UNIVERSITY OF SOUTHAMPTON

FACULTY OF ENGINEERING, SCIENCE & MATHEMATICS

School of Electronics & Computer Science

Endgame Optimisation

By

Nimet Patel

Thesis for the degree of Master of Philosophy

December 2008

Abstract

Within the defence industry, there is the need to provide an improvement in the efficiency (performance) of a missile system. The present generation of missile systems are sub-optimal in many currently considered scenarios. Scenarios include both anti-air and ground attack domains and these have to allow for an increased usage of stealth, more effective countermeasures, and better mission survivability by making use of redundancy in subsystems. There are many methods by which this improvement in efficiency can be achieved

The traditional approach to improving the lethality of a missile has been to concentrate efforts in the guidance and control systems to improve accuracy and agility.

This thesis considers how optimizing the endgame, the final few milliseconds before detonation, can yield improvements in overall lethality. This is achieved using traditional optimisation techniques and has investigated possible missile warhead fusing strategies which may be used in order to provide robust, high lethality engagement conditions for an air-to-air missile system.

The development of various fusing strategies has been performed based on observations made during the undertaking of this research. This included development of fusing rules used for the missile warhead and the development of advanced fusing algorithms that look at past missile fly-out and lethality data to aid the decision process of when to fuse the missile.

Contents

Figures

Tables

Acknowledgements

I would like to thank DSTL for their financial and technical support throughout the research study.

I would also like to acknowledge the support of my supervisors, Dr Andy Chipperfield and Prof. Andy Keane for their guidance and encouragement throughout the time that I spent working them.

Acknowledgements are also given to my parents, and family for their unconditional support throughout not just this undertaking, but in everything that I have done, especially to my nephew and niece who always provided a welcome distraction when required.

A final acknowledgement is given to the friends that I made during my time in Southampton who would always be there me whenever I would ask for their support and would always be on hand to brighten my spirits and provide humour in almost any situation.

Nomenclature

P_k	%	Lethality Probability
P_{kMean}	%	Mean Lethality
P_{kLB}	%	Lethality Lower Bound
V_m	m/s	Missile Velocity
V_t	m/s	Target Velocity
V_R	m/s	Relative Velocity
η	°	Engagement Angle
ω	°	Target Roll Angle
δ	°	Missile Yaw Angle
ε	°	Missile Pitch Angle
ψ	°	Missile Roll Angle
x_0, y_0, z_0	m	Missile Aim Points
ϕ	°	Dartboard Angle
S_r	m	Miss Distance/Dartboard Radius
z	m	Position Along Trajectory of Fuse Point
a		Gaussian Amplitude
b		Gaussian Position Vector
C		Gaussian Shape and Orientation Matrix
R		Rotational Matrix

1. Introduction

Within the defence industry, there is the need to provide an improvement in the efficiency (performance) of a missile system. The present generation of missile systems are sub-optimal in many currently considered scenarios. Scenarios include both anti-air and ground attack domains and these have to allow for an increased usage of stealth, more effective countermeasures, and better mission survivability by making use of redundancy in subsystems. There are many methods by which this improvement in efficiency can be achieved, such as optimisation of individual missile sub-systems (guidance/navigation, fuse delay etc). The missile system is made up of several sub-systems as shown in Figure 1.1.

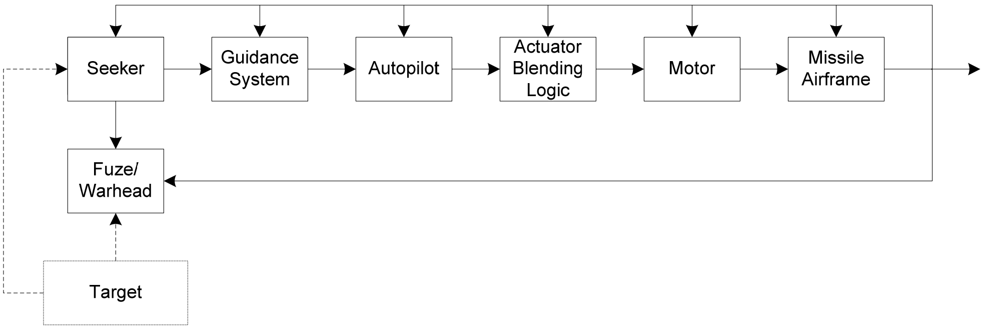


Figure 1.1: Missile Sub-Systems

A standard missile fly out [1] will involve an initial fly-out, and the endgame condition, or fusing stage. The initial fly-out is controlled using seeker, guidance and autopilot systems, in order to bring the missile close to the target. Once close enough to the target, control is passed over to the fusing system to determine when to trigger the fuse and detonate the warhead. This is achieved using a proximity fuse [2], which receiver a reflected transmission to determine the distance of the target to the missile to determine when to trigger the warhead, and is discussed in Section 6.

Traditional methods for optimisation of a missile system generally focus on the non lethal components of the missile. There has been ongoing research into the optimisation of these components, including the guidance system, autopilot system, motor, and missile airframe.

Improvements in missile lethality have been sought through improved guidance and control laws, for example, to optimize guidance for a specific control law such as the autopilot [3] and engagement conditions [4] or by solving receding horizon optimizations to achieve fast and realisable online target tracking [5]

The main aim of this research will be to investigate the possible methods for increasing the level of system performance of an advanced missile system, by improving the lethality probability based on the point at with the missile warhead fuses, rather than through improvements in guidance or missile control laws. This will be achieved by using optimisation techniques to improve the missile warhead aim points.

By optimizing the endgame geometry to achieve high levels of lethality probability, the missile fly-out endpoint is determined and a suitable guidance law can be developed using conventional approaches [6] or intelligent ones [7]

There are two software packages that have been made available for the research, MSTARS [8] and AGILE [9], which are described in more detail later in this thesis.

Munitions Simulation Tools and Resources (MSTARS, developed by DSTO, Australia) is a weapons modelling system for developing models and model libraries, and conducting simulations and analysis. The software is made up of Simulink model blocks for use in the Matlab environment. The objective of the tool is to perform weapons systems modelling, and includes within its libraries various models for missile launch-capable vehicles (such as aircraft and helicopters), and targets (both airborne and ground based). Many complex scenarios can be constructed using the package to simulate the fly-out conditions of varying engagement problems and provide a set of endgame parameters. These parameters can then be passed on to the second package, AGILE.

AGILE, (Analytic Gaussian Intersection for Lethality Engagement, Developed by QinetiQ, UK) is a lethality prediction tool that is designed to provide a value (probability) of engagement uncertainty, or ‘kill probability’. The AGILE software can be used stand-alone, or can be employed as a component embedded into another software package (MSTARS for example). AGILE incorporates many features including the prediction of damage inflicted upon a target (or component of a target) by warhead fragments. It also includes a simple blast damage model for close-burst conditions (i.e. where the warhead is in close proximity to the target). Uncertainties in the 'endgame geometry' (relative position, velocity and orientation) are represented, by using a set of Gaussian inputs, as are the uncertainties in target and warhead configuration. This final part of the missile-target engagement is assumed to be linear, for example all velocity vectors are static.

1.1. Chapter Overview

This thesis documents the research studies undertaken, and is split into the following sections:

2. AGILE and Engagement Modelling

This chapter will provide a basis for how lethality, P_k, can be evaluated. Lethality is a probability measurement of the likelihood that a target will be completely destroyed. This probability is used to base fusing decisions in simulations of fly-out scenarios, thus optimisation of this probability can yield improved performance of missile subsystems. A package, AGILE, which calculates this probability in a fast and efficient manner by using Gaussian function to define missile and target components, will be described. The coordinate system in which this process is performed will be defined, and the method by which the lethality values are calculated will be described and endgame entities will be specified and illustrated.

3. Lethality Optimisation

This chapter will discuss optimisation and the various types of optimisation methods. Sensitivity to disturbances of optimal parameters, known as robustness is then considered. Following this, a representative set of endgame scenarios will be optimised in order to assess if a maximum value of lethality lies close to the endgame parameter set, and a robustness measure of this maximum will be described and evaluated.

4. Modelling Fly-Out of Missile

This chapter will describe MSTARS, a missile fly-out simulation tool for Simulink. Its basic workings and how it can be used with AGILE will be described and some sample fly-out scenarios illustrated. Following this some fly-out scenarios will be analysed to see how lethality varies in the final stages of fly-out.

5. Endgame Optimisation Along Missile Trajectory

This chapter will examine the missile trajectory in more detail. Each point along the trajectory will be examined to see how the lethality varies along the trajectory prior to the fusing of the missile and whether the lethality value can be increased through optimisation. From these studies, potential fusing strategies will be assessed.

6. Fusing Methods for Missile Warhead

This Chapter will describe differing methods by which a decision on whether the missile should activate the trigger or not can be made. These will be split into three categories, simple decision processes, conditional decision processes, and knowledge based decision processes. Each will be described and a framework for how each will be implemented will be shown.

7. Analysis of Fusing Methods

Using the methods described in Chapter 6, a batch of 5,000 end game yielding scenarios will be evaluated, and the performance of each method compared. The strategies evaluated will include, minimum distance, lethality threshold level, fusing matrix with original and optimal lethality values, and advanced fusing algorithms.

8. Conclusions and Future Work

Conclusion, Contributions, and Future Work is discussed in this section.

A. Appendices

Supplementary work

2. AGILE and Engagement Modelling

This chapter will provide a basis for how lethality, P_k, can be evaluated. Lethality is a probability measurement of the likelihood that a target will be completely destroyed. This probability is used to base fusing decisions in simulations of fly-out scenarios, thus optimisation of this probability can yield improved performance of missile subsystems. A package, AGILE, which calculates this probability in a fast and efficient manner by using a Gaussian function to define missile and target components will be described. The coordinate system in which this process is performed will be defined, and the method by which the lethality values are calculated will be described and endgame entities will be specified and illustrated.

2.1. Engagement Geometry

The trajectories and orientations of the missile and target are collectively known as the endgame engagement geometry. Engagement geometry within AGILE is described using the GW372 coordinate system [10][11][12]. This coordinate system defines the relationship between two sets of Cartesian coordinates, using the relative data, such as position, orientation and velocity, between the target and missile to evaluate the lethality probability. Cartesian coordinate systems require more calculations to be applied to the parameters to enable AGILE to calculate the lethality value. The advantage of using the GW372 coordinate systems is that the key parameters can be varied independently to one another, reducing the computational load from performing conversions of standard Cartesian coordinate data each time an aspect of the endgame is varied.

2.1.1. GW372 Coordinates

The GW372 coordinate system, shown in Figure 2.1, defines the relationship between two Cartesian frames of reference, one for the missile and one for the target. These axes are different to the normal body axes that define *x* as longitudinal*,* *y* as latitude and *z* as vertical. In both frames of reference the *x*, *y* and *z* axes are usually defined as follows, in relation to the body of the vehicle:

- The *x* axis is to the left, for example in a fixed-wing aircraft along the port wing;
- The *y* axis is up (in level flight the direction of the pilot's torso);
- The *z* axis is ahead, along the centre line of the aircraft or missile body, i.e. in the direction of flight when there are zero incidences in pitch and yaw.

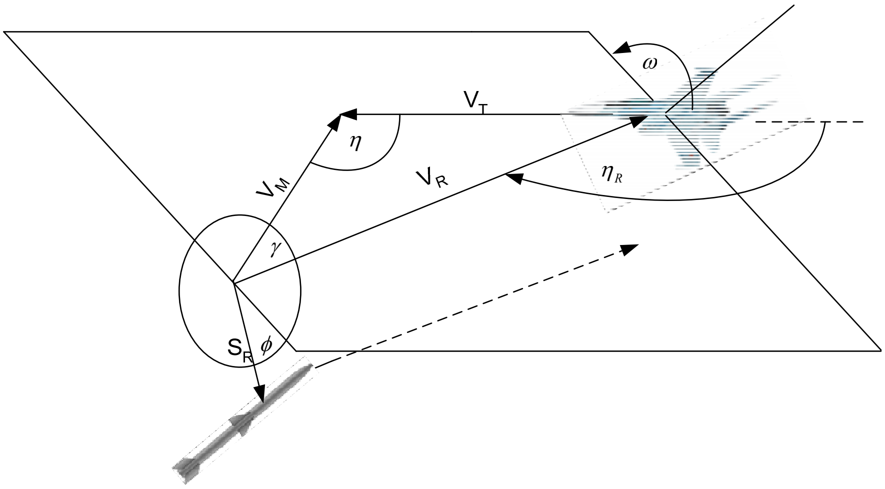


Figure 2.1: GW372 Coordinate System

GW372 coordinates only specify relative position, velocity and orientation; higher time derivatives (e.g. rotation rate and acceleration) are not specified because lethality is usually not sensitive to the latter. The lack of sensitivity of lethality to acceleration and rotation rate is due to the very short periods of time, typically less than a millisecond, involved in the damage mechanisms, such that the error in the position of fragment collision with the target due to a 10g target acceleration is approximately 0.05 mm, assuming that the fragment takes 1 ms to reach the target, so is in effect, negligible.

The relationships between the target and missile frames can be defined by two affine transformations, one for relative position and orientation, and the other for relative velocity. An affine transformation is a combination of a translation (shift) and a linear coordinate transformation. In this case the linear transformation is a pure rotation. The GW372 coordinate system describes these transformations in a way convenient for modelling the endgame.

AGILE is a software package that calculates a level of lethality for a particular endgame scenario. AGILE uses this GW372 coordinate system as its input and calculates a percentage value corresponding to the lethality. The main parameters of an engagement described in GW372 are summarised below.

- V_m and V_t are the speeds of the missile and target respectively in *meters per second*;

- η is the engagement angle in *degrees*: that subtended between the missile and target velocity vectors. $\eta = 0°$ implies a tail chase, whereas $\eta = 180°$ implies a head-on engagement;

- ω is the target roll, in *degrees*;

- δ, ε define missile yaw and pitch respectively, and ψ is missile roll, in *degrees*;

- x_0, y_0, z_0 define a missile aim point in the target's frame of reference. This point is used to define the burst (warhead detonation) point; it is the origin of a cylindrical polar coordinate system whose z axis is aligned with the missile velocity, the aim points define a vector, and each component is expressed in *meters*;

- ϕ, S_r, z are the above mentioned cylindrical polar coordinates used to define the burst points, (i.e. the point along the trajectory that the warhead is detonated). ϕ is known as the 'dartboard' angle expressed in *degrees* and S_r is known as the 'dartboard' radius, expressed in *meters*, and defines a polar coordinate away from the perpendicular of the trajectory of the missile. z specifies the position along the trajectory of the burst point, and is expressed in *meters*. The 'dartboard' is a graphical display produced by AGILE that displays the variances in lethality from the missile detonating at differing points to the actual calculated lethality from the original input parameters. An example of the dartboard is given in Figure 2.2.

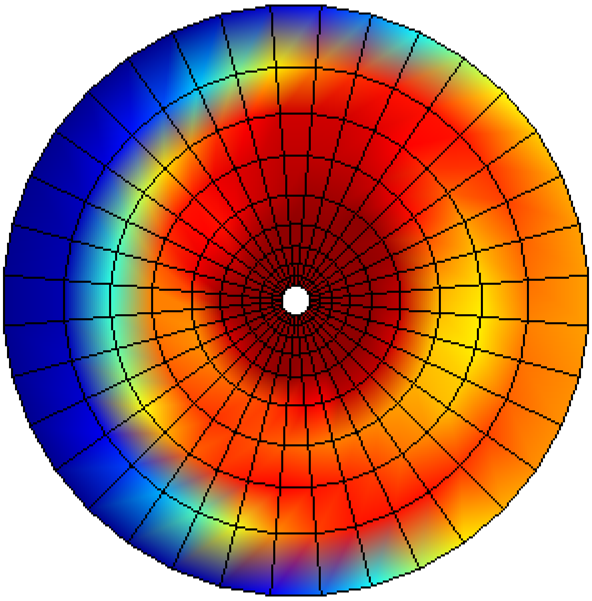


Figure 2.2: Example Dartboard

The dartboard's centre shows the lethality at the point of detonation. The dartboard shows the variance of lethality of points around the trajectory of the missile, i.e. the trajectory is perpendicular to the dartboard, with the dartboard showing lethality as the missile is translated to the various positions. It can be seen from the figure that lethality varies as the position of the missile is translated away from the original detonation point.

