_i, z_i), \ldots, (x_n, y_n, z_n)\), can now be transformed by using pure rotation. If \(pr_i = (x_{pri}, y_{pri}, 0)\) is the projection of point \(i\) in the plane, then

\[
x_{pri} = l_x \times x_i + m_x \times y_i + n_x \times z_i
\]

and

\[
y_{pri} = l_y \times x_i + m_y \times y_i + n_y \times z_i.
\]

The average \(x\) and \(y\) coordinates of the projected points are then calculated as

\[
x_{avg} = \frac{\sum_{i=1}^{n} x_{pri}}{n}
\]

and

\[
y_{avg} = \frac{\sum_{i=1}^{n} y_{pri}}{n}.
\]
When calculating each of the angles between the projected points and the positive x-axes, the points must first be translated so that the origin of the \(xy\)-plane and the average \(x\) and \(y\) coordinates of these points coincide. If \(t_i = (x_{ti}, y_{ti}, 0)\) is the translated-projection of point \(i\) in the \(xy\)-plane, then

\[
x_{ti} = x_{pi} - x_{avg}
\]

and

\[
y_{ti} = y_{pi} - y_{avg}.
\]

The angle of point \(i\) can now be calculated as

\[
a_i = \tan^{-1} \frac{y_{ti}}{x_{ti}}.
\]

The corner points are sorted in ascending order implementing the following algorithm.

1. for \(i \leftarrow 1\) to \((n - 1)\) do
2. \(key \leftarrow i\)
3. for \(j \leftarrow (i + 1)\) to \(n\) do
4. if \(a_j < a_{key}\) then
5. \(key \leftarrow j\)
6. \(k \leftarrow a_{key}\)
7. \(a_{key} \leftarrow a_i\)
8. \(a_i \leftarrow k\)

There are also other algorithms such as the Insertion sort, Merge sort, Heapsort, Quicksort, Counting sort, Radix sort, and Bucket sort. These have more efficient running times. However, since each side have no more than eight corner points, and the graphical display file for each target is only created once for each target, this is negligible for all practical purposes.

### 3.3.2 EVA

The target module has various functionalities within EVA. These are the creation and deletion of a target, as well as viewing the target graphically and viewing the original input file. These functionalities can be found on the target dropdown menu, see Figure 3.12.

#### 3.3.2.1 Create a new target

After a target has been modelled the target must be generated in EVA. This is done by naming a target and linking it to the appropriate critical description file (Input File (Critical)) and description file (Input File). Names for the output file to be used in Upshot (Output File) and the file used to display the target graphically (Target Graphic File) must also be given. This is done on the Create New Target form, Figure 3.13.

On creation of the target, the specified files are created from the input files and the locations of these files are stored in the database, Tree of Knowledge, used by EVA.
3.3.2.2 Delete an existing target

By selecting a target from the list of created targets in the Target deletion form, Figure 3.14a, and confirming the action, Figure 3.14b, a target will be deleted from the target list in EVA. In the event of a target being deleted the files that were created, namely the output file to be used in Upshot and the file used to display the target graphically, are deleted, as well as the target information stored in the database used by EVA.

3.3.2.3 View a target graphically

On selecting a target from the list of available targets in the View Target window, the target is displayed graphically, as can be seen in Figure 3.15. The graphical display is generated by EVA from the graphical display file that was created when the target was imported.

Once a specific target has been selected, the View Target functionality allows the user to perform various actions on this graphical display.

**Zoom action:** By selecting the magnifying glass buttons the user can zoom in and out of the target.

**Orientation action:** By selecting the orientation and centre adjusting arrows the orientation from which the target is viewed can be selected.

**Specific type:** The user can choose to view only the critical (red and green) or non-critical (blue) components or both.

**Specific part:** Each component can also be viewed individually.
3.3. The Target Model

3.3.2.4 View Target Input file

EVA allows the user to view and edit the original input file, see Figure 3.16. On saving any changes to the input file, the output file to be used in Upshot and the file used to display the target graphically will be re-created, and overwrite the previous files.

Should the target of which the input file is being changed, also be displayed graphically at that time, the changes are immediately applied for the user to view in the View Target window.

3.3.2.5 Improvements on SWEAT

In SWEAT the correct construction of a target model was a tedious task and errors were not easily identified as the person responsible for modelling a target was restricted to looking at the coordinates in the input file. EVA now creates the facility to stay within one environment
The power of the improvements, with regard to the target, discussed in this chapter lies within the fact that the target modeller can quickly pick up any mistakes made with the coordinates by looking at the target model graphically. Should something look suspect, the specific polyhedron can be identified by making use of the Specific part action. Once the specific polyhedron has been identified the user can make corrections and then view the effect of any changes in real time by using the View Target and View Input File functionalities in conjunction with one another.
3.3. The Target Model

Figure 3.16: Original Input File Window.
Chapter 4

Intercept Paths

4.1 Introduction

This chapter refers to Section 2.5 and in it all aspects of the intercept paths with regard to EVA are discussed. The aspects of the weapon system which influence the characteristics of the intercept scenarios are discussed in the section on propagators. These include a discussion on various platforms, propagators, and guidance systems which are applicable to Surface-to-Air Missiles and Air-to-Air Missiles. Another factor which influences the intercept scenario is the flight path, present paths and in particular variable paths are discussed in this chapter in this regard. The second last section of this chapter discusses how EVA generates random intercept scenarios, with the preceding section giving a brief overview of the terminology applicable to intercept scenarios. The chapter concludes discussing how these functionalities are accessed in EVA.

4.2 The Propagator

The intercept scenario is a function of the target and threat type. As was mentioned in Section 3.4, EVA focuses on pre-fragmented warheads. This is classified as a terminal threat where the important factors to consider with regard to intercept scenario, include the platform, propagator, and guidance system that make up the threat. A pre-fragmented warhead will be a subcomponent of a Surface-to-Air Missile (SAM) or an Air-to-Air Missile (AAM), with these two types having a missile as propagator. Missiles can either have a Surface Launcher or Airborne Interceptor as platform. Figure 4.1 shows where the threat type that is considered in this study fits into the general landscape of threat types. The specific components that are applicable to SAMs and AAMs are emphasised in Figure 4.1 and will be discussed in more detail [1].

4.2.1 Platforms

Surface Launcher: This platform comprises the launch and guidance equipment for SAMs. The primary purpose is thus to firstly launch SAMs and then guide the missile to an intercept point. Surface launchers vary greatly in size. The platform classification applies to platforms ranging from a single handheld launch tube to a semi-permanent complex (trailers, vans, launch units).
Tracking capabilities may include optical, as well as radar, target tracking systems. This technology is used in conjunction with the missile tracking and guidance computer which will be discussed later [1].

Airborne interceptor (AI): This platform is an aircraft, fixed or rotary wing, with the design and mission objective to engage and destroy airborne targets. AAMs is one of the weapon systems employed in achieving this objective. Other systems that complement the AAMs are air-to-air guns and the associated equipment for identifying, tracking, and firing the weapons.

![Threat Types Diagram](image)

Figure 4.1: Threat Types [1].

4.2.2 Propagators

Guided Missiles: A guided missile can be defined as an aerospace vehicle, with varying guidance capabilities, that is self-propelled through space for the purpose of inflicting damage on a designated target. Contrary to missiles, self-propelled aerospace vehicles with the purpose of inflicting damage on a designated target that is not guided, are known as rockets. In this study only missiles will be considered. Guided missiles can broadly be configured as having sensors, a warhead, control surfaces, a guidance section, and a propulsion section; the design of the missile is discussed in more detail in Appendix C.

As was briefly touched on, the two types of guided missiles posing a threat to aerial targets are SAMs and AAMs, the latter is also referred to as an air-to-air-intercept missile (AIM.)

The weight constraints result in AAMs typically employing a homing device for guidance. A
second result of the weight constraint is that the warhead itself is relatively small. The reasons for these weight constraints are the fact that the AAM is wing mounted, as well as to increase the speed and manoeuvrability of the AI platform.

With SAMs, on the other hand, weight is much less of a problem and often an advantage. SAMs are thus typically much larger than AAMs resulting in much larger warheads and ranges. SAMs theoretically have no space constraints, and therefore the guidance systems used can be more complex and bigger than with AAMs.

The various guidance systems will now be discussed.

### 4.2.3 Guidance Systems

![Missile guidance phases](image)

Figure 4.2: Missile guidance phases [14].

The section in a guided missile that is dedicated to guiding the missile on a course that will intercept the target is known as the guidance package. It is generally agreed that the missile guidance can be divided in three phases, see Figure 4.2. These are,

**Boost (Launch):** The period starting just after the missile has left the launcher and ending when all the fuel is burnt. Guidance may be active during this phase.

**Midcourse:** This phase comprises the largest part of the engagement process since the missile typically spends the longest time and travels the largest distance, compared to the other two phases, during midcourse.
Guidance may be used to bring the missile on the desired course, as well as keep it on that course.

Terminal: This is the last phase of guidance and includes the endgame scenario. High accuracy and fast reaction on the part of the guidance system is critical during this phase to ensure that the missile intercepts the target. During this phase the motor may be burnt out.

There are five main types of guidance, namely command, beam-rider, homing, retransmission, and navigation guidance.

**Command guidance:** The primary characteristic of this type of guidance is that the guidance instructions come from a source external to the missile. It follows that this type of threat requires a tracking system that is external to the missile and tracks both the missile and target. It is thus applicable to surface launchers and SAMs, and mostly focuses on short-range missile systems due to the relatively large tracking errors occurring should the range be long. Technology implemented in accomplishing the tracking are; radar, optical, laser, or infrared systems.

The basic principle behind this type of guidance is to determine the position and velocity of both the missile and target. Then computing the flight path the missile should follow to intercept the target. Next, the computed flight path and the predicted flight path of the missile are compared and the corrections required, for the missile to follow the computed path, determined.
The correction signals, also known as command guidance, are sent to the missile resulting in a change in the positioning of the control surfaces and, effectively, the flight path of the missile. Other information that might be sent is the fuze arming and warhead detonation. This guidance method is illustrated in Figure 4.3.

![Beam rider guidance](image)

**Beam-rider guidance:** The name follows from the fact that the missile constantly seeks the centre of the electromagnetic beam, transmitted from an offboard tracking system, and never "sees" the target. The correctional signals sent to the control surfaces are thus calculated based on the position of the missile relative to the centre of the tracking beam [14].

This type of guidance is executed by one of two systems. In the first case the missile rides the beam that tracks the target directly. In the second case, this missile rides a beam separate from the one tracking the target directly. This beam is directed towards a position, calculated by a computer, where the missile and the target are predicted to collide [15].

From an intercept perspective, the tracking beam must be relatively narrow to insure an intercept. The chances of the target losing the missile through manoeuvring and evasive actions are thus increased. This type of guidance is restricted to short-range missiles due to the tracking error becoming a factor with long ranges [15]. This guidance method is illustrated in Figure 4.4.

**Homing guidance:** This guidance system is located within the missile. It locates the target and calculates the command guidance, thus the missiles guides “itself” to the target. One of the advantages of this system is that the tracking error is reduced as the missile nears the intercept point. Homing guidance can be sub-divided into three major types; these are semi-active,
active, and passive systems [1]. These systems are illustrated in Figure 4.5 and will now be discussed briefly.

Semi-active: For this type of homing, the target is illuminated by a signal generated by tracking radar at a control point. It is important that this control point is separate from the missile. The missile itself is equipped with a radar receiver as part of the on board guidance system. This on board guidance system calculates the correction signals that are sent to the control surfaces using the data gathered by the radar receiver from the radar energy reflected off the target. Even though a radar energy source was used in this example, other energy sources, e.g. reflected laser, can also be used. With this type of system the target will “know” that it is being tracked, but not where the missile is, or whether or not there is a missile coming at all [15].

Active: In this instance the transmitter, as well as the receiver, is contained in the weapon. As a result it is said that this type of tracking makes use of a monostatic seeker, since the transmitted and reflected signal has the same angle with respect to the line of sight between the missile and the target. It is thus basically the same as the semi-active tracking with the difference being that the missile transmits, as well as receives the tracking energy. Due to weight and size constraints the output of the transmitter is restricted to high frequencies and low power output, resulting in a short seek and acquisition range. Because of the preceding, a missile implementing active homing is often called a “fire-and-forget” missile [15].

Passive: The passive homing system is characterised by the fact that the target is the only source of tracking energy. The energy can range from electromagnetic emissions to natural reflections from the target. As was the case with the active and semi-active homing systems the missile calculates its own correctional signals from the information obtained from the tracking energy. This type of guidance has its advantages, as well as disadvantages. The advantage
4.2. The Propagator

being that it reduces the problem of counter detection, whilst one disadvantage is that it is much more susceptible to counter measures [15].

Figure 4.6: Retransmission (TVM) guidance [14].

Retransmission guidance: This guidance technique is also known as track-via-missile (TVM). It can be described as a mix of control guidance and semi-active homing. As can be seen in Figure 4.6 the target is tracked by external radar, whilst the reflected signal is intercepted by the onboard receiver of the missile. One difference that distinguishes retransmission guidance from the homing guidance, already discussed, is that the missile has no onboard computer to calculate the corrective signals. The data are transmitted to the launch platform where the corrective signals are calculated and retransmitted to the missile [1].

Navigation guidance: There are various guidance systems in this guidance family, e.g. inertial, ranging, celestial, digital scene matching, and geophysical navigation guidance. The common trait of all these guidance systems is that it has a fixed position to which it guides [14]. It is thus typically used in the midcourse phase of the flight path. Since for the purposes of this study the interest is more focused on how the guidance system influences the endgame scenario; navigation guidance will not be discussed in more detail.

Composite guidance systems: From the remarks made, specifically in the brief discussion on Navigation guidance, it can be deduced that there is not a specific guidance system that is best suited for all three of the various guidance phases. The complete guidance system of a specific missile typically comprises two or three guidance systems, with the one dominating
the first phase or two and the other the latter phases [15]. Table 4.1 lists this for various missiles.

### 4.3 Flight paths

The guidance system plays a large role in the final intercept scenario. Flight paths can be grouped in two main groupings; namely preset flight paths and variable flight paths [15].

For preset paths the missile path is not influenced by any movements of the target after the missile has been launched. Variable paths, on the other hand, continuously takes cognisance of the position of the target; the direction of the missile is thus a function of the position and velocity of the target [15].

Due to the objective of EVA this study will focus on variable flight paths. The basic methodology of variable flight paths is as follows. The position and velocity of the target is continuously tracked. Assuming that the flight path of the target will remain unchanged until the next measurement of the parameters, the flight path of the missile is calculated and the control signals generated. There are four main types of variable flight paths. These are pursuit, constant-bearing, proportional navigation, and line of sight [15].

**Pursuit:** This is the most elementary of the variable flight paths as it involves always pointing the missile towards the target. In other words the missile follows the line-of-sight and it follows that the rate of turn must always equal the rate of turn of the line-of-sight. It is clear that for the missile to follow this method close to the target high manœuvrevability is essential. Most missiles are not capable of this level of manœuvrevability. In the event of manœuvrevability being too low, the algorithm followed is to continue to turn at the maximum rate until the manœuvrevability of the missile is sufficient again [15].

Typically this method is employed against slow targets, where the missile is launched from the rear of the target, or head on, directly towards the incoming target. Missiles employing this method typically end up in a tail chase scenario. As a result the missile must be faster than the target [15].

**Constant-bearing:** The objective is to keep the missile on a trajectory heading to a point just ahead of the target. This point, given the current information, will simultaneously be intercepted by both the target and missile. The missile path is as linear as is allowed by external forces, e.g. gravity and aerodynamic forces. From the nature of the method it follows, should any of the input parameters, such as the direction or the velocity of the target, change, that the flight path of the missile needs to be recomputed. Sufficient data-gathering and processing capabilities are thus required for this method. This is probably the major draw back of constant-bearing [15].

**Proportional navigation:** This method is typical of the more advanced homing missiles. The steering commands are generated, using the rate of change of the line-of-sight (LOS) as input parameter. The method employs a navigation ratio to calculate the rate of turn of the missile, e.g. for a ratio of 2:1 the rate of turn of the missile will be twice that of the target. The rate of turn of the missile is thus a fixed or variable multiple of the rate of turn of the target. Typically this ratio will be variable with the ratio being less than 1:1 in the early stages of flight to, for example, increase the range and as it nears the target it will increase to 2:1, 3:1, etc. This
<table>
<thead>
<tr>
<th>Missile</th>
<th>Launch Terminal</th>
<th>Mission</th>
<th>Guidance Type</th>
<th>Sensor Type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Midcourse</td>
<td>Terminal</td>
</tr>
<tr>
<td>Nike Hercules</td>
<td>Surf-to-Air</td>
<td>Anti-Air</td>
<td>Command Guidance</td>
<td>Command Guidance</td>
</tr>
<tr>
<td>Terrier</td>
<td>Surf-to-Air</td>
<td>Anti-Air</td>
<td>Beam Rider</td>
<td>Beam Rider</td>
</tr>
<tr>
<td>Sidewinder</td>
<td>Air-to-Air</td>
<td>Anti-Air</td>
<td>Passive</td>
<td>Passive</td>
</tr>
<tr>
<td>Sparrow</td>
<td>Air-to-Air</td>
<td>Anti-Air</td>
<td>Semi-Active</td>
<td>Semi-Active</td>
</tr>
<tr>
<td>Shrike</td>
<td>Air-to-Surface</td>
<td>Anti-Air</td>
<td>Inertial</td>
<td>Passive</td>
</tr>
<tr>
<td>Agile</td>
<td>Air-to-Air</td>
<td>Anti-Air</td>
<td>Passive</td>
<td>Passive</td>
</tr>
<tr>
<td>Standard Missile</td>
<td>Surf-to-Air</td>
<td>Anti-Air</td>
<td>Command Guided</td>
<td>Semi-Active</td>
</tr>
<tr>
<td>Hawk</td>
<td>Surf-to-Air</td>
<td>Anti-Air</td>
<td>Semi-Active</td>
<td>Semi-Active</td>
</tr>
<tr>
<td>Phoenix</td>
<td>Air-to-Air</td>
<td>Anti-Air</td>
<td>Inertial/Semi-Active</td>
<td>Inertial/Semi-Active</td>
</tr>
<tr>
<td>Harm</td>
<td>Air-to-Surface</td>
<td>Anti-Radar</td>
<td>Passive</td>
<td>Passive</td>
</tr>
<tr>
<td>ASROC</td>
<td>Surf-to-Surface</td>
<td>Anti-Submarine</td>
<td>Inertial</td>
<td>Active</td>
</tr>
<tr>
<td>Walleye</td>
<td>Air-to-Surface</td>
<td>Anti-Surface</td>
<td>Passive</td>
<td>Passive</td>
</tr>
<tr>
<td>Maverick</td>
<td>Air-to-Surface</td>
<td>Anti-Surface</td>
<td>Passive</td>
<td>Passive</td>
</tr>
<tr>
<td>Maverick</td>
<td>Air-to-Surface</td>
<td>Anti-Surface</td>
<td>Passive</td>
<td>Passive</td>
</tr>
<tr>
<td>Maverick</td>
<td>Air-to-Surface</td>
<td>Anti-Surface</td>
<td>Semi-Active</td>
<td>Semi-Active</td>
</tr>
<tr>
<td>Tomahawk</td>
<td>Surf-to-Surface</td>
<td>Anti-Surface</td>
<td>Inertial</td>
<td>Passive</td>
</tr>
<tr>
<td>Harpoon</td>
<td>Air/Ship/Sub-to-</td>
<td>Anti-Ship</td>
<td>Inertial</td>
<td>Active</td>
</tr>
<tr>
<td>Penguin</td>
<td>Ship-to-Surface</td>
<td></td>
<td>Active</td>
<td>Passive</td>
</tr>
<tr>
<td>Stinger</td>
<td>Surf-to-Air</td>
<td>Anti-Air</td>
<td>Passive</td>
<td>Passive</td>
</tr>
<tr>
<td>RAM</td>
<td>Surf-to-Air</td>
<td>Anti-Air</td>
<td>Passive RF</td>
<td>Passive IR</td>
</tr>
<tr>
<td>Patriot</td>
<td>Surf-to-Air</td>
<td>Anti-Air</td>
<td>Command</td>
<td>TFM</td>
</tr>
<tr>
<td>SLAM</td>
<td>Air-to-Surface</td>
<td>Anti-Surface</td>
<td>GPS Aided Inertial</td>
<td>Passive</td>
</tr>
<tr>
<td>AMRAAM</td>
<td>Air-to-Air</td>
<td>Anti-Air</td>
<td>Inertial</td>
<td>Active</td>
</tr>
<tr>
<td>JSOW</td>
<td>Air-to-Surface</td>
<td>Anti-Surface</td>
<td>GPS Aided Inertial</td>
<td>Inertial</td>
</tr>
<tr>
<td>JDAM</td>
<td>Air-to-Surface</td>
<td>Anti-Surface</td>
<td>GPS Aided Inertial</td>
<td>Inertial</td>
</tr>
<tr>
<td>BAT</td>
<td>Air-to-Surface</td>
<td>Anti-Surface</td>
<td>Inertial</td>
<td>Passive Acoustic/IR</td>
</tr>
<tr>
<td>Hellfire</td>
<td>Air-to-Surface</td>
<td>Anti-Surface</td>
<td>Inertial</td>
<td>Semi-Active Laser</td>
</tr>
<tr>
<td>Hellfire</td>
<td>Air-to-Surface</td>
<td>Anti-Surface</td>
<td>Inertial</td>
<td>Active</td>
</tr>
</tbody>
</table>

Table 4.1: Comparison of the midcourse and terminal guidance methods used on a variety of weapons [14].
ensures that the agility of the missile is sufficient to counter the evasive manoeuvres of a target in the terminal phase [15].

*Line-of-sight:* The basic idea behind the Line-of-sight method, also known as Three-point, is to guide the missile on the LOS between the launch station and target. This is typically associated with missiles employing the beam-rider method [15].

The result of the various guidance and flight paths are a number of intercept scenarios. These intercept scenarios, also known as endgame scenarios, are described by the velocity, roll, pitch, and yaw of the missile, as well as the miss distance and cylinder angle of the path, and velocity and roll of the target (these terms are defined in Section 4.3.1). Making use of this information the intercept scenario is stored as a text file of which the format is given in Appendix D.1.1.

![Figure 4.7: The intercept geometry.](image)

### 4.3.1 Intercept terminology

The *intercept path* can be defined as the path travelled by the missile, relative to the target. As discussed in Section 2.5 the zero point on the path is the point on the path with the closest vicinity to the predefined aim point on the target. The travel direction of the missile relative to the target is always in the direction of the positive values on the intercept path, as shown in Figure 4.7.

The *miss distance* is the shortest distance between the intercept path and the aim point, and is critical information when determining the zero point on the intercept path. Often the miss distance is notated as $R_{\text{min}}$; this is also the notation used in Figure 4.7.

The intercept path is furthermore defined by introducing an intercept- and cylinder angle. The *intercept angle* is the angle between the velocity vector of the missile relative to the target,
4.3. Flight paths

\( V_R, \) and the velocity vector of the target, \( V_T. \) This is illustrated in Figure 4.8, with the velocity vector of the missile being \( V_M. \)

![Diagram](image)

Figure 4.8: Intercept angle.

A longitudinal axis through the aim point on the target can be defined using the intercept angle illustrated in Figure 4.8. A virtual cylinder having this longitudinal axis as the centre axis and with a radius equal to the miss distance can be constructed. The zero point on the intercept path can thus be any point over a 360° scope with a perpendicular line between itself and the aim point lying on the surface of this virtual cylinder. The specific point is defined by the cylinder angle, \( \beta, \) illustrated in Figure 4.9. The cylinder angle is implemented by rotating the position vector lying in the \( xy \)-plane in an anti-clockwise direction through the specified number of degrees, \( \beta. \)

The velocity vector of the target is defined to be the axis running longitudinally through the centreline of the target. The roll angle is the angle with which the target is rotated around this centreline, as is illustrated in Figure 4.10.

The intercept path indicates the displacement of the missile as a whole, relative to the target. More information is required with regard to the orientation of the warhead, with the centre axis of the warhead and the missile coinciding. This is given in terms of the pitch, \( \theta, \) and yaw, \( \psi, \) angles of the missile. Figure 4.11 illustrates pitch and yaw with regard to a Cartesian co-ordinate system.

The pitch angle can be defined as the ‘vertical’ angle between the \( xy \)-plane of the missile co-ordinate system and the longitudinal axis of the missile. Should the missile be positioned in the direction of the positive \( z \)-axis the pitch angle is also positive. The positive pitch angle of
the missile is illustrated in Figure 4.12.

The orientation of the missile within the \( xy \)-plane is defined by the yaw angle. The \textit{yaw angle} is the angle between the \( x \)-axis of the missile coordinate system and the longitudinal axis of the missile with a positive yaw angle being in the direction of the negative \( y \)-axes as seen in Figure 4.13.

4.4 Generating random intercept scenarios

The user can manually create the text files describing a certain intercept scenario. Alternatively, EVA allows the user to specify the data of the various aspects of the endgame scenario. These aspects are:

\begin{itemize}
\item Missile: \hspace{1em} Closing Speed
\item Roll
\item Pitch
\item Yaw
\item Path: \hspace{1em} Miss Distance
\end{itemize}
4.4. Generating random intercept scenarios

Of these the closing speed, roll, pitch and yaw of the missile, as well as the miss distance, cylinder angle, velocity elevation and velocity heading can be described as continuous random variables.

This gives EVA the capability of randomly generating intercept scenarios that would be typical of the given target and missile combination.

4.4.1 Simulating a Continuous Random Variable

4.4.1.1 Techniques for simulating a Continuous random variables

The inverse Transform Method

Let $U$ be an uniform$(0, 1)$ random variable. For any continuous distribution function $F$ if we define the random variable $X$ by

$$X = F^{-1}(U).$$

Then the random variable $X$ has distribution function $F$. Thus $F^{-1}$ is defined to equal that value $x$ for which $f(x) = u$. 
CHAPTER 4. INTERCEPT PATHS

Figure 4.11: The pitch, $\theta$, and yaw, $\psi$, angles relative to the Cartesian co-ordinate system.

Rejection Method

Suppose we have a method for simulating a random variable having a density $g(x)$. We can use this as the basis for simulating from the continuous distribution having density $f(x)$ by simulating $y$ from $g$ and then accepting this simulated value with a probability proportional to $f(y)/g(U)$. Specifically let $c$ be a constant such that

$$\frac{f(y)}{g(y)} \leq c$$

for all $y$.

The following technique can be used for simulating a random variable having density function $f$.

Step 1: Simulate a $Y$ having a density function $g$ as well as a random number $u$.

Step 2: If $U \leq f(y)/(cg(Y))$ set $X = Y$. Otherwise return to Step 1.

The following formulae are used to generate a random variable for the respective distributions making use of one or more uniform(0, 1) random variables, $U_i$ [16].

4.4.1.2 Beta Distribution

The shape parameters for the beta distribution are $n$ and $m$. To calculate the random variable $X$, the negative of the log of the product of $n$ uniform(0, 1) random variables is divided by the negative of the sum of the log of the product of $n$ uniform(0, 1) random variables and the log of the product of $(m - n)$ uniform(0, 1) random variables. The formula for $X$ is thus

$$X = \frac{-\log \prod_{i=1}^{n} U_i}{\log \prod_{i=1}^{n} U_i - \log \prod_{i=n+1}^{m} U_i}.$$
4.4. Generating random intercept scenarios

4.4.1.3 Discrete Distribution

The parameters for the discrete distribution are \((n, P_1, V_1, P_2, V_2, \ldots, P_i, V_i, \ldots, P_n, V_n)\) where \(n\) is the number of possible values for the random variable \(X\), and \(P_i\) is the probability that \(X = V_i\).

A uniform(0, 1) random variable, \(U\), is generated. If,

\[
\sum_{j=1}^{i-1} P_j < U \leq \sum_{j=1}^{i} P_j,
\]

for \(U > P_1\), then

\[X = V_i;\]

else

\[X = V_1.\]

4.4.1.4 Exponential Distribution

The shape parameter for the exponential distribution is the mean of the data set. The random variable \(X\) is obtained by generating a uniform(0, 1) random variable, \(U\), and taking the product of the negative of the \(\ln(U)\) and the mean. Mathematically this is expressed as,

\[X = -\log(U) \times c = -\ln(U) \times \text{(mean)}.\]

4.4.1.5 Gamma Distribution

The gamma distribution has two parameters. The one is the scale parameter, \(\lambda\), and the other an integer shape parameter, \(n\). These two parameters are related to the mean, also known as the expected value, \(E(X)\), and the variance, \(V(X)\), of the gamma distribution with

\[E(X) = \frac{n}{\lambda}.\]
and

\[ V(X) = \frac{n}{\lambda^2}. \]

A gamma distribution can be simulated by generating \( n \) independent uniform(0, 1) random variables and taking the log of their product and multiplying it with the negative reciprocal of \( \lambda \), i.e.

\[ X = -\frac{1}{\lambda} \log \prod_{i=1}^{n} U_i. \]

### 4.4.1.6 Normal Distribution

The following inverse normal function is a curve fit solution [17], for which given an independent uniform(0, 1) random variable, \( U \), and

\[ T = \sqrt{\ln \left( \frac{1}{U^2} \right)}, \]

and

\[ \rho = T - \frac{2.515517 + 0.802853T + 0.010328T^2}{1 + 1.432788T + 0.189269T^2 + 0.001308T^3}. \]

Having the mean and the standard deviation, \( \sigma \), as input parameters, the random variable \( X \) for the normal distribution is given by the following formula

\[ X = \text{mean} - \rho \times \sigma. \]

### 4.4.1.7 Poisson Distribution

To simulate a poisson random variable with mean, \( \lambda \), independent uniform(0, 1) random variables \( U_1, U_2, \ldots \) must be generated, stopping at

\[ N + 1 = \min \left\{ n : \prod_{i+1}^{n} < e^{-\lambda} \right\}. \]
4.4. Generating random intercept scenarios

The random variable $N$ has the desired distribution, which can be seen by noting that

$$N = \max \left\{ n : \sum_{i=1}^{n} -\log U_i < \lambda \right\},$$

it follows, that

$$\Rightarrow N = \max \left\{ n : \sum_{i=1}^{n} -\ln U_i < \lambda \right\}$$

and

$$\Rightarrow N = \max \left\{ n : \sum_{i=1}^{n} \frac{-\ln U_i}{\lambda} < 1 \right\}.$$

4.4.1.8 Uniform Distribution

Let $\min$ be the minimum value and $\max$ the maximum value the random variable can have, these are also the parameters of the function. An independent uniform(0, 1) random variable is generated and multiplied by the difference of the maximum plus one and minimum value and then adding this result to the minimum value minus one. The formula for the uniformly distributed random variable, $X$, is,

$$X = (\min - 1) + (\max + 1 - \min)U.$$

The validity of the formulae, given in Sections 4.4.1.2 to 4.4.1.8, for generating random variables from independent uniform(0,1) random variables are verified in Appendix E.

4.4.2 The intercept path file

Once the random scenario information have been configured, EVA can generate path files that describe the intercept scenarios, the file format is given in Appendix D.

The following information is used to generate the paths:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Missile Closing Speed</td>
<td>$V_m$</td>
<td>Relative to the target.</td>
</tr>
<tr>
<td>Missile Roll</td>
<td>$\phi$</td>
<td>In the missile system.</td>
</tr>
<tr>
<td>Missile Pitch</td>
<td>$\theta$</td>
<td>In the missile system.</td>
</tr>
<tr>
<td>Missile Yaw</td>
<td>$\psi$</td>
<td>In the missile system.</td>
</tr>
<tr>
<td>Miss Distance</td>
<td>$R_{\min}$</td>
<td></td>
</tr>
<tr>
<td>Cylinder Angle</td>
<td>$\beta$</td>
<td></td>
</tr>
<tr>
<td>Velocity Elevation</td>
<td>$\chi$</td>
<td>Relative to the target.</td>
</tr>
<tr>
<td>Velocity Heading</td>
<td>$\gamma$</td>
<td>Relative to the target.</td>
</tr>
<tr>
<td>Target Roll</td>
<td>$\phi_{\text{target}}$</td>
<td>In the target system.</td>
</tr>
<tr>
<td>Target Velocity</td>
<td>$\mathbf{V}_t = (V_x, V_y, V_z)$</td>
<td>In the earth coordinate system.</td>
</tr>
<tr>
<td>Internal aim point</td>
<td>$(x_{ip}, y_{ip}, z_{ip})$</td>
<td>In the target system.</td>
</tr>
<tr>
<td>Drag</td>
<td>(Yes/No)</td>
<td></td>
</tr>
</tbody>
</table>
4.4.2.1 Miss Distance

As defined in Section 4.3.1 and Figure 4.7, $R_{\text{min}}$ is the distance from the aim point to the zero point on the intercept path. The coordinates of the zero point are $R_{\text{min}} = (x, y, z)$. $R_{\text{min}}$ is calculated by rotating the vector $R'_{\text{min}} = (x_{ip}, y_{ip} + R_{\text{min}}, z_{ip}) = (x_{ip}, y_{R_{\text{min}}}, z_{ip})$ around the $z$-axis through an angle of $\gamma$, which is the velocity heading of the path. The vector is then rotated around the ‘new’ $y$-axis through an angle of $\chi$, the velocity elevation of the path. Finally the vector is rotated around the ‘new’ $x$-axis through the cylinder angle, $\beta$.

Making use of the rotation matrixes discussed in Appendix F.1, $R_{\text{min}}$ is calculated as

$$R_{\text{min}} = R'_{\text{min}} \times \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} \cos \chi & 0 & \sin \chi \\ 0 & 1 & 0 \\ -\sin \chi & 0 & \cos \chi \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \beta & -\sin \beta \\ 0 & \sin \beta & \cos \beta \end{bmatrix},$$

also written as

$$R_{\text{min}} = \begin{bmatrix} x_{ip} \\ y_{R_{\text{min}}} \\ z_{ip} \end{bmatrix} \times \begin{bmatrix} \cos \gamma \cos \chi & \cos \gamma \sin \chi \sin \beta - \sin \gamma \cos \beta & \sin \beta \sin \gamma + \cos \gamma \sin \chi \\ \sin \gamma \cos \chi & \cos \gamma \cos \beta + \sin \gamma \sin \chi \sin \beta & \sin \gamma \sin \chi \cos \beta - \sin \beta \cos \gamma \\ \sin \chi & \cos \gamma \sin \beta \end{bmatrix}.$$

Thus,

$$R_{\text{min}} = \begin{bmatrix} x_R \\ y_R \\ z_R \end{bmatrix},$$

where

$$x_R = x_{ip} \cos \gamma \cos \chi + y_{R_{\text{min}}} \sin \gamma \cos \chi + z_{ip} \sin \chi,$$

$$y_R = x_{ip} (\cos \gamma \sin \chi \sin \beta - \sin \gamma \cos \beta) + y_{R_{\text{min}}} (\cos \gamma \cos \beta \sin \gamma \sin \chi \sin \beta) + z_{ip} \cos \chi \sin \beta,$$

and

$$z_R = x_{ip} (\sin \beta \sin \gamma + \cos \gamma \sin \chi \cos \chi) + y_{R_{\text{min}}} (\sin \gamma \sin \chi \cos \beta - \sin \beta \cos \gamma) + z_{ip} \cos \chi \cos \beta.$$

4.4.2.2 Aimpoint in target coordinates

Since the target coordinate system is superimposed onto the earth coordinate system the aimpoint coordinates is taken as given by the user.

4.4.2.3 Drag

If drag is enabled it is indicated by a ‘1’, otherwise a ‘0’. This influences the calculation of fragment velocities, e.g. magnitude and direction.

4.4.2.4 Missile Speed

The speed of the missile is given relative to the target. The speed of the missile given in the path file is thus the relative speed, $V_R$, seen in Figure 4.8, which is generated from the distribution specified by the user.
4.4. Generating random intercept scenarios

4.4.2.5 Missile travel direction

Since the velocity heading, $\gamma$, and the velocity elevation, $\chi$, of the travel direction, or path, of the missile is given relative to the target, the unit vector indicating the direction of the missile, $\mathbf{u}_{\text{path}}$, can be calculated using trigonometry. If it is assumed that $\mathbf{u}_{\text{path}}$ lies on the $x$-axis the vector is first rotated by $\gamma$ in the direction of the $y$-axis, and then $\chi$ in the direction of the $z$-axis, as illustrated in Figure 4.14. Given that $\mathbf{u}_{\text{path}} = (x_p, y_p, z_p)$, it can be shown that,

$$
\begin{bmatrix}
  x_p \\
  y_p \\
  z_p
\end{bmatrix}
= 
\begin{bmatrix}
  \cos \chi \cos \gamma \\
  \cos \chi \sin \gamma \\
  \sin \chi
\end{bmatrix}.
$$

Figure 4.14: Direction vector of missile path.

4.4.2.6 The direction of the missile system

The roll, pitch and yaw of the missile are given relative to the missile system. The calculations for calculating the transformation matrix, $S(\theta, \psi, \phi)$, are discussed in Appendix F.

Since the target coordinates are transformed, the effect of this must be transposed to the transformation matrix $S(\theta, \psi, \phi)$. First the roll, pitch and yaw of the target must be calculated. The roll is given and the information for the pitch and yaw is extracted from the target velocity, $\mathbf{V}_t = (V_x, V_y, V_z)$.

Three input options for roll are given in degrees, $\phi_{\text{deg}} = -90^\circ, 0^\circ, \text{ or } 90^\circ$, which must be converted to radians,

$$
\phi_{\text{target}} = \begin{cases}
  0, & \phi_{\text{deg}} = 0 \\
  -0.5 \times \pi, & \phi_{\text{deg}} = -90^\circ. \\
  0.5 \times \pi, & \phi_{\text{deg}} = 90^\circ
\end{cases}
$$

The pitch, $\theta_{\text{target}}$, is calculated as

$$
\theta_{\text{target}} = \arctan \left( \frac{V_z}{\sqrt{V_x^2 + V_y^2}} \right),
$$
and the yaw, $\psi_{\text{target}}$, as

$$\psi_{\text{target}} = \arctan\left(\frac{V_y}{V_x}\right).$$

The transformation matrix for the target is

$$T(\theta_{\text{target}}, \psi_{\text{target}}, \phi_{\text{target}}) = S(\theta_{\text{target}}, \psi_{\text{target}}, \phi_{\text{target}}).$$

The final transformation matrix, $S'$, that is written in the path file is

$$S' = T^{-1} \times S.$$

### 4.4.2.7 Coordinates of burst points

The number of burst points is the number of positions on the intercept path where the effectiveness of the warhead will be evaluated. The step size is the distance that the burst points are apart. Both the number of burst points and number of step sizes are specified by the user.

The coordinates of a specific burst point are calculated by multiplying the position of this burst point from the zero burst point, $R_{\text{min}}$, with the unit direction vector of the missile. This value is added to the $R_{\text{min}}$ coordinates, if it is in the positive direction from the burst point, or subtracted from the $R_{\text{min}}$ coordinates, if it is in the negative direction from the burst point.

The coordinates for the $i^{th}$ burst point, $b_i$, from $R_{\text{min}}$ are,

$$b_i = \begin{cases} 
  i \times u_{\text{path}} + R_{\text{min}}, & \text{for } b_i \text{ in the positive direction} \\
  -i \times u_{\text{path}} + R_{\text{min}}, & \text{for } b_i \text{ in the negative direction.}
\end{cases}$$

### 4.4.2.8 Target information

Since the intercept scenario is relative to the target, the target velocity is zero, as well as the target position, and the direction is $(-1, 0, 0)$. The matrix for the direction of the target system is

$$\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}.$$

### 4.5 EVA

Attack scenarios can now be generated by supplying EVA with a set of distributions that describes the various aspects of the endgame scenario probabilistically. EVA provides the functionalities to input these probabilistic distributions and randomly generates a specified number of files describing the attack paths. EVA also provides facilities to view and delete these files. These can be found on the Paths dropdown menu, see Figure 4.15.
4.5. EV A

4.5.1 Endgame Scenario Distributions

EVA allows the user to specify the distributions of the various aspects of the endgame scenario. These aspects are grouped under missile, path, target and general, and given in Section 4.4.

These scenarios can be set up on the Endgame Scenario Distribution form, Figure 4.16. Within this environment the user is provided with the following distribution types:

- Beta Distribution
- Normal Distribution
- Discrete Distribution
- Poison Distribution
- Exponential Distribution
- Uniform Distribution
- Gamma Distribution

Figure 4.17 (a) and (b) shows the input forms for the Beta and Discrete distributions respectively.

Once this information has been entered the user can either save the setup or have EVA generate the specified number of paths for the given number of burst points at the chosen step size.

4.5.2 View Path File

EVA allows the user to view and modify any chosen path file in a similar fashion as any other text file, e.g. target files, as can be seen in Figure 4.18.
4.5.3 View Scenario File

The functionality to view a scenario file works similar to that of viewing a path file. This file is created so that the user can later give probabilities to various endgame scenarios (e.g. head to head, tail chase, etc.) for a specific warhead/target simulation run. This contributes to the randomness of the simulation enabling EVA to run Monte Carlo simulations.

4.5.4 Deleting files associated with paths

Once again EVA incorporates this simple function so as to confine all the actions concerning the warhead design to one environment within Windows.

4.5.5 Improvements on SWEAT

This functionality was not available previously. Within the SWEAT environment the client needed to supply all of the attack path data. This caused two problems, namely an inconsistent file format and a lack of randomness. The first creating extra work in terms of converting the
4.5. EVA

Figure 4.17: Input forms for the Beta (a) and Discrete Distributions (b).

Data into the appropriate format and the second causing a complete lack of randomness within the simulation environment. This functionality creates the opportunity to implement Monte Carlo simulations and meta-heuristic search methods for the first time.
Figure 4.18: View the text file for a path.
Chapter 5

Warhead

5.1 Introduction

The WARHEAD program, developed as a part of SWEAT, calculates the velocity and direction of the fragments that are projected from a warhead. Given the purpose of this study the model used in the WARHEAD program was not studied in depth. It was rather implemented as a “black box” providing the data necessary to estimate the effectiveness of a warhead and optimising that warhead. It is, however, important for completeness to briefly discuss the mathematical theory on which WARHEAD is built. This theory, as well as the interface between EVA and WARHEAD, is discussed in this chapter.

5.2 Damage Mechanisms

The optimisation of fragments of a fragmentation warhead is a function of the target type, the effect of the fragments on the target and also the actual kill mechanisms [1, ?]. The factors involved are:

1. The penetration or perforation potential of the fragments.
2. The penetration depth along with the hole area.
3. The volume of the hole produced.
4. The momentum and kinetic energy of the fragments.
5. The rate at which the target can absorb the energy of the fragments, including secondary effects.
6. Failure of the structure of the target, or structural kill.

The destructive effect of fragments on a target is a function of the combined effect of the fragments. Therefore to increase the probability of destroying the target, the energy imparted to the target has to be increased. This can be achieved by increasing the number, size and/or velocity of the fragments. The combined effect of the fragments can also be increased by synergistic and cumulative effects. (See Figure 5.1)
A synergistic effect is one where the overall effect is greater than the sum of individual effects.

(1) Cumulative or Synergistic Effects.

- **Additive effects** are where a (target) component is weakened by many closely situated fragment perforations and/or penetrations, even when they arrive in succession.
- **Cumulative effects** are the special effects that are observed due to fragments hitting and penetrating the target simultaneously or within short periods of time. The mechanisms that can occur are (Figure 5.2):
  - Cumulative Mechanical Effect. When multiple fragments hit the target, special damage is caused by the overlapping of the pressures and shock waves caused by the impacting fragments. This can be observed when a large hole is knocked out of a plate impacted by numerous fragments on a relatively small area.
  - Cumulative Hydraulic Effect. When a fragment enters a compartment containing fluid, in addition to the shock waves, a cavitation bubble which causes specific pressure effects on the housing, connection lines and valves, are caused. These effects are enhanced by numerous fragment hits.
  - Cumulative Internal Blast Effect. An internal blast effect is caused by the interaction of the fragments and/or the target material with material originating from either the target or the fragment.

(2) Reaction Mechanisms. When multiple metal fragments hit a metal target at very high velocities incandescent metal particles or vapours, originated from the fragments and/or target material, are caused which upon ignition may create a cumulative blast effect, creating an inner pressure and damaging the components of the target.

Modern aircraft use composite materials such as carbon fibre reinforced plastics (CRP) [20] to reduce the weight of components. Depending on the layout of the material the cumulative
effects of multiple fragment hits on the target with CRP skin can be reduced considerably compared to that of aluminium alloys. Also, vaporific effects (from the target material) can also drastically be reduced. This decreases the internal critical overpressures which may cause structure kills. Therefore, from a warhead designer’s point of view, the fragments must be such that the maximum cumulative effect can be produced, such as using incendiary fragments, high fragment densities and velocities, etc.

Figure 5.2: Additive and Cumulative Effects.

A variety of materials are used for the manufacturing of fragments, of which steel, tungsten and depleted uranium (DU) are probably the most common. Depleted uranium, which is used for example in the AA-11, is an excellent choice for fragments for anti-aircraft applications. Not only does DU have a high density ($\geq 17.5 \text{g/cc}$) which enhances the fragment’s penetration capability, but it also exhibits good pyrophoric characteristics which improves the probability of igniting the target and cumulative blast effects. Materials that exhibit characteristics similar to those of DU, and may also be used for the manufacturing of fragments are typically zirconium and hafnium. However, DU fragments require extensive processing plants in the manufacturing environment.

Steel and tungsten fragments are still ‘favourite’ choices because of their availability, the former’s relative low cost and the latter’s excellent penetration capabilities. Copper preformed fragment rings are also used locally in air-target warheads due to their excellent cumulative mechanical effect, especially through multiple target plates, and also because of their high packing density.

The choice of fragments will, however, depend to a large extent on the following factors:

1. **Warhead constraints**: length, mass, diameter, costs, etc.
2. **Type of target**: missile, fighter aircraft, helicopter, etc.
3. **(Other) System constraints**: fuze capabilities, e.g. level of ‘intelligence’, angles and range, attack profiles, etc.
5.3 Fragment Acceleration

5.3.1 Introduction

Many calculations of velocities of explosively accelerated items are successfully done via sophisticated numerical procedures incorporating the equations of states of the material and strength models. However, it is often a tedious (and difficult) task to set-up the problem and even with the new generation software, solving these problems can be time-consuming. Therefore, empirical and/or analytical procedures are often used to estimate the initial velocities and ejection angles of explosively accelerated items. In the warhead simulation program of EVA the empirical equations discussed in the following section are used.

The well known Gurney formulas [13] which are even today used in many applications, give sufficiently accurate velocity predictions in certain configurations. This formula states that for long cylindrical warheads the fragment velocities can be predicted by

$$v_c = \sqrt{2E \sqrt{\frac{C/M}{1 + 0.5C/M}}}$$

where $E$ is the Gurney energy of the explosive and $C/M$ the ratio of the mass of the explosive to that of the surrounding material.

The fragment ejection angle $\alpha$ can be obtained by the Taylor equation,

$$\alpha = \arcsin \left(\frac{v_c}{2v_D}\right) \cos i,$$

where $v_c$ and $v_D$ are the fragment and detonation velocities respectively, and $i$ the angle between the detonation front and the normal to the explosive/metal interface.

It was found that for warheads with relatively small length to diameter ratios ($L/D < 2.0$) the velocities predicted by Equation 5.1 deviate considerably from actual measured data. The reason for this phenomenon is that the Gurney equation used in the calculations was derived for a long cylindrical warhead and the $C/M$ used is the ratio of the whole charge mass to the total mass of the surrounding material. Equation 5.1 therefore does not take any relaxation of detonation gasses at the ends of the warhead, which cause a reduction in fragment velocities in these regions, into account.

(In the predictions model WARHEAD the warhead is typically divided into discs, each with a length equal to the fragment length, of which the radial $C/M$ is then calculated and used in the Gurney equation to obtain the velocity of a fragment in that specific ring.)

5.3.2 Inclusion of end effects

The problem of handling the end-effects in cylindrical charges for fragment sleeves around the circumference of cylindrical sections has been addressed by a number of authors. Most notably the definition of the ‘relaxation coefficient’, $F_x$, by Hennequin [12], which broadened a concept
used by Randers-Pehrson [21] facilitated a direct reduction of the charge-to-metal mass ratio, \( C/M \), in the standard Gurney formula:

\[
v_c = \sqrt{2E} \left( \frac{M}{CF_x} + \frac{1}{2} \right)^{-1/2}.
\] (5.3)

The \( F_x \) factor in 5.3 was derived from numerical results of various configurations, and dependents on \( M/C \) as well as \( L/D \). Randers-Pehrson [21] proposed the following relaxation factor (see Figure 5.3):

\[
F_x = 1 - \left\{ \max \left[ 1 - \frac{x}{D}, 0, 1 - \frac{2(L-x)}{D} \right] \right\}^2.
\] (5.4)

He assumed that there are virtual conical volumes of explosive at the ends of the warhead which, due to relaxation, do not contribute to the acceleration of fragments.

![Diagram illustrating relaxation areas according Randers-Pehrson [21].](image)

This prediction methodology was further extended by König [22], Smit [23] and Smit et al. [24]. Hennequin [12] proposed the following for the relaxation factor (see Figure 5.4):

\[
F_x = \left[ 1 - \left( \frac{r_1}{R_1} \right)^2 \right] \left[ 1 - \left( \frac{r_2}{R_2} \right)^2 \right],
\] (5.5)

where

\[
\frac{r_1}{R_1} = 1 - n_1 \left( \frac{x}{D} \right)^n; \quad \frac{r_2}{R_2} = 1 - 2n_2 \left( \frac{L-x}{D} \right)^n.
\] (5.6)

The indices 1 and 2 in above equations represent the initiation end and the end opposite initiation, respectively. The numerical values of \( n_1 \) and \( n_2 \) depend on the charge-to-metal mass ratio \( (C/M) \) of the warhead. The heights of the release cones, i.e. the minimum distance from the respective ends of the warheads where the release waves would have no effect on the velocities of the fragments, are given by

\[
\frac{h_1}{D} k_1 \left( \frac{M}{C} + \frac{1}{2} \right)^{1/2}; \quad \frac{h_2}{D} k_2 \left( \frac{M}{C} + \frac{1}{2} \right)^{1/2},
\] (5.7)
where \( k_1 \) and \( k_2 \) are empirical constants (See Figure 5.4).

![Diagram illustrating relaxation areas according Hennequin [12].](image)

Figure 5.4: Diagram illustrating relaxation areas according Hennequin [12].

### 5.4 Fragment Retardation

The total drag force on a fragment can be written as [25]

\[
F_{\text{drag}} = \frac{c_w \rho_{\text{air}} v^2 A}{2},
\]

(5.8)

where,

- \( c_w \equiv \text{drag coefficient} \),
- \( \rho_{\text{air}} \equiv \text{air density} \),
- \( A \equiv \text{cross-sectional area of the body at right angles to the flight direction} \),
- \( v \equiv \text{speed of the projectile} \).

If it is assumed that the fragment only moves in the horizontal \( x \)-direction, the equation of motion is

\[
m \ddot{x} = -F_{\text{drag}} = -\frac{c_w \rho_{\text{air}} \dot{x}^2 A}{2},
\]

(5.9)

where \( m \) is the mass of the projectile, \( \ddot{x} \) the second time derivative of the position or acceleration and \( \dot{x} \) the speed of the projectile.

If it is assumed that \( c_w \) is constant (it is normally a function of speed), then from Equation (5.9) it follows that the speed as function of position can be obtained from the expression

\[
v \equiv \dot{x} = v_0 \exp \left( -\frac{c_w \rho_{\text{air}} A}{2m} x \right) = v_0 e^{-kx}.
\]

(5.10)

Equation (5.10) is a useful equation for estimation purposes for one-dimensional flight.
5.5 EVA

The warhead module has various functionalities within EVA. These are the creation of an original input file, running a pre-processor on this file to segment the warhead, and the creation of an input file for Upshot and a user file that summarises the data generated for Upshot into a more user friendly format. EVA also offers the functionality to view all of these files. These can be found on the dropdown menu, see Figure 5.5.

![Figure 5.5: Warhead dropdown menu.](image)

5.5.1 Create Warhead Pre-processor Input

Previously, the input file for Warhead or the Pre-processor needed to be created in a different package editor, e.g. Notepad, MS Word, etc. Apart from being time consuming it also created a greater opportunity for errors.

Within EVA the data contained in this input file were divided into three groups. These are general variables, variables specific to the various blocks, and data that typically stay constant for all pre-fragmented warheads.

The general variables included the name of the warhead, as well as the names of the various files to be created. These files are the input file for the pre-processor, the input for Warhead, the input file for Upshot, and the user summary file. Other variables found in this group are the variables that have a single value specific to the warhead. The general variables can be found on the Variables tab seen in Figure 5.6.
The non-cylindrical shape of a warhead and the fact that a warhead may not, for example, consist of one type of fragment is modelled by dividing it into various blocks or segments. The variable values for each of these segments are stored individually and can be found on the blocks tab seen in Figure 5.7.
Figure 5.7: Warhead Blocks Tab.
The data that typically stays constant for all pre-fragmented warheads can be found on the constants tab. These data are already completed when the process to create a new warhead is initiated, see Figure 5.8.

![Warhead Constants Tab](image)

Figure 5.8: Warhead Constants Tab.

The files of previously created warheads can also be opened making it easy to make changes to an existing file or generate a variation of the warhead under another name.

Once the data have been completed the data can be saved for future use, or the data can be saved and the input file for the pre-processor can be created.

This input for the pre-processor can be viewed from within EVA. Changes can also be made, but can only be saved as a new warhead, ensuring data integrity.

### 5.5.2 The Pre-processor

The pre-processor modifies the input file for Warhead by sub-dividing the segments into more segments as to accommodate a curved warhead profile. To initiate the pre-processor, a warhead needs to be selected, see Figure 5.9.
As with the input file for the pre-processor, the input file for Warhead can also be viewed from within EVA. The same restriction is also applicable.

![Selection window of warhead for pre-processor.](image)

Figure 5.9: *Selection window of warhead for pre-processor.*

### 5.5.3 Create Warhead

The Warhead programme calculates the information required by Upshot. This process is initiated in a similar fashion as the pre-processor.

The file created for Upshot contains information such as the size and weight of each fragment, as well as the ejection direction of each fragment relative to the warhead axes. As mentioned earlier these data are written into a file, specified by the user, for Upshot. Once again this file can be viewed from within EVA. The window used for this functionality can be seen in Figure 5.10. This figure also contains the data as written for Upshot.

The data, describing the warhead, is written in a user file that can be seen in Figure 5.11 and includes information such as:

1. Initiation point,
2. Single initiation,
3. Gurney velocity,
4. Detonation velocity,
5. Number of segments,
6. Total mass of the warhead,
7. Total explosive mass,
8. Total number of fragments.
Figure 5.10: View Warhead output file for Upshot.
The data also includes summative information on the segments, namely:

- Type of fragments,
- Fragment size,
- Number of fragments,
- Mass of fragments,
- Single layer,
- Density of fragment,
- Density of explosives,
- Density of filler,
- Density of casing outer,
- Density of casing inner.
5.5.4 Delete Warhead

By selecting a warhead from the list of created warheads in the Warhead deletion form, Figure 5.12, and confirming the action, Figure 5.13, a warhead will be deleted from the target list in EVA.

![Figure 5.12: Warhead deletion form.](image)

In the event of a warhead being deleted the information concerning the specific warhead are deleted. This includes all the information stored in the database as well as the files that were created.

5.5.5 Improvements on SWEAT

The whole process of generating a warhead has been incorporated into one Windows environment. Previously the pre-processor and Warhead was run as two separate programs in a Dos environment and the original input file needed to be generated in a much more manual fashion in a different word processing environment.

EVA also now keeps book of the various files associated with a specific Warhead. Within the SWEAT environment the designer needed to do this bookkeeping himself, this led to wasted time in finding the correct file and executing it.

The whole process is also much more streamlined with EVA serving as a “creation wizard” leading the user through the creation process once the creation action has been initiated at any point in the process.

Finally, the user friendliness of the process and ease of use has been increased dramatically by making use of the input sheet from which the first text file is created automatically.
Figure 5.13: Warhead deletion confirmation form.
Chapter 6

Fuze

6.1 Introduction

This chapter continues the discussion in Section 2.6. Background information on fuzes (target detection devices), such as their functions and various fuzing systems, are given in Section 6.2 of this chapter. The mathematical model used in EVA to simulate a specific fuzing design and calculate the detection position of it, given some basic input data, is discussed in Section 6.3. How these functionalities are accessed in EVA is explained in the last section of this chapter.

6.2 The Fuze

According to Szypula [26] the fuze can be defined as: “A weapon subsystem that activates the warhead mechanism and also maintains the warhead in a safe condition during all phases of the logistic and operational chain. A fuze is typically a mechanical or electronic device which will initiate/detonate the explosive train in a weapon.”

This definition indicates that the fuze and the warhead functions as a unit and that there are two distinct characteristics that differentiate it from other subsystems of the weapon system [26]:

- Firstly, it needs to remain functionally dormant until a “real” target is detected,
- and secondly, it must be able to perform its function reliably in a fraction of a millisecond.

A guidance system, for example, may temporarily malfunction and still recover; target-tracking radars may detect many “false” targets without compromising the weapon system; in other words, these systems have time to function, but once the fuze-warhead process is initiated, having only a small window of opportunity, it is momentary and irreversible.

The quality specifications for fuze designs are as a result high. Quality is related to functional- and safety reliability. In the case of missile fuzes of high complexity functional reliability is 95%, and for less complex missile fuzes, such as for projectile and bomb contact fuzes, as high as 99%. With regard to safety reliability, it is even higher with a failure rate of less than one out of a thousand being expected before it is cleared for service usage.
In summary, the main purpose of the fuze is twofold, firstly it keeps the weapon system safe when it is not needed, and secondly it triggers the explosive reaction when appropriate.

6.2.1 Functions of the Fuze

The fuze is a functional part of the weapon system. The functions of a fuze can be listed as [26]:

- keeping the weapon system safe,
- arming the weapon,
- recognising or detecting a target,
- initiating detonation, and,
- determining the direction of detonation in the case of special fuzes.

A warhead with directionality has the ability to direct the largest amount of energy in the direction of the target. Given the fact that Upshot is based on the ray tracing methodology, EVA can be extended to handle directionality. This study has, however, been confined to warheads with rigid axial symmetric warheads, making directionality irrelevant. From the above functionalities the detection of a target is the only one of relevance to the effectiveness analysis in this study.

6.2.2 Fuze Classification Systems

Fuzes can either be classified according to the weapon system in which they are used, or the fuze operation that is employed.

The following classifications are applicable when it is according to the weapon system it is used in [26]:

- Gun Projectile Fuzes:
  - Rotating Ammo,
  - Non Rotating Ammo,
- Rocket Fuzes,
- Bomb Fuzes,
- Mine Fuzes (Exploders),
- Missile Fuzes:
  - Air-to-Air,
  - Air-to-Surface,
  - Surface-to-Air,
  - Surface-to-Surface,
6.2. The Fuze

- Torpedo Fuzes (Exploders),

and when the classification is made according to the fuze operation it is:

- Proximity Fuzes (VT fuzes):
  - Active,
  - Semi-Active,
  - Passive,

- Time Fuzes,

- Impact (Point Detonation),

- Ambient,

- Delay Fuzes,

- Command Detonate Fuzes.

An active VT fuze generates a signal, e.g. electromagnetic, which is reflected by the target and detected by the fuze. In the case of the semi-active VT fuze the signal, which illuminates the target, is generated by an independent source from the target and fuze. A passive VT fuze detects the electromagnetic energy radiated by the target. Time fuzes detonate the warhead at a preset time after its launch. It is often used in conjunction with a proximity fuze to affect self-destruction. The impact fuze is also used with most proximity fuzes but is used independently as well, detonating the warhead in the event of a direct hit. An ambient fuze takes environmental factors, such as altitude through barometric switches, into consideration. A delay fuze is much like the time fuze, detonating after a preset time, the main difference being that the timer is triggered by some event, e.g. target detection, or target contact, in the endgame phase and not the launch of the missile. Finally, in the case of command detonated fuzes the warhead is command detonated by an external radio signal [1].

For the purposes of this study proximity fuzes, combined with delay fuzes, or alternatively missile fuzes, with respect to the two classifications, will be modelled. Some examples of proximity fuzes are:

- magnetic,

- optical,

- radio, and,

- lasers.