In reality the missile and target both move along their respective velocity vectors; however it is easier to think of the target as stationary with the missile moving along a vector V_R towards it. It is usually assumed that as the missile approaches the target along V_R all the other parameters remain constant (no manoeuvres take place). This assumption is justified because all the fusing and lethality events take place over a few milliseconds and thus within a very short distance along the trajectory length. The GW372 system has the advantage that the primary parameters can be changed independently of each other, and each has a clear physical meaning.

2.2 Agile Overview

AGILE (Analytical Gaussian Intersection for Lethality Engagement) is a computer lethality prediction tool that is designed primarily to provide fast representation of engagement uncertainties. The computational speed of AGILE gives it suitability to be used as a design tool (either for the weapon or to increase the survivability of the target), or as a component embedded in any computer simulation that would require a fast estimation of endgame lethality.

AGILE has the following features:

- Prediction of the damage inflicted by the fragments of a warhead on a target or a target component;
    - Fragment damage is the damage that occurs from the pieces of the warhead that scatter upon the fusing of the explosive material within the warhead. These fragments travel outwards and hit the target, causing the damage to the target. The amount of damage is related to the velocity and mass of the fragment.
    - This model is employed on research undertaken and a description follows, however, for completeness the following features are mentioned.
- A simple close-burst model that incorporates the effects of blast;
    - When a missile detonates there is an associated blast wave that propagates outwards. This wave can cause damage if it reaches the target before the dissipation of the energy in the wave. This component is not used in this study.
- A simple direct impact model;
    - Direct impact occurs when the missile hits the target. Within AGILE if the missile hits the target an automatic lethality value of 1 is assigned. This model is not used in this study.
- Representation of uncertainty in the endgame geometry (relative position, velocity and orientation), and in the configuration of the target and warhead.
    - As with all systems, there can be uncertainties resulting from sensor error and other noisy factors. AGILE can incorporate uncertainties from noise into the lethality calculation process.

AGILE can provide fast computations of lethality predictions due to the method by which endgame entities, namely the target and missile warhead, are modelled. They are represented using Gaussian functions which can be operated on quickly and easily to find intersections between them. The speed of AGILE can allow a solution to be formulated that can potentially be used to perform on-board calculations in an endgame scenario, and also incorporates uncertainty in the endgame parameters in its calculations.

2.2.1. Gaussian Functions

Gaussian functions have been used as they provide an efficient and fast method of defining missile and target entity components, and also allow for fast manipulation of the defined Gaussian functions in order to calculate points of intersection to analyse damage probabilities.

A 3-Dimensional Gaussian function, *f*, has the following form:

$$f(x) = a\exp\left(-\frac{1}{2}(x-b)^T C^{-1}(x-b)\right) \tag{2.1}$$

where *x* is a spatial position vector (with three Cartesian components), *a* is the maximum value of *f*, *b* is the position vector where *f* is maximal and *C* is a 3x3 positive-definite symmetric matrix representing the shape and orientation of level sets (surface contours) of *f*. The level sets of a Gaussian are ellipsoids, so the Gaussian itself can be thought of as a fuzzy ellipsoid; the value of *f* decays smoothly from *a* to zero as the distance from the centre *b* of the ellipsoids increases. Visualisation of a 3D function is difficult; it can be best thought of as an elliptical cloud whose density decays smoothly with distance from centre. Visualisation of a 2D Gaussian is easier, an example of such a 2D Gaussian is shown in Figure 2.3.

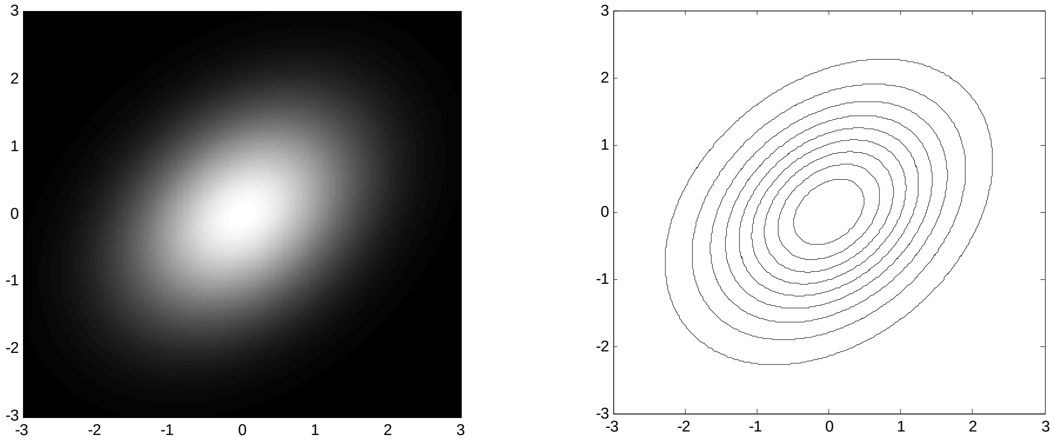


Figure 2.3: Intensity Plot and Contours of a 2D Gaussian.

The following objects are represented by sums of Gaussian functions in AGILE:

- Target vulnerability to warhead fragment damage.
  Regions of high vulnerability are close to the centres of one or more Gaussians, whilst regions of low or zero vulnerability are typically further away from the centres;

- Warhead fragment cluster density.
  This is not the density or mass of individual fragments, but the average number of fragments per unit volume, or ‘population density’. Where the target vulnerability and warhead fragment densities are both high, the level of damage, i.e. probability of target kill or component failure, will be high;

- Close-burst lethality and warhead damage blast damage.
  A set of ellipsoids and cylinders used to describe a neighbourhood of the target for which a ‘kill’ is certain. This region is the set of all points inside one or more of these Gaussian objects; the warhead blast damage are derived from level sets (contour surfaces) of Gaussian functions;

- Target shape is used by both the fusing and impact models.
  In the fusing model Gaussians are used to define the external shape of the target and its reflectivity to the radiation used by the fusing sensor. In the

impact model Gaussians are used to define the shape of both the missile and the target, so that the severity of a collision can be calculated;

- Radiation pattern of the fusing sensor.
  This information is used in conjunction with the shape and reflectivity of the target to predict the moment when the fuse is triggered.

Gaussian components are used for the following reasons:

- Their intersections can be computed very efficiently using an analytical approach;

- Uncertainty in the endgame geometry can be represented directly by Gaussian components, reducing or avoiding the need for Monte-Carlo methods [13] as sampling around the burst point is not required.

The reason for AGILE's speed is its ability to represent many warhead fragments simultaneously as a single entity; instead of computing each individual intersection of fragment and target, a single calculation can be applied to hundreds of fragments as an ensemble.

2.2.2. Gaussian Mixtures

Much of the input data specified for AGILE is in the form of Gaussian mixtures, a set of Gaussian functions each defined previously in Equation 2.1. The following sections describe how these mixtures are defined and what they will represent.

Each Gaussian function has three components:

- The amplitude, *a*, that is the peak value of the function. This is a single 'double-precision' scalar;

- The mean value, *b*, that is the 3-element position vector at which the peak value of the function occurs;

- The covariance matrix, *C*, which defines the size and shape of the function (and its contour surfaces). The covariance matrix is a 3x3 square matrix.

2.2.2.1. Visualisation

3D Gaussian functions are easiest to visualise by showing their contour surfaces, which are nested ellipsoids. Figure 2.4 shows a small number of contour surfaces, which are nested ellipsoids, in cross-section.

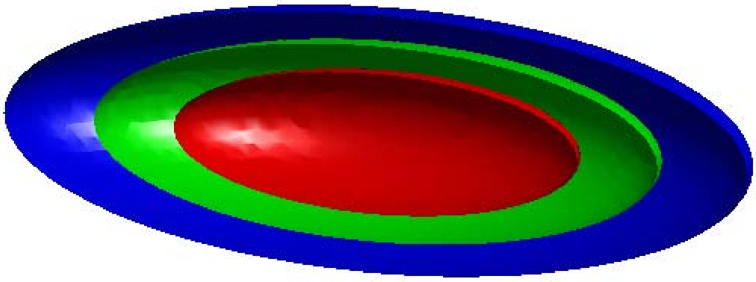


Figure 2.4: Cut Away Contour Surface of 3D Gaussians.

When viewing many Gaussians at the same time, for example when viewing the target, only a single contour surface is shown to reduce complexity. Figure 2.4 shows some examples of these.

2.2.2.2. Amplitude and Position

The amplitude, *a,* does not affect the contour surfaces except for their functional values, such that if the value of *a* is increased the value of each contour will increase, or equivalently the contour surfaces at the same value will dilate. This means that the mean value, *b,* is at the common centre of all the contour surfaces. Therefore changing *b* will result in a translation of all the contours by the same amount and direction. Figure 2.5 shows three Gaussians with the same *a* and *C*, but differing *b* values. In each case the contour surface with the same value is shown. Note that the contour surfaces are shown for each Gaussian separately, not the contour surfaces of all three added together. If the centres, *b,* are sufficiently similar, the contour surfaces can overlap, as shown by the red surface and blue surfaces in Figure 2.5.

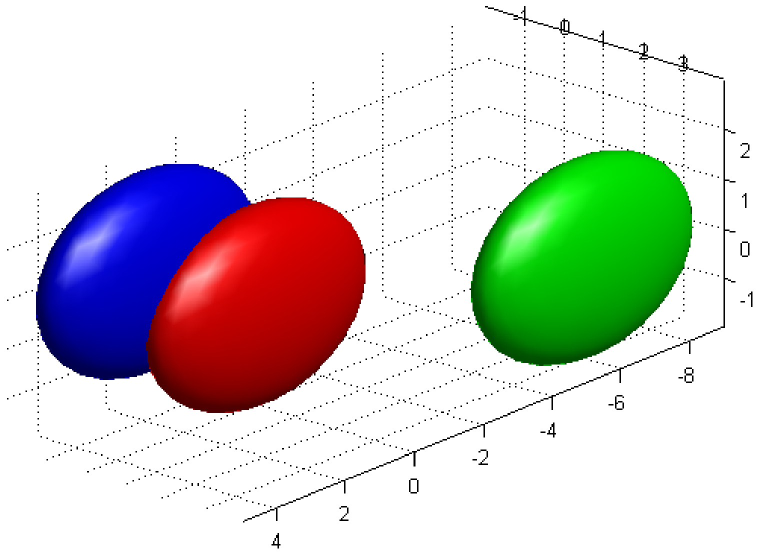


Figure 2.5: Three Gaussian contour surfaces with same *a, C* but different *b*.

2.2.2.3. Size, Shape and Orientation

The covariance matrix *C* affects the size, shape and orientation of the ellipsoid's contour surfaces. Multiplying all 9 coefficients of *C* by the same scalar causes each contour surface to dilate by the square root of the scalar value. The covariance matrix affects the shape and orientation of the contour surfaces in a more complex manner. There are two ways of describing this relationship:

- The three principal axes (directions of locally extreme curvatures) of the contour surfaces are given by the eigenvectors of the covariance matrix, and the lengths of these ellipsoids in each of these directions are proportional to the square roots of the corresponding eigenvalues. The constant of proportionality depends on the value of the contour surface chosen. In cases where two or more eigenvalues are identical the ellipsoids are circular in at least one cross-section, and when all three eigenvalues are identical the contour surfaces will be spherical. In such cases the eigenvectors are not unique, as the principal axes are not well defined. This is one reason why the covariance matrix describes the shape and orientation of the Gaussian, as it is always unique.

- Another method of describing this is more intuitive and is suitable for constructing a covariance matrix for a Gaussian of a given shape, size and orientation. The first step is to construct an ellipsoid whose principal axes are parallel to the coordinate axes *x*, *y*, and *z*; this is a diagonal matrix whose values are proportional to the squares of the diameter of the ellipsoid in the *x*, *y*, and *z* directions respectively. This step defines the size and shape of the ellipsoid, the next stage is to define the orientation. This is done by rotating the already constructed ellipsoid (parallel to *x*, *y*, and *z*) to the desired orientation. In general this involves defining a matrix of rotation, *R*, from which the required covariance matrix of the tilted Gaussian is:

$$C = RC_0R^T \tag{2.2}$$

  where C_0 is the original old diagonal matrix and *C* is the required covariance of the tilted Gaussian.

There are various ways of constructing this rotation matrix *R*. One approach is to define three consecutive rotations about each of the coordinate axes, R_x, R_y, R_z. A rotation through an angle θ about the *x, y* and *z* axis is given by:

$$R_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{pmatrix} \tag{2.3a}$$

$$R_y = \begin{pmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{pmatrix} \tag{2.3b}$$

$$R_z = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.3c}$$

The composite rotation R is given by the matrix product $R = R_x R_y R_z$ where the rotation about x is applied first and that about z is applied last.

In most practical cases it is sufficient to orientate the ellipsoid with the principal axes parallel to the coordinate axes, so the latter step of applying a rotation is not required. The diagonal elements of C are called variances and their square roots are the standard deviations. The standard deviations are proportional to the corresponding widths of the ellipsoids when C is a diagonal matrix.

Figure 2.6 shows examples of ellipsoids with different size, shapes and orientations. The red ellipsoid has semi-major axes of lengths 1, 2 and 3 in the x, y and z directions respectively. The green ellipsoid has corresponding lengths 1, 1, 2, and hence is circular in the xy plane. The blue ellipsoid has the same dimensions as the red but has been rotated in all three directions.

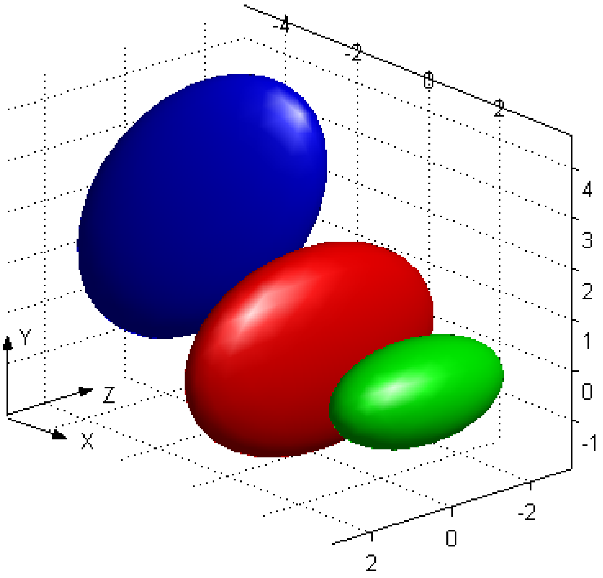


Figure 2.6: Ellipsoids with Varying Size, Shape and Orientation.

2.2.2.4. Spherical Polar Coordinates

Gaussian functions can be represented in any coordinate system, not just Cartesian coordinates. The suitability of the coordinate system to be used depends on the type of object being defined. Targets are most naturally represented in Cartesian coordinates, but objects that have natural symmetry in a radial direction or with respect to a set of rotations are sometimes better represented in spherical polar coordinates (SPC). Warhead fragments and active sensor beams both have radial symmetry and often rotational symmetry too. Therefore, these objects are represented by Gaussians in SPC.

SPC are composed of three coordinates:

- Azimuth, θ. This is the angle measured anti-clockwise between the *x*-axis and the vector (*x, y*), where (*x, y, z*) are the Cartesian coordinates.
- Elevation, ϕ. This is the angle between the *xy* plane and the vector (*x, y, z*), where the sign of ϕ is the sign of *z*.
- Radius, *r*. This is the Euclidian length of the vector (*x, y, z*), $r = \sqrt{x^2 + y^2 + z^2}$. In AGILE the natural logarithm or *r* is used as explained below.