The objective of the fusing model for EVA is thus to determine, given a specific intercept scenario and fuze design, at what position on the intercept path the fuze will detect the target. This is important since only the effect of a warhead detonation subsequent to detection is applicable to an effectiveness analysis. Also, a delay can be determined to increase the possibility of detonation at the point of maximum expected effectiveness after detection.
6.3 The Fuze Model

The model that has been developed is illustrated in the flow diagram in Figure 6.1. Firstly, various input data are required. These data include a description of the target, as discussed in Chapter 3, the various transformation matrices, and the parameters of the fuze. The fuze data are calculated and stored. Using the transformation matrices the fuze data are converted to the appropriate coordinate system. The next step is to determine at what coordinate the fuze will detect the target, given that it has an infinite reach. Once this has been determined, the reach of the fuze is considered. Should the target be in reach it is a positive detection. The model will now be discussed according to the various sections in the flow diagram except for the various transformation matrices which will be discussed where they are applied.

6.3.1 Calculate the fuze data

The fuze is described by three parameters, $(\theta, \alpha, d)$, where $\theta$ is half of the angle that the detection sensors of the fuze span, $\alpha$ is the number of degrees two sensors are apart (the density of the sensors), determining the accuracy, and $d$ is the reach distance of the sensors. This is illustrated in Figures 6.2 and 6.3. The coordinates of the detection sensors are taken as the coordinates of the warhead (considering the warhead as a point source).

The following information is thus available:

- $d_r$: Reach distance of the fuze detection sensors.
- $\alpha$: The distribution of detection points.
- $(x_0, y_0, z_0)$: Origin of the fuze detection sensors.
- $\theta$: The half width of the fuze detection sensor angle.
6.3. The Fuze Model

Figure 6.2: The \( \theta \) angle of the Fuze.

From this information the coordinates, \((x_i, y_i, z_i)\), at the maximum reach distance and in the detection direction is calculated for each sensor, \(i\), as follows.

As illustrated in Figure 6.4,

\[
y = d \cos \theta \quad \text{and} \quad x = d \sin \theta.
\]

Using this \(y\) and \(x\), see Figure 6.3, it can be shown that

\[
x' = x \cos \alpha \quad \text{and} \quad z' = x \sin \alpha.
\]

It follows that

\[
x_i = x_0 + d_r \sin \theta \cos \alpha,
y_i = y_0 + d_r \sin \theta,
z_i = z_0 + d_r \sin \theta \sin \alpha,
\]

and that the direction vector, \(r_i = (x_{ri}, y_{ri}, z_{ri})\), for the intercept point \(i\) is calculated as

\[
x_{ri} = \frac{x_i - x_0}{d_{\text{temp}}},
\]
CHAPTER 6. FUZE

Figure 6.3: The $\alpha$ angle of the Fuze.

\[
y_{ri} = \frac{y_i - y_0}{d_{\text{temp}}},
\]
\[
z_{ri} = \frac{z_i - z_0}{d_{\text{temp}}},
\]
where
\[
d_{\text{temp}} = \sqrt{x_{ri}^2 + y_{ri}^2 + z_{ri}^2}.
\]

Figure 6.4: Two dimensional side view of half the Fuze.

6.3.2 Transformation Matrix data

The data necessary to transform between the target, missile (fuze), and earth coordinate systems is read from the path file. This data and some of the calculations done are discussed below.

6.3.2.1 Missile data (Fuze)

The following information is read from the path file:

- The absolute velocity of the missile, $V_{\text{Abs},m}$,
• The direction of the missile, $\mathbf{L}_m = (x_m, y_m, z_m)$, in earth coordinates,

• The transformation matrix from earth coordinates to missile coordinates,

$$EM = \begin{bmatrix} em_{11} & em_{12} & em_{13} \\
em_{21} & em_{22} & em_{23} \\
em_{31} & em_{32} & em_{33} \end{bmatrix},$$

• The coordinates of the missile position on the path, $b_i = (x_{bi}, y_{bi}, z_{bi})$.

From this information the velocity of the missile can be calculated in earth coordinates by multiplying the velocity matrix with the missile-to-earth coordinate matrix,

$$v_{me} = V_{Abs,m} \times EM.$$

The transformation matrix from missile coordinates to earth coordinates is calculated by calculating the transpose of $EM$,

$$ME = EM^T = \begin{bmatrix} em_{11} & em_{21} & em_{31} \\
em_{12} & em_{22} & em_{32} \\
em_{12} & em_{23} & em_{33} \end{bmatrix}.$$

### 6.3.2.2 Target data

The following information is read from the target file:

• The absolute velocity of the target, $V_{Abs,t}$,

• The direction of the target, $\mathbf{L}_t = (x_t, y_t, z_t)$, in earth coordinates,

• The transformation matrix from earth coordinates to target coordinates,

$$ET = \begin{bmatrix} et_{11} & et_{12} & et_{13} \\
et_{21} & et_{22} & et_{23} \\
et_{31} & et_{32} & et_{33} \end{bmatrix},$$

• The coordinates of the position of the target, $(i, j, k)$.

From this information the velocity of the target can be calculated in earth coordinates by multiplying the velocity matrix with the target-to-earth coordinate matrix,

$$v_{te} = V_{Abs,t} \times ET.$$

The transformation matrix from target coordinates to earth coordinates is calculated by calculating the transpose of $ET$,

$$TE = ET^T = \begin{bmatrix} et_{11} & et_{12} & et_{13} \\
et_{12} & et_{22} & et_{32} \\
et_{12} & et_{23} & et_{33} \end{bmatrix}.$$
6.3.2.3 Fuze data

The following information is read in with regard to the fuze:

- The number of sensors, \( n_s \), which determines the accuracy,
- The search direction of each sensor in the fuze, \( \mathbf{r}_{si} = (x_{si}, y_{si}, z_{si}) \),
- The reach distance, \( d_r \), which is the same for all of the sensors.

The various sensor positions coincide with the missile, which is considered a point source, positions on the path.

Following this, the search direction of each sensor is converted to earth coordinates,

\[
\mathbf{r}_{si}' = \mathbf{ME} \times \mathbf{r}_{si}.
\]

The search directions are then converted from earth coordinates into target coordinates,

\[
\mathbf{r}_{si}'' = \mathbf{ET} \times \mathbf{r}_{si}'.
\]

The missile coordinates, as well as the sensor coordinates are in the target coordinate system at this stage of the algorithm.

6.3.3 Fuze

The vector from an internal point in the polyhedron, \( p_i \), to a point on the beam path, \( p_f \), is,

\[
\mathbf{r}_l = p_f - p_i = (x_f, y_f, z_f) - (x_i, y_i, z_i) = (x_f - x_i, y_f - y_i, z_f - z_i),
\]

as illustrated in Figure 6.5.

If \( d' \) is defined as the shortest distance that the beam can pass from the internal point, \( p_i \), then, using the theorem of Pythagoras, \( d'^2 \) is calculated as

\[
d'^2 = (\mathbf{r}_l \cdot \mathbf{r}_l) - (\mathbf{r}_l \cdot \mathbf{r}_{pf})^2
\]

(see Figure 6.6). If \( d_{max} \) is defined as the maximum distance from the internal point, \( p_i \), to any point on the surface of the polyhedron, then it follows that for,

\[
d'^2 > d_{max}^2,
\]

the beam will definitely not intercept the polyhedron. The inverse is, however, not true, namely that if

\[
d'^2 \leq d_{max}^2,
\]

the beam will definitely intercept the polyhedron. If this is the case, further analysis is required before a conclusion can be made.
6.3. The Fuze Model

Figure 6.5: The vector from an internal point in the polyhedron, $p_i$, to a point on the beam path, $p_f$.

Should $d$, in Figure 6.7, be defined as the perpendicular distance between the internal point, $p_i$, and the side of the polyhedron, and $\vec{n}$ is the normal vector from that side, it follows that the beam is not in the polyhedron if

$$\vec{n} \cdot \vec{r}_l > d$$

for one of the sides of the polyhedron.

Let $\epsilon$ be a small value ($\epsilon$ is used instead of zero due to the floating point error in software.) If the absolute value of the component of the direction vector, which is parallel to the normal vector of a side of the polyhedron, is less than $\epsilon$,

$$||\vec{r}_{pf} \cdot \vec{n}|| < \epsilon,$$

it follows that the side and the beam path is parallel. It will thus be a definite miss if $\vec{n} \cdot \vec{r}_l > d$.

Figure 6.8 illustrates “all” the possible orientations of the direction of a beam relative to the normal vector $\vec{n}$ from a two dimensional perspective.

All of the possible angles between the direction vector, $\vec{r}_{pf}$, and the normal vector, $\vec{n}$, are shown in Table 6.1. Since $Q_2$ is calculated from the dot product of $\vec{n}$ and $\vec{r}_{pf}$, it follows that the sign is that of $\cos \alpha$, where $\alpha$ is the angle between $\vec{n}$ and $\vec{r}_{pf}$.

It can be deduced from Table 6.1 and Figure 6.8 that in all the cases where $Q_2 < 0$ the plane, in which the side for which the dot product is taken with the normal vector lies, will potentially be this first plane intercepted by the beam, or not at all. The inverse is also the case when $Q_2 > 0$. In this case the plane will either be intercepted after another plane in which one of the sides of the polyhedron lies, or not at all.
Figure 6.6: Calculating $d^2$ using the theorem of Pythagoras.

Figure 6.7: The perpendicular distance, $d$, between the internal point, $p_i$, and the side of the polyhedron.

From Figure 6.8 it can be seen that the plane, in which side $n$ lies, will be intercepted first by the beams originating from the following point sources 1, 2, 5, 6, 29 and 30. By referring to Table 6.2 a number of characteristics of these point sources (sensors) can be identified.

All the beams that can possibly intercept have a positive value for $Q_1(= n \cdot r_i)$. Another characteristic is the fact that $Q_1 > d$. The beams that will not intercept the polyhedron, but for which $Q_1$ has a positive value, on the other hand, has a $Q_1 < d$. All the other beams that will not intercept the polyhedron have a negative $Q_1$, $Q_1 < 0$. It follows that $d - Q_1 < 0$ for all beams that have a possibility of intercepting, and $d - Q_1 > 0$ otherwise. From this the “time”
6.3. The Fuze Model

Figure 6.8: A point source and the relative direction of a beam from it to the normal vector of a plane.

that the beam takes to reach side \( l \) can be calculated as

\[
T_l = \frac{d - Q_1}{Q_2} \geq 0,
\]

should it intercept the side. In the case of side \( n \) in Figure 6.8 the beams that will potentially intercept are in the blue shaded area.

Following the same argument it follows that the beams that may potentially intercept side \( m \) are 5, 6, 7, 8, 11 and 14. These beams are in the yellow shaded area as shown in Figure 6.8.

Sides \( m \) and \( n \) can potentially be intercepted by beam 5 or 6. Should the beam intercept the planes of both sides, it must be determined which one will be intercepted first.

From Figure 6.9 it can be seen that should the beam move in the general direction labelled 5 the plane in which side \( m \) lies will be intercepted first. It follows that \( t_m > t_n \), given that the beam is a distance \( d \) from both of the planes for the respective sides. The plane with the largest \( T_l \) is the plane that will be intercepted.

Following the same argument, but changing all the signs, the plane that is intercepted second can be determined. Consequently,

\[
T(1) = T_{min}, \quad \text{the position where the polyhedron is intercepted first;}
\]
\[
T(2) = T_{max}, \quad \text{the position where the polyhedron is intercepted second;}
\]
Table 6.1: Cases for which the beam will potentially intercept the side first or not at all.

<table>
<thead>
<tr>
<th>#</th>
<th>The angle between ( n ) and ( r_{pf} )</th>
<th>( Q_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( 90^\circ &lt; \alpha \leq 180^\circ )</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>( 90^\circ &lt; \alpha \leq 180^\circ )</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>( 0^\circ &lt; \alpha \leq 90^\circ )</td>
<td>+</td>
</tr>
<tr>
<td>4</td>
<td>( 0^\circ &lt; \alpha \leq 90^\circ )</td>
<td>+</td>
</tr>
<tr>
<td>5</td>
<td>( 90^\circ &lt; \alpha \leq 180^\circ )</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>( 90^\circ &lt; \alpha \leq 180^\circ )</td>
<td>-</td>
</tr>
<tr>
<td>7</td>
<td>( 0^\circ &lt; \alpha \leq 90^\circ )</td>
<td>+</td>
</tr>
<tr>
<td>8</td>
<td>( 90^\circ &lt; \alpha \leq 180^\circ )</td>
<td>-</td>
</tr>
<tr>
<td>9</td>
<td>( 0^\circ &lt; \alpha \leq 90^\circ )</td>
<td>+</td>
</tr>
<tr>
<td>10</td>
<td>( 90^\circ &lt; \alpha \leq 180^\circ )</td>
<td>-</td>
</tr>
<tr>
<td>11</td>
<td>( 0^\circ &lt; \alpha \leq 90^\circ )</td>
<td>+</td>
</tr>
<tr>
<td>12</td>
<td>( 90^\circ &lt; \alpha \leq 180^\circ )</td>
<td>-</td>
</tr>
<tr>
<td>13</td>
<td>( 0^\circ &lt; \alpha \leq 90^\circ )</td>
<td>+</td>
</tr>
<tr>
<td>14</td>
<td>( 0^\circ &lt; \alpha \leq 90^\circ )</td>
<td>+</td>
</tr>
<tr>
<td>15</td>
<td>( 0^\circ &lt; \alpha \leq 90^\circ )</td>
<td>+</td>
</tr>
<tr>
<td>16</td>
<td>( 0^\circ &lt; \alpha \leq 90^\circ )</td>
<td>+</td>
</tr>
<tr>
<td>17</td>
<td>( 0^\circ &lt; \alpha \leq 90^\circ )</td>
<td>+</td>
</tr>
<tr>
<td>18</td>
<td>( 0^\circ &lt; \alpha \leq 90^\circ )</td>
<td>+</td>
</tr>
<tr>
<td>19</td>
<td>( 90^\circ &lt; \alpha \leq 180^\circ )</td>
<td>-</td>
</tr>
<tr>
<td>20</td>
<td>( 90^\circ &lt; \alpha \leq 180^\circ )</td>
<td>-</td>
</tr>
<tr>
<td>21</td>
<td>( 0^\circ &lt; \alpha \leq 90^\circ )</td>
<td>+</td>
</tr>
<tr>
<td>22</td>
<td>( 0^\circ &lt; \alpha \leq 90^\circ )</td>
<td>+</td>
</tr>
<tr>
<td>23</td>
<td>( 90^\circ &lt; \alpha \leq 180^\circ )</td>
<td>-</td>
</tr>
<tr>
<td>24</td>
<td>( 90^\circ &lt; \alpha \leq 180^\circ )</td>
<td>-</td>
</tr>
<tr>
<td>25</td>
<td>( 90^\circ &lt; \alpha \leq 180^\circ )</td>
<td>-</td>
</tr>
<tr>
<td>26</td>
<td>( 0^\circ &lt; \alpha \leq 90^\circ )</td>
<td>+</td>
</tr>
<tr>
<td>27</td>
<td>( 90^\circ &lt; \alpha \leq 180^\circ )</td>
<td>-</td>
</tr>
<tr>
<td>28</td>
<td>( 0^\circ &lt; \alpha \leq 90^\circ )</td>
<td>+</td>
</tr>
<tr>
<td>29</td>
<td>( 90^\circ &lt; \alpha \leq 180^\circ )</td>
<td>-</td>
</tr>
<tr>
<td>30</td>
<td>( 90^\circ &lt; \alpha \leq 180^\circ )</td>
<td>-</td>
</tr>
<tr>
<td>31</td>
<td>( 0^\circ &lt; \alpha \leq 90^\circ )</td>
<td>+</td>
</tr>
<tr>
<td>32</td>
<td>( 0^\circ &lt; \alpha \leq 90^\circ )</td>
<td>+</td>
</tr>
</tbody>
</table>

Table 6.1: Cases for which the beam will potentially intercept the side first or not at all.

and,

\[
Cov(1) = \frac{-Q_2}{\sqrt{r_{pf}^T r_{pf}}} = \frac{-\cos \alpha \|r_{pf}\|\|z\|}{\|r_{pf}\|} = -\cos \alpha, \quad \text{the direction cosine;}
\]

\[
Cov(2) = \frac{Q_2}{\sqrt{r_{pf}^T r_{pf}}} = \frac{\cos \alpha \|r_{pf}\|\|z\|}{\|r_{pf}\|} = \cos \alpha, \quad \text{the direction cosine.}
\]

If \( T_{\min} - T_{\max} = 0 \) it follows that the beam intercepts the polyhedron in the side.

Should the direction of the detection sensor signal an intercept, the next step is to determine whether or not the side is within the reach distance of the sensor. Let

\[
Length_s = (x_s, y_s, z_s)
\]
6.4 EVA

EVA provides the functionalities to calculate where a certain fuze design will detect a specific target on a given path. EVA also provides facilities to view and delete the files containing the detection position information. These can be found on the dropdown menu, seen in Figure 6.10.

<table>
<thead>
<tr>
<th>#</th>
<th>The angle between $\mathbf{n}$ and $\mathbf{r}_f$</th>
<th>$Q_1$</th>
<th>$Q_1 &gt; d$</th>
<th>$d - Q_1$</th>
<th>$r_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$0^\circ &lt; \alpha \leq 90^\circ$</td>
<td>+</td>
<td>Yes</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>2</td>
<td>$0^\circ &lt; \alpha \leq 90^\circ$</td>
<td>+</td>
<td>Yes</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>3</td>
<td>$0^\circ &lt; \alpha \leq 90^\circ$</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>$0^\circ &lt; \alpha \leq 90^\circ$</td>
<td>+</td>
<td>Yes</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>5</td>
<td>$0^\circ &lt; \alpha \leq 90^\circ$</td>
<td>+</td>
<td>Yes</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>6</td>
<td>$0^\circ &lt; \alpha \leq 90^\circ$</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>$0^\circ &lt; \alpha \leq 90^\circ$</td>
<td>+</td>
<td>No</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>8</td>
<td>$90^\circ &lt; \alpha \leq 180^\circ$</td>
<td>-</td>
<td>Yes/No</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>9</td>
<td>$90^\circ &lt; \alpha \leq 180^\circ$</td>
<td>+</td>
<td>No</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>10</td>
<td>$90^\circ &lt; \alpha \leq 180^\circ$</td>
<td>+</td>
<td>Yes/No</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>11</td>
<td>$90^\circ &lt; \alpha \leq 180^\circ$</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>$90^\circ &lt; \alpha \leq 180^\circ$</td>
<td>+</td>
<td>Yes/No</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>13</td>
<td>$90^\circ &lt; \alpha \leq 180^\circ$</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>$90^\circ &lt; \alpha \leq 180^\circ$</td>
<td>+</td>
<td>Yes/No</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>15</td>
<td>$90^\circ &lt; \alpha \leq 180^\circ$</td>
<td>+</td>
<td>Yes/No</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>16</td>
<td>$90^\circ &lt; \alpha \leq 180^\circ$</td>
<td>+</td>
<td>Yes/No</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>17</td>
<td>$90^\circ &lt; \alpha \leq 180^\circ$</td>
<td>+</td>
<td>Yes/No</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>18</td>
<td>$90^\circ &lt; \alpha \leq 180^\circ$</td>
<td>+</td>
<td>Yes/No</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>19</td>
<td>$90^\circ &lt; \alpha \leq 180^\circ$</td>
<td>+</td>
<td>Yes/No</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>20</td>
<td>$90^\circ &lt; \alpha \leq 180^\circ$</td>
<td>+</td>
<td>Yes/No</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>21</td>
<td>$90^\circ &lt; \alpha \leq 180^\circ$</td>
<td>+</td>
<td>Yes/No</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>22</td>
<td>$90^\circ &lt; \alpha \leq 180^\circ$</td>
<td>+</td>
<td>Yes/No</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>23</td>
<td>$90^\circ &lt; \alpha \leq 180^\circ$</td>
<td>+</td>
<td>Yes/No</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>24</td>
<td>$90^\circ &lt; \alpha \leq 180^\circ$</td>
<td>+</td>
<td>Yes/No</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>25</td>
<td>$90^\circ &lt; \alpha \leq 180^\circ$</td>
<td>+</td>
<td>Yes/No</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>26</td>
<td>$90^\circ &lt; \alpha \leq 180^\circ$</td>
<td>+</td>
<td>Yes/No</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>27</td>
<td>$90^\circ &lt; \alpha \leq 180^\circ$</td>
<td>+</td>
<td>Yes/No</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>28</td>
<td>$90^\circ &lt; \alpha \leq 180^\circ$</td>
<td>+</td>
<td>Yes/No</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>29</td>
<td>$90^\circ &lt; \alpha \leq 180^\circ$</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>$90^\circ &lt; \alpha \leq 180^\circ$</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6.2: Beam characteristics

be the distance from the sensor to the point on side that is intercepted. Then,

$$x_s = T_{\min} \times x_{rpf},$$

$$y_s = T_{\min} \times y_{rpf},$$

$$z_s = T_{\min} \times z_{rpf},$$

and

$$Length_s = \sqrt{x_s^2 + y_s^2 + z_s^2}.$$ 

It follows that the specific polyhedron will be detected if

$$Length_s \leq d_r$$ (the reach distance of the sensor).

6.4 EVA

EVA provides the functionalities to calculate where a certain fuze design will detect a specific target on a given path. EVA also provides facilities to view and delete the files containing the detection position information. These can be found on the dropdown menu, seen in Figure 6.10.
Figure 6.9: The beam moving in the general direction labelled 5 will intercept the plane in which side m lies first.

Figure 6.10: Fuzing dropdown menu.

6.4.1 Run Fuze Simulation

Figure 6.11: Fuze angle, step size, and reach distance.

On the fuze form, Figure 6.12, the user must specify the target, fuze angle, the step size and reach distance of the fuze, as well as the paths for which to evaluate the fuze against the target.
Figure 6.11 shows what is defined by the fuze angle, the step size and reach distance of the fuze.

### 6.4.2 View Fuze File

After evaluating a specific fuze against a target a fuze file for each of the selected paths is created. This file can be viewed, and edited from within EVA, Figure 6.13. The fuze file contains the following information:

<table>
<thead>
<tr>
<th>General</th>
<th>Fuze</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target name</td>
<td>Fuze reach distance</td>
</tr>
<tr>
<td>Name of path file</td>
<td>Fuze Angle</td>
</tr>
<tr>
<td>Detection position on the path</td>
<td>Step Size</td>
</tr>
<tr>
<td>Type of detection</td>
<td></td>
</tr>
</tbody>
</table>

### 6.4.3 Deletion of fuze file

EVA also allows the user to delete a fuze file, Figure 6.14. Once again ensuring that all of the required functionalities are contained within the EVA environment.

### 6.4.4 Improvements on SWEAT

SWEAT also has a module for calculating the detection position for a certain fuze design that works on the same principle of modelling the fuze with a cone. This package does, however, not always give the expected result.
The program was redeveloped within EVA and gives satisfactory results. This is important since it also enables the implementation of optimisation methods through the use of heuristic search algorithms which will be discussed in Section 9.4.

Once again SWEAT made use of various programs to implement this functionality including a Dos environment. EVA provides the user with all the necessary functionalities within one program and in a Windows environment.
Chapter 7

Effectiveness Model

7.1 Introduction

Upshot is briefly discussed in Section 1.2.2.2 and 2.7. Upshot can be seen as the heart of EVA and utilises the target, warhead, and path models to do the effectiveness analysis. The result derived at by these models, however, only serve as input data. Within Upshot, which vital parts are hit by which fragment and the effect this has on the probability of kill are calculated. These results are presented as the probability of an event occurring for each intercept scenario, the number of hits on the target, as well as the number of hits on each vital part during the encounter. These results are then averaged out over all the intercept scenarios and presented as the mean probability of an event occurring, the mean number of hits on the target, and the mean number of hits on each vital part over all the intercept scenarios.

7.2 Upshot

The effectiveness model, even though a mathematical model, is implemented in a programme named Upshot. The structure of Upshot can be viewed in Figure 7.1. The model will first be discussed at a high level, looking at the methodology rather than the mathematics for implementing it.

The required data are firstly read in from the various data files. These data include the attack geometry (the intercept scenarios), the warhead and target description, the penetration and crater data, and the target system characteristics.

The effect of the warhead detonation is evaluated at each of the burst points for a specific intercept scenario and the mean values for the various events calculated over the number of burst points.

At a specific burst point the path of each of the fragments is individually followed. First it is determined whether or not the fragment will actually hit the target. To effectively calculate whether the fragment will hit the target, the effect of drag is taken into account. Having taken drag into account the fragment is now evaluated against each polyhedron that is part of the outer skin of the target. As part of the evaluation a cylinder is constructed around each of
the polyhedron and it is determined whether or not the fragment will intersect this cylinder, and if the fragment path intersects the cylinder, at what point in “time”. Time is written in quotation marks, because it is not real time. It is only an amount which is used to determine that should the fragment travel from either of two points, from which of the two points it will travel in the shortest time. It is thus “time” relative to one another. There is one of four (4) possible intercept scenarios as illustrated in Figure 7.2. These are:

(1) the fragment enters from the top flat side of the cylinder and exits the cylinder through the curved side;

(2) the fragment enters and exits the cylinder through the curved side;

(3) the fragment enters from the cylinder through the curved side and exits it through the flat side; and

(4) the fragment enters from the bottom flat side of the cylinder and exits the cylinder through the curved side.

Once it has been determined that the fragment will hit the target it is necessary to calculate which polyhedrons are hit, when and where. This is done using the methodology discussed in
Chapter 6. The polyhedrons in the outer skin influences the momentum of the fragment, whilst it is important to record which vital parts are hit, and when.

Having calculated which polyhedrons where hit, by which fragment, and knowing the velocity of the fragment at the time of the hit, the probability of each component failing due to the cumulative effect of all the fragments that hit it is calculated. Having these probabilities the probabilities of the various systems, and resultantly a specific event occurring, breaking down due to the breakdown of a critical part, can be calculated. These events are:

- \( F \) - Aircraft returns to base (No crash);
- \( G \) - Mission accomplished;
- \( H \) - Mission accomplished and aircraft returns to base.

Using the probabilities for these three events the probabilities for the following events:

- \( A \) - Mission made impossible, immediate crash;
- \( B \) - Mission made impossible, no crash;
- \( C \) - Mission made possible, delayed crash;
- \( D \) - Mission possible, no crash, repairs must be made,

due to the damage of all the fragments ejected from the warhead at the specific burst point are calculated.

Finally, as mentioned earlier, the mean of the probabilities of the events is calculated over all the burst points and reported.

The Upshot programme, and effectively the effectiveness model, will be now be discussed modularly in accordance with the model seen in Figure 7.1.

### 7.2.1 Read Data

The following input is required by the effectiveness model: the attack geometry of the missile and the target, the data describing the warhead, the penetration data for various fragment types through alluminium and crater calculations, the data describing the target with regard
to the target geometry as well as the systems it is built up of.

7.2.1.1 Read the attack geometry

Within this subroutine data, describing both the geometry of the missile and the target, are read in. The missile data are:

- \( I_{\text{drag}} \): A boolean variable indicating whether drag is taken into account or not in the calculations. If drag is taken into account \( I_{\text{drag}} = 1 \), and if not, \( I_{\text{drag}} = 0 \).
- \( v_{\text{am}} \): The absolute velocity of the missile.
- \( r_m \): The direction of the missile in the earth-system.
- \( ME \): A transformation matrix converting coordinates from the missile coordinate system to the earth coordinate system.
- \( n_{\text{is}} \): The number of intercept scenarios.
- \( C_p \): The coordinates of missile path \( p \).

The target data:

- \( v_{\text{at}} \): The absolute velocity of the target.
- \( r_t \): The direction of the target in the earth-system.
- \( ET \): A transformation matrix converting coordinates from the earth coordinate system to the target coordinate system.
- \( C_t \): The coordinates of the target.

The velocity vectors of the missile and target, \( \mathbf{v}_{\text{me}} \) and \( \mathbf{v}_{\text{te}} \), in the earth-system are calculated by multiplying the absolute velocities with the direction of the target and missile respectively,

\[
\mathbf{v}_{\text{me}} = v_{\text{am}} \times r_m
\]

and

\[
\mathbf{v}_{\text{te}} = v_{\text{at}} \times r_t.
\]

Calculating the transpose of the transformation matrices, the transformation matrices from missile- and target-to earth coordinates, \( ME \) and \( TE \), are calculated respectively,

\[
EM = ME^T
\]

and

\[
TE = ET^T.
\]

7.2.1.2 Read the Warhead description

All the information describing the warhead is read in, in this subroutine. This information is,

- \( n_f \): The number of fragments in the warhead.
- \( C_f(i) \): The coordinates of fragment \( i \).
- \( r_f(i) \): The direction of fragment \( i \).
- \( v_f(i) \): The absolute velocity of fragment \( i \).
- \( m_f(i) \): The mass of fragment \( i \).
- \( l_f(i) \): The length of fragment \( i \).
As was indicated in Section 5.5.1, one of three fragment types are simulated. These are balls, cubes, and rectangles. Upshot calculates the projected area for each fragment according to the formula developed by Janzon [25], as well as the equivalent ball diameter of this area. These formulae are given in Table 7.1. (Note that \( \pi = 3.141592654 \) in these calculations)

<table>
<thead>
<tr>
<th>Fragment Shape</th>
<th>Projected Area of fragment ( i, A_f(i) )</th>
<th>Ball Diameter of fragment ( i, D_f(i) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ball</td>
<td>( \frac{\pi l_f(i)}{4} )</td>
<td>( \sqrt{\frac{A_f(i)}{\pi}} )</td>
</tr>
<tr>
<td>Cube</td>
<td>( \frac{3l_f(i)}{2} )</td>
<td>( \sqrt{3l_f(i)} )</td>
</tr>
<tr>
<td>Rectangle</td>
<td>( \frac{(l_f(i)b_f(i)+l_f(i)h_f(i)+b_f(i)h_f(i))}{2} )</td>
<td>( \sqrt{2A_f(i)} )</td>
</tr>
</tbody>
</table>

Table 7.1: Formula for the average projected area (\( m^2 \)) and equivalent ball diameter (\( m \)) of this area.

By multiplying the average projected area and the equivalent ball diameter by a thousand (1000) it is converted from meter to millimeter,

\[
A_f(i) = A_f(i) \times 1000
\]

and

\[
D_f(i) = D_f(i) \times 1000.
\]

The fragment velocity vector, for fragment \( i, v_f(i) \), is calculated in the missile coordinate system by multiplying the direction of fragment \( i, r_f(i) \), with the absolute velocity of the fragment, \( v_f(i) \),

\[
v_f(i) = r_f(i) \times v_f(i).
\]

7.2.1.3 Read the target description

All the data describing the target, e.g. the outer geometry described by polyhedrons, and the vital parts described by both polyhedrons and line-elements, are read in, in this subroutine. The target model has already been discussed in Section 2.3 and Chapter 3, and the format in which the data are stored, after being converted for Upshot, is given in Appendix A.3.1.