The logarithm of radius is used as the third SPC coordinate in AGILE because it is unbounded, whereas the conventional radius is bounded below zero. The use of the natural logarithm prevents a Gaussian in SPC from violating this bound, i.e. assuming negative values of (conventional) radius. Shifts in log radius (the third element of the Gaussian mean *b*) correspond to dilations in the corresponding Cartesian coordinates.

Some examples of various Gaussians in SPC, showing their shapes (surface contours) in Cartesian coordinates are now described. Figure 2.7 shows three narrow jets with narrow distributions in azimuth and elevation, but a relatively wide distribution in radius. The red and green jets have zero mean elevation so are in the *xy* plane. The red jet also has zero mean azimuth, and so points along the *x* axis and has small equal standard deviations in azimuth and elevation, and hence is narrow and circular in section. The blue jet has 3π/4 azimuth, so points along the line with direction vector

(-1, 0, 1). It has a mean log radius of 1, whereas the red and green jets have a mean log radius of 0, which means the blue jet has been dilated by a factor of *e* in the radial direction, which is why it is longer.

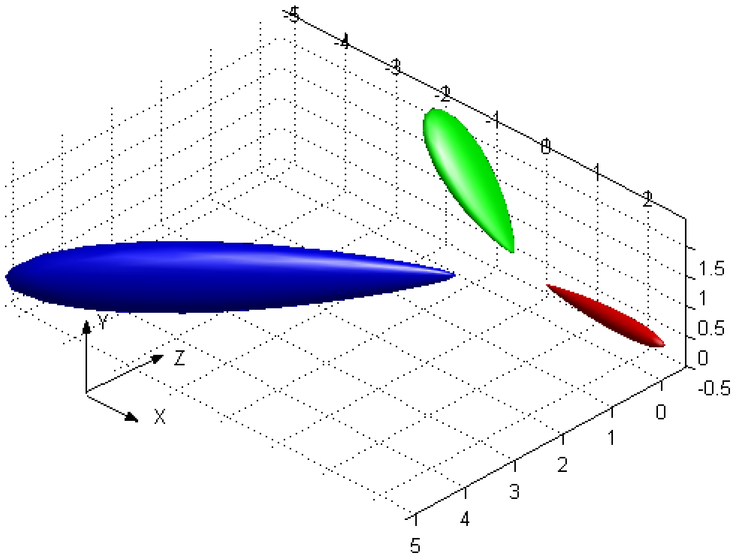


Figure 2.7: Narrow Jets Described in SPC

Figure 2.8 shows two Gaussians with a small standard deviation in elevation, but much wider deviations in azimuth. The wider deviation in azimuth results in the ‘stretching’ of the jet shape out across the *xy* plane in a circular arc from a fixed centre. Therefore these shapes are similar to conical shapes than ellipsoids because the Gaussians are defined in SPC, not Cartesian coordinates.

In both cases the mean elevation is approximately half the angle at the base of the cone; the mean elevations are π/6 and π/3 for the red and green objects respectively. The red Gaussian contour surface has an distribution in azimuthal range of (0. π), or 180°, and as such the contour displayed does not wrap all the way round in a loop, whereas the green surface has a very large azimuthal range of (-1000, 1000). In the green object’s case the azimuthal range far exceeds the 2π that corresponds to a full circle, so the Gaussian distributions in azimuth is almost exactly a uniform distribution in the range (-π, π), or 360°; this is why in Cartesian coordinates this Gaussian is almost a perfect cone whose thickness is the range in elevation.

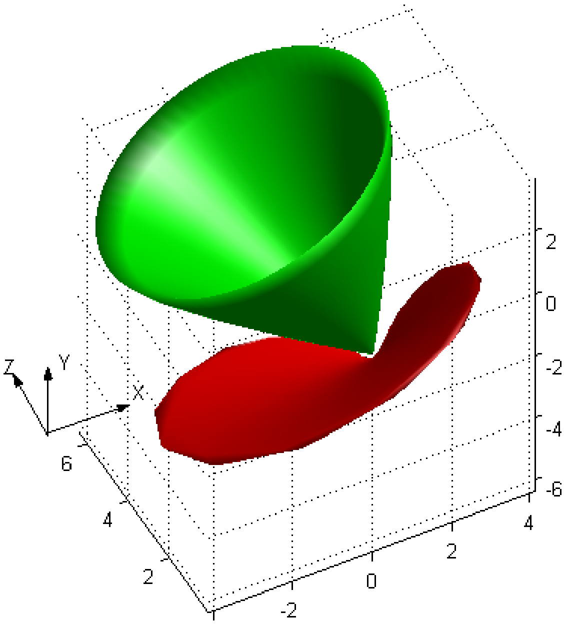


Figure 2.8: Gaussians Defined using SPC with Large Standard Deviation in Azimuth.

Therefore, this is a very convenient configuration method to enable the description of a radial-symmetric fragment distribution emanating from a warhead cone.

2.2.3. Mathematical Operations on Gaussians

The main reasons why Gaussian mixtures are used to describe the various objects involved in lethality prediction are because the following operations on Gaussians can be computed analytically and efficiently:

- **Point-wise multiplication**. Here we define the product of two functions *h* = *fg* as the function whose value at each point *x* is *h*(x) = *f*(x)*g*(x); this is known as point-wise multiplication. The product of two Gaussians is also a Gaussian and its amplitude, mean and covariance can be calculated analytically by completing the square of a quadratic function. The product of a Gaussian in Cartesian coordinates and another in SPC is not exactly a Gaussian in either system, but an approximation can be computed using a numerical approximation technique that is almost as fast as the calculation of the product in a common coordinate system.

- **Integration**. A Gaussian can be integrated analytically in one, two or three dimensions provided the domain of this integral is unbounded, i.e. the whole real line, real plane or 3D space. The integral is given by $a\sqrt{\pi^n \det(C)}$ where *n* is the number of dimensions being integrated.

- **Affine transformation**, i.e. the combination of translation and any linear transformation. Affine transformations include any combination of translation, rotation, dilation and skew transformations.

- **Convolution**. A convolution is an integral that expresses the amount of overlap of one function *g,* as it is shifted over another function *f.* It therefore "blends" one function with another. Convolution of two functions of real numbers can be defined as *h(x) =f*g* = $\int_{-\infty}^{\infty} f(x)g(\tau - x)dx$.

  Two Gaussians can be convolved, simply by adding their mean's and covariance's respectively, where each addition is weighted by the Gaussian amplitude, thus the resultant convolution is also a Gaussian.

- **Maximal projection**. This is defined to be the maximum value of a Gaussian with respect to one or more of its coordinates (independent variables). This operation reduces the dimension of the Gaussian. For example, maximal projection in the third coordinate (*z*) reduces a 3D Gaussian (a function of *x,y,z*) to a 2D Gaussian (a function of *x,y*): $f_{\max}(x, y) \equiv \max_{z}\{f(x, y, z)\}$. Projection is implemented by deleting the appropriate rows and columns from the mean vector *b* and covariance matrix *C*. For example for the maximal projection in *z* the 3rd row and column is deleted.

An overview of how these operations are applied in lethality prediction is given next.

2.2.3.1. Geometric Intersection

This is the operation to find a region of intersection between two geometric objects, and is implemented by point-wise multiplication. The most important example is finding where warhead fragments reach target components. Here the fragment cluster density is multiplied by the target's fragment vulnerability, and the product is a vulnerability probability density. The integral of this is related to the component's damage probability. Target component damage is most likely to occur only in regions where both the fragment density and target vulnerability are high.

Another example is the reflection of radiation from a fusing sensor off the target's surface. The strength of reflected radiation at each target point is proportional to the radiation power density multiplied by the reflectivity of the target. These functions are represented as a Gaussian in AGILE. The total power of the radiation reflected is proportional to the integral of the above product, assuming the radiation is incoherent. A third example is the damage from the collision of the missile and the target, a direct hit. This is calculated as the total kinetic energy of the missile that intersects with each target component, weighted by a target component vulnerability function. This is proportional to an area integral, in the plane perpendicular to the direction of the relative velocity vector, or the product of the missile mass density and the target vulnerability.

2.2.3.2. Frames of Reference

When calculating Gaussian intersections between missile and target components it is necessary to take account of the relative position, velocity and orientation of these systems. All the components of the missile and target respectively are specified in separate coordinate frames so that different endgame geometry can be incorporated efficiently. Affine transformations are applied to both the target and missile components in order to work with a common frame of reference when calculating Gaussian intersections.

2.2.3.3. Uncertainty

Uncertainties in either the endgame trajectory or target/missile configurations can occur for any given scenario. Examples of uncertainty are:

- Fuse timing, caused by errors in seeker measurements;

- Relative position and velocity, either because of seeker measurement, target evasive manoeuvres or air turbulence.

- Relative orientation, e.g. because of low seeker imaging resolution.

- Target configuration, either because of target identification error, or because future target concepts are being assessed that are inherently uncertain.

- Warhead fragment distribution, e.g. because of manufacturing tolerance, or limited knowledge of the explosive dynamics.

A general treatment of uncertainty typically involves Monte-Carlo simulation, but where uncertainty can be approximated by a Gaussian mixture, convolution of the target or missile components by Gaussians is appropriate, as this will allow the uncertainty to be modelled within AGILE, and thus reducing the number of computations required if Monte Carlo methods are used. For example, if the relative position of the target and warhead at the moment of warhead detonation has a Gaussian probability distribution, the target Gaussians can be convolved by this Gaussian to account for this uncertainty.

2.3. Target Lethality Modelling in AGILE

There are a number of damage models that can be employed by AGILE in lethality computation. These include the lethality caused by fragment damage; lethality caused by the blast wave of the warhead, lethality caused by a close burst of the warhead, and a blast damage model. Also included is a simple fusing model to determine when to detonate the warhead for given endgame trajectory. These can be used independently or different damage models can be activated for use simultaneously. In this study only the fragment damage model is considered.

2.3.1. Fragment Damage Model

2.3.1.1. Background

The level of damage to a very small region of a target is assumed to be proportional to the mean number of fragments that intersect that region. This is similarly assumed to be proportional to the solid angle that the region subtends with the centre of the warhead blast. Therefore the greater the distance between the target and warhead the smaller this angle will be, and hence the smaller of probability of fragment damage.

Each target component within the fragment damage model is partitioned into regions of infinitesimal solid angle to derive a Poisson stochastic process. The probability of each region of the target being damaged is proportional to a known probability density multiplied by the solid angle subtended by this region. If *h* is this probability density, the probability of component failure is modelled as:

$$P = 1 - \exp\left(-\iint_S h(\theta,\phi) d\Omega \right) = 1 - \exp\left(-\int_{-\pi/2}^{\pi/2} h(\theta,\phi)\cos(\phi) d\theta d\phi \right) \qquad (2.4)$$

where *S* is the unit sphere, *dΩ* is the solid angle element, and the RHS is an alternative form of the integral in spherical polar coordinates.

The probability density, *h*, is calculated from the product of the following functions:

- The conditional probability, *f*, of target damage from a single fragment;

- The fragment cluster density, *g*, i.e. the average number of fragments per unit solid angle.

Since a target has three-dimensional geometry, the dependence of target damage on the radial dimension (distance travelled by a fragment) needs to be considered as well as the direction the fragment travelled which is characterised by θ and ϕ. Also requiring consideration is how much damage is dependant on the depth of fragment penetration through the target.

One method for this would be to assume that *h* is proportional to the depth of penetration. In such a case *h* could be given using the following integral:

$$h(\theta,\phi)=\int_{-\infty}^{\infty} f(\theta,\phi,r)g(\theta,\phi,r)\exp(r)dr \tag{2.5}$$

where *r* is *log* radius (see Section 2.2.4) and *g* could incorporate an optimal radial dependence such as allowing for the decay of target vulnerability with distance travelled, for example, due to loss of kinetic energy. The function *f* is now the conditional probability density of target damage with respect to log radius. The exponential term is the Jacobian (first derivative) resulting from a change of variable from radius to log radius, and thus would not exist in conventional SPC.

An advantage of this model is that the integral in Equation 2.4 becomes a 3D volume integral of the product *fg*, and this integral would be easy to calculate analytically in AGILE using the operations of point-wise multiplication and 3D integration.

2.3.1.2. Implementation

To calculate the component damage probability, *P* in Equation 2.4, the functions *f* and *g* are represented as Gaussian mixtures where *f* describes the target's conditional fragment damage probability and *g* describes fragment cluster density (fragments per Steradian). *f* is in Cartesian coordinates and *g* is in SPC. Each permutation of Gaussians in *f* and *g* is then point-wise multiplied and the product is another Gaussian. The maximum *h* of the product with respect to log radius is then derived as a projection in the radial direction. *h* is a 2D Gaussian in the coordinates *θ*, *ϕ*. *h* can be integrated analytically, but the product *h*cos(*ϕ*) cannot. Consider the following analytical approximation to Equation 2.4:

$$P=1-\exp\left(-\cos(\phi_0)\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(\theta,\phi)J(\phi-\phi_0)d\theta d\phi\right) \tag{2.6}$$

where ϕ_0 is the mean value of elevation angle at the Gaussian function *h* (the second element of the mean vector *b*) and *J* is given by:

$$J(\phi) = \exp\left(-\tfrac{1}{2}\phi^2\right) \tag{2.7}$$

Because both *J* and *h* are Gaussians, Equation 2.6 can be evaluated analytically. This approximation is given by the following identity:

$$h(\theta,\phi)\cos(\phi) = h(\theta,\phi)\left(\cos(\phi_0)\cos(\phi-\phi_0) - \sin(\phi_0)\sin(\phi-\phi_0)\right) \tag{2.8}$$

When the right-hand-side is integrated, the sin terms integrate to zero because *h* is symmetric about ϕ_0 and the sine function is anti-symmetric. The Gaussian function *J* is an approximation to the cosine function that has the same third-order Taylor expansion, so the error in Equation 2.6 is only fourth-order in ϕ, so is only significant if the range in elevation in *h* is very large, which only happens when the target component is very close to the burst point of the warhead compared to its diameter. In such cases the target is likely to be defeated either by the blast wave, or by direct impact. For example, if the standard deviation of ϕ is half a radian (which means the burst point is effectively at most a diameter from the target component), the relative error in Equation 2.6 is at most 1.25%. Even when the standard deviation of ϕ is a full radian (so the burst point is virtually touching the target component), the relative error in Equation 2.6 is at most 14%

2.3.1.3. Fragment Velocities

The fragment cluster density, *g*, in Section 2.3.1 described the spatial distribution of fragment cluster density in the target's frame of reference over all times, but did not explicitly define the fragment velocities. This additional temporal information of velocity is not required if calculations are carried out in the target's frame of reference.

However, in order to study the impact on lethality of changes to the missile's velocity, it is better to specify the fragments in the missile's frame of reference. In order to transform *g* from the missile frame to the target frame information about the fragment velocity distribution is required.

The most general way of specifying the fragment distribution would be a density function in space-time: a 4D function that would require 4D Gaussians to describe it. A simpler approach is adopted in AGILE, which is to specify the spatial distribution of the fragment velocity vector. This is a 3D distribution, and therefore involves 3D Gaussians. It is assumed that all fragments originate from a single burst point and each travels with constant speed. This purely spatial 3D fragment distribution will therefore dilate uniformly with time.

The fragment velocity distribution is thus specified in the missile's frame of reference. To convert this to the spatial fragment cluster density *g* the following operations are performed:

- The distribution is converted to Cartesian coordinates and the relative velocity of the missile and target is added to the mean velocity vector.
- Uncertainty in fragment velocity is incorporated by convolution (Section 2.2.3.3).
- The new velocity distribution in the target's frame is converted back to SPC.
- Uncertainty in orientation is incorporated by convolution.
- The velocity distribution is converted into the spatial cluster density *g* by setting the variance in log radius to infinity (which in practice uses a very large value). This effectively convolves the fragment distribution over all times after the time of detonation.

As an example of how the above transformations work, consider fragments distributed uniformly in a thin disc, with a small standard deviation in elevation (Figure 2.9, shown in semi-transparent green). The true cluster density *g* is unbounded in radius, as the fragments do not decelerate, but to aid visualisation, the radial coordinate is bounded in this illustration. The velocity distribution in missile coordinates is shown in red. The standard deviation in fragment speed is small, so the distribution is rather disc-like. This distribution is equivalent to the spatial cluster density *at a fixed time*, 1 second after detonation.

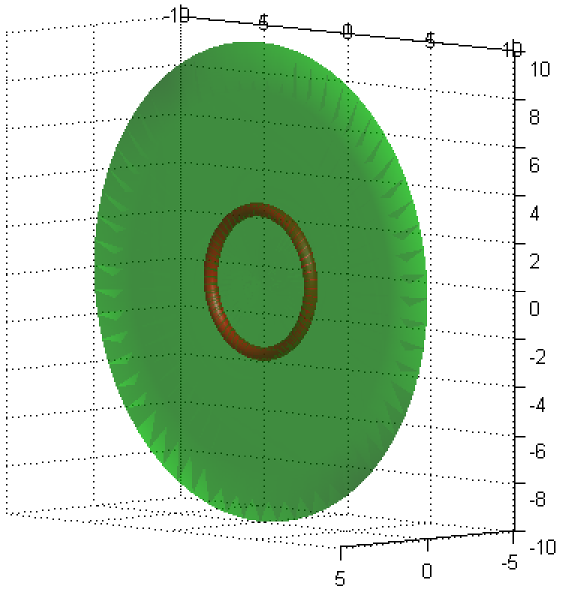


Figure 2.9: Fragment Spatial and Velocity Distribution in missile Coordinates.

Figure 2.10 shows the corresponding spatial and velocity distributions in target coordinates. The disc has the same shape but has been shifted to incorporate the motion of the missile relative to the target. The effect of the relative motion is to transform the fragment disc into a cone.

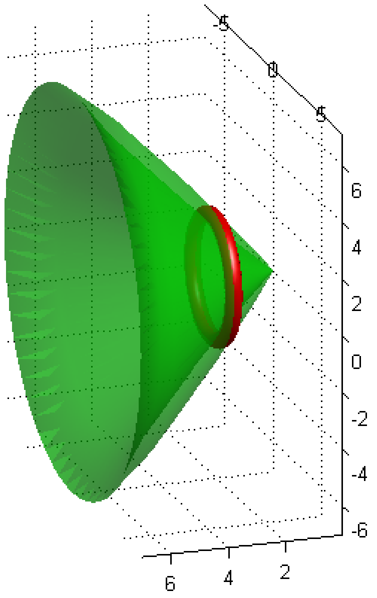


Figure 2.10: Spatial and Velocity Distribution in target Coordinates.

2.4. Target and Warhead Definitions

Running the fragment damage model requires the target and warhead's probability densities to be defined using Gaussian mixtures in Cartesian and SPC coordinates respectively. These are defined in text files that are read in at the start of an endgame lethality computation. Each component is defined using an amplitude, mean, and covariance matrix as described in Section 2.2.2. The amplitude term is prefixed with the label 'amp', and the mean is prefixed with 'mean'. For a full covariance matrix six terms are required, prefixed using the term 'covar'. Similarly, for covariance matrices that have off diagonal terms that are zero, only three terms are required and the prefix 'stdev' is used.

2.4.1. Warhead Definition

The peak fragment density is 1000 fragments per Steradian. A uniform density in azimuth is defined by using a very large standard deviation (1000 Steradians). The value of the corresponding mean is therefore unimportant in this case (hence it's zero value). The disc is thin, so the standard deviation in elevation is small (0.01 radians). The mean elevation of zero radians shows that the disc is perpendicular to the *z* axis, i.e. that fragments are flying outwards from the missile in a radial pattern. If this mean elevation value is non-zero in either direction the fragments would have a conical distribution as shown previously in Figure 2.9.

The radial coordinate represents the length of the vector, i.e. the speed of the fragments. This third coordinate is the log radius, so a value of 7.6 is the mean of the logarithm of fragment speed. In this case the actual fragment speed is almost 2000m/s. The standard deviation of log speed of 0.1 corresponds to an average variation in speed of 10%, so the standard deviation in fragment speed is approximately 2000x0.1 = 200m/s.

The warhead fragment velocity density is defined in a text file using the label ‘`fragment_density`’ and is of the form shown in Table 2.1:

```
% Warhead fragment disc
fragment_density
amp            1000
mean           0  0  7.6
stdev          1000  0.01  0.1
```

Table 2.1: Warhead Fragment Definition

2.4.2. Target Aircraft

Only one target is defined in this section, the simple fixed-wing aircraft. Other target craft, an airliner size craft and a helicopter are described in Appendix A.

The component definitions of the Gaussians required for defining a simple fixed-wing aircraft for are shown below in Table 2.2 Each component is labelled using ‘`fragment_damage`’ followed by an integer as these values will be used by the AGILE fragment damage model. Following this label are the values defining the Gaussian. The order of components is not important.

% **Cockpit**: highly vulnerable fragment_damage 1 amp 0.1 mean 0 1 3 stdev 0.35 0.53 0.71	% **Engine**: medium vulnerability fragment_damage 3 amp 0.03 mean 0 -1 -4 stdev 0.53 0.53 1.06
% **Fuselage**: low vulnerability fragment_damage 2 amp 0.01 mean 0 0 0 stdev 0.71 0.71 2.47	% **Wings**: low vulnerability fragment_damage 4 amp 0.01 mean 0 0 0 stdev 2.47 0.088 0.71

Table 2.2: Simple Fixed Wing Target Definition

Figure 2.11 shows the target defined above using contour surfaces at $\sqrt{8}$ standard deviations. This measure of $\sqrt{8}$ standard deviations enables one to view the surface containing 87.5% of the defined 3D Gaussian function to enable a visualisation of how the target is defined. The colour coding is as follows: blue for the cockpit; green for the fuselage; red for the engine; and cyan for the wings. The position of components in this case are relative to the fuselage ('mean 0 0 0'), therefore the cockpit is defined along the centre (x axis), 1 metre above (y coordinate) and 3 meters ahead (z coordinate, 'mean 0 1 3').

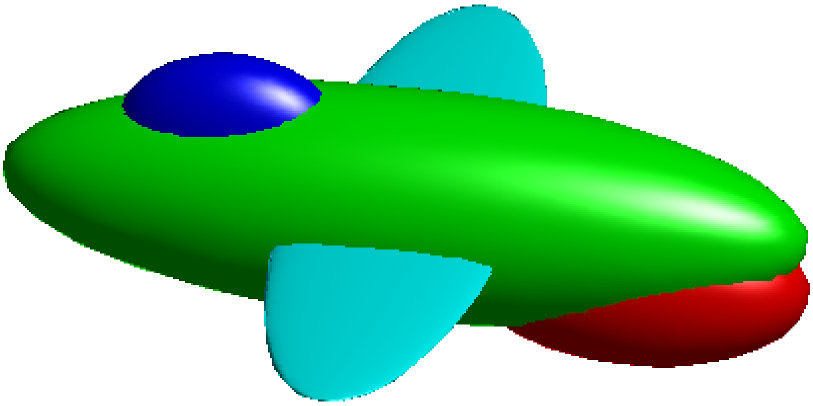


Figure 2.11: Simple Fixed Wing Aircraft Model

2.5. AGILE Examples

This section will cover some examples of lethality calculated using AGILE for a set of basic endgame scenarios.

2.5.1. Simple Example

For a simple endgame example the following parameter values, shown in Table 2.3, were used to exercise AGILE.

Parameter	Value	Units
V_M	1000	m/s
V_T	500	m/s
η	30	°
S_R	15	m
ω	100	°

δ	15	°
ε	35	°

Table 2.3: Example Endgame Parameters

All other parameters were set to zero, so there was no missile aiming, only the direction of travel, and no fuse delay. The missile is travelling at 500 m/s at and angle of 30° to the target (from behind) and at fusing is orientated such that the pitch and yaw are 15° and 35°.

The resulting lethality calculation yields a lethality value, P_k, of 35%. Figure 2.12 shows the geometry of the fragments and target. The blue regions show where the fragments intersect the target. As can be seen, portions of the cockpit, fuselage and wing are damaged.

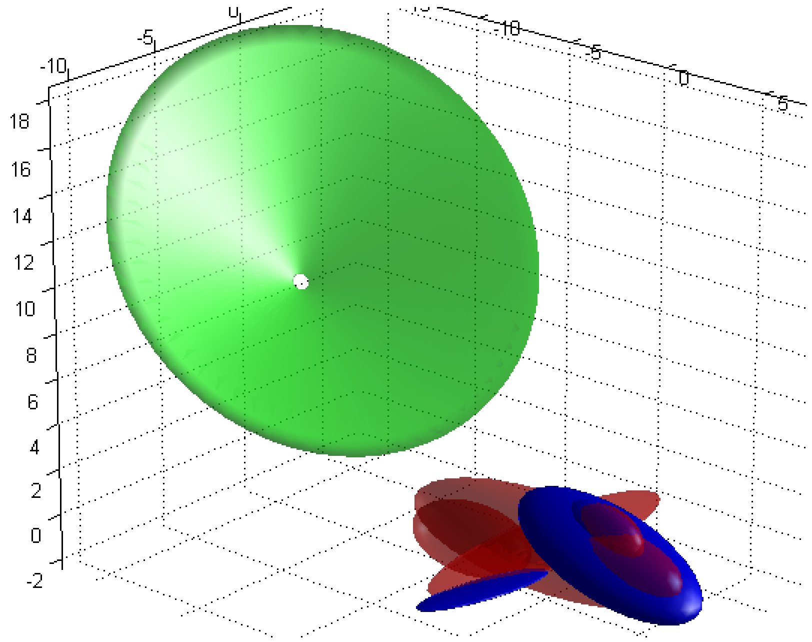


Figure 2.12: Example Endgame Geometry

2.5.2. Random Search

AGILE was exercised using random parameter inputs, in the ranges shown below in Table 2.4. As well the parameters used above the missile aim points are included. For completeness missile roll, Ψ, is also included, however due to the symmetrical nature of the warhead cone the value is not required.

Parameter	Min	Max	Units
V_M	0	2000	m/s
V_T	0	V_M	m/s
η	0	90	°
S_R	0	50	m
ω	0	180	°
δ	0	30	°
ε	0	30	°
Ψ	0	360	°
X_0	-5	5	m
Y_0	-5	5	m
Z_0	-5	5	m

Table 2.4: Agile Parameter Ranges

10000 random searches were undertaken and the distribution plotted into lethality probability bounds as shown in Figure 2.13.

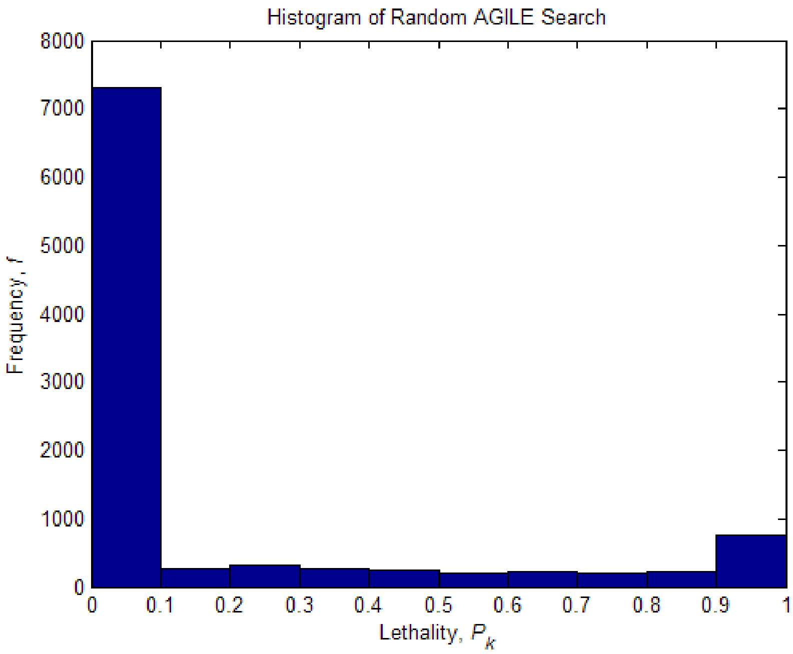


Figure 2.13: Random Search Results

The histogram in Figure 2.13 shows that for randomly distributed missile and target coordinates it is extremely difficult to achieve a high value of P_k. In fact, only 1620 of the solutions yielded lethality above 0.5. This is expected as there are many inputs to the system that can influence the outcome of the lethality model, such as the missile and target velocities and the angles between the missile and target, and fusing delay.

From these results it can be seen that some form of search is required to find sets of endgame parameters that yield higher lethality values. It is also useful to see if poorer lethality values can be improved upon by local optimisation of the endgame parameters. This optimisation would take place within close bounds of the original parameters.

Summary

This chapter has provided background on lethality and how it is calculated. The GW372 coordinate system and its associated parameter set has been illustrated and identified. The AGILE package has been described and its method of using Gaussians to represent endgame entities (target and warhead) has been covered and defined, further to which an example target is specified. The fragment damage model used by AGILE has been described. An example endgame has been shown and a random search performed to highlight the vast problem space that exists and the difficulty in finding high lethality scenarios has been exposed.

3. Lethality Optimisation

This chapter will discuss optimisation and the various types of optimisation methods. Sensitivity to disturbances of optimal parameters, known as robustness is then considered. Following this, a representative set of endgame scenarios will be optimised in order to assess if a maximum value of lethality lies close to the endgame parameter set, and a robustness measure of this maximum will be described and evaluated.

3.1. Optimisation Techniques

Within practical engineering fields, many problems may be encountered for which a superior solution may be found via optimisation of an existing design. These existing designs would be sub-optimal and would only reflect past design experiences. These kinds of problems are usually complex in nature, consisting of multiple variables and multiple design objectives and/or constraints. Some typical objectives may include performance, cost, and safety etc., which are functionally dependent on a set of input parameters (design variables).

The search space of such a multi-variable problem may contain numerous solutions. Many of the potential solutions found during a design might be local optima. There will be a global optimum or, as is usually the case with a complex multi-variable system, many solutions that lie across the search space each of which would offer competing optimal designs in some respect. An example showing these areas of the design space is shown in Figure 3.1

This is because some designs may provide better attainment of a particular objective whilst not achieving as high an attainment in other objectives. These different solutions would then need to be compared and a trade–off decision made to decide which solution to use. The set of solutions that provide an improvement in one objective whilst degrading another objective is called the Pareto-set.

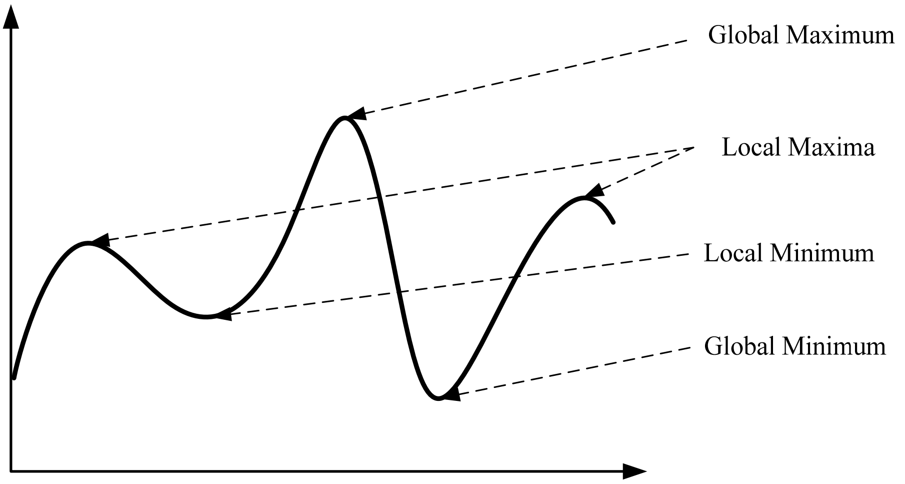


Figure 3.1: Local and Global Maxima and Minima

Similarly within a missile-target endgame scenario there will be many different sets of conditions that result in the same lethality, or P_k, level, for instance, the same lethality probability may occur from a tail chase condition, as well as from an angled head on condition depending on the relative positions and speeds of the target and missile. Therefore the problem set would contain many solutions depending on the endgame scenario (start point) and it may not be feasible to achieve the global solution from a given start point.