After the name of the target, Target_Name, is read in, the remainder of the data are read in in three sections; the polyhedrons describing the outer geometry of the target, the polyhedrons describing the vital parts, and the line elements describing the vital parts.

Data for the outer description of the target

\( N_{pol} \) The number of polyhedrons describing the outer geometry.

The following is read in for each of the \( N_{pol} \) polyhedrons:
\( c_i = (x_i, y_i, z_i) \)
The coordinates of the internal point of the polyhedron.

\( d_{\text{max}}(i) \)
The maximum distance from the internal point to a corner of polyhedron \( i \).

\( pt_p(i, j), j = 1, 2, 3 \)
The internal structure in \textit{mm} Aluminium per meter length (three (3) values).

\( pt_f(i, j), j = 1, 2, 3 \)
The fraction, in percentage, for each value, \( pt_p(i, j) \), of the structure.

\( NS_{\text{pol}}(i) \)
The number of surfaces for polyhedron \( i \).

\( n_{\text{pol}}(m, i) \)
The normal vector for surface \( m \) of polyhedron \( i \).

\( d_{\text{pol}}(m, i) \)
The perpendicular distance from the internal point, \( c_i \), to surface \( m \) of polyhedron \( i \).

\( th_{\text{pol}}(m, i) \)
The thickness of surface \( m \) of polyhedron \( i \).

The surface specific information is read in for each of the \( NS_{\text{pol}}(i) \) surfaces of polyhedron \( i \).

The expected value for the internal protection of polyhedron \( i \), \( \text{Prot}_{\text{internal}}(i) \), is calculated by multiplying the probability of hitting a certain internal structure with the thickness of that structure and taking the sum of these values,

\[
\text{Prot}_{\text{internal}}(i) = pt_p(1, i) \times pt_f(1, i) + pt_p(2, i) \times pt_f(2, i) + pt_p(3, i) \times pt_f(3, i). \tag{7.1}
\]

\textit{Polyhedron data describing the vital parts of the target}

\( N_{\text{vpol}} \)
The number of polyhedrons describing the vital parts of the target.

The following is read in for each of the \( N_{\text{vpol}} \) polyhedrons:

\( C_{\text{kill,crit}}(i) \)
Category for the kill criteria. (1 - Personal, 2 - Ordinary Component, 3 - Fuel Tank, 4 - Wiring)

\( P_k(i) \)
Information to calculate the probability of kill.

\( Ip_i = (x_i, y_i, z_i) \)
The coordinates of the internal point of the polyhedron.

\( d_{\text{cmax}}(i) \)
The maximum distance from the internal point to a corner of polyhedron \( i \).

\( \text{Int}_{\text{prot}}(i) \)
Internal protection in \textit{mm} Al for vital part \( i \).

\( NS_{\text{vpol}}(i) \)
The number of surfaces for the vital part polyhedron \( i \).

\( n_{\text{vpol}}(m, i) \)
The normal vector for surface \( m \) of the vital part polyhedron \( i \).

\( d_{\text{vpol}}(m, i) \)
The perpendicular distance from the internal point, \( Ip_i \), to surface \( m \) of polyhedron \( i \).

\( th_{\text{vpol}}(m, i) \)
The thickness of surface \( m \) of the vital part polyhedron \( i \).

The surface specific information is read in for each of the \( NS_{\text{vpol}}(i) \) surfaces of the vital part polyhedron \( i \).

\textit{Line-element data describing the vital parts of the target}

\( N_{\text{vline}} \)
The number of lines describing the vital parts of the target.

The following is read in for each of the \( N_{\text{vline}} \) lines:
7.2. Upshot

\[ v_{\text{begin}}(i) = (x_{bi}, y_{bi}, z_{bi}) \] Coordinates for the starting point of the line-element.

\[ v_{\text{end}}(i) = (x_{ei}, y_{ei}, z_{ei}) \] Coordinates for the end point of the line-element.

\[ D_{\text{line}}(i) \] The diameter of line-element \( i \).

\[ P_{\text{rot out}}(i) \] The outer protection of line-element \( i \) in millimeter Aluminium.

\[ P_{\text{rot in}}(i) \] The internal protection of line-element \( i \) in millimeter Aluminium.

\[ C_{\text{kill line}}(i) \] Category for the kill criteria. (1 - Personal, 2 - Ordinary Component, 3 - Fuel Tank, 4 - Wiring)

\[ P_{\text{kill}}(i) \] The kill probability of line-element \( i \).

The line-element information is read in for each of the \( N_{\text{line}} \) line-elements.

Finally the total number of vital components for the target, \( n_{\text{components}} \), is calculated,

\[ n_{\text{components}} = N_{\text{pol}} + N_{\text{line}}. \]

### 7.2.1.4 Read the penetration data

The penetration and crater data is read in from two data files, “Pentrat.dat” and “penet_cr.dat” respectively, in this routine. This data include the constants, \( C_{ki} \), used in the penetration formula as empirically derived by Thor [27]. Each fragment type has five constants, \( i \in \{1, 2, 3, 4, 5\} \), associated with it. The fragment types accommodated and the respective constants for penetration through aluminium are given in Table 7.2.

<table>
<thead>
<tr>
<th>( k )</th>
<th>Type</th>
<th>( 1 )</th>
<th>( 2 )</th>
<th>( 3 )</th>
<th>( 4 )</th>
<th>( 5 )</th>
<th>Density (g/cc)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Uranium</td>
<td>9.992</td>
<td>1.486</td>
<td>1.531</td>
<td>1.518</td>
<td>0.484</td>
<td>18.676988</td>
</tr>
<tr>
<td>2</td>
<td>Tungsten Alloy</td>
<td>11.620</td>
<td>1.530</td>
<td>1.640</td>
<td>1.500</td>
<td>0.928</td>
<td>16.8189</td>
</tr>
<tr>
<td>3</td>
<td>Steel</td>
<td>6.571</td>
<td>0.944</td>
<td>0.990</td>
<td>1.130</td>
<td>0.076</td>
<td>7.76873</td>
</tr>
<tr>
<td>4</td>
<td>Aluminium</td>
<td>7.233</td>
<td>1.017</td>
<td>1.141</td>
<td>1.186</td>
<td>0.174</td>
<td>2.80315</td>
</tr>
</tbody>
</table>

Table 7.2: The constant values for the various fragment types as per Thor [27].

Thor’s report mainly deals with the comparison of the performance of tungsten and steel fragments. The crater data \( P_{ijv} \) and \( P_{ijd} \) for these two fragment materials that are imported are given in Table 7.3, where \( i \) indicates the type and \( j \) is the data set number.

<table>
<thead>
<tr>
<th>( l )</th>
<th>Type</th>
<th>( j )</th>
<th>Impact velocity (m/s)</th>
<th>Crater diameter / Fragment diameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Steel</td>
<td>1</td>
<td>0.00</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>1300.00</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>4000.00</td>
<td>3.05</td>
</tr>
<tr>
<td>2</td>
<td>Tungsten</td>
<td>1</td>
<td>0.00</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>1080.00</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>4000.00</td>
<td>3.50</td>
</tr>
</tbody>
</table>

Table 7.3: The data for crater calculations.
7.2.1.5 Read the system characteristics

All the data describing the target system according to, how the target is assembled by subsystems, and how these subsystems are assembled by components (vital parts) are read in in this subroutine. What this data comprise of and how it is read in will now be discussed.

\[ N_{syst} \]

The number of systems:

The following is read in for each of the \( N_{syst} \) systems.

\[ S_{Description}(i) \]
The description of system \( i \).

\[ P_A(i) \]
The probability that the mission will be aborted and the target will be killed should system \( i \) fail.

\[ P_B(i) \]
The probability that the mission will be aborted, but the target is not killed should system \( i \) fail.

\[ P_C(i) \]
The probability that the mission will be accomplished and the target will be killed should system \( i \) fail.

\[ P_D(i) \]
The probability that the mission will be accomplished, the target is not killed, but does need repairs should system \( i \) fail.

\[ N_{comp}(i) \]
The number of components in system \( i \):

The following is read in for each of the \( N_{comp}(i) \) components in system \( i \).

\[ Comp_{i,j} \]
The number of the \( j \)th component in system \( i \).

\[ P_{hit}(i,j) \]
The probability of component \( j \) in system \( i \) breaking down given a hit.

\[ Comp(k) \]
The subsystem of which component \( k \) is a part of.

7.2.2 Initiation of Variables

There are two subroutines, Init_1 and Init_2, for setting the values of various variables to zero.

Init_1 sets the sum of the probability for a specific event occurring, see Section 2.7, over all the paths, to zero,

\[ P(A) = P(B) = P(C) = P(D) = 0. \]

The hit counters are also set to zero in Init_1. The counters are the number of hits on the outer skin of the target during intercept scenario (path) \( i \), \( n_{ohit}(i) \); the number of hits on the outer skin of the target during all the intercept scenarios, \( N_{ohit} \); the number of hits on critical component \( j \), \( N_{vhit}(j) \), over all paths; and the number of hits on vital component \( j \) of the target for intercept scenario (path) \( i \), \( n_{vhit}(i,j) \).

The subroutine Init_1 is called once, before the loop that runs over all the paths, and as can be seen from the definitions of the variables in the previous paragraph, the variables are applicable to all the hits on the target or vital part for each path or all the intercept scenarios.

Init_2 sets the “damage-variables”, the probability of a specific component failing, \( P_{comp}(i) \),
to zero for each intercept scenario (path) that is simulated. It is the first subroutine in the loop over all paths which also include the loop over all fragments and all systems, as can be seen in Figure 7.1. In the following sections these two loops will be discussed.

### 7.2.3 The loop over all Fragments

#### 7.2.3.1 Initiate Fragment

This subroutine calculates the fragment velocity relative to the target velocity, in the target coordinate system, for each new fragment.

First the fragment velocity of fragment \( i \), relative to the missile, is transformed from the missile coordinate system to the earth coordinate system,

\[
\mathbf{v}_{fe}(i) = \mathbf{ME} \times \mathbf{v}_f(i). \tag{7.2}
\]

The fragment velocity in earth coordinates is now added to the missile velocity, which is in earth coordinates,

\[
\mathbf{v}_{fme}(i) = \mathbf{v}_{me} + \mathbf{v}_{fe}(i). \tag{7.3}
\]

The target velocity is now subtracted from the fragment velocity relative to that of the missile,

\[
\mathbf{v}_{fte}(i) = \mathbf{v}_{fme}(i) - \mathbf{v}_{te}. \tag{7.4}
\]

The velocity of the fragment in then transformed from the earth coordinate system to the target coordinate system,

\[
\mathbf{v}_{ftt}(i) = \mathbf{ET} \times \mathbf{v}_{fte}(i), \tag{7.5}
\]

and the absolute velocity of fragment \( i \) relative to the target,

\[
V_f(i) = \sqrt{\mathbf{v}_{ftt}(i) \cdot \mathbf{v}_{ftt}(i)}. \tag{7.6}
\]

Next the relative fragment coordinates need to be calculated. First the fragment coordinates must be transformed from missile coordinates to earth coordinates,

\[
\mathbf{C}_{fe}(i) = \mathbf{ME} \times \mathbf{C}_f(i). \tag{7.7}
\]

and the missile coordinates, on path \( p \), must be transformed from the target coordinates system to the earth coordinates system,

\[
\mathbf{C}_{pe} = \mathbf{TE} \times \mathbf{C}_p. \tag{7.8}
\]

By adding the fragment coordinates to the coordinates of the missile and subtracting the coordinates of the target, and multiplying this vector with the transformation matrix \( \mathbf{ET} \), the coordinates of fragment \( i \) relative to the position of the target, in earth coordinates, is calculated,

\[
\mathbf{f}_{pos}(i) = (\mathbf{C}_{pe} + \mathbf{C}_{fe}(i) - \mathbf{C}_{te}) \times \mathbf{ET}. \tag{7.9}
\]

In other words, \( \mathbf{f}_{pos}(i) \) is the start coordinates of the fragment in the target coordinate system.
7.2.3.2 Fragment Trajectory

The purpose of this subroutine is to determine whether or not the fragment will hit the target. If it is the case it is recorded.

The following variables are input parameters of the subroutine:

- $v_{ftt}$: The relative fragment velocity between the target and the fragment in the target coordinate system, and
- $f_{pos}$: the position of the fragment in the target coordinate system.

The results of the subroutine are four output parameters:

- $\text{Hit\_Target}$: Is a logical flag (a Boolean variable) which is true if the fragment will hit the target,
- $\text{In\_Target}$: a logical flag which indicates a direct hit, in other words the warhead explodes within the target,
- $\underline{v}_{ftt}$: the new fragment velocity, in the target coordinate system, taking drag into account and positioned near the first polyhedron to be hit by the fragment, and
- $F_{pos}$: the new fragment position near the first polyhedron to be hit.

First the two logical flags, $\text{Hit\_Target}$ and $\text{In\_Target}$, are assigned a False value. It is thus excepted to be false unless proven, or calculated, to be true.

To effectively do so the impact of drag, should it be considered, is calculated in a subroutine, vel\_polyh. The calculation of the influence of drag will now be discussed.

Calculating the velocity of a fragment after drag has been taken into account

For this calculation the fragment velocity relative to the earth, $\underline{v}_{fme}(i) = (x_{vf}(i), y_{vf}(i), z_{vf}(i))$, and not the target is used, as well as the fragment direction in earth coordinates, $\underline{r}_{f}(i) = (x_{rf}(i), y_{rf}(i), z_{rf}(i))$

\[
\begin{align*}
x_{rf}(i) &= \frac{x_{vf}(i)}{v_{fme,abs}(i)}, \\
y_{rf}(i) &= \frac{y_{vf}(i)}{v_{fme,abs}(i)} \\
z_{rf}(i) &= \frac{z_{vf}(i)}{v_{fme,abs}(i)},
\end{align*}
\]

where

\[v_{fme,abs}(i) = \|\underline{v}_{fme}(i)\|\).

The drag coefficient, $C_w$, is read in from the tables provided in Appendix G which is given in terms of the Mach number and the fragment shape. The Mach number is calculated by dividing the absolute velocity by the constant value 343.64,

\[V_{mach} = \frac{v_{fme,abs}(i)}{343.64}.
\]
Within this subroutine the air density, \( \rho \) (kg/cubic m), is calculated at a height of \( h = 305m \) (1000 ft) above sea level, making use of Janzon’s formula [25] which is valid for \( h < 1000m \). This is necessary since the density of the air is one of the factors in the formula for the drag force, \( D = \frac{1}{2}C_\rho A v^2 \). In this subroutine a number of the factors are multiplied giving \( \alpha \). The air density at a height \( h \) is,

\[
\rho = \frac{350}{278 - 0.006h} \exp\left(\frac{-0.034h}{278 - 0.006h}\right),
\]

and

\[
\alpha = \frac{C_w \rho A f(i)}{m_f(i)}.
\]

The next step in the subroutine vel_polyh is to calculate the fragment coordinates in earth coordinates. This is done by transforming the fragment coordinates, that are relative to the target, \( f_{\text{pos}}(i) \), to coordinates in the earth coordinate system, and adding this to the target coordinates in the earth coordinate system, \( C_t \),

\[
f_{\text{epos}}(i) = (TE \times f_{\text{pos}}(i)) + C_t.
\]

The following section within the subroutine is repeated for each of the \( N_{\text{pol}} \) polyhedrons describing the outer skin of the target.

Calculate the coordinates of the internal point of polyhedron \( i \) in the earth coordinate system, \( \mathcal{L}_{\text{earth}}^i \), by transforming the coordinates of the internal point, \( c_i \), from target coordinates to earth coordinates and adding the position of the target in earth coordinates to it,

\[
\mathcal{L}_{\text{earth}}^i = (c_i \times TE) + C_t.
\]

The next step is to define a cylinder around the polyhedron of which the curved area is parallel to the direction of the target path, as illustrated in Figure 7.3. The dimensions of the cylinder are calculated as follows:

First the diameter, \( d_{\text{cyl}} \) is calculated by multiplying two with the maximum distance from the internal point to a corner of the polyhedron. In this subroutine the value that is used is \( r_{sq} = d_{\text{max}}^2 \), thus

\[
d_{\text{cyl}} = 2 \sqrt{r_{sq}}.
\]

The absolute size of the direction of the target in earth coordinates, \( r_{\text{abs}}^{te} \), is now calculated,

\[
r_{\text{abs}}^{te} = \sqrt{r_t \cdot r_{\text{abs}}}. 
\]

The coordinates of the centre point on both ends of the cylinder can now be calculated using the above calculated values. It can be seen as the beginning, \( C_s \), and ending, \( C_e \), coordinates. The formulas for \( C_s \) and \( C_e \) are

\[
C_s = c_{\text{earth}}^i - \left[ \frac{d_{\text{cyl}}}{2} \right] \times \frac{r_t}{r_{\text{abs}}^{te}}
\]

and

\[
C_e = C_s + \left[ \frac{t \times v_{at} \times r_{\text{abs}}^{te}}{r_{\text{abs}}^{te}} \right] + \left[ \frac{d_{\text{cyl}}}{2} \right] \times \frac{r_{\text{abs}}}{r_{\text{abs}}^{te}}
\].
Figure 7.3: The cylinder around a polyhedron.

where \( t = 60 \text{s} \).

Whether or not the fragment will intersect this cylinder, and if the fragment line intersects the cylinder, at what point in “time” is now calculated as per the discussion in Section 7.2 referring to Figure 7.2.

The length, \( l_{cyl} \), of the cylinder is calculated by subtracting the beginning, \( C_s \), from the ending, \( C_e \), coordinates of the cylinder, \( \vec{k} \), calculating the dot product of this vector with itself, and taking the square root of this value,

\[
\vec{k} = (C_e - C_s) \\
l_{cyl} = \sqrt{\vec{k} \cdot \vec{k}}.
\]

The direction vector of the cylinder is then normalised,

\[
\vec{k}_u = \frac{\vec{k}}{l_{cyl}}.
\]

The shortest distance between the path (line) the fragment is traveling on and the beginning, \( C_s \), and ending, \( C_e \), coordinates of the cylinder is now calculated, these are intercept scenarios 1 and 4 from above. First the coordinate of a point on the line is subtracted from the point from which the shortest distance to line is being calculated, the subroutine in Upshot that calculates the distance from a point to a line is \( \text{Dist\_point\_line} \). The point on the line is taken as the coordinate of the fragment, \( f_{\text{epos}}(i) \),

\[
\vec{u}_s = C_s - f_{\text{epos}}(i)
\]

and

\[
\vec{u}_e = C_e - f_{\text{epos}}(i).
\]
The magnitude of the vector, $\mathbf{u}$, is calculated,

$$u = \sqrt{\mathbf{u} \cdot \mathbf{u}}.$$ 

If $u_s$ or $u_e$ is equal to zero it follows that the point under investigation and the point on the line lie on the same coordinate, which implies that the distance between the two points is zero, $d = 0$, and that the shortest “time” between the line and point is also zero, $t = 0$.

If $u_s$ or $u_e$ is not equal to zero, $u_s$ or $u_e \neq 0$, more calculations are required to determine the “time”, $t$, and distance, $d$. To calculate the “time” the absolute size of the fragment direction vector is calculated,

$$r_{f,Abs} = \sqrt{\mathbf{r}_f \cdot \mathbf{r}_f}.$$ 

From the definition of the Euclidean inner product and trigonometric identities the cosines and sinus of the angle $\theta$, which is indicated in Figure 7.4 where $\theta = \theta_s$ and $\theta_e$ for $\mathbf{u} = \mathbf{u}_s$ and $\mathbf{u}_e$ respectively, are respectively calculated as,

$$\cos(\theta) = \frac{\mathbf{u} \cdot \mathbf{r}_f}{u \times r_{f,Abs}}$$

and

$$\sin(\theta) = \sqrt{1 - \cos^2(\theta)},$$

with the absolute value of $1 - \cos^2(\theta)$ being taken as a precautionary measure in preventing floating point errors due to small numbers.

![Figure 7.4](image)

**Figure 7.4: The cosines and sinus of the angle $\theta$.**

The distance, $d$, and the “time”, $t$, are now calculated as,

$$d = u \times \sin(\theta)$$

and

$$t = \frac{u \times \cos(\theta)}{r_{f,Abs}}.$$
As was mentioned the above calculations are done for both the beginning, \(C_s\), and ending, \(C_e\), coordinates of the cylinder respectively returning \((d_s, t_s)\) and \((d_e, t_e)\). Before this calculation is done the variable indicating the hit status, \(ic\), with zero indicating no hit and one indicating a hit, is set to zero, as well as the variable indicating the minimum time, \(t_{min}\), is set to infinity, \(t_{min} = \infty\). Subsequent to first \((d_s, t_s)\) being calculated, and then \((d_e, t_e)\) the following process is followed to determine whether or not there will be a hit as well as the “time” to that hit.

If \(d \leq d_{cyl}^2\) and \(t \leq t_{min}\), then

\[
ic = 1, \\
t_{min} = t
\]

and

\[
d_{min} = d,
\]

where \(d_{min}\) is the minimum distance of the two.

Also important to calculate is the distance between the centreline of the cylinder and the path of the fragment, the subroutine in Upshot in which this is calculated is \(Dist_{\text{line\_line}}\). The result of the subroutine is the “shortest time” and distance for intercept scenarios 2 and 3 in Figure 7.2. First a vector, \(n_k\), which is normal to the two lines is calculated by taking the cross product of the direction vector of centerline, \(k_u\), and the path, \(r_f\),

\[
n_k = r_f \times k_u,
\]

followed by the calculation of the absolute value of this normal vector,

\[
n_{k,abs} = \sqrt{n_k \cdot n_k}.
\]

Also to be calculated is the vector between the two points of which one is on each line. The two points used are \(C_s\) and \(f_{\text{epos}}\) with the resultant vector from taking the difference being \(u_s\). If \(n_{k,abs}\) is equal to zero, it follows that the lines are parallel. If this is the case, then the distance between the coordinate of the fragment and the cylinder axes, \(d_c\), as well as the time, \(t_c\), is calculated in a similar fashion as earlier in \(Dist_{\text{point\_line}}\). Also \(t_f\) and \(t_c\) are set equal to infinity, \(t_f = t_c = \infty\), where \(t_f\) and \(t_c\) is the shortest time to a point on the fragment path and cylinder axis respectively. If \(n_{k,abs} \neq 0\) the distance, \(d_c\), is calculated as,

\[
d_c = \frac{u_s \cdot n_k}{n_{k,abs}}.
\]

The resultant vector, \(B\), of the vector between the fragment coordinate and the point on the cylinder axis, and the vector that has the magnitude of the shortest distance between the fragment path and axes and that lies normal on both of these two lines, as shown in Figure 7.5, is calculated next,

\[
B = (C_s - f_{\text{epos}}) + d_c = \begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix},
\]

where,

\[
d_c = d_c \times \left( \frac{n_k}{n_{k,abs}} \right).
\]
The problem that needs to be solved now is to find the \( x = (x_1, x_2) \) in the equation \( A \times x = B \). Where,

\[
A = \begin{bmatrix} x_{rf} & -k_x \\ y_{rf} & -k_y \end{bmatrix}
\]

and

\[
B = \begin{bmatrix} x_B \\ y_B \end{bmatrix}
\]

or

\[
A = \begin{bmatrix} x_{rf} & -k_x \\ z_{rf} & -k_z \end{bmatrix}
\]

and

\[
B = \begin{bmatrix} x_B \\ z_B \end{bmatrix}
\]

or

\[
A = \begin{bmatrix} y_{rf} & -k_y \\ z_{rf} & -k_z \end{bmatrix}
\]

and

\[
B = \begin{bmatrix} y_B \\ z_B \end{bmatrix}
\]

for \( k_u = (k_x, k_y, k_z) \).

The two-equation system is solved making use of Cramer’s rule. To use this rule it must be shown that the system has a unique solution, that the matrix \( A \) is not singular. This is the case if the determinate of matrix \( A \) is not equal to zero, \( \det(A) \neq 0 \), where,

\[
\det(A) = a_{11}a_{22} - a_{12}a_{21}.
\]

Given that the system has a unique solution, \( x_1 \) and \( x_2 \) is calculated as follows,

\[
x_1 = \frac{b_1a_{22} - b_2a_{12}}{\det(A)}
\]
and
\[ x_2 = \frac{b_2a_{11} - b_1a_{21}}{\det(A)}. \]

The results of the above calculations are,
\[ t_f = x_1, \]
\[ t_c = x_2 \]
and
\[ d = |d|. \]

If \( x_2 \) is directed into the cylinder, \( x_2 \geq 0 \), and the length of \( x_2 \) is less than that of the cylinder, \( x_2 \leq l \), indicating that it ends within the cylinder, there is a possibility that the fragment hits the cylinder in a similar scenario than intercept scenario 2 or 3. The “time” is now compared to the shortest of the previously tested intercept scenarios and if \( d \leq \frac{d_{cyl}}{2} \) and \( t_f \leq t_{min} \), then
\[ ic = 1, \]
\[ t_{min} = t_c \]
and
\[ d_{min} = d. \]

Having the above information the range, \( R \), which is the distance from the original coordinate of the fragment to where the hit takes place is calculated,
\[ X_s = L_f \times t_{min} \]
and
\[ R = \sqrt{X_s \cdot X_s} - \sqrt{\left(\frac{d_{cyl}}{2}\right)^2 - d_{min}^2}, \]
as is illustrated in Figure 7.6.

![Figure 7.6: A diagrammatic illustration of the various variables used to calculate the range, R.](image)

If the above calculations indicate that the fragment did hit the target, \( ic = 1 \), then the new velocity, \( v_x \), is calculated as
\[ v_x = v_{fme, abs} \times e^{-\alpha R}, \]
as well as the average velocity, $v_{\text{avg}}$, taking the effect of drag into account,

$$v_{\text{avg}} = \frac{-v_{\text{free,abs}}(e^{-\alpha R} - 1)}{\alpha R}.$$ 

It is important to note that the fragment velocity only changed in magnitude and not in direction, thus,

$$v_x = \mathcal{L}_f \times v_x$$

and

$$v_{\text{avg}} = \mathcal{L}_f \times v_{\text{avg}}.$$ 

This new fragment velocity that is in the earth coordinate system must be relative to the target. To calculate this velocity the target velocity must be subtracted from the new fragment velocity and this velocity must be transformed to target coordinates,

$$V_x(i) = ET \times (v_x - v_{\text{te}})$$

and

$$V_{\text{avg}}(i) = ET \times (v_{\text{avg}} - v_{\text{te}}).$$

This is calculated for each of the polyhedron in the outer skin, where $V_x(i)$ is the velocity of the fragment just before hitting polyhedron $i$ in the outer skin.

If the above calculations indicate that the fragment did, however, not hit polyhedron $i$, $ic = 0$, then

$$V_x(i) = 0$$

and

$$V_{\text{avg}}(i) = 0.$$ 

As indicated earlier, these velocities are calculated for each polyhedron with regard to the specific fragment of which the trajectory is being followed at this stage given that drag needs to be taken into account, $I_{\text{drag}} = 1$. If this is the case, then

$$v_{\text{frag}} = V_{\text{avg}}(i),$$

subsequent to this velocity being determined for each polyhedron, otherwise

$$v_{\text{frag}} = v_{\text{ftt}}(i).$$

The velocity at which a fragment will hit the target if drag is taken into account has now been calculated. Having all of the above information it can now be calculated whether a fragment path will intersect a polyhedron, in other words whether the fragment hits the polyhedron, and if so, where. This is done following the same methodology as in Section 6.3.3. For the model which is referred to in Section 6.3.3 a detection sensor is the same as a fragment, and a beam the same as the path of the fragment. The variables used in the two applications of the model are given in Table 7.4.

One of four possible results is returned, these are given in Table 7.5.

If IC = 3 it follows that the warhead is in the target, which means a direct hit, and no more testing is required. If $T(1) < 0$ it follows that the fragment has not yet passed the polyhedron
Table 7.4: The relation between the variables used in Upshot and Fuze.

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Description</th>
<th>Variable Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_i$</td>
<td>Internal Point in Polyhedron $i$</td>
<td>$c_i$</td>
<td>Same as for Fuze</td>
</tr>
<tr>
<td>$p_f$</td>
<td>The coordinate of a point on the beam path.</td>
<td>$f_{pos}(i)$</td>
<td>Coordinates of the fragment in the target coordinate system.</td>
</tr>
<tr>
<td>$r_{pf}$</td>
<td>The direction of the beam</td>
<td>$v_{frag}$</td>
<td>Fragment velocity in the target coordinate system (the velocity is the relative velocity between the target and fragment).</td>
</tr>
<tr>
<td>$d_{max}$</td>
<td>The maximum distance from $p_i$ to any point on the surface of the polyhedron $i$.</td>
<td>$d_{max}(i)$ and $d_{cmax}(i)$</td>
<td>Same as for Fuze with $max$ to the outer skin and $cmax$ the critical parts being for the polyhedrons describing.</td>
</tr>
<tr>
<td>$d$</td>
<td>The distance between inside point and each side on the polyhedron.</td>
<td>$d_{pol}(m, i)$ and $d_{vpol}(m, i)$</td>
<td>The orthogonal distance from the internal point, $c_i$, to surface $m$ of polyhedron $i$ for the surface and vital part polyhedrons respectively.</td>
</tr>
<tr>
<td>$n$</td>
<td>The normal vector of a side of the polyhedron.</td>
<td>$n_{pol}(m, i)$ and $n_{vpol}(m, i)$</td>
<td>The normal vector for side $m$ of surface- and vital part polyhedron $i$ respectively.</td>
</tr>
</tbody>
</table>

Table 7.5: The IC Flag indicating the hit status.

<table>
<thead>
<tr>
<th>ic</th>
<th>Hit status</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No hit</td>
</tr>
<tr>
<td>1</td>
<td>Hit in edge</td>
</tr>
<tr>
<td>2</td>
<td>Hit</td>
</tr>
<tr>
<td>3</td>
<td>Inside polyhedron</td>
</tr>
</tbody>
</table>

which means that the fragment moved away from the polyhedron. It thus follows that if IC = 0 or IC = 3, and $T(1) \geq 0$ the target was hit from the outside and that $n_{ohit}(i)$ is increased by one as well as $N_{ohit}$.

7.2.3.3 Check which components are hit

Given that the target was hit from the outside, various calculations follow to determine the effect of a the fragment on the target. These are to:

- Calculate the events of all the hits on the outer skin of the target and sort according to the “time to hit” in ascending order;
- Calculate the events of all the hits in the vital parts (both polyhedrons and line-elements) and sort according to the “time to hit”;

118
• Calculate the magnitude of the fragment velocity and store the outside velocity; and
• Analyse the fragment path through the target.