There are many methods by which the domain can be searched to find the optimal set of design or process parameters. Within the Matlab environment there exists the Optimisation Toolbox [14] which contains a representative range of search methods. The principles behind these search methods will be discussed in this section. Also discussed will be a multiple objective approach, the Multi Objective Genetic Algorithm, MOGA [15], a multi-objective form of the genetic algorithm.

Optimisation techniques are employed to seek out a set of optimal design parameters ($x=\{x_1, x_2,..., x_n\}$) that can be said to be optimal in some manner. In the most simplistic case this may be the minimisation (or maximisation) of a system that is dependant on one variable, say x. In a more complex situation $f(x)$ may contain many design parameters and may be subject to equality constraints, $G_i(x)=0$ ($i=1,...,m_e$); or inequality constraints, $G_i(x)\leq 0$ ($i= m_e+1,...,m$); or the design parameters may be limited to some boundary conditions, x_l, x_u.

The general optimization problem can be stated as:

$$\text{minimise} \quad f(x) \qquad x \in \Re^n \tag{3.1}$$

subject to

$$\begin{aligned} &G_i(x) = 0, && i = 1,\ldots,m_e \\ &G_i(x) \le 0, && i = m_e + 1,\ldots,m \\ &x_l \le x \le x_u \end{aligned} \tag{3.2}$$

where x is a vector of n design parameters, $f(x)$ is the scalar objective function to be minimised, and the vector function $G_i(x)$ returns an m length vector that contains the values of the constraints evaluated at x.

The solution to such a problem is dependant on the number and types of constraints as well as the limitations on the design parameters in the form of the bounds. The characteristics of the objective function also affect the efficiency and effectiveness of an optimiser.

When the objective function and constraints are linear functions of the design variables, the problem is known as a 'linear programming' (LP) problem [16]. Linear programming is a technique used for the optimisation of a linear objective function subject to a number of constraints on the variable parameters of the function.

A 'quadratic programming', or QP minimisation or maximisation problem is one that has a quadratic objective function, subject to linear constraints on the design parameters [17]. Reliable solutions to the LP and QP problems exist, for example the Simplex algorithm [18], and Broyden- [19] Fletcher- [20] Goldfarb- [21] Shanno [22] or BFGS method which will be covered below.

A more complex problem arises when the objective function and/or constraints are nonlinear functions of the design variables. This is known as a 'Nonlinear

Programming', or NP problem. A solution to the NP problem would generally require an iterative approach. This is so a search direction can be established at each iteration in order to try and converge to a solution. This can be achieved, for example, by reformulating the problem and using a LP, QP or an unconstrained solver.

3.2. Unconstrained Optimisation

Many methods exist for unconstrained problems; however most can be defined by the derivative information that is used by the method. For example the simplex search methods of Nelder-Mead [18], which only use the function evaluations, are useful for problems that are highly nonlinear or contain many discontinuities to the search space. Another class of method, the gradient based searches, are better suited to a problem set with an objective function that has continuous first derivative information. Higher order methods are only really suited when the second order function derivatives are readily computed as it is usually computationally expensive to do so numerically.

3.2.1. Gradient-based Methods

Gradient-based methods use derivative information to influence the direction, or an approximation of the direction, of search where the minimum is believed to be. These will be described below.

3.2.1.1. Steepest descent

The simplest of these methods is the steepest descent method, which searches in the direction $-\nabla f(x)$, where $\nabla f(x)$ is the gradient of the objective function.

3.2.2. Quasi-Newton

Quasi-Newton methods [19] use the derivative information at each iteration to generate a quadratic model problem such that:

$$\frac{\min}{x} \quad \frac{1}{2}x^T Hx + c^T x + b \tag{3.3}$$

where H is the Hessian Matrix, a positive definite symmetric matrix, c is a constant vector and b is a constant. The optimum of this problem is found when the partial derivatives of x are zero, so:

$$\nabla f(x^*) = Hx^* + c = 0 \tag{3.4}$$

where x^* is the optimal solution point, and can be written as,

$$x^* = -H^{-1}c \tag{3.5}$$

Quasi-Newton methods use the behaviour of *f(x)* and ∇*f(x)* to calculate information on the curvature and use an updating technique to approximate the value of *H*. Newton methods calculate *H* numerically which can be computationally demanding.

There are many updating techniques available to obtain a value for *H*. Of these, the BFGS formulation [19-22] is considered to be quite effective in the general case. The BFGS formulation is:

$$H_{k+1} = H_k + \frac{q_k q_k^T}{q_k{}^T s_k} - \frac{H_k^T s_k^T s_k H_k}{s_k^T H_k s_k} \tag{3.6}$$

where

$$\begin{aligned} s_k &= x_{k+1} - x_k \\ q_k &= \nabla f(x_{k+1}) - \nabla f(x_k) \end{aligned} \tag{3.7}$$

H_0 can be set to any positive definite symmetric matrix, such as *I*,

In order to avoid the inversion of *H*, Davidon [23], Fletcher and Powell [24] devised the DFP updating method. The DFP method uses the same formula as BFGS except that q_k is used instead of s_k.

The gradient information is either obtained through calculating the gradients analytically, or it is determined by partial derivatives using a numerical differentiation method with finite differences. This is done by perturbing each of the design variables, *x*, in turn and calculating the rate of change in the objective function. At each major iteration, *k*, a line search is performed in the direction:

$$d = -H_k^{-1}.\nabla f(x_k) \tag{3.8}$$

3.2.2.1. Line Search

The search direction in which a solution is estimated to lie is usually calculated by solving a sub-problem. The minimum along the line formed is approximated using a search procedure (e.g. Fibonacci) or by a polynomial method involving interpolation or extrapolation (e.g. quadratic, cubic). Polynomial methods approximate a number of points with a univariate polynomial whose minimum can be found easily. Interpolation refers to the condition that the minimum lies within the area spanned by the available points, and extrapolation refers to a minimum that is located outside the range of the spanned points. Extrapolation methods are considered generally unreliable for estimating minima for nonlinear functions. However, extrapolation methods are useful for estimating step length when trying to approach a region close to the solution. Polynomial interpolation methods are usually efficient when the function is continuous. The problem is to find a new iterate x_{k+1} of the form:

$$x_{k+1} = x_k + \alpha * d \tag{3.9}$$

where the current iteration is denoted by x_k, *d* is the search direction (obtained by an appropriate method), and α* is a scalar step parameter that is the distance to the minimum.

3.2.2.2. Quadratic Interpolation

Quadratic interpolation involves a data fit to a univariate function of the form:

$$m_q(\alpha) = a\alpha^2 + b\alpha + c \tag{3.10}$$

where the extremes occur at a step length of

$$\alpha^* = \frac{-b}{2a} \tag{3.11}$$

This point is a minimum when interpolation is performed (i.e. a bracketed minimum) or when a is positive. The coefficients a and b can be calculated using any combination of three function or gradient evaluations, or from two gradient evaluations. The coefficients are found by formulating and solving a set of linear simultaneous equations. Simplifications to these equations can be achieved by using particular characteristics of the points, for example, the first point can usually be taken as $\alpha = 0$. Other simplifications can be achieved when the points are evenly spaced. For example, assume there are three unevenly spaced points $\{x_1, x_2, x_3\}$ and their respective function values are $\{f(x_1), f(x_2), f(x_3)\}$, the minimum resulting from a second-order fit is given by

$$x_{k+1} = \frac{1}{2}\frac{\beta_{23}f(x_1) + \beta_{31}f(x_2) + \beta_{12}f(x_3)}{\gamma_{23}f(x_1) + \gamma_{31}f(x_2) + \gamma_{12}f(x_3)} \tag{3.12}$$

where,

$$\begin{aligned} \beta_{ij} &= x_i^2 - x_j^2 \\ \gamma_{ij} &= x_i - x_j \end{aligned} \tag{3.13}$$

for interpolation to be used, as opposed to extrapolation, the minimum must be bracketed so that the points can be arranged to give

$$f(x_2) < f(x_1) \quad \text{and} \quad f(x_2) < f(x_3) \tag{3.14}$$

3.2.2.3. Cubic Interpolation

This is more useful when gradient information is readily available or when three or more function evaluations have been computed. It also involves a data fit to the univariate function

$$m_c(\alpha) = a\alpha^3 + b\alpha^2 + c\alpha + d \tag{3.15}$$

where the local extrema are roots of the solution to the derivative i.e.

$$3a\alpha^2 + 2b\alpha + c = 0 \tag{3.16}$$

In order to find the minimum of the above equation, the root that gives $6a\alpha+2b$ as positive should be used. The coefficients a and b can be found using any combination of four function or gradient evaluations or with just three gradient evaluations. These coefficients are found by formulating and solving a set of linear simultaneous equations as before.

Given two points, $\{x_1, x_2\}$, with their corresponding gradients with respect to x, $\{\nabla f(x_1), \nabla f(x_2)\}$, and respective function values, $\{f(x_1), f(x_2)\}$, the update is

$$x_{k+1} = x_2 - (x_2 - x_1)\frac{\nabla f(x_2) + \beta_2 + \beta_1}{\nabla f(x_2) - \nabla f(x_1) + 2\beta_2} \tag{3.17}$$

where,

$$\begin{aligned} \beta_1 &= \nabla f(x_1) + \nabla f(x_2) - 3\frac{f(x_1) - f(x_2)}{x_1 - x_2} \\ \beta_2 &= \left(\beta_1^2 - \nabla f(x_1)\nabla f(x_2)\right)^{1/2} \end{aligned} \tag{3.18}$$

3.2.2.4. Quasi-Newton Implementation

A Quasi-Newton method is available to be used in Matlab. This algorithm is made up of two stages:

1. Hessian Update (calculating the search direction, such as BFGS or DFP)
2. Line Search Procedures (quadcubic or cubicpoly)

3.2.2.4.1. Hessian Update

The Hessian, *H*, is always kept positive definite to ensure that the direction of search, *d*, is always in a descent direction. Therefore some arbitrarily small step in the descent direction will result in the objective function decreasing by some magnitude. Positive definiteness of *H* is achieved by seeing that *H* is initialized to be positive definite and thereafter $q_k^T s_k$ (from Equation 3.19) it is always positive. The $q_k^T s_k$ term is a product of the line search step length parameter $\alpha_{k,}$ and a combination of the search direction *d* with past and present gradient evaluations such,

$$q_k^T s_k = \alpha_k \left(\nabla f(x_{k+1})^T d - \nabla f(x_k)^T d \right) \tag{3.19}$$

The value of $q_k^T s_k$ can be kept positive by varying the accuracy of the line search. The search direction is a descent, therefore, α_k and - $\nabla f(x_k)^T d$ are positive. Thus, the possible negative term, $\nabla f(x_{k+1})^T d$, can be made as small as possible by using a more accurate line search.

3.2.2.4.2. Line Search

There are two types of line search that can be used by this function depending on whether the gradient information is easily obtainable or not. If gradient information is not readily available it is best to use a cubic polynomial method, and if the gradient information is more difficult to evaluate (for instance if it needs to be found using finite difference methods) it is better to use a mixed quadratic and cubic polynomial method.

3.2.2.5. Cubic Polynomial Method.

When the cubic polynomial method is used, the gradient and function evaluation is calculated at each iteration *k*. At each iteration the update is performed when a new point, x_{k+1}, is found that satisfies the following condition

$$f(x_{k+1}) < f(x_k) \tag{3.20}$$

At each iteration, a step, α_k, is used to form a new iterate of the form

$$x_{k+1} = x_k + \alpha_k d \tag{3.21}$$

If this step does not conform to the condition in Equation 3.20, then the value of α_k is reduced to form a new x_{k+1}. The general rule for the reduction in α_k is to continually reduce it by factor of 0.5 until there is a reduction in $f(x)$. This technique is quite slow, however, compared to a method that uses function and gradient information together with cubic interpolation/extrapolation to determine the estimates for step length.

When a point that satisfies the condition in equation 3.20, an update is performed if $q_k^T s_k$ is positive, otherwise cubic interpolations are performed until the univariate gradient term $\nabla f(x_{k+1})^T d$ is small enough for $q_k^T s_k$ to be positive. Following each update procedure a step length of α_k is tried, after which a number of possible outcomes may occur.

At each iteration, a cubically interpolated step length α_c is calculated and then used to adjust the step length parameter α_{k+1}. Occasionally, for highly nonlinear functions, the value of α_c can be calculated to be negative, in which case α_c is given a value of $2\alpha_k$. and the next iteration computed. Some robustness measure can also be included so that, even in the case when false gradient information is supplied, a reduction in $f(x)$ can be obtained by taking a negative step. This is achieved by setting

$\alpha_{k+1} = -\alpha_k/2$ when α_k falls below a certain threshold value. This is critical if a high level of accuracy is required, if only finite difference gradients are available.

3.2.2.6. Mixed Cubic/Quadratic Polynomial Method

When gradient information is not readily available, a mixed method may be appropriate. This interpolation/extrapolation method is implemented so that gradients are not needed at every iteration. The approach involves using quadratic interpolation and the minimum is usually bracketed using some form of bisection. This method does not use all the available information.. For example, the gradient is always calculated for the Hessian update for each major iteration. Therefore, given three points that bracket the minimum, it is possible to use cubic interpolation, which would provide a much more accurate calculation than quadratic interpolation. Hence, the method that is used in Matlab is to find three points that bracket the minimum and to use cubic interpolation to estimate the minimum at each line search.

If the interpolated point is greater than any of the three used for the interpolation, then it is replaced with the point with the smallest function value. Following the line search procedure, the Hessian update procedure is performed as for the cubic polynomial line search method.

3.3. Constrained Optimisation

The aim of constrained optimisation is to transform the problem into an easier sub-problem. This can then be solved and used as the base for an iterative function. Earlier methods of optimisation focused on converting the problem to an unconstrained one and then applied penalty functions to constraints that were near or beyond the boundary constraints. The constrained problem can then be solved using a sequence of parameterised unconstrained optimisations. These methods are generally inefficient and have been replaced by methods that involve the solution of the Kuhn-Tucker (KT) equations [25]. The KT equations are fixed conditions required for the optimality of a constrained optimisation problem. If the problem is a convex programming one, such that *f(x)* and $G_i(x), \{i = 1,\ldots,m\}$ are convex functions, then the KT equations are both necessary and enough for a global solution point.

Using Equation 3.1 the Kuhn-Tucker equations can be given as

$$
\begin{aligned}
\nabla f(x*) + \sum_{i=1}^{m} \lambda_i * \cdot \nabla G_i(x*) &= 0 && && (a) \\
\lambda_i * \cdot G_i(x*) &= 0 && i = 1,\ldots,m && (b) \qquad (3.22) \\
\lambda_i * &\geq 0 && i = m_e + 1,\ldots,1 && (c)
\end{aligned}
$$

Equation 3.22(a) provides a cancelling of the gradients of the objective function and the constraints that are active at the solution point. In order to achieve this cancellation, Lagrange multipliers (λ_i, i=1,…,m) are used to balance the changes in magnitude of the objective function and constraint gradients. Only active constraints are included in the cancelling operation, therefore any inactive constraints have their Lagrange multipliers set to zero.

The solution to the KT equation provides the structure for many nonlinear programming algorithms. The algorithms try to calculate the Lagrange multipliers directly. Constrained quasi-Newton methods ensure convergence of the solution by gathering second order data of the KT equations using a quasi-Newton updating procedure. These methods are usually referred to as Sequential Quadratic Programming (SQP), because a QP sub-problem is solved at each major iteration. SQP is also known as Iterative Quadratic Programming, Recursive QP and Constrained Variable Metric Methods.

The SQP mimics the Newton method used in unconstrained optimisation. At each major iteration the Hessian of the Lagrangian function is estimated using a quasi-Newton updating method. This is then used to generate a sub-problem which is solved to provide a search direction for a line search procedure.

Given the general problem described earlier (Equation 3.1), the QP sub-problem is formulated using a quadratic approximation of the Lagrangian function,

$$L(x,\lambda) = f(x) + \sum_{i=1}^{m} \lambda_i \cdot g_i(x) \tag{3.23}$$

The general problem is simplified using the assumption that boundary constraints are expressed as inequalities. The QP sub-problem is obtained by linearising the non-linear constraints. The sub problem can be written as,

$$\begin{array}{ll} \text{minimise} & \frac{1}{2} d^T H_k d + \nabla f(x_k)^T d \\ d \in \Re^n & \end{array} \tag{3.24}$$

$$\begin{array}{ll} \nabla g_i(x_k)^T d + g_i(x_k) = 0 & i = 1,\ldots,m_e \\ \nabla g_i(x_k)^T d + g_i(x_k) \leq 0 & i = m_e + 1\ldots,m \end{array}$$

This sub-problem can be solved using a QP algorithm. The solution gives an iterate of the form:

$$x_{k+1} = x_k + \alpha_k d_k \tag{3.25}$$

The step length parameter, α_k, is determined using a suitable search method such that a sufficient decrease in the merit function is attained. The Hessian matrix, H_k, is a positive definite matrix and is calculated by any of the quasi-Newton methods, such as BFGS.

A non-linear constrained problem can be solved in fewer iterations than an unconstrained problem using SQP. This is due to the limits on the feasible region to be searched, the optimiser can make more informed decisions on the search direction and step length change.

3.4. Sensitivity to Disturbance (Robustness)

In order to establish how sensitive the optimal solution is to parameter variations the optimal parameter set can be perturbed by some set amount. This amount can either be within a fixed bound, or a percentage of the parameter value, or a percentage of the maximum value that the parameter range lies in, as shown for a three parameter system in Figure 3.2 below.

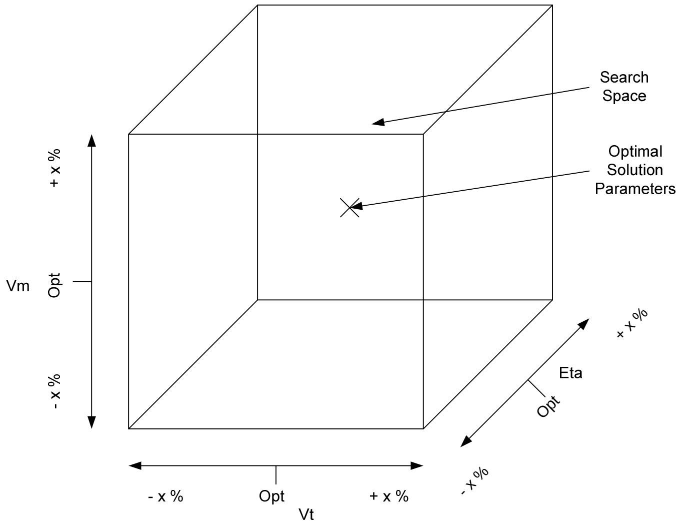


Figure 3.2: Perturbation Search Space

These parameters are then perturbed by this amount about their nominal values. Following the perturbation a second evaluation of the objective function is found to identify the deviation from the optimal solution.

Multiple samples can be taken for each optimisation, and a sensitivity measure can be assigned, for example the mean of the perturbed sample evaluations or the standard deviation of the samples, to establish how the robust a solution is to disturbances..

3.5. Agile Analysis Using the Matlab Optimisation Toolbox

Following the initial study of AGILE, where a random search using all design variables was explored (Section 2.5), analysis of endgame lethality using methods from the Matlab Optimisation Toolbox was undertaken.

The tolerance of an objective function is the sensitivity of the evaluation of the objective function. It measures how many decimal places the optimiser will use to decide whether or not to terminate the search. In order to decide the tolerance setting for the objective function for this study, a simple three parameter objective function was evaluated for ten start points, using missile and target velocities and engagement angle as the parameters, such that:

$$P_k = f(V_M, V_T, \eta) \tag{3.26}$$

The results of running the optimiser with different tolerance settings of the objective function are shown below in Figure 3.3.

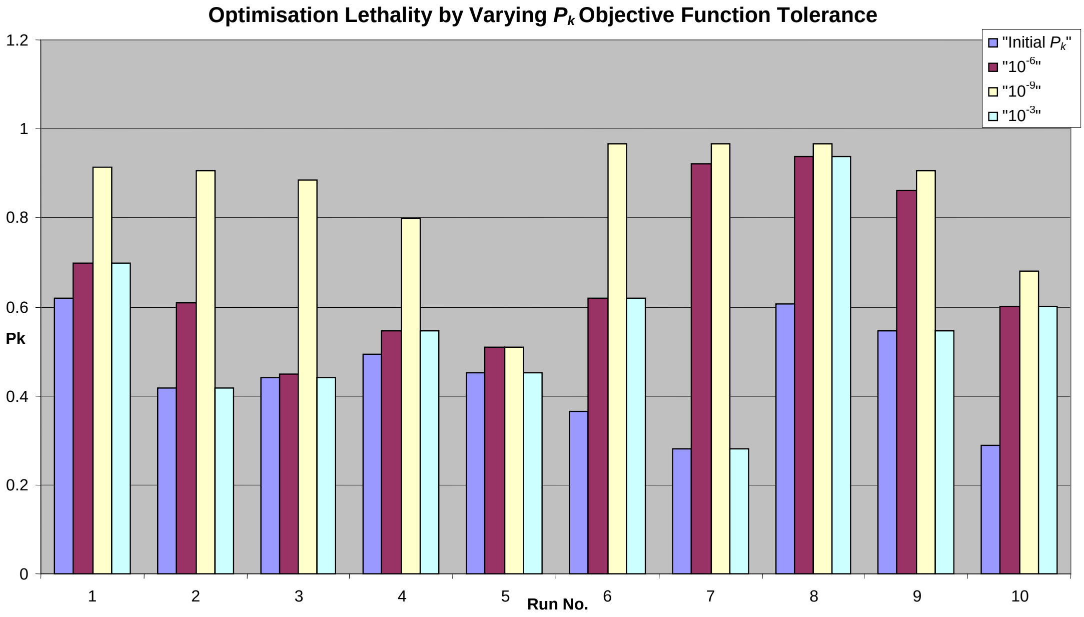


Figure 3.3: Varying Tolerances of Optimiser

From the graph above it can be seen that the most improvement occurs when the tolerance is set to 10^{-9}, as would be expected as this is the most sensitive. Therefore the 'options' structure for the optimiser will use this value for future optimisation purposes.

Following each optimisation a sensitivity function is run in order to gauge how sensitive the optimal solution is to disturbances of parameters. Initially, two measures are calculated, the mean of all perturbed lethality values, and the standard deviation of these perturbed lethality variations.

A sample of 500 random start points were implemented and the lethality calculations performed, for each of three categories based on the engagement angle, η. The three categories were for front on, side on and rear on fusing points. These are defined as follows:

Front On: 135° - 225°
Side On: 45° - 135° and 225° - 315°
Rear On: 0° - 45° and 315° - 360°

In addition to these categories, for each optimisation only 'controllable' parameters are optimised, i.e. those that one can adjust, for example the missile parameters such as missile orientation and warhead aim point are perturbed and not the target velocity and orientation.

Bounds were placed on the parameters to be optimised such that any solution found does not result in a set of parameters which would be unrealisable in a practical situation. The bound on the range of parameters during the optimisation from the initial start points will ensure that for an end-game scenario the optimised parameters are feasible in terms of being able to realise an increase in lethality through small deviations from the initial search points.

The bounds used for the optimisation were defined as shown in Table 3.1.

Parameter	Bounds	Units
V_M	±50	m/s
η	±15	°
δ	±15	°
ε	±15	°
X_0	±5	m
Y_0	±5	m
Z_0	±5	m

Table 3.1: Optimisation Bounds

For each optimal solution, 1000 different perturbations are evaluated to calculate the robustness of each solution by calculating the standard deviation and mean of the lethality values found from the lethality achieved from perturbations of the missile parameters within AGILE.

For each category 500 valid start points, i.e. those that initially yield lethality values greater than zero, are assessed. The results are displayed for the simple aircraft model in Figure 3.4. The graphs on the left show optimal lethality vs. standard deviation, and the right side shows optimal vs. mean lethality.

As can be seen, the results show that for all three scenario categories, the optimiser can yield both low and high lethality probabilities, and that these consist of solutions that range from robust solutions, showing a very low standard deviation and high mean from the perturbations of parameters from the optimal found; to those that are extremely sensitive to variations in the optimal parameters, whereby the mean lethality can drop by as much as 95% in the case of rear endgame scenarios.

How ever there are many endgame scenarios that do not yield high lethality values, or are not robust, possibly due to unlikely endgame scenarios. Therefore a fly out analysis will be performed in order to understand how lethality varies during the missile fly out.

Small changes to say, delta or epsilon, can reduce the probability of lethality dramatically. Therefore it would be ideal to look for solutions that not only give high lethality probability, but also provides a robust solution, or that will not deteriorate significantly if the parameters are perturbed slightly. This led to the investigation of the lethality problem using a multi-objective approach. The research and analysis into multi-objective optimisation using a genetic algorithm is discussed in Appendix B, and also Appendix C, a paper published at the 16th IFAC World Congress, as this is an extension of the main body of the research. A summary of this is provided below.

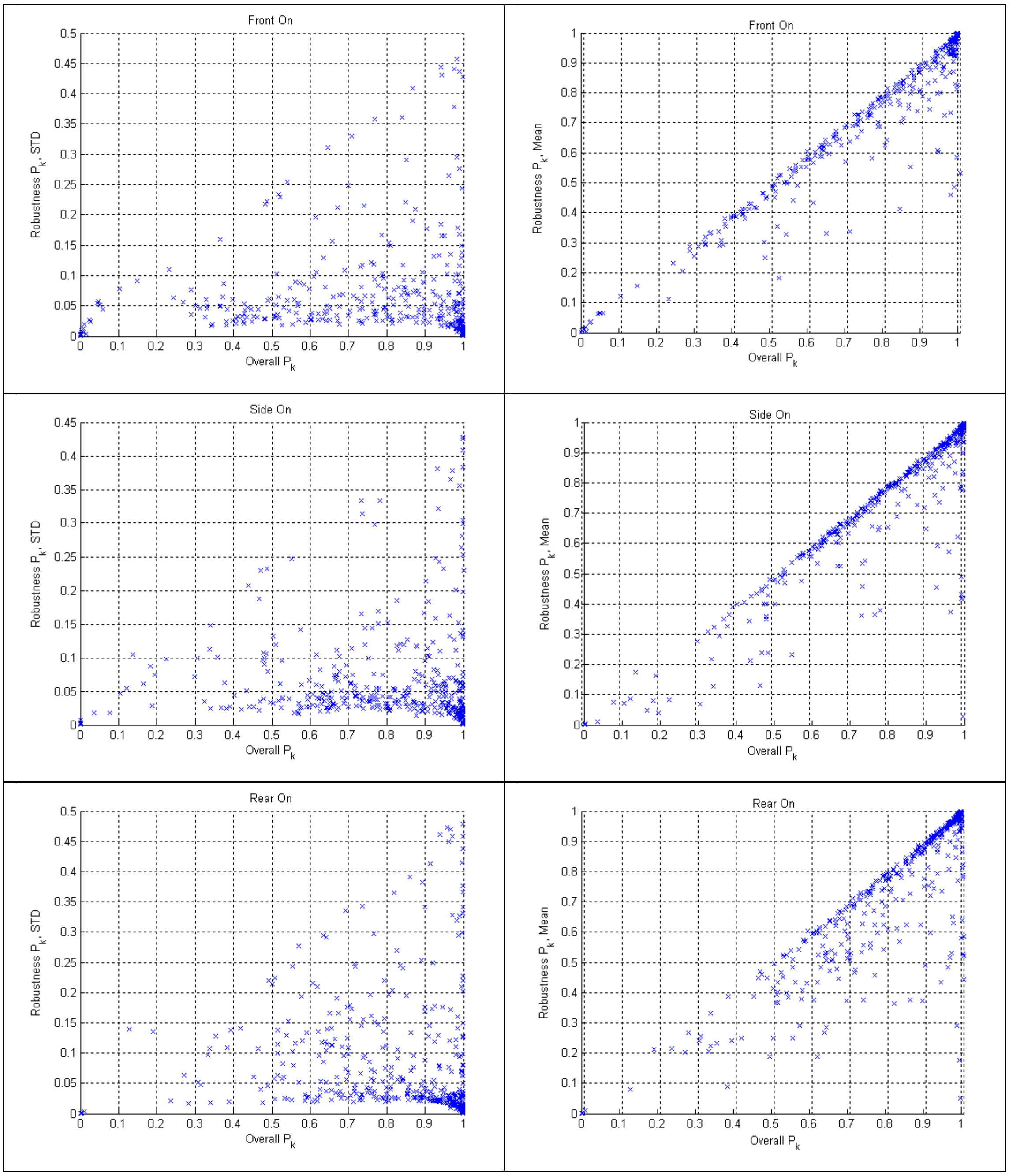


Figure 3.4: Optimal Lethality and Corresponding Sensitivity Measure.

3.6. Multi Objective Optimisation

The use of multi-objective optimisation (MO) in engineering design recognises that most practical problems involve a number of design criteria that need to be satisfied simultaneously, such that:

$$\min_{x \in \Omega} \quad G(x) \tag{3.27}$$

where x =[x_1, x_2,..., x_n] and Ω define the set of free variables, x, subject to any constraints and $\boldsymbol{G(x)}$ = [$g_1(x)$, $g_2(x)$,..., $g_n(x)$] are the design objectives to be optimised.

For this set of functions, $\boldsymbol{G(x)}$, it can be seen that there is no one ideal optimal solution, but rather a set of solutions for which an improvement in one design objective will lead to a degradation in one or more of the other objectives. This set is known as the Pareto-optimal solution set. These solutions are also known as non-dominated solutions to the MO optimisation problem.

These solutions can be sought after using the NP methods discussed earlier by means of applying weighting and goal attainment functions for the defined objectives; however these approaches require precise expression of a usually not well understood set of weights and goals. In addition to this NP methods can not handle multimodality and discontinuities in the function space well, and so are likely to find local solutions only.

Because of the stochastic nature of the search mechanism, genetic algorithms (GA) are capable of searching the entire solution space with more likelihood of finding the global optimal than conventional methods. Conventional methods usually require the objective function to be well behaved, whereas the generational nature of GAs can tolerate noisy, discontinuous and time-varying function evaluations. Furthermore EAs allow the use of mixed decision variables (binary, n-ary and real-values) that allows the parameterisation to closely match the nature of the problem.

It has been shown that EAs can offer an advantage over conventional methods in optimal design problems and the related field of performance seeking control [26].

3.6.1. Multi Objective Genetic Algorithm

The idea of the fitness of an individual solution estimate and the associated objective function value are closely related in a single objective framework. The objective function characterises the problem domain and cannot be changed at will, whereas the fitness of an individual can change depending on the solutions ability to reproduce and as such can be treated as part of the GA search strategy. However with the multi-objective case, these two values cannot be linked so closely, and the distinction between them becomes more important. As described by Fleming and Fonseca [15], this distinction becomes important when performance is measured as a vector of the objectives, because the fitness value must remain a scalar. Individual are assigned a measure of utility dependant on whether they perform better, worse, or similar to others in the population.