*Calculate the events of all the hits on the outer skin of the target and sort according to the “time to hit”*

If $I_{\text{drag}} = 1$, then

$$v_{\text{frag}} = V_{\text{avg}}(i)$$
and

$$V_{\text{frag}} = V_x(i),$$

otherwise

$$v_{\text{frag}} = v_{\text{ftt}}(i)$$
and

$$V_{\text{frag}} = v_{\text{ftt}}(i).$$

Reset the number of polyhedrons hit by the fragment, $n_p$, to zero (0). The following loop runs over all the polyhedrons in the outer skin.

First it is checked whether or not the fragment will hit the current polyhedron under investigation. This is done using the same model discussed above for which the reader was referred to Section 6.3.3. Given the results of this model, it follows that polyhedron $i$ was hit if $ic = 1$ or $ic = 2$ and $T(1) \geq 0$. If this is the case

$$n_p = n_p + 1.$$ 

The magnitude of the velocity of the fragment, when it is going into and out of the polyhedron on the $n_p^{th}$ hit, is set as

$$v_{\text{in}}(n_p) = v_{\text{out}}(n_p) = \sqrt{V_{\text{frag}} \cdot V_{\text{frag}}}.$$ 

The time when the fragment hits and leaves the polyhedron on the $n_p^{th}$ hit is set as

$$t_{\text{in}}(n_p) = T(1)$$
and

$$t_{\text{out}}(n_p) = T(2).$$

The skin thickness on the side entered, $S_{\text{thick}}^{\text{in}}(n_p)$, and the exit side, $S_{\text{thick}}^{\text{out}}(n_p)$, must also be set. For this calculation $th_{\text{pol}}(m, i)$, where $m$ is the side of outer skin polyhedron $i$, is divided by $Cov(1) = -\cos \alpha$ and $Cov(2) = \cos \alpha$, calculated in Section 6.3.3, to calculate the “skew” as illustrated in Figure 7.7. That is,

$$S_{\text{thick}}^{\text{in}}(n_p) = \frac{th_{\text{pol}}(s, i)}{-\cos \alpha}$$
and

$$S_{\text{thick}}^{\text{out}}(n_p) = \frac{th_{\text{pol}}(t, i)}{\cos \alpha}.$$
Figure 7.7: The distance the fragment needs to travel through the sides of the polyhedron.

Finally, the internal protection going into the polyhedron, $Prot_{in}(n_p)$, and before exiting the polyhedron, $Prot_{out}(n_p)$, is set as

$$Prot_{in}(n_p) = 0$$

and

$$Prot_{out}(n_p) = Prot_{internal}(i).$$

The above is done for each of the polyhedrons with respect to the particular fragment. Once this is complete the “hits” are sorted according to the time, $t_{in}$, of the hit. It also follows that the number of surfaces that where hit, $n_{hit}^{p}$, is twice the number of polyhedrons hit, $n_p$, thus,

$$n_{hit}^{p} = 2 \times n_p.$$

At the end of this section the magnitude of the entering and leaving velocities of each of the fragments which hit the target, and for each of the polyhedrons that where hit by each fragment, is stored. Also, the thickness of the side with respect to each hit, as well as the “time” of the hit, and the protection in front of the side.

**Calculate the events of all the hits in the vital parts (both polyhedrons and line-elements) and sort according to the “time” to hit**

For the polyhedrons describing the vital parts exactly the same process is followed as for the polyhedrons describing the outer skin to calculate,

- $v_{in}$ and $v_{out}$ The velocity of the fragment entering and leaving the polyhedron.
- $t_{in}$ and $t_{out}$ The time entering and leaving the vital part.
- $S_{thick}^{in}$ and $S_{thick}^{out}$ The thickness of the polyhedron skin going into and exiting the polyhedron.
- $Prot_{in}$ The internal protection.

Three more results are however stored, namely,
7.2. Upshot

\[ v_{\#}(n_{vp}) \] The number of the vital part that is hit by the \( n_{vp} \)th fragment hit.

\[ v_{\text{Cat}}(n_{vp}) \] The category in which component number \( v_{\#}(n_{vp}) \) falls.

\[ v_{\text{val}}(n_{vp}) \] The probability of kill of the part given a hit.

For the line elements describing the vital parts the same process as for the polyhedrons defining the vital part is followed, excluding the section where the cylinder around the polyhedron is defined and taken into account, and information stored as for the polyhedrons describing the vital parts.

Finally the vital parts (polyhedrons and line-elements) that are hit are sorted according to the time each part was hit.

**Calculate the magnitude of the fragment velocity and stores the outside velocity**

At this stage the magnitude of the fragment velocity after all the hits, \( v_{\text{final}} \), is calculated,

\[
v_{\text{final}} = \sqrt{v_{\text{out}}^2}.
\]

**Analyse the fragment path through the target**

If \( v_{\text{final}} > 0 \) it is confirmed whether the fragment will hit the outer skin, vital part(s), and that there is nothing left to hit at the end of the fragment path. This is done for each fragment after it has been determined which polyhedrons in the outer skin and which vital parts will be hit and at what time. First where the fragment is in “time”, is set to less than zero, \( f_{\text{time}} = -1 \). The last used polyhedron for the outer skin, \( PO_{\text{last}} \), and last used vital part, \( VP_{\text{last}} \) is reset for each new fragment, \( f_{\text{time}} < 0 \), to \( PO_{\text{last}} = 0 \) and \( VP_{\text{last}} = 0 \).

Next the “times” of all the outer polyhadron that were hit are compared to \( f_{\text{time}} \) to find the first one that was hit by the fragment, that is the first one for which \( t_{\text{in}}(i) \geq f_{\text{time}} \), since the hits where earlier sorted in ascending order with regard to the “time” of the hit. If this is the case for any of the hits then \( PO_{\text{time}} = t_{\text{in}}(i) \) and \( PO_{\#} = i \), where \( PO_{\text{time}} \) is the time it was hit and \( PO_{\#} \) which of the recorded outer skin hits actually hit. If \( t_{\text{in}}(i) < f_{\text{time}} \) for all of the recorded hits, it follows that the fragment passed through the target without hitting any part of relevance and \( H_{\text{flag}} = 0 \) is returned.

Should it be found that the specific fragment hit one of the polyhedrons describing the outer skin the first vital part that is hit, if any, is confirmed using the same algorithm as for the outer skin. If this is the case for any of the hits, then \( PV_{\text{time}} = t_{\text{in}}(j) \) and \( VP_{\#} = j \), where \( PV_{\text{time}} \) is the time it was hit and \( VP_{\#} \) which of the recorded hit vital parts was hit first. If \( t_{\text{in}}(j) < f_{\text{time}} \) for all of the recorded hits, it follows that the fragment only hit a part of the outer skin and missed all of the vital parts and \( H_{\text{flag}} = 1 \) is returned, as well as \( f_{\text{old}} = f_{\text{time}} \) and \( f_{\text{time}} = PV_{\text{time}} \).

Should a hit, however, also be confirmed for the vital parts, the “time” at which it occurred needs to be compared to that of the hit on the outer skin, and if \( PV_{\text{time}} \leq PO_{\text{time}} \), then

\[
H_{\text{flag}} = 2, \quad f_{\text{old}} = f_{\text{time}} \quad \text{and} \quad f_{\text{time}} = PV_{\text{time}},
\]
are returned, otherwise

\[ H_{flag} = 1, \]
\[ f_{old} = f_{time} \] and
\[ f_{time} = PO_{time}, \]

are returned.

Should it be found that only the outer skin was hit, \( H_{flag} = 1 \), then by how much the velocity of the fragment is reduced by the internal protection and skin thickness of the polyhedron describing the outer skin that was hit is calculated and the new velocity returned.

First the distance, \( d_{int} \), over which the fragment is exposed to the internal protection is calculated as

\[ d_{int} = (f_{time} - f_{old}) \times v_{in} \times Prot_{internal}(k). \]

This is used to calculate the velocity after penetration which will be set to zero if it is not possible to penetrate the internal protection.

The density of the fragment, \( \rho_f(j) \), will determine the metal values to be used for the calculations, as given in Table 7.6.

<table>
<thead>
<tr>
<th>Density ( \rho_f(j) )</th>
<th>Metal</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \geq 17.8 g/cc )</td>
<td>Uranium</td>
</tr>
<tr>
<td>( 12.0 g/cc \geq \rho_f(j) &lt; 17.8 g/cc )</td>
<td>Tungsten Alloy</td>
</tr>
<tr>
<td>( 5.0 g/cc \geq \rho_f(j) &lt; 12.0 g/cc )</td>
<td>Steel</td>
</tr>
<tr>
<td>( \rho_f(j) &lt; 5.0 g/cc )</td>
<td>Aluminium</td>
</tr>
</tbody>
</table>

Table 7.6: The metal type of the fragment.

Using this the final velocity of the fragment can now be calculated as

\[ v_{frag}(i) = 0.3048 (3.28084v_{in} - A_1 A_2 A_3 A_4), \]

(7.10)

where

\[ A_1 = 10^{C_{k1}} (61.023d_{int}A_f)^{C_{k2}}, \]
\[ A_2 = (15432.36m_f)^{-C_{k3}}, \]
\[ A_3 = 1^{C_{k4}} \]

and

\[ A_4 = (3.28084v_{in})^{-C_{k5}}. \]

If \( v_{frag}(i) < 0 \), then the fragment velocity is set to zero, \( v_{frag}(i) = 0 \).

The same algorithm is used to calculate \( v_{frag}(i) \) after the reduction due to the penetration
of the outer skin itself. The only difference is the variable $v_{in}$ is replaced by the value of $v_{frag}(i)$, just calculated, and the variable $d_{int}$ is replaced by $S_{thick}$.

Should both the outer skin and some vital component have been hit, $H_{flag} = 2$, then the reduction in velocity due to both the internal protection in the outer polyhedron, as well as the outer and internal protection of the vital part must be taken into account. Should the fragment successfully penetrate the internal protection of the vital part, the damage of the part also needs to be calculated.

Once again the same algorithm is used to calculate the decrease in velocity of the fragment due to the internal protection of the outer skin polyhedron, outer protection of the vital part, the inner protection of the vital part, and as a result of the outer protection of the vital part. The same values, $v_{in}$ and $d_{int}$, are used for the calculations with regard to the internal protection of the outer skin polyhedron. For calculating the velocity reduction as a result of the outer protection of the vital part, the new value of $v_{frag}(i)$ is used as the input velocity, and $d_{int} = S_{out}^{int}$. For the velocity penetrating the internal protection of the vital part, the new $v_{frag}(i)$ is once again used as the input velocity, and $d_{int} = S_{in}^{thick}$.

Should there still be a positive velocity, $v_{frag}(i) > 0$, after the calculation of the velocity reduction due to the inner protection of the vital part, the damage (kill probability) of the component is first calculated, before calculating the effect of the outer protection of the vital part on the velocity of the fragment. Also, the number of hits on vital part $j$ is increased by one, inc($N_{vhit}(i,j)$), as well as each time a vital part is hit, inc(#vhit).

Depending on the kill category that the vital component falls into the probability of component $j$ failing, given a hit by fragment $i$, $P_{kill}(X_{ij})$ is calculated as described in Section 7.2.4. Should the component be categorised as being in Kill Category 3 the crater hole area as a result of the fragment is also calculated.

Whether the fragment must be treated as Steel ($l = 1$) or Tungsten ($l = 2$) when calculating the crater size must first be determined. If $\rho_f < 10\, \text{g/cm}^3$ then $l = 1$, otherwise $l = 2$. Knowing this the data set, $P_{ljv}$ and $P_{ljd}$, as defined in Section 7.2.1.4, can be accessed. Using the values in this data set and $v_{frag}(i)$ as input, the size of the crater per fragment diameter can be determined through interpolation, $P_{ld}$. Having this value the crater diameter caused by fragment $i$, $A_i$, can be calculated as

$$A_i = P_{ld} \times D_f(i).$$

After calculating the damage, the effect of the outer protection of the vital part on the velocity of the fragment is calculated as explained above.

### 7.2.4 Kill probabilities for each of the event that occur given a hit

These formulae for the various events were determined empirically and was taken from the SWEAT package for which it has been verified.
7.2.4.1 Kill Category 1

The kill probability of soft targets, e.g. the pilot, are calculated as

\[ P_{\text{kill}}(X_{ij}) = 1 - e^{-\sqrt{\frac{m_f(i) v_{in}(i)^{2.34}}{21}}} , \quad (7.11) \]

where \( m_f(i) \) is the mass of fragment \( i \), \( v_{in}(i) \) is the velocity of fragment \( i \) before penetrating the inner protection of the vital part, and \( P_{\text{kill}}(X_{ij}) \) is the probability of component \( j \) failing, given a hit by fragment \( i \). Having this value, the accumulated probability of component \( j \) failing, taking into account the effect of all the fragments that have been found to hit the component up to this point is,

\[ P_{\text{kill}}(j) = 1 - e^{-\sum_i P_{\text{kill}}(X_{ij})} . \quad (7.12) \]

7.2.4.2 Kill Category 2

The kill probability for hydraulics or corresponding parts are calculated as

\[ P_{\text{kill}}(j) = 1 - (1 - P_k(j))^n , \quad (7.13) \]

where \( P_{\text{kill}}(j) \) is the probability of component \( j \) being killed given that \( n \) out of the \( n_f \) fragments in the warhead hit component \( j \).

7.2.4.3 Kill Category 3

The kill probability for parts like fuel tanks are calculated as

\[ P_{\text{kill}}(j) = \min\left(\frac{\sum A_i}{A_j}, 1\right) , \quad (7.14) \]

where \( A_j \) is the area of the opening in component \( j \) that will cause a hundred percent failure of the component, and \( \sum A_i \) is the sum of the areas of the holes caused by each of the fragments that have hit component \( j \). Since the maximum value of a probability is one \( P_{\text{kill}}(j) \) is equal to one should the sum of the areas of the holes caused by each of the fragments that have hit component \( j \) be larger than \( A_j \).

7.2.4.4 Kill Category 4

The kill probability for vital parts, such as wiring, is calculated as

\[ P_{\text{kill}}(j) = 1 - e^{-\sum \left(\frac{D_f(i)}{\text{Prot}_{\text{int}}(j)}\right)^2} , \quad (7.15) \]

where \( \text{Prot}_{\text{int}}(j) \) is the critical diameter of line element \( j \) that will cause a hundred percent failure of the element, also described as the internal protection of line-element \( i \), and \( D_f(i) \) is the diameter of fragment \( i \).
7.2.5 The loop over all Systems

As can be seen in Figure 7.1, the loop over all the systems is the last iterative process in the loop over all the paths. The objective of this section of the model is to calculate the probability of a specific system, e.g. system \( i \), breaking down, \( P_{\text{Breakdown}}(i) \), due to components in the target that have been damaged. The formula for this is

\[
P_{\text{Breakdown}}(i) = 1 - \prod_{j=1}^{n_{\text{components}}} (1 - P_{\text{kill}}(j) P_{\text{kill}}(i, j)),
\]

where \( P_{\text{kill}}(i, j) \) is the probability of component \( j \) breaking down as part of system \( i \), if component \( j \) is not part of system \( j \) then \( P_{\text{kill}}(i, j) = 0 \). Furthermore, \( P_{\text{kill}}(j) \) is the accumulated kill probability of component \( j \), and it follows that

\[
1 - P_{\text{kill}}(j) P_{\text{kill}}(i, j),
\]

is the probability of system \( i \) not failing as a result of damage to component \( j \). The product of all the probabilities of system \( i \) not failing due to the damage on components \( 1 \ldots n_{\text{components}} \) is the probability of system \( i \) not failing as a result of the damaged caused at a specific burst point of an intercept scenario. For this to be true the events must be independent. This is insured through the construction of the critical component data file where each component can only fail for part of one system. Subtracting this probability from one gives the probability of the system failing, \( P_{\text{Breakdown}}(i) \), as the result.

Having the probability that system \( i \) will fail, the probability that a specific mission criterion will occur given that system \( i \) fails, \( P_{\text{mc}}(\text{Event}) \), can be calculated. This is calculated by multiplying the probability that event \( A, B, C, \) or \( D \) will occur given that system \( i \) breaks down, \( P_e(i) \), where \( e = \{A, B, C, D\} \), with the probability of system \( i \) breaking down,

\[
P_{\text{mc}}(\text{Event}) = P_e(i) P_{\text{Breakdown}}(i)
\]

and

\[
P_{\text{mc}}(E) = 1 - P_{\text{Breakdown}}(i),
\]

where event \( E \) is the event of the target being undamaged, and the others as defined in Section 2.7. Referring to the event diagram in Figure 2.8, three additional events will now be defined. These are:

- \( F \) - Aircraft returns to base (No crash);
- \( G \) - Mission accomplished;
- \( H \) - Mission accomplished and aircraft returns to base,

where,

\[
P_i(E) = P_{\text{mc}}(E),
\]

\[
P_i(F) = P_{\text{mc}}(B) + P_{\text{mc}}(D) + P_{\text{mc}}(E),
\]

\[
P_i(G) = P_{\text{mc}}(C) + P_{\text{mc}}(D) + P_{\text{mc}}(E)
\]

and

\[
P_i(H) = P_{\text{mc}}(D) + P_{\text{mc}}(E)
\]
CHAPTER 7. EFFECTIVENESS MODEL

for \( P_i(\text{Event}) \) defined as the probability that an event will occur as a result of damage on system \( i \). For equations (7.19) - (7.22) to be true mutual exclusivity must apply. By once again referring to the event diagram in Figure 2.8 this mutual exclusivity can be shown. The events are

- **A** - The event where the mission is aborted and the aircraft is lost;
- **B** - The event where the mission is aborted but the aircraft is not lost;
- **C** - The event where the mission is not aborted but the aircraft is lost;
- **D** - The event where the mission is not aborted and the aircraft is not lost but there are some damages to be repaired, and;
- **E** - The event where the mission is not aborted and the aircraft is not lost and there are no damages to be repaired (outside of the large circle in Figure 2.8).

### 7.2.6 Calculate the probability of the various events

This is the last subroutine in the loop over all burst points and is run once all the other calculations with regard to a specific burst point have been completed. The probability of an event occurring is the result of the various system breakdown probabilities.

If there is a direct hit (the warhead is in the target, \( \text{ic} = 3 \)), then

\[
P^i(A) = 1, \quad P^i(B) = 0, \quad P^i(C) = 0, \quad P^i(D) = 0 \quad \text{and} \quad P^i(E) = 0
\]

for burst point \( i \). Otherwise, \( P_{\text{total}}(E) = P_{\text{total}}(F) = P_{\text{total}}(G) = P_{\text{total}}(H) = 1 \), where \( P_{\text{total}}(E \ldots F) \) is the probability that the respective events will occur, at burst point \( i \), taking all of the systems into account, and is calculated as

\[
P_{\text{total}}(E) = P_{\text{total}}(E) \prod_{j=1}^{N_{\text{sys}}} P_j(E),
\]
\[
P_{\text{total}}(F) = P_{\text{total}}(F) \prod_{j=1}^{N_{\text{sys}}} P_j(F),
\]
\[
P_{\text{total}}(G) = P_{\text{total}}(G) \prod_{j=1}^{N_{\text{sys}}} P_j(G),
\]
\[
P_{\text{total}}(H) = P_{\text{total}}(H) \prod_{j=1}^{N_{\text{sys}}} P_j(H).
\]

Once this has been completed for all the systems the probability of each event occurring for burst point \( i \) is calculated,

\[
P^i(A) = 1 - P_{\text{total}}(F) - P_{\text{total}}(G) + P_{\text{total}}(H),
\]
\[
P^i(B) = P_{\text{total}}(F) - P_{\text{total}}(G),
\]
\[
P^i(C) = P_{\text{total}}(G) - P_{\text{total}}(H),
\]
\[
P^i(D) = P_{\text{total}}(H) - P_{\text{total}}(E),
\]
\[
P^i(E) = P_{\text{total}}(E).
\]

Finally the sum of the probability of a specific event occurring for all the burst points that have been determined up to this point is calculated,
\[ P(A) = \sum_{k=1}^{i-1} P^k(A) + P^i(A), \]
\[ P(B) = \sum_{k=1}^{i-1} P^k(B) + P^i(B), \]
\[ P(C) = \sum_{k=1}^{i-1} P^k(C) + P^i(C), \]
\[ P(D) = \sum_{k=1}^{i-1} P^k(D) + P^i(D), \]
and,
\[ P(E) = \sum_{k=1}^{i-1} P^k(E) + P^i(E). \]

Once all the burst points over all the intercept scenarios have been analysed, the mean probabilities for the various events are calculated and the summative results are presented. These are shown in the following two sections.

### 7.2.7 Calculate the mean probability for each of the mission criteria over all the intercept scenarios

The mean probability of an event occurring is calculated by dividing the total probability by the number of scenarios, \( n_{is} \),

\[ P_{\text{mean}}(A) = \frac{P(A)}{n_{is}}, \]
\[ P_{\text{mean}}(B) = \frac{P(B)}{n_{is}}, \]
\[ P_{\text{mean}}(C) = \frac{P(C)}{n_{is}} \]
and
\[ P_{\text{mean}}(D) = \frac{P(D)}{n_{is}}. \]

### 7.2.8 The results

The final results of the calculations performed in Upshot are presented as the results for each intercept scenario (or missile) as well as the mean results for all the intercept scenarios. The results are presented in the following format:

#### 7.2.8.1 Mission criteria for each missile

Missile number: \( i \)

\( P(a) = P^i(A), \) Mission made impossible, immediate crash.
P(b) = \( P^i(B) \), Mission made impossible, no crash.

P(c) = \( P^i(C) \), Mission made possible, delayed crash.

P(d) = \( P^i(D) \), Mission possible, no crash, repairs must be made.

Number of hits on the target outer skin: \( n_{ohit}(i) \)

<table>
<thead>
<tr>
<th>Vital Part no.</th>
<th>Number of Fragments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( N_{vhit}(i,1) )</td>
</tr>
<tr>
<td>2</td>
<td>( N_{vhit}(i,2) )</td>
</tr>
<tr>
<td>.</td>
<td>( N_{vhit}(i,.) )</td>
</tr>
<tr>
<td>( j )</td>
<td>( N_{vhit}(i,j) )</td>
</tr>
<tr>
<td>.</td>
<td>( N_{vhit}(i,.) )</td>
</tr>
<tr>
<td>( n_{components} )</td>
<td>( N_{vhit}(i,n_{components}) )</td>
</tr>
</tbody>
</table>

Table 7.7: The number of hits on each vital part during intercept scenario \( i \).

7.2.8.2 Summed up mission criteria results

P(a) = \( P_{\text{mean}}(A) \), Mission made impossible, immediate crash.

P(b) = \( P_{\text{mean}}(B) \), Mission made impossible, no crash.

P(c) = \( P_{\text{mean}}(C) \), Mission made possible, delayed crash.

P(d) = \( P_{\text{mean}}(D) \), Mission possible, no crash, repairs must be made.

Mean number of hits on target outer skin: \( N_{vhit} \)

<table>
<thead>
<tr>
<th>Vital Part no.</th>
<th>Number of Fragments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( n_{vhit}(1) )</td>
</tr>
<tr>
<td>2</td>
<td>( n_{vhit}(2) )</td>
</tr>
<tr>
<td>.</td>
<td>( n_{vhit}(.) )</td>
</tr>
<tr>
<td>( j )</td>
<td>( n_{vhit}(j) )</td>
</tr>
<tr>
<td>.</td>
<td>( n_{vhit}(.) )</td>
</tr>
<tr>
<td>( n_{components} )</td>
<td>( n_{vhit}(n_{components}) )</td>
</tr>
</tbody>
</table>

Table 7.8: The mean number of fragments which penetrates the outer and internal protections for each vital part \( j \).
7.3 EVA

EVA provides the functionalities to set up a simulation run, initiate it, and delete it. It also keeps book of the files created for the user. These functions can be found on the dropdown menu, see Figure 7.8.

![Figure 7.8: Upshot dropdown menu.](image)

7.3.1 Standard Simulation Setup

To set up a simulation run the user needs to provide a name for the simulation, the warhead and target to be used, the file name for the simulation results, as well as the paths for which to do the evaluation. Figure 7.9 shows the form in which the data must be entered and the simulation setup saved. The fact that EVA keeps book of the files associated with each target and warhead simplifies this process and shows one of the user-friendly improvements from SWEAT to EVA.

7.3.2 Running the standard simulation

Once a simulation has been set up the user can initiate the simulation by selecting it from the list of simulations provided and selecting the OK button, see Figure 7.10.

On completion of the simulation run the results can either be viewed as text or graphically, as a graph. These functionalities can be activated by selecting the View Upshot File or View Graph options from the Design tab, see Figure 7.11, respectively.
7.3.3 View Graph

An output file of a simulation run can be viewed in graphical format by selecting the file and specifying the event type required (see Section 2.7 for a brief discussion on events). This infor-
Information must be entered in the form seen in Figure 7.12. EVA allows the user to open up to five graphs simultaneously enabling the designer to compare outputs with one another easily.

The graph is displayed on the form seen in Figure 7.14. The top panel displays information concerning the intercept scenario and general information namely the name of the output file displayed the event shown.

The graph area shows two graphs, the one shows the number of hits with respect to the burst point whilst the other shows the probability of the selected event with respect to the burst point.

The form also allows the user to import various fuze evaluation files to compare the different fuze designs with respect to the effectiveness of the warhead design on a specific path.

### 7.3.4 Deleting a simulation run

EVA has a facility to delete all information concerning the setup of the simulation run, see Figure 7.13.

### 7.3.5 View Upshot Output File (text)

The user can also view the output file in text format, see Figure 7.15. This functionality has the same properties as in the other cases where data have been viewed in text.
CHAPTER 7. EFFECTIVENESS MODEL

7.3.6 Improvements on SWEAT

Once again all of the operations regarding the simulation is confined to the program that runs within a Windows environment where previously SWEAT made use of various programs some of which still executed in a Dos environment.
EVA also makes it simple to display the results graphically. Within SWEAT the user needed to import the data into a spreadsheet package, e.g. MS Excel, and process the data to an appropriate format within that package.

As part of the Upshot section of the project both the methodology and mathematical model were reviewed, reconsidered, and verified. The program code implementing the model was also verified as being correct and in the process minor changes were made for easier analyses of the code in future. Through this process knowledge transfer in an area for which this is depleting in South Africa took place resulting in improvement and contingency in this discipline.
Chapter 8

Optimisation

8.1 Introduction

In this chapter the heuristic model that was developed as a part of this study to search for a warhead that has the largest possible diameter and total mass for which the momentum of the fragment will be larger than a specified minimum, given user defined constraints, is discussed. In the first section the algorithm and the second section how this functionality is accessed within EVA are addressed.

8.2 The warhead search Algorithm

The search area and initiation point for the search are defined by the warhead designer. This is done by selecting an existing warhead. When selecting this warhead, the following are taken into consideration:

(1) the desired shape and density of the fragments;
(2) the number of fragment layers, given that the fragments are cubes;
(3) the density of the explosive, the outer and inner casing, and;
(4) the shape of the of the warhead.

The boundaries of the search area are defined by the following constraints:

\[
\begin{align*}
    m_{\text{min}} & \quad \text{Minimum fragment mass;} \\
    m_{\text{max}} & \quad \text{Maximum fragment mass;} \\
    L/D_{\text{min}} & \quad \text{Minimum length over diameter;} \\
    L/D_{\text{max}} & \quad \text{Maximum length over diameter;} \\
    L_{\text{max}} & \quad \text{Maximum length;} \\
    D_{\text{max}} & \quad \text{Maximum diameter;} \\
    M_{\text{max}} & \quad \text{Maximum warhead mass;} \\
    H_{\text{min}} & \quad \text{Minimum hole diameter;} \\
    C/M_{\text{min}} & \quad \text{Minimum ratio of the mass of the explosive to the surrounding mass.}
\end{align*}
\]

The accuracy of the search is determined by the following three variables by which the search
CHAPTER 8. OPTIMISATION

for the warhead is conducted:

\[ m_{\text{step}} \quad \text{Fragment mass step size;} \]
\[ L/D_{\text{step}} \quad \text{Length over diameter step size;} \]
\[ D_{\text{step}} \quad \text{Diameter step size.} \]

These variables respectively set the search intervals. The objective of the warhead search is to find:

**The warhead that has the largest possible diameter and total mass for which the momentum of the fragment will be larger than a specified minimum for a given fragment size, given the constraints.**

The maximum warhead length, \( L_{\text{max}} \), maximum warhead diameter, \( D_{\text{max}} \), and maximum warhead mass, \( M_{\text{max}} \), are constraints as a result of the missile design which only allows for a maximum warhead size and weight.

The larger the warhead diameter, \( D \), the more space there is for explosives as well as fragments per segment. This will increase the velocity of the fragments. The length of the warhead, \( L \), allows for a larger fragment flux. This is because more fragments can be fitted next to each other.

The ratio of warhead length and the warhead diameter is also an important relationship in the design of a warhead. The \( L/D \) of the warhead has an influence on the velocities and ejection angles of the warhead fragments. The warhead designer strives for \( L/D \) values in the order of 1.5 to 2.

Hardware, e.g. SAD, wires, needs to fit inside and/or pass through the warhead. To accommodate for this a hole with a minimum diameter, \( H_{\text{min}} \), is required.

For \( C/M \), known as the charge-to-metal mass ratio, a minimum is also specified since the amount of explosive charge, relative to the mass of the damage causing material to be accelerated, will determine the acceleration. Since the minimum momentum is known, \( C/M \) can be calculated and provided to EVA.