3.6.2. Decision Strategies

In the absence of any information regarding the relative importance of design objectives, Pareto-dominance is the only method of determining the relative performance of solutions. Non-dominated individuals are all therefore considered to be the best performers and are thus assigned the same fitness, e.g. zero. However determining the fitness of dominated solutions is a more subjective matter. An approach that can be used is to assign a cost proportional to the number of individuals in a population that dominate a given individual, as illustrated in Figure 3.5. In this instance non-dominated individual are treated as desirable.

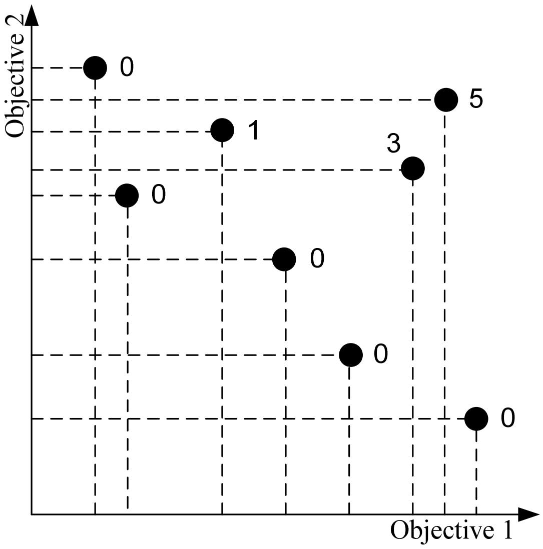


Figure 3.5: Pareto Ranking

If goal and/or priority information is available for the design objectives, then it may be possible to differentiate between some non-dominated solutions. For example, if degradation in an individual's objectives still allow those goals to be maintained but also allow the attainment of some goals in other non-satisfied objectives, then these degradations should be accepted. In cases where different priority levels are set for each objective then it is important to improve the high priority objective, such as hard constraints, after which the lower priority objectives may be improved.

3.6.3. Initial MOGA Analysis

A multi objective optimiser, MOGA, was initially employed using each individual component's lethality probability as individual objectives (wings, engine fuselage, cockpit), with the overall lethality value as a fifth objective. A population of fifty individuals per generation was initialised, using the three parameter setup employed in the previous studies (V_M, V_T, η). The MOGA then generated a generation of solutions that provided a measure of how the individual components of P_k interacted with each other. This showed that if the cockpit P_k value was high for example, then the engine P_k was lower, due to its relative position on the aircraft itself.

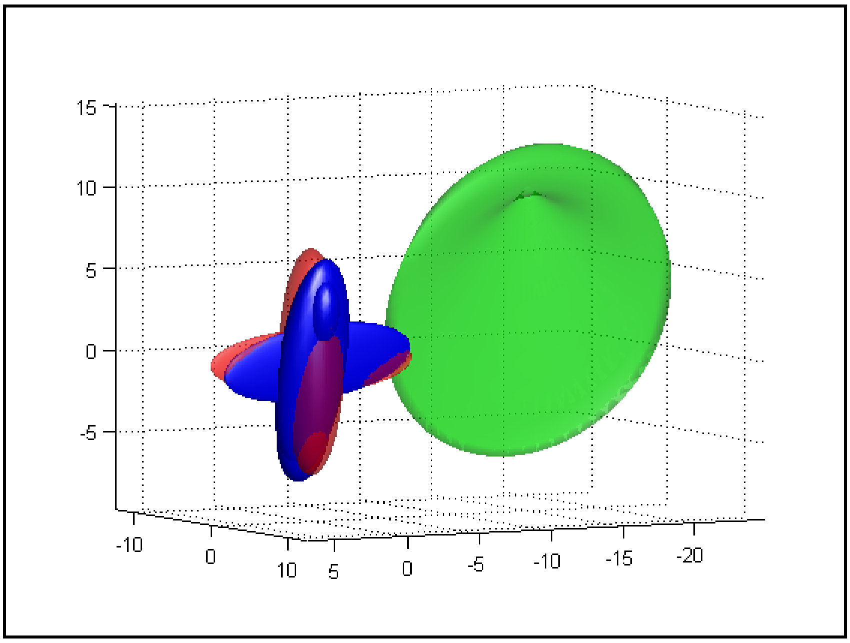


Figure 3.6: Example MOGA Optimised Endgame Geometry

An example of such a case is shown in Figure 3.6. the MOGA attempts to optimise maximum damage to all components of the target aircraft and as a result, the cone of fragments is a flat disc shape that hits the aircraft diagonally along the length of the craft in order to inflict damage upon all four components.

3.6.2. Robust MOGA Optimisation

The next stage for the MOGA software was to set the objectives as total P_k and the robustness value from the previous work with the standard Matlab optimisers. This configuration was set such that for each suitable individual (solution) a routine was run that sampled ten deviant (from optimal parameters) solutions as for the Matlab optimiser and the worst case was used as the sensitivity measure. For this setup the minimum value of perturbed samples was used. However, a more suitable measure for this sensitivity is a standard deviation of the perturbed samples. Another implementation of MOGA explored this measure of sensitivity as the second objective, and also a third objective of maximising the mean was implemented, although this is closely connected to the standard deviation, it gives a slightly easier visual of the performance of individuals in the population.

All the endgame parameters were considered for this implementation. The Trade-Off window is coded so that selecting an individual's line would display the corresponding engagement geometry using the AGILE GUI.

Three runs were undertaken, for front on, side on and rear on scenarios using engagement angle constraints ($-45 < \eta < 45$ for rear on, $45 < \eta < 135$ for side on, and $135 < \eta < 225$ for front on), using 50 individual per generation, for 200 generations, and for each individual, 50 perturbed samples are taken to establish the sensitivity measure of standard deviation and mean. The sensitivity measure is calculated by perturbing only those variables that are controllable, i.e. the missile parameters, δ, ε, x_0, y_0, z_0, Z_{delay}.

As can be seen there are many competing solutions present that offer high lethality probabilities which are also robust to perturbations in the missile parameters. These solutions are shown in Figure 3.7. The scatter plots of the Pareto Solutions (left), accompanied by the scatter plot of all solution found in 200 generations (right), are shown as overall P_k (nominal) vs. standard deviation.

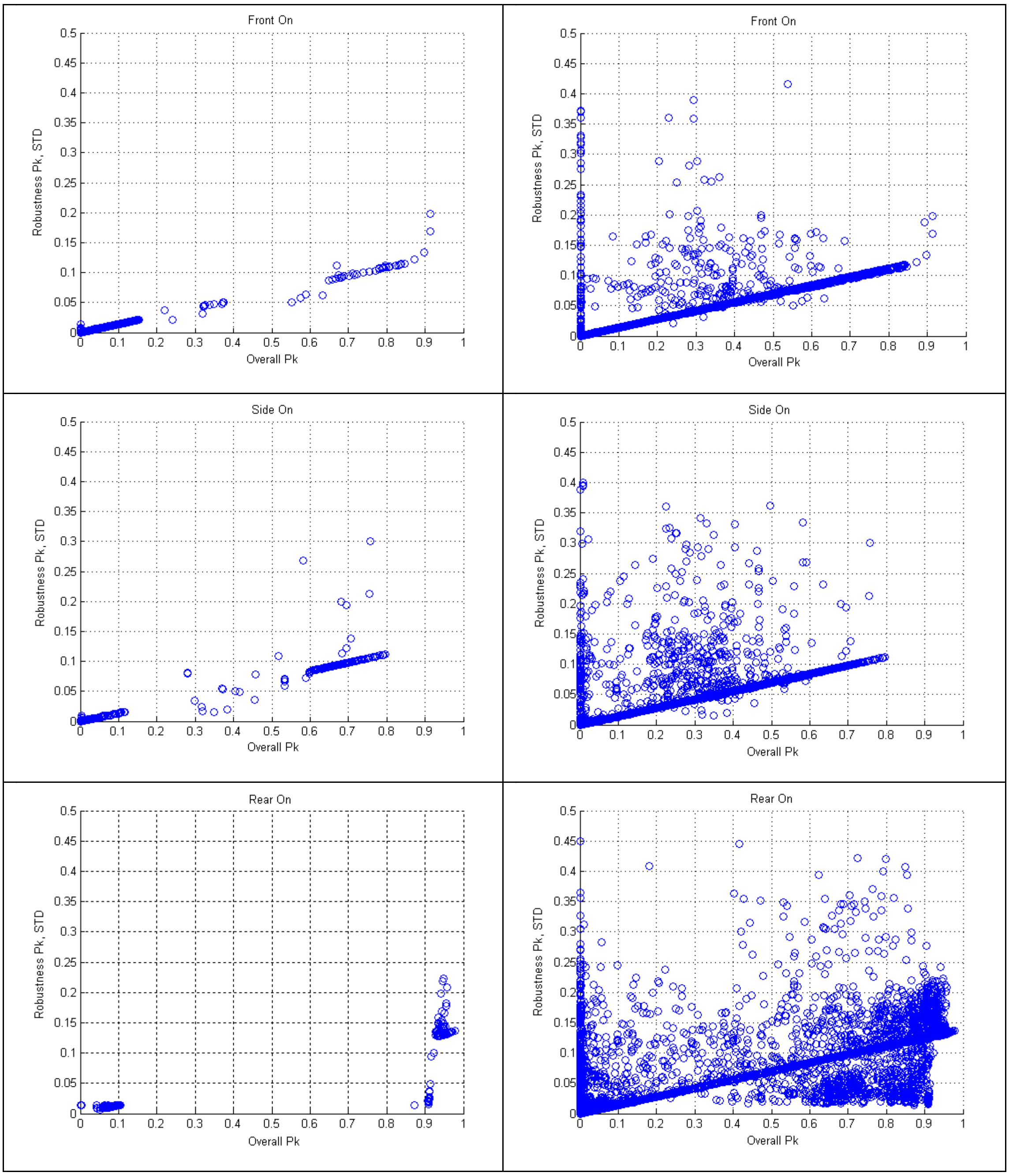


Figure 3.7: MOGA Optimisations for 200 Generations

As can be seen many solutions exist, however most are dominated by the Pareto set, and for each case the middle region of P_k yields sparser solutions. Looking at the graphs on the right, for all solutions, a definite trend can be seen showing the increase in standard deviation as overall optimal probability increases, however there do exist some solutions that can provide a good robustness measure, and it is these that show on the Pareto Front.

It was found that whilst the MOGA yielded some interesting results, it did not present a viable practical solution, due the time taken by the optimiser to find solutions. Hence it was decided to pursue the regular optimisation techniques to find robust solutions due the speed at which optimisations can be performed on the initial endgame data.

Summary

This section has covered the topic of optimisation and robustness. Different optimisation types have been discussed. AGILE has been used in the objective function for an optimiser from the Matlab Toolbox, and its tolerance setting established. Many scenarios have been optimised and related robustness measures found.

4. Modelling Fly-Out of Missile

This chapter will describe MSTARS, a missile fly-out simulation tool for Simulink. Its basic workings and how it can be used with AGILE will be described and some sample fly-out scenarios illustrated. Following this some fly-out scenarios will be analysed to see how lethality varies in the final stages of fly-out.

4.1. MSTARS Overview

Munition simulation tools and resources is a simulation toolbox for use within Matlab's Simulink environment. It allows the modelling and simulation of weapon systems for analysis purposes. Various models can be employed, including launch-capable airborne vehicles such as aircraft, helicopters, as well as ships and ground-based vehicles. MSTARS allows a user to construct various missile fly out scenarios using such vehicles as the missile launch vehicle and as the target.

The advantages of using MSTARS include

- A reduction in time required for analysis compared to previous methods
    - MSTARS runs faster than traditional approaches for building models & conducting analysis, such as
- Graphical approach enables a user to build models in an easier manner
- Reusable models and components make construction of complex scenarios easy
- Ability export data to Matlab workspace
    - Allows Matlab functions and compiled c-code to be used for analysis, i.e. using the AGILE package.

A simple simulation setup is shown in Figure 4.1, using two generic aircraft, and a generic missile launched from one at the other. Initial conditions for the positions and velocities of the fighters can be defined prior to running a simulation. Once the simulation has been run the final conditions can be exported to Matlab in order for a lethality calculation to be made using the AGILE software, and also to enable the plotting of flight paths.

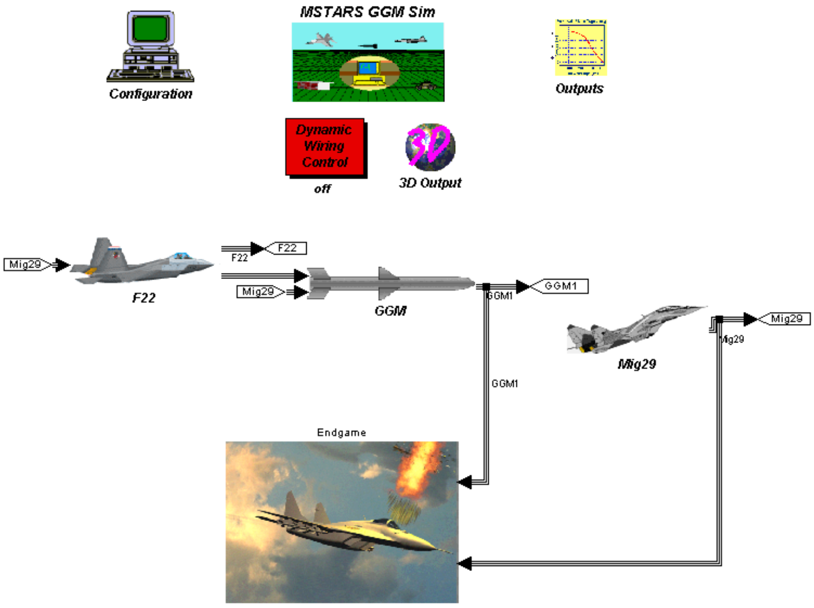


Figure 4.1: Top Level MSTARS Simulation

The above Simulink setup can be compiled using a built-in utility in allowing simulations to be processed faster using the command line interface rather than through the GUI. This can then be used by Matlab scripts to run batches of simulations for analysis of various fly out conditions.

4.1.1. Simulation Components

As can be seen in Figure 4.1, a simulation consists of a set of blocks, each representing a component of the simulation. In this example the components of the simulation are the target and fighter aircrafts, the missile system and the endgame component. The connections between components represent the data flow of the system's states and signals.

4.1.1.1. Fighter and Target Aircraft Model

Both the fighter and target aircraft models are made up using a simple generic fighter model. This generic model, shown in Figure 4.2, contains the kinematics for the aircraft, i.e. the state equations and equations of motion of an aircraft, including angles and positions and rates of change of these, as well as the missile launch data and logic system. This data can be used to plot the path of the aircrafts and the missile. The missile launch data need only be contained in the attacking aircraft model; it can disabled in the target aircraft.

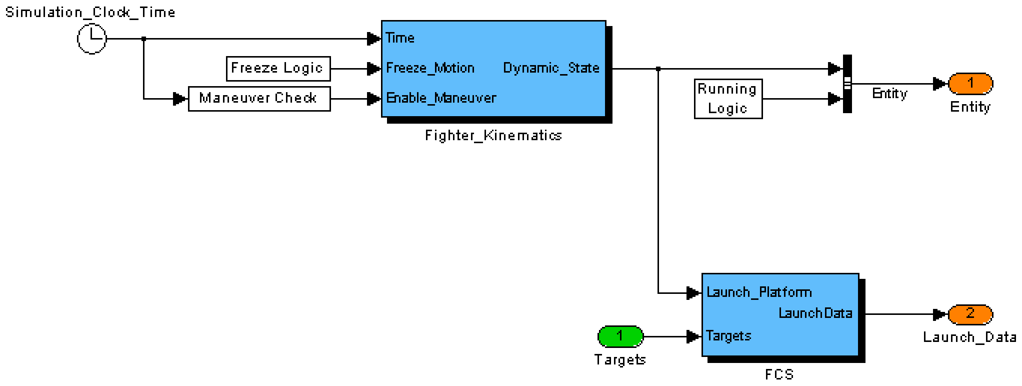


Figure 4.2: Fighter and Target Aircraft Model

4.1.1.2. Missile Model

The missile model consists of various components for each of the missile subsystems as shown in Figure 4.3. These are described below:

- Kinematic Seeker

  The seeker model provides calculations of range, closing velocity, azimuth and elevation line-of-sight angles, and relative position and velocity values.
- Target Filter Kinematic Seeker

  The targeting filter works with the kinematic seeker and provides estimates of the line-of-sight angle rates, range, closing velocity, target position and relative position of the target with respect to the missile.
- Biased Proportional Navigation Guidance

  Missile guidance is provided by a simple implementation of a biased proportional navigation (BPN) system. It is used with the seeker and inertial navigation system, and directs the missile towards the target using.