Given the pre-selected warhead the following information can be read in:

\[ \rho_{\text{frag}} \quad \text{The density of the fragment;} \]
\[ S \quad \text{The shape of the fragment;} \]
\[ C_{\text{layer}} \quad \text{The number of layers of fragments the warhead has, given that the fragments are cubes;} \]
\[ \rho_{\text{expl}} \quad \text{The density of the explosive;} \]
\[ C_{\text{outer}} \quad \text{The density of the outer casing, and;} \]
\[ C_{\text{inner}} \quad \text{The density of the inner casing.} \]

The algorithm runs from the minimum fragment size, \( m_{\text{min}} \), up to the maximum fragment size, \( m_{\text{max}} \), as specified. A warhead is generated for each of the fragments using the specified warhead as the basis to generate the warhead from. The various fragments that are generated
8.2. The warhead search Algorithm

If, given the three parameters, a total of \( n \) fragments can then be generated,

\[
m_1 = m_{\text{min}}, m_2 = m_{\text{min}} + m_{\text{step}}, \ldots, m_i = m_{\text{min}} + (i-1)m_{\text{step}}, \ldots, m_n = m_{\text{max}}.
\]

Having the shape, density, and mass of the fragment, the volume, \( V_{\text{frag}} \), and dimension of the fragment can be calculated where the dimensions are the length, \( L_{\text{frag}} \), height, \( H_{\text{frag}} \), and breadth, \( B_{\text{frag}} \). If the fragment is ball shaped, then

\[
L_{\text{frag}} = H_{\text{frag}} = B_{\text{frag}} = 10^{\frac{3}{2}}\sqrt{\frac{6}{\pi}V_{\text{frag}}},
\]

else, if the shape is a cube, then

\[
L_{\text{frag}} = H_{\text{frag}} = B_{\text{frag}} = 10^{\frac{2}{3}}V_{\text{frag}}^{\frac{1}{3}}.
\]

From the objective of the heuristic search algorithm it follows that the algorithm aims to find the warhead with the largest possible diameter, \( D \), and length, \( L \), combination. The starting point is consequently the maximum diameter, \( D = D_{\text{max}} \), as well as maximum length, \( L = L/D \times D \), given the maximum \( L/D = L/D_{\text{max}} \). This will result in the maximum number of fragments and explosives if only the constraints with regard to the diameter, length, and the \( L/D \) relationship are taken into account.

To find a combination of the largest possible diameter, \( D \), and length, \( L \), the following process is repeated until \( L \leq L_{\text{max}} \). First the length, \( L \), is calculated by multiplying the diameter with the \( L/D \) relationship, \( L = L/D \times D \).

Should \( L > L_{\text{max}} \), then the length must be set to the maximum it can be, \( L = L_{\text{max}} \), and the diameter recalculated, \( D = \frac{L}{L_{\text{max}}} \), with the maximum \( L/D \) being preferred. If the diameter is less than or equal to zero at this stage, \( D \leq 0 \), then it follows that if \( L/D = L/D_{\text{min}} \) the warhead cannot be generated, given the set of constraints. Otherwise the \( L/D \) ratio is decreased by the step size, \( L/D_{\text{step}} \), and the diameter recalculated, \( D = \frac{L}{L_{\text{max}}} \), until the diameter is larger than zero, \( D > 0 \).

The diameter of the explosive, \( D_{\text{exp}} \), is calculated by subtracting the height of the fragments and the inner and outer casing, \( T_{\text{inner}} \) and \( T_{\text{outer}} \) respectively, from the warhead diameter,

\[
D_{\text{exp}} = D - 2C_{\text{layer}}H_{\text{frag}} - 2(T_{\text{inner}} + T_{\text{outer}}).
\]

If the diameter of the explosive is larger than the diameter of the minimum hole size, \( D_{\text{exp}} > H_{\text{min}} \), it is decreased to \( H_{\text{min}} \). Using all of the above calculated information and setting the hole size to the minimum hole size, \( H = H_{\text{min}} \), an input file for WARHEAD is generated and the warhead is “constructed” by WARHEAD.

Once the warhead has been generated the following information is available:

- \( M_{\text{total}} \) The total mass of the warhead;
- \( C_{\text{max}} \) The maximum mass of explosives that can be fitted into the warhead;
- \( N_{\text{frag}} \) The number of fragments.

The first constraint that is checked with regard to the generated warhead is whether or not the
total warhead mass is less than or equal to the maximum mass that is allowed, $M_{total} \leq M_{max}$. If this is the case, whether or not the ratio of the mass of the explosive and the surrounding mass is greater than or equal to the minimum that was indicated, $\frac{C}{M} \geq C/M_{min}$. If both of these requirements are met, then the warhead is accepted for the particular fragment mass.

If the total warhead mass is greater than the maximum mass that is allowed, $M_{total} > M_{max}$, the following adjustments are made to the variables and a new warhead is generated. If the warhead complied to this constraint on a previous iteration for this fragment size and now does not, as a result of the diameter being increased to increase the $C/M$ ratio after $L/D$ was found to be equal to $L/D_{min}$, it is impossible to generate a warhead given the selected constraints. Otherwise, if the warhead mass minus the explosive mass is larger than the maximum warhead mass, $(M_{total} - C_{max}) > M_{max}$, the diameter is decreased by the step size, $D_{step}$. Should both of the previous two conditions not apply, the maximum possible explosive mass, $M_{exp}$, is calculated by subtracting the calculated surrounding mass from the maximum warhead mass that is allowed,

$$M_{exp} = M_{max} - (M_{total} - C_{max}).$$

If the $C/M$ ratio for this maximum explosive mass is less than the allowed minimum, the surrounding mass of the explosive must be decreased, which, as previously, is done by decreasing the diameter. Should the $C/M$ ratio not be less than the minimum, the mass of the explosive might be to large to have a warhead within the mass constraints. The square of the maximum possible radius of the explosive, $r_{max}$, given the maximum explosive mass, $M_{exp}$, allowed can be calculated by dividing this mass by the product of the density of the explosive, the length of the warhead, and $\pi$,

$$r_{max}^2 = \frac{M_{exp}}{L \rho \pi}.$$

Should $D_{exp}^2 < r_{max}^2$ (note that the direction of the inequality changed as a result of the division), then the diameter is once again decreased by $D_{step}$ to decrease the warhead mass. Should it not be the case, the hole size must be increased to the difference between the explosive diameter and the calculated maximum is

$$H = \sqrt{D_{exp}^2 - r_{max}^2}.$$

Should the total warhead mass be less than or equal to the maximum warhead mass that is allowed and the second main constraint, namely that the ratio of the mass of the explosive and the surrounding mass is greater than or equal to the minimum that was indicated, $\frac{C}{M} \geq C/M_{min}$, is found not to be satisfied, the following adjustments are made to the input variables for the warhead and a new warhead is generated.

Should $L/D = L/D_{min}$, the diameter must be increased. It can, however, not be done in the same size increment as before since that will deliver a previous warhead, for this fragment size, that was found to be outside of the constraint boundaries. As a result the diameter is increased by a tenth of the decreased increment, $D = D - \frac{D_{step}}{10}$. It is important to note that once this has taken place, and the constraint of the total warhead mass being less than or equal to the maximum warhead mass that is allowed, is found to be not satisfied, the process will be aborted as no warhead can be generated given the specific set of constraints and initiation warhead.

Alternatively, if $L/D > L/D_{min}$ the ratio of the warhead length over the diameter is decreased by the selected increment, $L/D_{step}$. Should the result be less than $L/D_{min}$ it is set to $L/D_{min}$.
8.2. The warhead search Algorithm

As indicated earlier, the maximum length is preferred. Changing the $L/D$ ratio will increase the diameter as a result, $D = \frac{L_{\text{max}}}{L/D}$. Should the diameter be larger than the allowed maximum it is set equal to it, $D = D_{\text{max}}$.

The above algorithm is repeated until the warhead that is generated by WARHEAD either complies with all the constraints or it is found to be impossible to generate a warhead within these constraints, given the initial warhead. This is done for each of the $i$ specified fragment sizes and the following set of results is stored in a database table, $\{m_i, L/D, C/M, M_{\text{total}}, C, N_{\text{frag}}\}$.

The above process is illustrated in a process flow diagram in Figure 8.1 and 8.2.

![Figure 8.1: The warhead search algorithm flow diagram, part 1.](image)
8.3 EVA

8.3.1 Variable Simulation

The Variable Simulation must first be set up in EVA to access the heuristic warhead search functionality. This is done on the Variable Simulation Form which is displayed in Figure 8.3. On this form the constraint boundaries and search intervals, as was defined in the previous section, as well as an existing warhead for EVA to use as an initiation point for the search are given as input.

Figure 8.2: The warhead search algorithm flow diagram, part 2.
Also specified in the Variable Simulation Form are the paths for, and the target against which, the warheads must be evaluated. This facility allows the designer to evaluate the effectiveness trends for the various fragment sizes.

### 8.3.2 Variable simulation warhead generation

On the Variable Simulation form, as seen in Figure 8.3, the parameters for the simulation only, are specified. Before the user can access the warhead effectiveness evaluation tools, which will be discussed below, for the specified fragment size spectrum, these warheads must first be
generated. This is done by selecting one of the variable warhead simulations, created and stored in the Variable Simulation setup functionality discussed in Section 8.3.1, from the dropdown list on the Variable Simulation Warhead Generation form seen in Figure 8.4.

![Variable Simulation Warhead Generation]

Figure 8.4: Warhead generation for variable simulation.

8.3.3 Variable simulation run

Once the simulation has been set up and the warheads generated, the effectiveness evaluation can be initiated, Figure 8.5. The effectiveness analysis as discussed in Chapter 7 are done for each of the generated warheads, on all of the specified intercept scenarios, against the target selected on the form in Figure 8.3.

![Variable Simulation Run]

Figure 8.5: Select a variable simulation to run.
8.3.4 Variable Simulation Information

Once the simulation run has been completed the user can view the following information:

**Variable Simulation Warhead Mass Graph**

This information sheet, as seen in Figure 8.7, is accessed by selecting the Mass Data button on the Variable Simulation Information form shown in Figure 8.6. In the top panel of the form shown in Figure 8.7 the following information are displayed:

- Simulation Name;
- Maximum Warhead Mass;
- The minimum and maximum length over diameter ratio;
- Maximum Diameter;
- Maximum Length;
- Minimum Hole Size, and
- Minimum $C/M$ ratio.

On the Variable Simulation Warhead Mass Graph three graphs are displayed, these are:

- Calibre over mass relationship to fragment mass of the warhead;
Total mass of the warhead in relation to the fragment mass, and Explosive mass of the warhead in relation to the fragment mass.

Figure 8.7: Variable Simulation Warhead Mass Graph.

Variable Simulation Warhead Fragment Graph

This information sheet, as seen in Figure 8.8, is accessed by selecting the Fragment Data button on the Variable Simulation Information form shown in Figure 8.6. In the top panel of the form shown in Figure 8.8 the same information is displayed as is in the top panel of the form which displays Variable Simulation Warhead Mass Graph. The graph displayed as the Variable Simulation Warhead Fragment Graph is the number of fragments relative to fragment mass.
Warhead information

This information sheet, as seen in Figure 8.9, is accessed by selecting the fragment size of the warhead for which the information must be displayed from the drop down menu and then selecting the View Warhead button on the Variable Simulation Information form shown in Figure 8.6.

At the top of the View Variable Simulation Warhead Output File for User form the name of the simulation of which the warhead forms a part and the name of the text file in which the warhead information is stored, are displayed. In the text box the information for the selected warhead is displayed in text format.

Figure 8.8: Variable Simulation Warhead Fragment Graph.
Figure 8.9: View Variable Simulation Warhead Output file for user.

Variable Simulation Graph

This information sheet, as seen in Figure 8.10, is accessed by selecting the path, event, and OK button on the Variable Simulation Information form shown in Figure 8.6.

In the top panel the Miss distance, Relative speed, file name, and the event for which the probability is graphed, are displayed. This functionality should be familiar to the reader, since it functions in a similar fashion as was discussed in Section 7.3.3 with regard to viewing the information of the effectiveness analysis. The one big difference being the sliding bar on the right that allows the user to view the effectiveness of the different fragment sizes dynamically, helping the designer to see the trend.
8.3.5 Improvements on SWEAT

The heuristic search algorithm and process as a whole has been developed for EVA with none of these capabilities being available previously. The tools discussed in this chapter enable the designer to see and analyse various trends as the result of the fragment size changing for a warhead with certain fixed characteristics. These characteristics include the shape, fragment type, and if applicable, the number of fragment layers. The trends that can be visualised are:

- Calibre over mass relative to the fragment mass of the warhead;
- Total mass of the warhead relative to the fragment mass;
- Explosive mass of the warhead relative to the fragment mass;
- The number of fragments relative to the fragment mass, and
• The effectives of for the various fragment sizes against a specific target on a specific intercept scenario.
Chapter 9

Conclusion

9.1 Introduction

An overview of the thesis is given in the first section of this chapter, and the contributions made as a result of this study are discussed in the second section. In the third section possible future developments, as a result of the new developments that followed out of this study, are presented. As a part of this presentation a number of meta-heuristic optimisation search algorithms, associated with simulation, are briefly discussed and a possible model to be implemented in future suggested.

9.2 Summary of the study

The aim of this study was the development of an Integrated Effectiveness Model for Aerial Targets. Some background is given on the discipline of effectiveness modelling, relating effectiveness to terminology, such as survivability, susceptibility, vulnerability and endgame, orientating the reader to where effectiveness modelling resides with regard to MED and WCDS.

The two main methodologies, the Vulnerable Area- and Ray-tracing models, are explained, compared, and why the Ray-tracing model was implemented justified. As a part of this discussion the models and capabilities of other countries were also discussed.

Target modelling is explained, starting from the target analysis to the setting up of the original input and the critical file. Various techniques and tools were investigated and are presented with regard to target analysis, e.g. CCA, Kill tree, FMEA, DMEA, FTA, and kill expressions. Previously the construction of the target model was open to error due to the analyst not being able to visualise it without a time consuming process of physically plotting and connecting the data points. An algorithm was developed to extract the information from the data in the target input file necessary to display the target visually. Upshot, the effectiveness model, also needs the data in another format. The calculations done to convert it and the results of the calculations were verified and discussed in this study.

Various propagators and guidance systems are discussed as to briefly explain the reason for numerous flight paths and, as a result, intercept scenarios occurring. The various variables used to describe an intercept scenario were defined. How the data are stored for use in the
effectiveness model is also defined, with the calculations for converting the variables into the required data explained. Also explained and verified are the algorithms used to generate random variables for different distribution types, e.g. Gamma- and Exponential distributions. These are employed to randomly generate the variables that describe an intercept scenario.

The algorithm (and programme) that calculates the velocity and ejection angles for each of the fragments in a warhead is effectively treated as a “black box”. However, a brief background is provided on the factors (e.g. fragment volume, mass, momentum, and Cumulative or Synergistic effects) that determine the effect of a damage mechanism. The mathematics implemented, such as the Gurney formulas and Taylor equation, the modifications to the formulas to accommodate end effects, and fragment retardation are also briefly discussed for completeness.

The fuze (target detection device) is discussed with regard to its functionality, models and classification. The model developed to simulate a fuze, which is an approximation of a fuze design to be employed with the missile for which a specific warhead is being designed, and calculate the approximate detection position of this fuze design against a specified target is discussed. This model is also extended to the section that discusses the effectiveness analysis model. In both models the methodology used with regard to detecting a target is similar.

All of the models are brought together in the module named Upshot. Upshot is the effectiveness model that calculates the probability of an event occurring. The reader is orientated on where the information that is required by this model is sourced from and the method used to manipulate and transform it into the required format. The calculation of which parts of the outer skin, as well as which vital parts are hit, and at what velocity are explained. The four models used to calculate the probability of kill for a vital part due to one or more hits, is explained for each of the categories. Combining the effect of the various events, taking factors such as mutual independence and dependence into account, and calculating the probability of each of the selected events are discussed. This is coupled with providing the reader with more insight into the various events and where these events intersect.

As part of this study a heuristic search algorithm was developed with the objective of finding the warhead that has the largest possible diameter and total mass for which the momentum of the fragment will be larger than a specified minimum for a given fragment size, given the constraints. The algorithm and the integration of the module WARHEAD are discussed. Since it is a heuristic search algorithm, optimal warhead designs given the constraints and input, such as the initiation warhead, cannot be guaranteed. However, it provides the warhead designer with good approximations which help identify the resulting trend of changing certain variables, i.e. the fragment size.

Throughout the study how the various functionalities are accessed, supported by a description of the GUI interface, is provided. The underlying theory, implementation, and finally interface application thereof are given for each sub-model and functionality.

### 9.3 Contributions made

The calculations done to the target data in the pre-processor for Upshot were verified as being correct. Due to the fact that the target can now be viewed graphically the target modeller can
The advancement of technology has enabled the missile designers to, through simulation, determine and provide the distributions for the various variables that describe an intercept scenario. Because of this, it justified the development of the capability to generate random intercept scenarios. This has a couple of advantages, namely the fact that the inconsistency in which the intercept scenarios are supplied by various missile designers and the time consuming conversion thereof can now be circumvented. More importantly, for the first time the capability was developed to generate random intercept scenarios, and also effectiveness simulations. This makes the implementation of Monte Carlo simulations, and subsequently meta-heuristic search algorithms, possible. The calculations done on these variables as to convert it to the required format for Upshot were also verified.

The fuze detection position model used in EVA was developed independently of previous models which sometimes gave unexpected results. This model was verified by doing manual calculations of extreme cases and comparing the results. It was further verified by comparing the results to another model that was developed subsequently and independently of the fuze model used by EVA. Having this capability supports work done on the generation of random intercept scenarios and the implementation of meta-heuristic search algorithms. Knowing the detection position in this regard is important since it plays an important role in calculating the value of the objective function that is required for implementing a meta-heuristic search algorithm.

The methodology and mathematical model implementing the effectiveness model was reviewed, reconsidered, and verified. The program code of Upshot implementing the model was also verified as being correct and in the process minor changes were made for easier analyses of the code in future.

The heuristic search algorithm for warheads is a capability that was not previously available. New design tools based on this capability enables the designer to see and analyse various trends as the result of the fragment size changing for a warhead with certain fixed characteristics.

As a part of the study the previous DOS based environment was replaced with a GUI based user interface, localising the various operations into what is perceived by the user to be one program that runs within a Windows environment. Not only is the generation of various results now confined to one interface, but also the tools used to view and analyse it. Previously, various packages such as MS Word and MS Excel needed to be used to access and analyse the information, often being associated with the development of a pre-processor as to convert the data into the correct format to be imported. A further improvement that is a result of the development of EVA is that the bookkeeping with regard to various targets, warheads, intercept scenarios, etc., and the associated files is now done by the package.

The development of EVA, and this study as a whole, has made the technology more accessible to users. Firstly, this document serves as the first complete documentation of the simulation model employed, simplifying the transfer of, and up-skilling into the knowledge. Secondly, since EVA manages the implementation of the actions requested, the user need not be as knowledgeable in the development history of this capability and the slight variations that needs to be taken into account when selecting one of the many versions that have been developed over the years. The user need not be as highly skilled and qualified as was previously the case. The need for package modification, and development of pre-processors should have diminished dramatically.
The easier transferability of this knowledge, as a result of it being comprehensively documented, not only makes it more accessible to users, but also protects the technology, satisfying one of the original project objectives of ensuring contingency planning.

9.4 Future developments as a result of this study

As indicated earlier the development of the capability to generate random intercept scenarios, and also effectiveness simulations, has made the implementation of Monte Carlo simulations, and subsequently meta-heuristic search algorithms, possible. But why use meta-heuristics?

According to Groves [28], there are two main reasons for using heuristic methods:

1. The execution time of some types of problems grows extremely quickly in accordance with the size of the problem, heuristics are then only practical option for large instances of these problems, and;

2. Heuristics are often more capable of modelling the complicated assumptions inherent in real-life problems.

Heuristics are, however, not always the most suitable tool to use. Some combinatorial problems are, in fact, easy to solve optimally and the relevant literature should always be consulted before deciding to use a heuristic on a problem.

The fact that the effectiveness of the warhead design is estimated through simulation, and the problem having the same characteristics as described above by Groves, it is the opinion of the author that meta-heuristics are the preferred method for finding a good warhead design. Note that meta-heuristic methods do not necessarily yield an optimal design. Since the effectiveness analysis is, however, not an exact science, and as a result, optimisation methods that calculate an optimal solution would not with certainty give one, this is more than sufficient.

Some examples of meta-heuristics are:

- Generic Algorithms;
- Population-Based Incremental Learning Procedures (PBIL);
- Harmonic Search Algorithms;
- Ant System Algorithms, and
- Simulated annealing.

It will not be attempted to decide on which of these meta-heuristics are the most appropriate for the problem at hand and subsequently the various types named will not be discussed as a part of this study. According to Bekker [29], one technique will be more suited to a specific problem than another. It is, however, also acknowledged that how this is determined is still open to debate and that much of what determines a successful implementation is an art developed
future developments as a result of this study

What all of the techniques do, however, have in common is the fact that an objective function is required. The following factors are suggested for the object function; $A_p$ the average probability area, $N_P$ the number of high probability regions in the detonation vicinity, and $N_{ohit}$ the number of hits on the target.

The average probability area, $\bar{A}_p$, is the average area on the effectiveness graph under the probability of kill line and after the target detection for the various intercept scenarios.

A high probability region is defined as the positions (or regions) on the intercept path for which the kill probability is above 0.75. Since the detonation occurs a fixed time after detection the high probability region may not coincide with the detonation region. The number of high probability regions in the detonation vicinity, $N_P$, is the maximum number of times this can happen should the most appropriate delay time be selected for all the intercept scenarios.

The number of hits on the target, $N_{ohit}$, is the average number of fragments that hit the outer skin of the target during the $n_p$ intercept scenarios.

It is important that the objective function is constructed as to allow the designer the ability to steer the warhead being designed towards a specific effectiveness characterisation. A possible approach is a multi-objective linear programming (MOLP) approach. The average probability area describes the ability of the warhead to, during all intercept scenarios, effect some level of kill probability. Since the detonation is a fixed time after detection the number of high probability regions in the detonation vicinity presents the number of times that, during all intercept scenarios, the warhead will be able to deliver the most effective damage it can on that intercept scenario. Whilst the number of hits on the target indicates the possibility of structural damage or at least mission abortation.

There are many factors for which consideration cannot be build in to a model such as this at this stage, these can only be considered through human intervention. A typical example of this will be the knowledge of the production infrastructure already available. The aim of the meta-heuristic search is, however, to find a good warhead design, and should not be to find the final warhead. Rather, the aim should be to decrease the solution area for the designer, as to increase the effectiveness of the human intervention.
References


Appendix A

Target data files

A.1 Geometric Description File

A.1.1 File Format

The format of the input file that is used by TargetModeller and EVA to respectively generate the input file for effect and the graphical display of the target is given below.

- Title.
- Number of polyhedrons.
- Polyhedron number.
- Internal structure in mm Al per length meter. (Three values)
- The fraction, in percentage, for each value of the structure description. (Three values)
- Number of corners for the specific polyhedron.
- (x, y, z) - coordinates for corner number 1.
- (x, y, z) - coordinates for corner number 2.
- Repeat for each corner.
- Number of surfaces for the specific polyhedron.
- Corner 1, corner 2, corner 3, thickness for surface number 1.
- Corner 1, corner 2, corner 3, thickness for surface number 2.
- Repeat for each surface.
- Repeat for each polyhedron.

After the outer polyhedrons have been described, the vital components are also described as polyhedrons. This format is given below.

- Number of vital part polyhedrons.
• Vital part polyhedron number.
• Category for the kill criteria. (1 - Personal, 2 - Ordinary Component, 3 - Fuel Tank, 4 - Wirings)
• Kill probability, Internal protection in mm Al.
• Number of corners for the specific vital part polyhedron.
• \((x, y, z)\) - coordinates for corner number 1.
• \((x, y, z)\) - coordinates for corner number 2.
• Repeat for each corner.
• Number of surfaces for the specific vital part polyhedron.
• Corner 1, corner 2, corner 3, thickness for surface number 1 in mm Al.
• Corner 1, corner 2, corner 3, thickness for surface number 2 in mm Al.
• Repeat for each surface.
• Repeat for each polyhedron.

Some vital parts are also described as a line element. The format is given below.

• Number of line elements.
• Line element number.
• Category for the kill criteria. (1 - Personal, 2 - Ordinary Component, 3 - Fuel Tank, 4 - Wirings)
• Kill probability, Diameter of the specific line element.
• Outer protection, Internal protection in mm Al.
• \((x, y, z)\) - coordinates for the startpoint.
• \((x, y, z)\) - coordinates for the endpoint.
• Repeat for each vital part line element.

A.1.2 Example of Geometric Description File

RPG 7
2 ! Number of components
1 ! W/H forward parallel section
0.0 3.0 3.0
1.0 0.0 0.0
12
0.0 0.0000 0.0420
0.0 -0.0364 0.0210
0.0 -0.0364 -0.0210
0.0 0.0000 -0.0420
0.0 0.0364 -0.0210
0.0 0.0364 0.0210
0.028 0.0000 0.0420
0.028 -0.0364 0.0210
0.028 -0.0364 -0.0210
0.028 0.0000 -0.0420
0.028 0.0364 -0.0210
0.028 0.0364 0.0210
8
1 2 3 4.0
1 2 7 2.0
2 3 8 2.0
3 4 9 2.0
4 5 10 2.0
5 6 11 2.0
6 7 12 2.0
7 8 9 0.0
2
0.0 3.0 3.0
1.0 0.0 0.0
12
0.028 0.0000 0.0420
0.028 -0.0364 0.0210
0.028 -0.0364 -0.0210
0.028 0.0000 -0.0420
0.028 0.0364 -0.0210
0.028 0.0364 0.0210
0.13 0.0000 0.0215
0.13 -0.0186 0.0108
0.13 -0.0186 -0.0108
0.13 0.0000 -0.0215
0.13 0.0186 -0.0108
0.13 0.0186 0.0108
8
1 2 3 0.0
1 2 7 2.0
2 3 8 2.0
3 4 9 2.0
4 5 10 2.0
5 6 11 2.0
6 7 12 2.0
7 8 9 4.0
3
! NUMBER OF CRITICAL COMPONENTS
1
! Explosive forward part
2
100.0 0.0
12
0.004 0.0000 0.0380
0.004 -0.0329 0.0190
0.004 -0.0329 -0.0190
0.004 0.0000 -0.0380
0.004 0.0329 -0.0190
0.004 0.0329 0.0190
0.028 0.0000 0.0380
0.028 -0.0329 0.0190
0.028 -0.0329 -0.0190
0.028 0.0000 -0.0380
0.028 0.0329 -0.0190
0.028 0.0329 0.0190
0.028 0.0000 0.0380
0.028 -0.0329 0.0190
0.028 -0.0329 -0.0190
0.028 0.0000 -0.0380
0.028 0.0329 -0.0190
0.028 0.0329 0.0190
0.028 0.0000 0.0380
0.028 -0.0329 0.0190
0.028 -0.0329 -0.0190
0.028 0.0000 -0.0380
0.028 0.0329 -0.0190
0.028 0.0329 0.0190
0.028 0.0000 0.0380
0.028 -0.0329 0.0190
0.028 -0.0329 -0.0190
0.028 0.0000 -0.0380
0.028 0.0329 -0.0190
0.028 0.0329 0.0190
0.028 0.0000 0.0380
0.028 -0.0329 0.0190
0.028 -0.0329 -0.0190
0.028 0.0000 -0.0380
0.028 0.0329 -0.0190
0.028 0.0329 0.0190
8
1 2 3 0.0
1 2 7 0.0
2 3 8 0.0
3 4 9 0.0
4 5 10 0.0
5 6 11 0.0
6 7 12 0.0
7 8 9 0.0
2
1
100.0 0.0
12
0.028 0.0000 0.0380
0.028 -0.0329 0.0190
0.028 -0.0329 -0.0190
0.028 0.0000 -0.0380
0.028 0.0329 -0.0190
0.028 0.0329 0.0190
0.096 0.0000 0.0260
0.096 -0.0225 0.0130
0.096 -0.0225 -0.0130
0.096 0.0000 -0.0260
0.096 0.0225 -0.0130
0.096 0.0225 0.0130
8
1 2 3 0.0
1 2 7 0.0
2 3 8 0.0
3 4 9 0.0
4 5 10 0.0
5 6 11 0.0
6 7 12 0.0
7 8 9 0.0
3
1
100.0 0.0

!Explosive rearward part

! SAD
A.2 Critical Components and Subsystems File

A.2.1 File Format

The format of the input file describing the critical components and subsystems is given below:

- Number of systems.
- Name of system.
- Probability of a certain event occurring should the system fail.
- Number of components in system.
- Component 1, Probability of component breakdown given a hit.
- Component 2, Probability of component breakdown given a hit.
- Repeat for each component in the system.
- Repeat for each system.

A.2.2 Example of Critical File

```
2 ! NUMBER OF SYSTEMS
Warhead ! SYSTEM
```

12
0.096 0.0000 0.0170
0.096 -0.0147 0.0085
0.096 -0.0147 -0.0085
0.096 0.0000 -0.0170
0.096 0.0147 -0.0085
0.096 0.0147 0.0085
0.126 0.0000 0.0170
0.126 -0.0147 0.0085
0.126 -0.0147 -0.0085
0.126 0.0000 -0.0170
0.126 0.0147 -0.0085
0.126 0.0147 0.0085
8
1 2 3 0.0
1 2 7 0.0
2 3 8 0.0
3 4 9 0.0
4 5 10 0.0
5 6 11 0.0
6 7 12 0.0
7 8 9 0.0
0
<table>
<thead>
<tr>
<th>Component No.</th>
<th>Prob of Breakdown</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.0</td>
</tr>
<tr>
<td>2</td>
<td>1.0</td>
</tr>
</tbody>
</table>

**System:**
0.0 1.0 0.0 0.0

! MISSION PROB.
! NUMBER OF COMPONENTS
! COMP. NO, PROB OF BREAKDOWN
! SYSTEM
! MISSION PROB.
! NUMBER OF COMPONENTS
! COMP. NO, PROB OF BREAKDOWN
A.3 Target Input File for Effect

A.3.1 File Format

The format of the input file that is generated by TargetModeller for effect is given below.

- Title.
- Number of polyhedrons.
- Polyhedron number.
- Internal Point.
- Radius.
- Internal structure in mm Al per length meter. (Three values)
- The fraction, in percentage, for each value of the structure description. (Three values)
- Number of surfaces for the specific polyhedron.
- Normal vector, distance from internal point, thickness for surface number 1.
- Normal vector, distance from internal point, thickness for surface number 2.
- Repeat for each surface.
- Repeat for each polyhedron.

After the outer polyhedrons have been described, the vital components are also described as polyhedrons. This format is given below.

- Number of vital part polyhedrons.
- Vital part polyhedron number.
- Category for the kill criteria. (1 - Personal, 2 - Ordinary Component, 3 - Fuel Tank, 4 - Wirings)
- Information to calculate probability of kill.
- Internal Point.
- Radius.
- Internal protection in mm Al.
- Number of surfaces for the specific vital part polyhedron.
- Normal vector, distance from internal point, thickness for surface number 1 in mm Al.
- Normal vector, distance from internal point, thickness for surface number 2 in mm Al.
- Repeat for each surface.
Repeat for each polyhedron.

Some vital parts are also described as a line element. The format is given below.

- Number of line elements.
- Line element number.
- \((x, y, z)\) - coordinates for the startpoint.
- \((x, y, z)\) - coordinates for the endpoint.
- Diameter of the specific line element.
- Outer protection in mm Al.
- Internal protection in mm Al.
- Category for the kill criteria. (1 - Personal, 2 - Ordinary Component, 3 - Fuel Tank, 4 - Wirings)
- Kill probability.
- Repeat for each vital part line element.

A.3.2 Example of Target Input File for Effect

RPG 7
2
1
0.014 0.000 0.000
0.000 3.000 3.000
1.000 0.000 0.000
8
-1.000 0.000 0.000 0.014 4.000
0.000 -0.500 0.866 0.036 2.000
0.000 1.000 0.000 0.036 2.000
0.000 -0.500 -0.866 0.036 2.000
0.000 0.500 -0.866 0.036 2.000
0.000 1.000 0.000 0.036 2.000
0.000 0.500 0.866 0.036 2.000
1.000 0.000 0.000 0.014 4.000
2
0.079 0.000 0.000
0.000 3.000 3.000
1.000 0.000 0.000
8
-1.000 0.000 0.000 0.051 0.000
0.172 -0.492 0.853 0.027 2.000
0.172 -0.985 0.000 0.027 2.000
A.3. Target Input File for Effect

0.171 -0.492 -0.853 0.027 2.000 ! Norm vect, dist, out prot (mm Dural)
0.172 0.492 -0.853 0.027 2.000 ! Norm vect, dist, out prot (mm Dural)
0.172 0.985 0.000 0.027 2.000 ! Norm vect, dist, out prot (mm Dural)
0.171 0.491 0.854 0.027 2.000 ! Norm vect, dist, out prot (mm Dural)
1.000 0.000 0.000 0.051 4.000 ! Number of vital part polyh.
3
1
2
0.100E+03
0.016 0.000 0.000
0.040
0.000
8
-1.000 0.000 0.000 0.012 0.000
0.000 -0.500 0.866 0.033 0.000
0.000 -1.000 0.000 0.033 0.000
0.000 -0.500 -0.866 0.033 0.000
0.000 0.500 -0.866 0.033 0.000
0.000 1.000 0.000 0.033 0.000
0.000 0.500 0.866 0.033 0.000
1.000 0.000 0.000 0.012 0.000
2
2
0.100E+03
0.062 0.000 0.000
0.051
0.000
8
-1.000 0.000 0.000 0.034 0.000
0.151 -0.494 0.856 0.027 0.000
0.151 -0.989 0.000 0.027 0.000
0.151 -0.494 -0.856 0.027 0.000
0.151 0.494 -0.856 0.027 0.000
0.151 0.989 0.000 0.027 0.000
0.151 0.495 0.856 0.027 0.000
1.000 0.000 0.000 0.034 0.000
3
1
0.100E+03
0.111 0.000 0.000
0.023
0.000
8
-1.000 0.000 0.000 0.015 0.000
0.000 -0.501 0.866 0.015 0.000
0.000 -1.000 0.000 0.015 0.000
0.000 -0.501 -0.866 0.015 0.000
0.000 0.501 -0.866 0.015 0.000
0.000 1.000 0.000 0.015 0.000

165
0.000 0.501 0.866 0.015 0.000  ! Norm vect, dist, out prot (mm Dural)
1.000 0.000 0.000 0.015 0.000  ! Norm vect, dist, out prot (mm Dural)
0                              ! Number of line-elements
A.4 Graphical Output File

A.4.1 File format

The graphic output file gets created out of the data in the input file. This file is used to display the target graphically. The file has the following format.

- Title.
- Number of polyhedrons.
- Polyhedron number.
- Number of corners.
- \((x, y, z)\) - coordinates for corner number 1.
- \((x, y, z)\) - coordinates for corner number 2.
- Repeat for each corner.
- Number of surfaces.
- Number of corners in surface, corner 1, corner 2, corner 3,..., corner \(n\) for surface number 1.
- Number of corners in surface, corner 1, corner 2, corner 3,..., corner \(n\) for surface number 2.
- Repeat for each surface.
- Repeat for each polyhedron.

The corners of every surface is ordered in such a manner that if a line is drawn between corner 1 and corner 2, corner 2 and corner 3,..., corner \(n\) and corner 1, the lines will form the edge of the surface. After the outer polyhedrons have been described, the vital components are also described as polyhedrons. This format is given below.

- Number of vital part polyhedrons.
- Vital part polyhedron number.
- Number of corners for the specific vital part polyhedron.
- \((x, y, z)\) - coordinates for corner number 1.
- \((x, y, z)\) - coordinates for corner number 2.
- Repeat for each corner.
- Number of surfaces.
- Number of corners in surface, corner 1, corner 2, corner 3,..., corner \(n\) for surface number 1.
• Number of corners in surface, corner 1, corner 2, corner 3,..., corner n for surface number 2.

• Repeat for each surface.

• Repeat for each vital part polyhedron.

Finally the vital parts that are described as a line element is written in the following format.

• Number of line elements.

• Line element number.

• \( (x, y, z) \) - coordinates for the startpoint.

• \( (x, y, z) \) - coordinates for the endpoint.

• Repeat for each vital part line element.

A.4.2 Example of Graphical Output File for EVA

RPG 7
2
1
12
0.0 0.0000 0.0420
0.0 -0.0364 0.0210
0.0 -0.0364 -0.0210
0.0 0.0000 -0.0420
0.0 0.0364 -0.0210
0.0 0.0364 0.0210
0.028 0.0000 0.0420
0.028 -0.0364 0.0210
0.028 -0.0364 -0.0210
0.028 0.0000 -0.0420
0.028 0.0364 -0.0210
0.028 0.0364 0.0210
8
5 11 6 12 1 7 8 2 9 3 10 4
7 11 12 6 5 1 4 2 3 9 8 10
8 12 7 1 6 2 5 3 4 10 11 9
7 9 8 2 3 1 4 6 5 11 12 10
10 8 9 3 2 4 1 5 6 12 11 7
11 9 10 4 3 5 6 2 1 7 8 12
10 12 11 5 6 4 1 3 2 8 9 7
5 11 6 12 7 1 2 8 9 3 4 10
2
12
0.028 0.0000 0.0420
0.028 -0.0364 0.0210
0.028 -0.0364 -0.0210
A.4. Graphical Output File

0.028 0.0000 -0.0420  
0.028 0.0364 -0.0210  
0.028 0.0364 0.0210  
0.13 0.0000 0.0215  
0.13 -0.0186 0.0108  
0.13 -0.0186 -0.0108  
0.13 0.0000 -0.0215  
0.13 0.0186 -0.0108  
0.13 0.0186 0.0108  
8  
5 6 1 2 3 4  
7 12 1 2 9 8  
8 7 2 3 10 9  
9 8 3 4 11 10  
10 9 4 5 12 11  
11 10 5 6 7 12  
5 6 1 2 8 7 9 10 12 11  
11 5 6 12 1 7 8 2 3 9 10 4  
3  
1  
12  
0.004 0.0000 0.0380  
0.004 -0.0329 0.0190  
0.004 -0.0329 -0.0190  
0.004 0.0000 -0.0380  
0.004 0.0329 -0.0190  
0.004 0.0329 0.0190  
0.028 0.0000 0.0380  
0.028 -0.0329 0.0190  
0.028 -0.0329 -0.0190  
0.028 0.0000 -0.0380  
0.028 0.0329 -0.0190  
0.028 0.0329 0.0190  
8  
5 11 6 12 1 7 8 2 9 3 10 4  
11 7 12 6 1 5 2 4 3 9 10 8  
8 12 7 1 6 2 5 3 4 10 11 9  
9 7 8 2 1 3 6 4 5 11 10 12  
8 10 9 3 4 2 5 1 6 12 7 11  
11 9 10 4 3 5 6 2 1 7 8 12  
12 10 11 5 4 6 3 1 2 8 7 9  
5 11 6 12 7 1 2 8 9 3 4 10  
2  
12  
0.028 0.0000 0.0380  
0.028 -0.0329 0.0190  
0.028 -0.0329 -0.0190  
0.028 0.0000 -0.0380  
0.028 0.0329 -0.0190  
0.028 0.0329 0.0190  
169
<p>| | | | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.028</td>
<td>0.0329</td>
<td>0.0190</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.096</td>
<td>0.0000</td>
<td>0.0260</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.096</td>
<td>-0.0225</td>
<td>0.0130</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.096</td>
<td>-0.0225</td>
<td>-0.0130</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.096</td>
<td>0.0000</td>
<td>-0.0260</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.096</td>
<td>0.0225</td>
<td>-0.0130</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.096</td>
<td>0.0225</td>
<td>0.0130</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>11 5 6 12 7 1 8 2 3 9 10 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7 12 6 1 2 3 9 8</td>
<td>8 7 1 2 3 4 10 9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9 8 2 3 4 5 11 10</td>
<td>10 9 3 4 5 6 12 11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11 10 4 5 6 1 7 12</td>
<td>5 6 1 2 8 7 9 10 12 11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11 5 6 12 7 1 8 2 3 9 4 10</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

8
| 12   | 0.096 | 0.0000 | 0.0170 |     |     |     |     |     |     |     |     |     |     |     |
| 0.096 | -0.0147 | 0.0085 |     |     |     |     |     |     |     |     |     |     |     |     |
| 0.096 | -0.0147 | -0.0085 |     |     |     |     |     |     |     |     |     |     |     |     |
| 0.096 | 0.0000 | -0.0170 |     |     |     |     |     |     |     |     |     |     |     |     |
| 0.096 | 0.0147 | -0.0085 |     |     |     |     |     |     |     |     |     |     |     |     |
| 0.096 | 0.0147 | 0.0085 |     |     |     |     |     |     |     |     |     |     |     |     |
| 0.126 | 0.0000 | 0.0170 |     |     |     |     |     |     |     |     |     |     |     |     |
| 0.126 | -0.0147 | 0.0085 |     |     |     |     |     |     |     |     |     |     |     |     |
| 0.126 | -0.0147 | -0.0085 |     |     |     |     |     |     |     |     |     |     |     |     |
| 0.126 | 0.0000 | -0.0170 |     |     |     |     |     |     |     |     |     |     |     |     |
| 0.126 | 0.0147 | -0.0085 |     |     |     |     |     |     |     |     |     |     |     |     |
| 0.126 | 0.0147 | 0.0085 |     |     |     |     |     |     |     |     |     |     |     |     |
| 8    | 5 11 6 12 1 7 8 2 9 3 10 4 |     |     |     |     |     |     |     |     |     |     |     |     |     |
| 11 7 12 6 1 5 2 4 3 9 10 8 | 11 7 12 6 1 5 2 4 3 9 10 8 |     |     |     |     |     |     |     |     |     |     |     |     |     |
| 8 12 7 1 6 2 5 3 4 10 11 9 | 9 7 8 2 1 3 6 4 5 11 10 12 |     |     |     |     |     |     |     |     |     |     |     |     |     |
| 8 10 9 3 4 2 5 1 6 12 7 11 | 11 9 10 4 3 5 6 2 1 7 8 12 |     |     |     |     |     |     |     |     |     |     |     |     |     |
| 5 4 6 3 1 2 8 7 9 12 10 11 | 5 4 6 3 1 2 8 7 9 12 10 11 |     |     |     |     |     |     |     |     |     |     |     |     |     |
| 5 11 6 12 7 1 2 8 9 3 4 10 | 0     |     |     |     |     |     |     |     |     |     |     |     |     |     |
Appendix B

Proof of Equation 1.1

This proof follows the work done by Ball [1] in his discussion on the vulnerability to externally detonating HE warheads.

B.1 Symbols

- $A_{vi}$: Vulnerable area for the $i$th component, given a hit.
- $A_{pi}$: The presented area for the $i$th component in the plane normal to the direction of the approaching fragment.
- $P_{k/hi}$: The kill probability of the $i$th component given that it was hit.
- $P_{K/H}^{(j)}$: The probability that the target is killed on the $j$th hit after surviving the first $(n-1)$.
- $A_{V}^{(j)}$: The vulnerable area for the $j$th hit.
- $P_{K}$: The probability of a target kill, given specific encounter conditions, due to the burst of a specific warhead.
- $n$: Number of fragments independently, randomly, hitting the presented area of the target.
- $\phi$: The average number of fragments per unit area for the presented area.

B.2 Assumptions

1. The fragments travel in parallel shotlines.
2. The fragments hit the target randomly.
3. $A_{V}^{(j)}$ is constant for all hits.
4. The effect of the $j$th fragment hitting the target is independent from the damage caused by the previous $(j - 1)$ fragments.
B.3  Proof

A vulnerable area can mathematically be defined as,

\[ A_{vi} = A_{pi} \cdot P_{k/h}. \quad (B.1) \]

From Equation B.1 and the definition of \( P^{(j)}_{K/H} \) it follows that,

\[ A^{(j)}_V = A_P \cdot P^{(j)}_{K/H}. \quad (B.2) \]

and thus,

\[ P^{(j)}_{K/H} = \frac{A^{(j)}_V}{A_P}. \quad (B.3) \]

The probability that a target will not be killed given the \( j \)th random hit on it is,

\[ 1 - P^{(j)}_{K/H}. \quad (B.4) \]

The probability that the target is not killed is therefore the product of the individual events for all the independent random hits,

\[ \prod_{j=1}^{n} (1 - P^{(j)}_{K/H}). \quad (B.5) \]

Hence, \( P_K \), can be defined as,

\[ P_K = 1 - \prod_{j=1}^{n} (1 - P^{(j)}_{K/H}). \quad (B.6) \]

From Equation B.5 it follows that,

\[ e^{(\ln \prod_{j=1}^{n} (1 - P^{(j)}_{K/H}))} = e^{\sum_{j=1}^{n} \ln (1 - P^{(j)}_{K/H})}. \quad (B.7) \]

For a small \( P^{(j)}_{K/H} \), as can be seen in Figure B.1,

\[ \ln (1 - P^{(j)}_{K/H}) \approx -P^{(j)}_{K/H}, \quad (B.8) \]

and thus,

\[ \prod_{j=1}^{n} (1 - P^{(j)}_{K/H}) \approx e^{-\sum (P^{(j)}_{K/H})}. \quad (B.9) \]

(Note: A similar assumption have been made in Ball and by other researchers in the field.)

It now follows that Equation B.6 can be rewritten as,

\[ P_K = 1 - e^{-\sum (P^{(j)}_{K/H})}. \quad (B.10) \]

Following Equation B.3,

\[ \sum_{j=1}^{n} P^{(j)}_{K/N} = \sum_{j=1}^{n} \frac{A^{(j)}_V}{A_P}. \quad (B.11) \]

Substituting Equation B.11 into Equation B.10,

\[ P_K = 1 - e^{-\sum (\frac{A^{(j)}_n}{A_{vi}})}. \quad (B.12) \]
The number of hits on the target is calculated by multiplying the presented area and average number of hits per unit area,

\[ n = \phi A_P, \quad \text{(B.13)} \]

and thus,

\[ A_P = \frac{n}{\phi}. \quad \text{(B.14)} \]

Substituting Equation B.14 into Equation B.12, \( P_K \) simplifies to

\[ P_K = 1 - e^{-\frac{\phi}{A_V} \sum_{j=1}^{n} A_V}. \quad \text{(B.15)} \]

Then according to Assumption 3 it follows that,

\[ P_K = 1 - e^{-A_V \phi}. \quad \text{(B.16)} \]

Figure B.1: \( \ln(1-p) \) vs \(-p\) where \( p \) is very small.
APPENDIX B. PROOF OF EQUATION 1.1
Appendix C

The Missile

A typical guided missile is build up by various building blocks. These are the warhead and fuzing or armament section, the propulsion section (rocket motor), control surfaces (e.g. wings and fins), and guidance section. The arrangement of these building blocks varies from missile to missile, Figure C.1 and C.2 shows two examples of missiles.

The guidance section of the missile can probably be considered as its brain. It is typically a small computer which most of the time comprise of very powerful processors to effect the extremely fast calculations and guidance signals required for the resultant speeds at which a missile needs to perform. The guidance section thus comprise of the sensing system as well as processing section.

From the above it can be deduced that the guidance and control sections are closely integrated. The control surfaces, e.g. the wings and fins, that make up the control section thus effects the calculations signalled from the guidance section. The various guidance models are discussed in detail in Section 4.2.3.[30]

As mentioned, the armament section comprise of the payload, fuzing and firing system, safety
and arming (S&A) devices, as well as target-detecting devices (TDDs.) *Payload* is the explosive charge carried in the warhead. This study focuses mainly on high-explosive warheads.[30] This is discussed in more detail in Chapter 2.4.

The fuze and firing system is typically located next to the warhead. The system has three main components, namely, the fuze, S&A device, and the TDD. The two most common fuze types in guided missiles are proximity (VT fuze from its code name in WW II implying variable time) and contact fuzes, used individually or in conjunction (known as multifuzing). These fuzes are normally subject to delayed arming with acceleration force serving as the arming trigger. Arming thus takes place after the fuze have been subjected to a certain level of acceleration for a given length of time. The S&A devices are normally electromagnetic, explosive control devices that keep the explosive train of the fuze system unaligned until specific requirements, e.g. the acceleration requirements, are met. Upon impact the force causes a firing switch within the contact fuze to close, completing the firing circuit, and detonating the warhead. A proximity fuze on the other hand is triggered by some device detecting that the missile is in the proximity of the target. Apart from the detecting of the target by a TDD, the TDD also need to calculate the moment of fire referred to as the delay. The arming of a proximity fuze is based on the fuze logic and a combination of the magnitude and change rate of the missile. The three fuze TDDs, namely active, semi-active, and passive, are discussed in Section 4.2.3. When proximity-fuzed warheads are used there is typically also a time-fuze which initiate self-destruction after a given period of time to neutralise the warhead in the case of a miss. [30]

The propulsion system is typically jet powered for guided missiles. The atmospheric- and thermal jet systems are the two basic types employed. The main difference between the two systems being that the oxygen required for initiating and sustaining the burning of the fuel is obtained from the surrounding atmosphere in the case of the atmospheric jet system whilst the thermal jet system sustains combustion with onboard oxygen. An atmospheric jet propulsion system can be a turbojet, pulsejet, or ramjet engine. The turbojet is the most common with one of its advantages being that it does not require boosting and can begin operating at zero acceleration. It comprises of an air intake, a turbine driven mechanical compressor, a combustion chamber, and an exhaust. Thermal jets can either be solid propellant, liquid propellant, or combined propellant. The solid and combined propellant missiles are the most common in air-launched guided missiles. [30]
Appendix D

Intercept Scenarios data files

D.1 Endgame Scenario Distributions Save File

D.1.1 File Format

The format of the file to which a specific endgame scenario distribution setup is saved is given below.

• Closing Speed Distribution Type
• Repeat for each Closing Speed Distribution Parameters
• Roll Distribution Type
• Repeat for each Roll Distribution Parameters
• Yaw Distribution Type
• Repeat for each Yaw Distribution Parameters
• Pitch Distribution Type
• Repeat for each Pitch Distribution Parameters
• Miss Distance Distribution Type
• Repeat for each Miss Distance Distribution Parameters
• Cylinder Angle Distribution Type
• Repeat for each Cylinder Angle Distribution Parameters
• Velocity Elevation Distribution Type
• Repeat for each Velocity Elevation Distribution Parameters
• Velocity Heading Distribution Type
• Repeat for each Velocity Heading Distribution Parameters
• Target Roll degrees
• Target Roll x-direction
• Target Roll y-direction
• Target Roll z-direction
• Internal point x co-ordinate
• Internal point y co-ordinate
• Internal point z co-ordinate
• Drag (Yes / No)

### D.1.2 Distribution Type/Parameters setups

<table>
<thead>
<tr>
<th>Distribution Type</th>
<th>Beta</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribution Parameters</td>
<td>Parameter 1</td>
</tr>
<tr>
<td></td>
<td>Parameter 2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Distribution Type</th>
<th>Discrete</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribution Parameters</td>
<td>Number of Intervals</td>
</tr>
<tr>
<td></td>
<td>Probability, Value - Repeated for each of the intervals</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Distribution Type</th>
<th>Exponential</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribution Parameters</td>
<td>Parameter 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Distribution Type</th>
<th>Gamma</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribution Parameters</td>
<td>Parameter 1</td>
</tr>
<tr>
<td></td>
<td>Parameter 2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Distribution Type</th>
<th>Normal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribution Parameters</td>
<td>Parameter 1</td>
</tr>
<tr>
<td></td>
<td>Parameter 2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Distribution Type</th>
<th>Poison</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribution Parameters</td>
<td>Parameter 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Distribution Type</th>
<th>Uniform</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribution Parameters</td>
<td>Parameter 1</td>
</tr>
<tr>
<td></td>
<td>Parameter 2</td>
</tr>
</tbody>
</table>

### D.1.3 Example of Endgame Scenario Distributions Save File

```
Beta ! Closing Speed
1    ! Parameter 1
2    ! Parameter 2
Uniform ! Roll
0    ! Parameter 1
```
1.57 ! Parameter 2
Discrete ! Yaw
4 ! No of Intervals
0.25 0 ! Prob., Value
0.25 0.57 ! Prob., Value
0.25 1 ! Prob., Value
0.25 1.57 ! Prob., Value
Normal ! Pitch
1 ! Parameter 1
0.5 ! Parameter 2
Normal ! Miss Distance
2 ! Parameter 1
0.5 ! Parameter 2
Uniform ! Cylinder Angle
0 ! Parameter 1
1.57 ! Parameter 2
Discrete ! Velocity Elevation
3 ! No of Intervals
0.33 0 ! Prob., Value
0.34 0.76 ! Prob., Value
0.33 1.23 ! Prob., Value
Discrete ! Velocity Heading
4 ! No of Intervals
0.2 0 ! Prob., Value
0.3 0.5 ! Prob., Value
0.3 1 ! Prob., Value
0.2 1.5 ! Prob., Value
90 deg ! Target Roll
1 ! x-direction
0 ! y-direction
0 ! z-direction
8 ! Internal point x co-ordinate
0 ! Internal point y co-ordinate
0 ! Internal point z co-ordinate
False ! Drag

D.2 Path File

D.2.1 File Format

- File name - Title
- Coordinates defining Rmin
- Aimpoint in target coordinates
- Fragment drag option: 0 no drag; 1 drag
- Missile Speed (m/s)
• Missile travel direction (target system)
• The direction of the missile system (x-axis)
• The direction of the missile system (y-axis)
• The direction of the missile system (z-axis)
• Number of burst point on the path
• Distance from minimum miss distance position, (x, y, z) co-ordinates of burst point in target system
• Repeat for each burst point
• Target Speed (m/s)
• Target travel direction (global system)
• The direction of the target system (x-axis)
• The direction of the target system (y-axis)
• The direction of the target system (z-axis)
• Target position in global system

D.2.2 Example of Path File

C:\Akademie\Eva\Program\Test_Paths\Path_1
0.0000 0.0000 0.0000 ! Coordinates defining Rmin
8 0 0 ! Aimpoint in target coordinates
0 ! Idrag
0.4659465 ! Missile Speed
1 0 0 ! Missile vel. vector dir.
0.07600258 -0.4051109 1.039283 ! Missile vel. vector dir.
-0.2485917 0.8721308 -1.085168 ! Missile vel. vector dir.
-1.278732 0.5822767 -0.1606169 ! Missile vel. vector dir.
20! Number of missiles
-9 -1 0.319794 1.707821 1
-8 0 0.319794 1.707821 1
-7 1 0.319794 1.707821 1
-6 2 0.319794 1.707821 1
-5 3 0.319794 1.707821 1
-4 4 0.319794 1.707821 1
-3 5 0.319794 1.707821 1
-2 6 0.319794 1.707821 1
-1 7 0.319794 1.707821 1
0 8 0.319794 1.707821 1
1 9 0.319794 1.707821 1
2 10 0.319794 1.707821 1
3 11 0.319794 1.707821 1
4 12 0.319794 1.707821 1
<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>13</td>
<td>0.319794</td>
<td>1.707821</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>14</td>
<td>0.319794</td>
<td>1.707821</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>15</td>
<td>0.319794</td>
<td>1.707821</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>16</td>
<td>0.319794</td>
<td>1.707821</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>17</td>
<td>0.319794</td>
<td>1.707821</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>18</td>
<td>0.319794</td>
<td>1.707821</td>
<td>1</td>
</tr>
</tbody>
</table>

0.000 ! Target Velocity

-1.0000 0.0000 0.0000 ! Target travel direction

1.0000 0.0000 0.0000 ! The dir. of target syst. (ex)

0.0000 1.0000 0.0000 ! The dir. of target syst. (ey)

0.0000 0.0000 1.0000 ! The dir. of target syst. (ez)

0.0000 0.0000 0.0000 ! Target position in earth system
Appendix E

Validation of distribution generation functions

A number of tests were run to confirm the validity of the values generated by the functions describing the various distributions used to generate various intercept paths. This was done using a program called Input Analyzer created by Rockwell Software and part of the Arena suite. The data points generated from a specific distribution with specific input parameters were given as input for the Input Analyzer. This data was then analysed and the analysis results given as various summaries. The summaries that will be given for each test in this document are the Chi Square Test, Data Summary, Distribution Summary, Fit All Summary, and Histogram Summary. The more important of these are the Fit All Summary where various distributions are returned in a descending order, best to least appropriate, according to the Square Error, as well as the specific distribution summary and Chi Square Test. The test results for the various distributions will be given in this summarised form.
### E.1 Beta

#### E.1.1 Input parameters for EVA: Beta(1, 6)

**Input Analyzer Results**

<table>
<thead>
<tr>
<th>Distribution Summary</th>
<th>Data Summary</th>
<th>Fit All Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribution: Beta</td>
<td>Number of Data Points = 50000</td>
<td>Function</td>
</tr>
<tr>
<td>Expression: BETA(0.966, 5.38212)</td>
<td>Min Data Value = 1.27e-005</td>
<td>Beta</td>
</tr>
<tr>
<td>Square Error: 0.000063</td>
<td>Max Data Value = 0.856</td>
<td>Gamma</td>
</tr>
<tr>
<td></td>
<td>Sample Mean = 0.143</td>
<td>Weibull</td>
</tr>
<tr>
<td></td>
<td>Sample Std Dev = 0.125</td>
<td>Erlang</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Exponential</td>
</tr>
</tbody>
</table>

**Chi Square Test**

- Number of intervals = 32
- Degrees of freedom = 29
- Test Statistic = 53.5
- Corresponding p-value < 0.005

**Histogram Summary**

- Histogram Range = 0 to 0.94
- Number of Intervals = 40

<table>
<thead>
<tr>
<th>Function</th>
<th>Sq Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lognormal</td>
<td>0.0045</td>
</tr>
<tr>
<td>Normal</td>
<td>0.0179</td>
</tr>
<tr>
<td>Triangular</td>
<td>0.0276</td>
</tr>
<tr>
<td>Uniform</td>
<td>0.052</td>
</tr>
</tbody>
</table>

**Table E.1:** Data analyses summary for Beta(1, 6).

**Figure E.1:** Histogram of data and fit for Beta(0.966, 5.38212).
E.1.2 Input parameters for EVA: Beta(6, 15)

Input Analyzer Results

<table>
<thead>
<tr>
<th>Distribution Summary</th>
<th>Data Summary</th>
<th>Fit All Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribution: Beta</td>
<td>Number of Data Points = 50000</td>
<td>Function</td>
</tr>
<tr>
<td>Expression: BETA(5.78, 13.2188)</td>
<td>Min Data Value = 0.0185</td>
<td>Beta</td>
</tr>
<tr>
<td>Square Error: 0.000053</td>
<td>Max Data Value = 0.73</td>
<td>5.26e-005</td>
</tr>
<tr>
<td></td>
<td>Sample Mean = 0.286</td>
<td>Gamma</td>
</tr>
<tr>
<td></td>
<td>Sample Std Dev = 0.0967</td>
<td>0.000261</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Erlang</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.000273</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Weibull</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.00043</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Normal</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.000855</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Lognormal</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.00134</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Triangular</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.0204</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Uniform</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.0435</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Exponential</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.0456</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chi Square Test</th>
<th>Histogram Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of intervals = 27</td>
<td>Histogram Range = 0 to 0.94</td>
</tr>
<tr>
<td>Degrees of freedom = 24</td>
<td>Number of Intervals = 40</td>
</tr>
<tr>
<td>Test Statistic = 56.8</td>
<td></td>
</tr>
<tr>
<td>Corresponding p-value &lt; 0.005</td>
<td></td>
</tr>
</tbody>
</table>

Table E.2: Data analyses summary for Beta(6, 15).

Figure E.2: Histogram of data and fit for Beta(5.78, 13.2188).
### E.1.3 Input parameters for EVA: Beta(10, 10)

**Input Analyzer Results**

<table>
<thead>
<tr>
<th>Distribution Summary</th>
<th>Data Summary</th>
<th>Fit All Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribution: Beta</td>
<td>Number of Data Points = 50000</td>
<td>Function</td>
</tr>
<tr>
<td>Expression: BETA(9.26, 8.12886)</td>
<td>Min Data Value = 0.0124</td>
<td>Beta</td>
</tr>
<tr>
<td>Square Error: 0.000041</td>
<td>Max Data Value = 0.893</td>
<td>Normal</td>
</tr>
<tr>
<td></td>
<td>Sample Mean = 0.501</td>
<td>Gamma</td>
</tr>
<tr>
<td></td>
<td>Sample Std Dev = 0.109</td>
<td>Erlang</td>
</tr>
<tr>
<td><strong>Chi Square Test</strong></td>
<td><strong>Histogram Summary</strong></td>
<td>Weibull</td>
</tr>
<tr>
<td>Number of intervals = 30</td>
<td>Histogram Range = 0 to 0.94</td>
<td>Lognormal</td>
</tr>
<tr>
<td>Degrees of freedom = 27</td>
<td>Number of Intervals = 40</td>
<td>Triangular</td>
</tr>
<tr>
<td>Test Statistic = 77.7</td>
<td></td>
<td>Uniform</td>
</tr>
<tr>
<td>Corresponding p-value &lt; 0.005</td>
<td></td>
<td>Exponential</td>
</tr>
</tbody>
</table>

*Table E.3: Data analyses summary for Beta(10, 10).*

*Figure E.3: Histogram of data and fit for Beta(9.26, 8.12886).*
E.1.4 Input parameters for EVA: Beta(7, 2)

Input Analyzer Results

<table>
<thead>
<tr>
<th>Distribution Summary</th>
<th>Data Summary</th>
<th>Fit All Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribution: Beta</td>
<td>Number of Data Points = 45953</td>
<td>Function</td>
</tr>
<tr>
<td>Expression: BET A(6.17, 1.43731)</td>
<td>Min Data Value = 0.0172</td>
<td>Beta</td>
</tr>
<tr>
<td>Square Error: 0.000403</td>
<td>Max Data Value = 0.94</td>
<td>Weibull</td>
</tr>
<tr>
<td></td>
<td>Sample Mean = 0.761</td>
<td>Normal</td>
</tr>
<tr>
<td></td>
<td>Sample Std Dev = 0.125</td>
<td>Erlang</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Gamma</td>
</tr>
<tr>
<td>Chi Square Test</td>
<td></td>
<td>Lognormal</td>
</tr>
<tr>
<td>Number of intervals = 31</td>
<td></td>
<td>Triangular</td>
</tr>
<tr>
<td>Degrees of freedom = 28</td>
<td></td>
<td>Uniform</td>
</tr>
<tr>
<td>Test Statistic = 363</td>
<td></td>
<td>Exponential</td>
</tr>
<tr>
<td>Corresponding p-value &lt; 0.005</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Histogram Range = 0 to 0.94</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Number of Intervals = 40</td>
<td></td>
</tr>
</tbody>
</table>

Table E.4: Data analyses summary for Beta(7, 2).

Figure E.4: Histogram of data and fit for Beta(6.17, 1.43731).
E.1.5 Input parameters for EVA: Beta(16, 4)

Input Analyzer Results

<table>
<thead>
<tr>
<th>Distribution Summary</th>
<th>Data Summary</th>
<th>Fit All Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribution: Beta</td>
<td>Number of Data Points = 48801</td>
<td>Function</td>
</tr>
<tr>
<td>Expression: BETA(12.6, 2.28084)</td>
<td>Min Data Value = 0.363</td>
<td>Weibull</td>
</tr>
<tr>
<td>Square Error: 0.000912</td>
<td>Max Data Value = 0.94</td>
<td>Beta</td>
</tr>
<tr>
<td></td>
<td>Sample Mean = 0.796</td>
<td>Normal</td>
</tr>
<tr>
<td></td>
<td>Sample Std Dev = 0.0847</td>
<td>Lognormal</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Erlang</td>
</tr>
<tr>
<td>Chi Square Test</td>
<td></td>
<td>Gamma</td>
</tr>
<tr>
<td>Number of intervals = 24</td>
<td></td>
<td>Triangular</td>
</tr>
<tr>
<td>Degrees of freedom = 21</td>
<td>Histogram Range = 0 to 0.94</td>
<td>Uniform</td>
</tr>
<tr>
<td>Test Statistic = 1.19e+003</td>
<td>Number of Intervals = 40</td>
<td>Exponential</td>
</tr>
<tr>
<td>Corresponding p-value &lt; 0.005</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table E.5: Data analyses summary for Beta(16, 4).

Figure E.5: Histogram of data and fit for Beta(12.6, 2.28084).
E.2 Exponential

E.2.1 Input parameters for EVA: Expo(1.25)

Input Analyzer Results

<table>
<thead>
<tr>
<th>Distribution Summary</th>
<th>Data Summary</th>
<th>Fit All Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribution: Exponential</td>
<td>Number of Data Points = 50000</td>
<td><strong>Function</strong></td>
</tr>
<tr>
<td>Expression: EXPO(1.25)</td>
<td>Min Data Value = 2.05e-005</td>
<td>Exponential</td>
</tr>
<tr>
<td>Square Error: 0.000009</td>
<td>Max Data Value = 14.2</td>
<td>Erlang</td>
</tr>
<tr>
<td></td>
<td>Sample Mean = 1.25</td>
<td>Weibull</td>
</tr>
<tr>
<td></td>
<td>Sample Std Dev = 1.25</td>
<td>Beta</td>
</tr>
</tbody>
</table>

**Chi Square Test**

- Number of intervals = 26
- Degrees of freedom = 24
- Test Statistic = 23.8
- Corresponding p-value = 0.475

**Histogram Summary**

- Histogram Range = 0 to 15
- Number of Intervals = 40

**Table E.6:** Data analyses summary for Expo(1.25).

![Histogram of data and fit for Expo(1.25).](image)

**Figure E.6:** Histogram of data and fit for Expo(1.25).
APPENDIX E. VALIDATION OF DISTRIBUTION GENERATION FUNCTIONS

E.2.2 Input parameters for EVA: Expo(2.5)

Input Analyzer Results

<table>
<thead>
<tr>
<th>Distribution Summary</th>
<th>Data Summary</th>
<th>Fit All Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribution: Exponential</td>
<td>Number of Data Points = 50000</td>
<td>Function</td>
</tr>
<tr>
<td>Expression: EXPO(2.52)</td>
<td>Min Data Value = 2.58e-005</td>
<td>Exponential</td>
</tr>
<tr>
<td>Square Error: 0.000009</td>
<td>Max Data Value = 29.3</td>
<td>Erlang</td>
</tr>
<tr>
<td></td>
<td>Sample Mean = 2.52</td>
<td>Weibull</td>
</tr>
<tr>
<td></td>
<td>Sample Std Dev = 2.52</td>
<td>Beta</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Lognormal</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chi Square Test</td>
<td></td>
<td>Histogram Summary</td>
</tr>
<tr>
<td>Number of intervals = 27</td>
<td></td>
<td>Normal</td>
</tr>
<tr>
<td>Degrees of freedom = 25</td>
<td></td>
<td>Triangular</td>
</tr>
<tr>
<td>Test Statistic = 21</td>
<td></td>
<td>Uniform</td>
</tr>
<tr>
<td>Corresponding p-value = 0.692</td>
<td></td>
<td>Gamma</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table E.7: Data analyses summary for Expo(2.5).

Figure E.7: Histogram of data and fit for Expo(2.52).
### E.2.3 Input parameters for EVA: Exp(11.11)

**Input Analyzer Results**

<table>
<thead>
<tr>
<th>Distribution Summary</th>
<th>Data Summary</th>
<th>Fit All Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribution: Exponential</td>
<td>Number of Data Points = 50000</td>
<td>Function</td>
</tr>
<tr>
<td>Expression: EXP(11.1)</td>
<td>Min Data Value = 9.52e-005</td>
<td>Exponential</td>
</tr>
<tr>
<td>Square Error: 0.000026</td>
<td>Max Data Value = 133</td>
<td>Erlang</td>
</tr>
<tr>
<td></td>
<td>Sample Mean = 11.1</td>
<td>Gamma</td>
</tr>
<tr>
<td></td>
<td>Sample Std Dev = 11.2</td>
<td>Weibull</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Beta</td>
</tr>
<tr>
<td><strong>Chi Square Test</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of intervals = 26</td>
<td>Histogram Summary</td>
<td></td>
</tr>
<tr>
<td>Degrees of freedom = 24</td>
<td></td>
<td>Lognormal</td>
</tr>
<tr>
<td>Test Statistic = 28.1</td>
<td></td>
<td>Normal</td>
</tr>
<tr>
<td>Corresponding p-value = 0.258</td>
<td></td>
<td>Triangular</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Uniform</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Sq Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exponential</td>
<td>2.63e-005</td>
</tr>
<tr>
<td>Erlang</td>
<td>2.63e-005</td>
</tr>
<tr>
<td>Gamma</td>
<td>3.19e-005</td>
</tr>
<tr>
<td>Weibull</td>
<td>0.000217</td>
</tr>
<tr>
<td>Beta</td>
<td>0.00119</td>
</tr>
<tr>
<td>Lognormal</td>
<td>0.00513</td>
</tr>
<tr>
<td>Normal</td>
<td>0.0458</td>
</tr>
<tr>
<td>Triangular</td>
<td>0.0971</td>
</tr>
<tr>
<td>Uniform</td>
<td>0.125</td>
</tr>
</tbody>
</table>

Table E.8: Data analyses summary for Exp(11.11).

![Histogram of data and fit for Exp(11.1).](image)

Figure E.8: Histogram of data and fit for Exp(11.1).
E.3 Gamma

E.3.1 Input parameters for EVA: Gamma(1, 3)

Input Analyzer Results

<table>
<thead>
<tr>
<th>Distribution Summary</th>
<th>Data Summary</th>
<th>Fit All Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribution: Gamma</td>
<td>Number of Data Points = 50000</td>
<td>Function</td>
</tr>
<tr>
<td>Expression: GAMM(0.995, 3.01)</td>
<td>Min Data Value = 0.027</td>
<td>Erlang</td>
</tr>
<tr>
<td>Square Error: 0.000021</td>
<td>Max Data Value = 14.2</td>
<td>Gamma</td>
</tr>
<tr>
<td></td>
<td>Sample Mean = 3</td>
<td>Beta</td>
</tr>
<tr>
<td>Chi Square Test</td>
<td>Sample Std Dev = 1.73</td>
<td>Weibull</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Lognormal</td>
</tr>
<tr>
<td>Number of intervals = 33</td>
<td></td>
<td>Normal</td>
</tr>
<tr>
<td>Degrees of freedom = 30</td>
<td></td>
<td>Triangular</td>
</tr>
<tr>
<td>Test Statistic = 38.5</td>
<td></td>
<td>Exponential</td>
</tr>
<tr>
<td>Corresponding p-value = 0.147</td>
<td>Number of Intervals = 40</td>
<td>Uniform</td>
</tr>
</tbody>
</table>

Table E.9: Data analyses summary for Gamma(1, 3).

Figure E.9: Histogram of data and fit for Gamma(0.995, 3.01).
E.3.2 Input parameters for EVA: Gamma(0.2, 11)

Input Analyzer Results

<table>
<thead>
<tr>
<th>Distribution Summary</th>
<th>Data Summary</th>
<th>Fit All Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribution: Gamma</td>
<td>Number of Data Points = 50000</td>
<td>Function:</td>
</tr>
<tr>
<td>Expression: GAMM(0.199, 11)</td>
<td>Min Data Value = 0.382</td>
<td>Erlang: 1.56e-005</td>
</tr>
<tr>
<td>Square Error: 0.000031</td>
<td>Max Data Value = 5.82</td>
<td>Gamma: 3.1e-005</td>
</tr>
<tr>
<td></td>
<td>Sample Mean = 2.2</td>
<td>Lognormal: 0.000379</td>
</tr>
<tr>
<td>Chi Square Test</td>
<td>Sample Std Dev = 0.66</td>
<td>Beta: 0.000447</td>
</tr>
<tr>
<td>Number of intervals = 32</td>
<td></td>
<td>Normal: 0.00123</td>
</tr>
<tr>
<td>Degrees of freedom = 29</td>
<td></td>
<td>Weibull: 0.00282</td>
</tr>
<tr>
<td>Test Statistic = 41.6</td>
<td>Histogram Range = 0 to 6</td>
<td>Triangular: 0.0171</td>
</tr>
<tr>
<td>Corresponding p-value = 0.0633</td>
<td>Number of Intervals = 40</td>
<td>Uniform: 0.0411</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Exponential: 0.0476</td>
</tr>
</tbody>
</table>

Table E.10: Data analyses summary for Gamma(0.2, 11).

![Histogram of data and fit for Gamma(0.199, 11).](image)
E.4 Normal

E.4.1 Input parameters for EVA: Normal(10, 2.5)

Input Analyzer Results

<table>
<thead>
<tr>
<th>Distribution Summary</th>
<th>Data Summary</th>
<th>Fit All Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribution: Normal</td>
<td>Number of Data Points = 50000</td>
<td>Function</td>
</tr>
<tr>
<td>Expression: NORM(9.95, 2.43)</td>
<td>Min Data Value = 0.516</td>
<td>Normal</td>
</tr>
<tr>
<td>Square Error: 0.000082</td>
<td>Max Data Value = 16.2</td>
<td>Beta</td>
</tr>
<tr>
<td></td>
<td>Sample Mean = 9.95</td>
<td>Weibull</td>
</tr>
<tr>
<td></td>
<td>Sample Std Dev = 2.43</td>
<td>Erlang</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Gamma</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Lognormal</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Triangular</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Uniform</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Exponential</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chi Square Test</th>
<th>Histogram Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of intervals = 37</td>
<td>Histogram Range = 0 to 17</td>
</tr>
<tr>
<td>Degrees of freedom = 34</td>
<td>Number of Intervals = 40</td>
</tr>
<tr>
<td>Test Statistic = 346</td>
<td></td>
</tr>
<tr>
<td>Corresponding p-value &lt; 0.005</td>
<td></td>
</tr>
</tbody>
</table>

Table E.11: Data analyses summary for Normal(10, 2.5).

Figure E.11: Histogram of data and fit for Normal(9.95, 2.43).
E.4.2 Input parameters for EVA: Normal(7, 1)

Input Analyzer Results

<table>
<thead>
<tr>
<th>Distribution Summary</th>
<th>Data Summary</th>
<th>Fit All Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribution: Normal</td>
<td>Number of Data Points = 50000</td>
<td>Function</td>
</tr>
<tr>
<td>Expression: NORM(6.98, 0.967)</td>
<td>Min Data Value = 2.25</td>
<td>Normal</td>
</tr>
<tr>
<td>Square Error: 0.000097</td>
<td>Max Data Value = 9.49</td>
<td>Beta</td>
</tr>
<tr>
<td></td>
<td>Sample Mean = 6.98</td>
<td>Weibull</td>
</tr>
<tr>
<td></td>
<td>Sample Std Dev = 0.967</td>
<td>Erlang</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Gamma</td>
</tr>
<tr>
<td>Chi Square Test</td>
<td></td>
<td>Histogram Summary</td>
</tr>
<tr>
<td>Number of intervals = 33</td>
<td></td>
<td>Lognormal</td>
</tr>
<tr>
<td>Degrees of freedom = 30</td>
<td>Histogram Range = 2 to 10</td>
<td>Triangular</td>
</tr>
<tr>
<td>Test Statistic = 387</td>
<td>Number of Intervals = 40</td>
<td>Uniform</td>
</tr>
<tr>
<td>Corresponding p-value &lt; 0.005</td>
<td></td>
<td>Exponential</td>
</tr>
</tbody>
</table>

Table E.12: Data analyses summary for Normal(7, 1).

Figure E.12: Histogram of data and fit for Normal(6.98, 0.967).
E.5 Poisson

E.5.1 Input parameters for EVA: Poisson(10)

Input Analyzer Results

<table>
<thead>
<tr>
<th>Distribution Summary</th>
<th>Data Summary</th>
<th>Fit All Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribution: Poisson</td>
<td>Number of Data Points = 50000</td>
<td>Function</td>
</tr>
<tr>
<td>Expression: POIS(10)</td>
<td>Min Data Value = 0</td>
<td>Sq Error</td>
</tr>
<tr>
<td>Square Error: 0.000011</td>
<td>Max Data Value = 27</td>
<td>Poisson 1.1e-005</td>
</tr>
<tr>
<td></td>
<td>Sample Mean = 10</td>
<td>Beta 0.000126</td>
</tr>
<tr>
<td>Chi Square Test</td>
<td>Sample Std Dev = 3.16</td>
<td>Gamma 0.00047</td>
</tr>
<tr>
<td>Number of intervals = 23</td>
<td></td>
<td>Normal 0.000515</td>
</tr>
<tr>
<td>Degrees of freedom = 21</td>
<td></td>
<td>Erlang 0.000551</td>
</tr>
<tr>
<td>Test Statistic = 16.1</td>
<td>Histogram Range = -0.5 to 27.5</td>
<td>Weibull 0.00129</td>
</tr>
<tr>
<td>Corresponding p-value &gt; 0.75</td>
<td>Number of Intervals = 28</td>
<td>Lognormal 0.00188</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Triangular 0.0206</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Uniform 0.0541</td>
</tr>
</tbody>
</table>

Table E.13: Data analyses summary for Poisson(10).

Figure E.13: Histogram of data and fit for Poisson(10).
E.5.2 Input parameters for EVA: Poisson(20)

Input Analyzer Results

<table>
<thead>
<tr>
<th>Distribution Summary</th>
<th>Data Summary</th>
<th>Fit All Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribution: Poisson</td>
<td>Number of Data Points = 50000</td>
<td>Function</td>
</tr>
<tr>
<td>Expression: POIS(20)</td>
<td>Min Data Value = 3</td>
<td>Poisson</td>
</tr>
<tr>
<td>Square Error: 0.000027</td>
<td>Max Data Value = 44</td>
<td>2.7e-005</td>
</tr>
<tr>
<td></td>
<td>Sample Mean = 20</td>
<td>Beta</td>
</tr>
<tr>
<td></td>
<td>Sample Std Dev = 4.5</td>
<td>9.44e-005</td>
</tr>
<tr>
<td>Chi Square Test</td>
<td></td>
<td>Normal</td>
</tr>
<tr>
<td>Number of intervals = 33</td>
<td></td>
<td>0.000246</td>
</tr>
<tr>
<td>Degrees of freedom = 31</td>
<td></td>
<td>Gamma</td>
</tr>
<tr>
<td>Test Statistic = 47.9</td>
<td></td>
<td>0.000269</td>
</tr>
<tr>
<td>Corresponding p-value = 0.0271</td>
<td></td>
<td>Erlang</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.000294</td>
</tr>
<tr>
<td>Histogram Summary</td>
<td></td>
<td>Weibull</td>
</tr>
<tr>
<td>Number of Intervals = 42</td>
<td></td>
<td>0.000514</td>
</tr>
<tr>
<td>Histogram Range = 2.5 to 44.5</td>
<td></td>
<td>Lognormal</td>
</tr>
<tr>
<td>Number of Intervals = 42</td>
<td></td>
<td>0.000998</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Triangular</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.0158</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Uniform</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.0391</td>
</tr>
</tbody>
</table>

Table E.14: Data analyses summary for Poisson(20).

![Histogram of data and fit for Poisson(20).](image)

Figure E.14: Histogram of data and fit for Poisson(20).
E.6 Uniform

E.6.1 Input parameters for EVA: Uniform(0, 10)

Input Analyzer Results

<table>
<thead>
<tr>
<th>Distribution Summary</th>
<th>Data Summary</th>
<th>Fit All Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribution: Uniform</td>
<td>Number of Data Points = 50000</td>
<td>Function</td>
</tr>
<tr>
<td>Expression: UNIF(0, 10)</td>
<td>Min Data Value = 9.61e-006</td>
<td>Uniform</td>
</tr>
<tr>
<td>Square Error: 0.000017</td>
<td>Max Data Value = 10</td>
<td>Beta</td>
</tr>
<tr>
<td></td>
<td>Sample Mean = 4.99</td>
<td>Normal</td>
</tr>
<tr>
<td></td>
<td>Sample Std Dev = 2.89</td>
<td>Weibull</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Gamma</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chi Square Test</td>
<td>Histogram Summary</td>
<td></td>
</tr>
<tr>
<td>Number of intervals = 40</td>
<td>Histogram Range = 0 to 10</td>
<td>Erlang</td>
</tr>
<tr>
<td>Degrees of freedom = 39</td>
<td>Number of Intervals = 40</td>
<td>Exponential</td>
</tr>
<tr>
<td>Test Statistic = 34.8</td>
<td></td>
<td>Lognormal</td>
</tr>
<tr>
<td>Corresponding p-value = 0.656</td>
<td></td>
<td>Triangular</td>
</tr>
</tbody>
</table>

Table E.15: Data analyses summary for Uniform(0, 10).

Figure E.15: Histogram of data and fit for Uniform(0, 10).
### E.6.2 Input parameters for EVA: Uniform(2.33, 7.77)

**Input Analyzer Results**

<table>
<thead>
<tr>
<th>Distribution Summary</th>
<th>Data Summary</th>
<th>Fit All Summary</th>
<th>Function</th>
<th>Sq Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribution: Uniform</td>
<td>Number of Data Points = 50000</td>
<td>Beta</td>
<td>0.000876</td>
<td></td>
</tr>
<tr>
<td>Expression: UNIF(2, 8)</td>
<td>Min Data Value = 2.33</td>
<td>Uniform</td>
<td>0.00229</td>
<td></td>
</tr>
<tr>
<td>Square Error: 0.002270</td>
<td>Max Data Value = 7.77</td>
<td>Weibull</td>
<td>0.00327</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sample Mean = 5.06</td>
<td>Normal</td>
<td>0.0035</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sample Std Dev = 1.57</td>
<td>Gamma</td>
<td>0.00393</td>
<td></td>
</tr>
<tr>
<td>Chi Square Test</td>
<td></td>
<td>Uniform</td>
<td>0.00229</td>
<td></td>
</tr>
<tr>
<td>Number of intervals = 40</td>
<td></td>
<td>Weibull</td>
<td>0.00327</td>
<td></td>
</tr>
<tr>
<td>Degrees of freedom = 39</td>
<td></td>
<td>Normal</td>
<td>0.0035</td>
<td></td>
</tr>
<tr>
<td>Test Statistic = 4.54e+003</td>
<td></td>
<td>Gamma</td>
<td>0.00393</td>
<td></td>
</tr>
<tr>
<td>Corresponding p-value &lt; 0.005</td>
<td></td>
<td>Uniform</td>
<td>0.00229</td>
<td></td>
</tr>
</tbody>
</table>

**Histogram Summary**

<table>
<thead>
<tr>
<th>Function</th>
<th>Sq Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Erlang</td>
<td>0.00418</td>
</tr>
<tr>
<td>Lognormal</td>
<td>0.00574</td>
</tr>
<tr>
<td>Triangular</td>
<td>0.00602</td>
</tr>
<tr>
<td>Exponential</td>
<td>0.0102</td>
</tr>
</tbody>
</table>

Table E.16: *Data analyses summary for Uniform(2.33, 7.77).*

![Histogram of data and fit for Uniform(2, 8)](image)

Figure E.16: *Histogram of data and fit for Uniform(2, 8).*
Appendix F

Transformation Matrices

F.1 Rotation Operators

To effect counterclockwise rotation about the positive $x$-axis through an angle of $\phi$ degrees the vector is multiplied by the following matrix,

$$
\begin{bmatrix}
1 & 0 & 0 \\
0 & \cos \theta & -\sin \theta \\
0 & \sin \theta & \cos \theta
\end{bmatrix}.
$$

(F.1)

To effect counterclockwise rotation about the positive $y$-axis through an angle of $\theta$ degrees the vector is multiplied by the following matrix,

$$
\begin{bmatrix}
\cos \theta & 0 & \sin \theta \\
0 & 1 & 0 \\
-\sin \theta & 0 & \cos \theta
\end{bmatrix}.
$$

(F.2)

To effect counterclockwise rotation about the positive $z$-axis through an angle of $\psi$ degrees the vector is multiplied by the following matrix,

$$
\begin{bmatrix}
\cos \theta & -\sin \theta & 0 \\
\sin \theta & \cos \theta & 0 \\
0 & 0 & 1
\end{bmatrix}.
$$

(F.3)

F.2 The Transformation Matrix

It is important that the axes be rotated in the reverse sequence of which the axes were rotated when the original angles were calculated. The sequence specified for determining the rotation angles used as input for EVA is roll ($\phi$), pitch ($\theta$), and yaw ($\psi$). It follows that the sequence in which the rotation matrixes must be multiplied to affect the correct rotation is yaw, pitch, and roll.

The first step is to rotate the co-ordinate system of the intercept path around the $z$-axis through the angle $\psi$. The rotation is always done in the positive, anti-clockwise direction. Making use of the rotation matrix F.1 the direction of the old $x$- and $y$-axis ($x$, $y$, $z$) in the new system ($X$, $Y$, $Z$) can be obtained as follows:
\[i = (i \cdot I)I + (i \cdot J)J + (i \cdot K)K,\]
\[i = \cos \psi I - \sin \psi J,\]  
\[(F.4)\]

\[j = (j \cdot I)I + (j \cdot J)J + (j \cdot K)K,\]
\[j = \sin \psi I + \cos \psi J,\]  
\[(F.5)\]

\[k = (k \cdot I)I + (k \cdot J)J + (k \cdot K)K,\]
\[k = K,\]  
\[(F.6)\]

The system is then rotated around the Y-axis \(\text{(see rotation matrix } F.2)\) through an angle \(\theta\) to the new coordinate system \((X', Y', Z')\):

\[I = (I \cdot I')I' + (I \cdot J')J' + (I \cdot K')K',\]
\[I = \cos \theta I' + \sin \theta K',\]  
\[(F.7)\]

\[J = (J \cdot I')I' + (J \cdot J')J' + (J \cdot K')K',\]
\[J = J',\]  
\[(F.8)\]

\[K = (K \cdot I')I' + (K \cdot J')J' + (K \cdot K')K',\]
\[K = -\sin \theta I' + \cos \theta K',\]  
\[(F.9)\]

The system is then rotated around the \(X'\)-axis \(\text{(see rotation matrix } F.3)\) through an angle \(\phi\) to the new coordinate system \((x', y', z')\):

\[I' = (I' \cdot i')i' + (I' \cdot j')j' + (I' \cdot k')k',\]
\[I' = \cos \phi j' - \sin \phi k',\]  
\[(F.10)\]

\[J' = (J' \cdot i')i' + (J' \cdot j')j' + (J' \cdot k')k',\]
\[J' = -\sin \phi,\]  
\[(F.11)\]

\[K' = (K' \cdot i')i' + (K' \cdot j')j' + (K' \cdot k')k',\]
\[K' = \sin \phi j' + \cos \phi k'.\]  
\[(F.12)\]

The following equations are obtained by substituting equations F.10, F.11, and F.12 into equations F.7, F.8, and F.9,

\[I = \cos \theta I' + \sin \theta (\sin \phi j' + \cos \phi k'),\]
\[I = \cos \theta I' + \sin \theta \sin \phi j' + \sin \theta \cos \phi k',\]  
\[(F.13)\]

\[J = \cos \phi j' - \sin \phi k',\]  
\[(F.14)\]

\[K = -\sin \theta I' + \cos \theta (\sin \phi j' + \cos \phi k'),\]
\[K = -\sin \theta I' + \cos \theta \sin \phi j' + \cos \theta \cos \phi k'.\]  
\[(F.15)\]

The transformation matrix is now obtained by substituting equations F.13, F.14, and F.15 into equations F.4, F.5, and F.6, and can be written as:
The Transformation Matrix

\[ S(\theta, \psi, \phi) = \begin{bmatrix} S_{11} & S_{12} & S_{13} \\ S_{21} & S_{22} & S_{23} \\ S_{31} & S_{32} & S_{33} \end{bmatrix}. \] (F.16)

where,

\[ S_{11} = \cos \psi \cos \theta, \] (F.17)
\[ S_{12} = \cos \psi \sin \theta \sin \phi - \sin \psi \cos \phi, \] (F.18)
\[ S_{13} = \cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi, \] (F.19)
\[ S_{21} = \sin \psi \cos \theta, \] (F.20)
\[ S_{22} = \sin \psi \sin \theta \sin \phi + \cos \psi \cos \phi, \] (F.21)
\[ S_{23} = \sin \psi \sin \theta \cos \phi - \sin \phi \cos \psi, \] (F.22)
\[ S_{31} = - \sin \theta, \] (F.23)
\[ S_{32} = \cos \theta \sin \phi, \] (F.24)
\[ S_{33} = \cos \theta \cos \phi. \] (F.25)
Appendix G

Drag Coefficients

These tables have been constructed from the graphs given by Janzon [25] which relates the Drag Coefficient to the Mach Number for a specific fragment shape.

G.1 Ball shaped fragments

<table>
<thead>
<tr>
<th>$V_{mach}$</th>
<th>$C_w$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{mach} \leq 0.33$</td>
<td>0.47</td>
</tr>
<tr>
<td>$0.33 &lt; V_{mach} \leq 0.8$</td>
<td>$0.48 + 0.13(V_{mach} - 0.33)$</td>
</tr>
<tr>
<td>$0.8 &lt; V_{mach} \leq 1.4$</td>
<td>$0.54 + 0.899(\log V_{mach}(x) + 0.22)$</td>
</tr>
<tr>
<td>$1.4 &lt; V_{mach} \leq 5.0$</td>
<td>$1.04 - 0.094(\log V_{mach}(x) - 0.336)$</td>
</tr>
<tr>
<td>$5.0 &lt; V_{mach}$</td>
<td>0.92</td>
</tr>
</tbody>
</table>

Table G.1: Ball shaped fragments.

G.2 Rectangularly (Cylindrical) shaped fragments

<table>
<thead>
<tr>
<th>$V_{mach}$</th>
<th>$C_w$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{mach} \leq 0.33$</td>
<td>0.8</td>
</tr>
<tr>
<td>$0.33 &lt; V_{mach} \leq 0.8$</td>
<td>$0.8 + 0.17(V_{mach} - 0.33)$</td>
</tr>
<tr>
<td>$0.8 &lt; V_{mach} \leq 1.2$</td>
<td>$0.88 + 0.996(\log V_{mach}(x) + 0.22)$</td>
</tr>
<tr>
<td>$1.2 &lt; V_{mach} \leq 3.6$</td>
<td>$1.28 - 0.164(\log V_{mach}(x) - 0.182)$</td>
</tr>
<tr>
<td>$3.6 &lt; V_{mach}$</td>
<td>1.1</td>
</tr>
</tbody>
</table>

Table G.2: Rectangularly (Cylindrical) shaped fragments.
G.3 Cubed shaped fragments

<table>
<thead>
<tr>
<th>$V_{mach}$</th>
<th>$C_w$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{mach} \leq 0.5$</td>
<td>0.8</td>
</tr>
<tr>
<td>$0.5 &lt; V_{mach} \leq 0.9$</td>
<td>$0.8 + 1.26(V_{mach} - 0.0625)$</td>
</tr>
<tr>
<td>$0.9 &lt; V_{mach} \leq 1.3$</td>
<td>$0.875 + 1.07(V_{mach} - 0.9)$</td>
</tr>
<tr>
<td>$1.3 &lt; V_{mach} \leq 1.6$</td>
<td>1.31</td>
</tr>
<tr>
<td>$1.6 &lt; V_{mach} \leq 2.5$</td>
<td>$1.31 - 0.211(V_{mach} - 1.6)$</td>
</tr>
<tr>
<td>$2.5 &lt; V_{mach} \leq 3.5$</td>
<td>$1.12 - 0.04(V_{mach} - 2.5)$</td>
</tr>
<tr>
<td>$3.5 &lt; V_{mach} \leq 7.0$</td>
<td>$1.08 - 0.114(V_{mach} - 3.5)$</td>
</tr>
<tr>
<td>$7.0 &lt; V_{mach}$</td>
<td>1.04</td>
</tr>
</tbody>
</table>

Table G.3: Cubed shaped fragments.