- Autopilot

  A skid-to-turn linear dynamic compensator autopilot model is implemented. This component produces control surface deflection commands for the missile in roll, pitch and yaw directions.

- Four Fin Control System

  A simple four-fin control surface model that uses 1st order actuator models. Commands from the autopilot are implemented here.

- 6 Degrees of Freedom model

  This model contains the equations of motion for a symmetric body assuming a flat Earth.

- One Stage Motor

  A simple motor model which has been incorporated that allows the missile to burn for a specified time, and provides thrust force data for the model.

- Inertial Measurement Unit (IMU)

  A generic IMU model uses actual accelerometer specific force and rate gyro measurements and outputs either an ideal force and rate data, or incorporates an element of noise to the data.

- Unaided Inertial Navigation System (INS)

  The INS model gives the missile estimates of position, velocity, acceleration, and the rotational equivalents. The model bases its estimates on the outputs from the IMU. The IMU values are integrated and added to the missile's knowledge of its original state to produce the estimates.

- Stick Cone Fuse

  A basic model that provides a ‘warhead enable’ flag if the target is within the cone range.

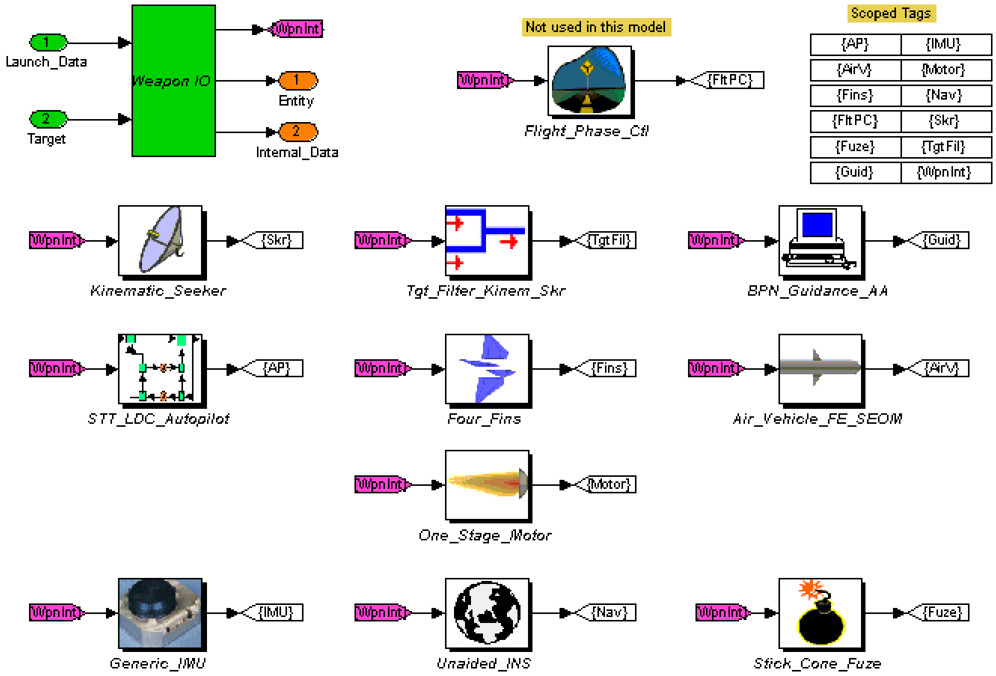


Figure 4.3: Missile Sub-Systems

4.1.2. Simulation Parameters

There are two methods that can used to provide inputs for a simulation. They can either be entered using the MSTARS GUI in Simulink, or they can be entered using the Matlab command line interface, and hence by a script file. The advantage of using a script files is that it can allow for a batch of multiple runs to be processed at once.

4.1.2.1. Simulation Inputs

Inputs to the simulation can be of two parameter types. The first type is that of constant inputs which are kept identical throughout different simulations, such as missile start and end masses, gains for various systems (for example the autopilot system), and manoeuvre acceleration limits etc. The other is of specified inputs that may vary for each simulation. Primarily these will be the starting positions and velocities of the fighter and target aircrafts and will be classified dependent on the engagement scenario to be evaluated.

4.1.2.2. Simulations Outputs

Outputs from each simulation are stored in the Matlab workspace. These are connected to the simulation using standard ‘to workspace’ blocks from the Simulink model library. The data from the simulation that are recorded for analysis purposes are made up of two sets (missile and target) of twenty one parameters, containing position (3), speed (3), acceleration (3), angular velocity (3) and orientation data (direction cosine matrix) (9).

$$[x, V, \dot{V}, \theta, R] = f(\ ) \tag{4.1}$$

The data that is returned from an MSTARS simulation can not be directly used in AGILE as MSTARS uses inertial coordinates for fly-out simulations. As described in Section 2.1, AGILE requires the input parameters to be in GW372 notation. This will require a conversion to take place to generate the thirteen GW372 input parameters for AGILE from the twenty one used by MSTARS.

4.1.2.3. Conversion to GW372 Geometry

The conversion of parameters to the GW372 coordinate system from the inertial system used by MSTARS simulations requires the following steps:

- Initially the 21 parameters for each endgame entity are reduced to fifteen, using the direction cosine matrix to extract the Euler angles φ, θ, and ψ.

- These are then used in a conversion function [27] to calculate the required GW372 parameters. This function finds the speeds and angles required for AGILE to evaluate the probability of lethality.

4.1.3. Fly-Out Scenarios

Fly-out is the progress from the launch of the missile to either detonation or until the missile expends it fuel.

Fly-out scenarios were categorised into three classes: front on, side on and tail on.

For each simulation in a particular category the starting positions and velocities are varied to provide a range of fly-outs that will yield differing endgame conditions to evaluate how lethality probabilities vary. These classes are illustrated in Figure 4.4.

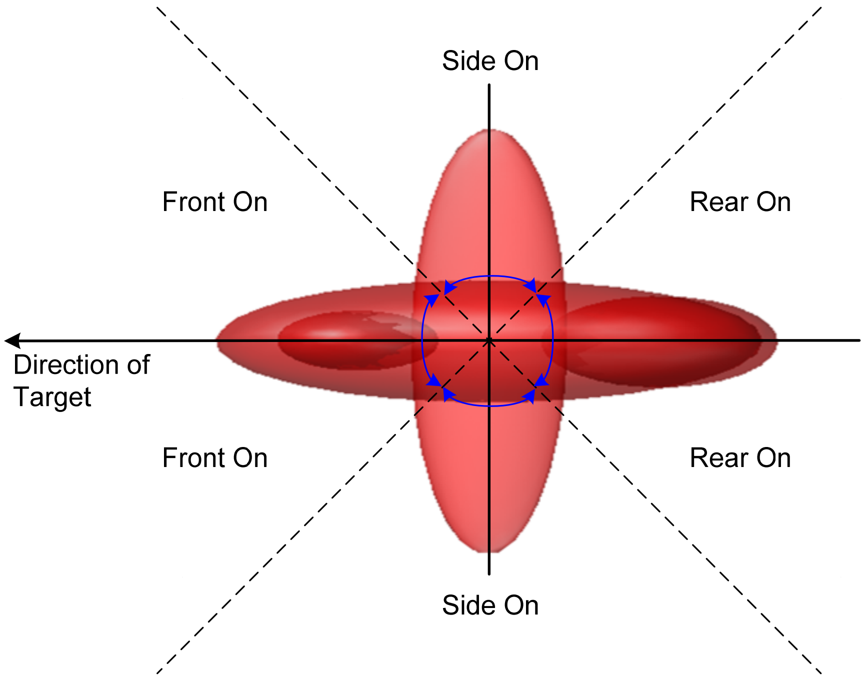


Figure 4.4: Fly-Out Categories

- **Front On Scenario**

  Front on scenarios involve a head-on engagement of fighter craft and target. This occurs when the two aircraft are moving towards each other. This results in scenarios where the engagement angle varies from 135° to 225° (i.e. ±45° from 180°).

- **Side On Scenario**

  A side on scenario occurs when the engagement angle lies in the ranges 45° to 135° and 225° to 315°. This will involve the fighter aircraft moving toward the target from either side of the target.

- **Rear On Scenario**

  Rear on scenarios consists of tail chase conditions of engagement. This arises from the fighter chasing the target aircraft from behind and results in engagement angles of ±45°, or 0° to 45° and 315° to 360°.

4.2. Initial Fly-Out Simulations

Using these three categories, simulations can be run using MSTARS to fly-out a missile and calculate the final lethality probability from AGILE. This evaluates P_k from a fly-out/engagement strategy. This process can be automated such that several simulations can be run as a batch and then post processed to yield the endgame lethality values.

For these batch runs the target initial start point and velocity is kept the same for each run, and the fighter aircraft has its start position and velocity varied randomly within set bounds to generate a set of endgame conditions that can be used for lethality analysis. Not all simulations yield a valid endgame scenario, due to the random nature of the start points of each run, and also due to the simulation time used, of 60 seconds. The runs were categorised into front on, side on and rear on engagements for a fixed altitude for both the attack and target aircraft. For each category, 100 random start points were evaluated to determine the endgame scenario allowing the results to be processed by AGILE and a lethality calculation performed. A summary of these results is given below.

4.2.1. Front On

For a front on set of 100 start points, 71 resulted in a valid endgame scenario, whereas the remaining 29 did not reach the target within the simulation time. It was found that his was due to the start points for the fly out not permitting the missile to reach an endgame condition within the time parameters set for the simulations. Figure 4.5 shows the lethality probability distribution for each of the 71 valid endgames.

From the graph it can be seen that there is a wide range of lethality values ranging from less than 10% all the way up to 100%. This may be a result of the missile fusing with an orientation that is not pointing the warhead's fragment cone appropriately toward the target. Although the missile may be in close vicinity, the orientation of the warhead is important in order to maximise the lethality.

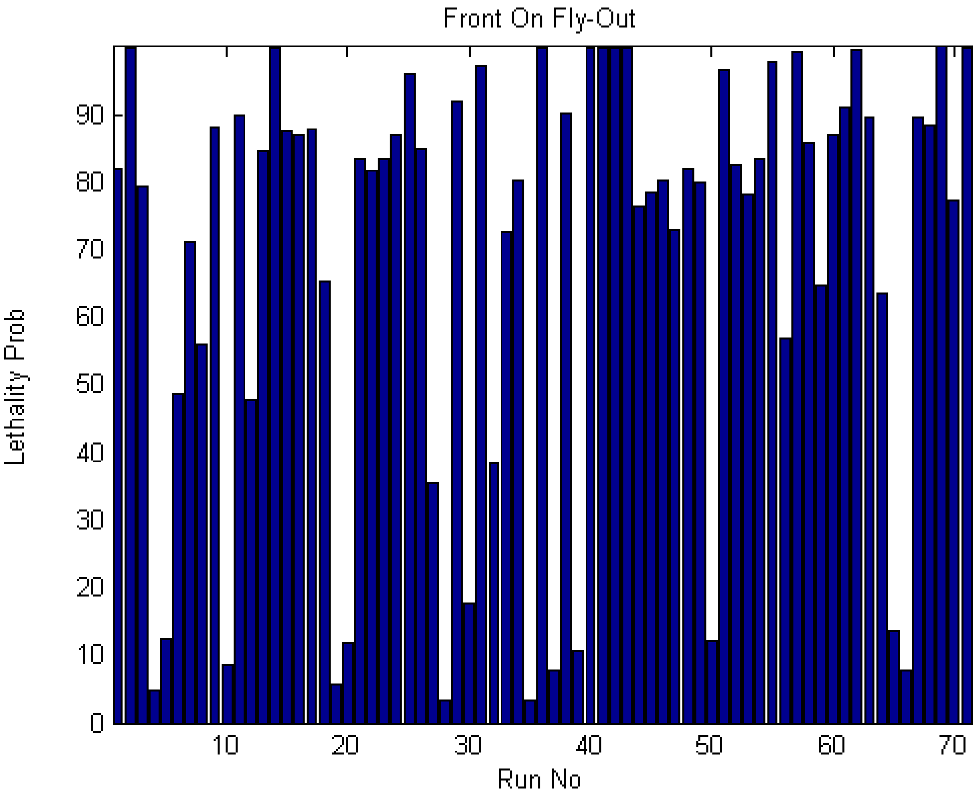


Figure 4.5: Front On Endgame Lethality

4.2.2. Side On

Of the 100 simulation runs performed for side on cases, 58 of these runs resulted in completed simulations, with 42 timing out due to the length of the simulation. The lethality values for these solutions are shown in the graph in Figure 4.6.

As the graph shows there is a wide range in probability values for lethality, from around 35% up to 100%. This is slightly better than the front on cases in terms of average lethality; however fewer runs yielded an endgame condition compared to the front on scenarios.

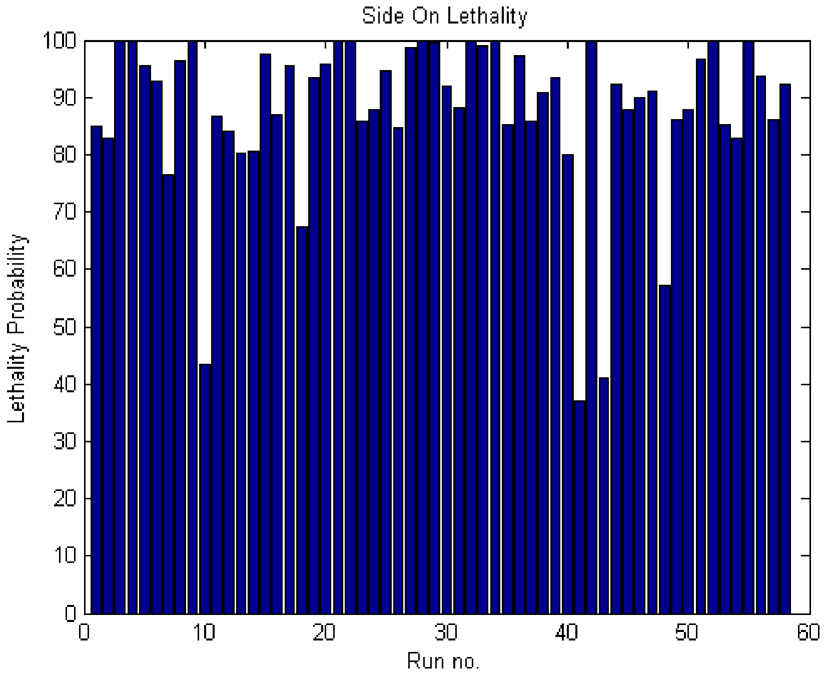


Figure 4.6: Side On Endgame Lethality

4.2.3. Rear On

Similarly, for a rear on engagement scenario a series of 100 simulations were conducted resulting in a total of 54 completed runs, whose lethality is shown in Figure 4.7. It was noted that 46 runs timed out due to the length of the simulations. This was due to the tail chase nature of the start conditions, as for some of these runs the missile did not start in a position from which it could reach the target within the bounds of the simulation time.

From the graph again it can be seen that there is quite a large range of lethality probabilities, from less than 5% all the way up to 100%. This is due to the tail chase nature of the fly out and also the orientation of the warhead cone at fusing.

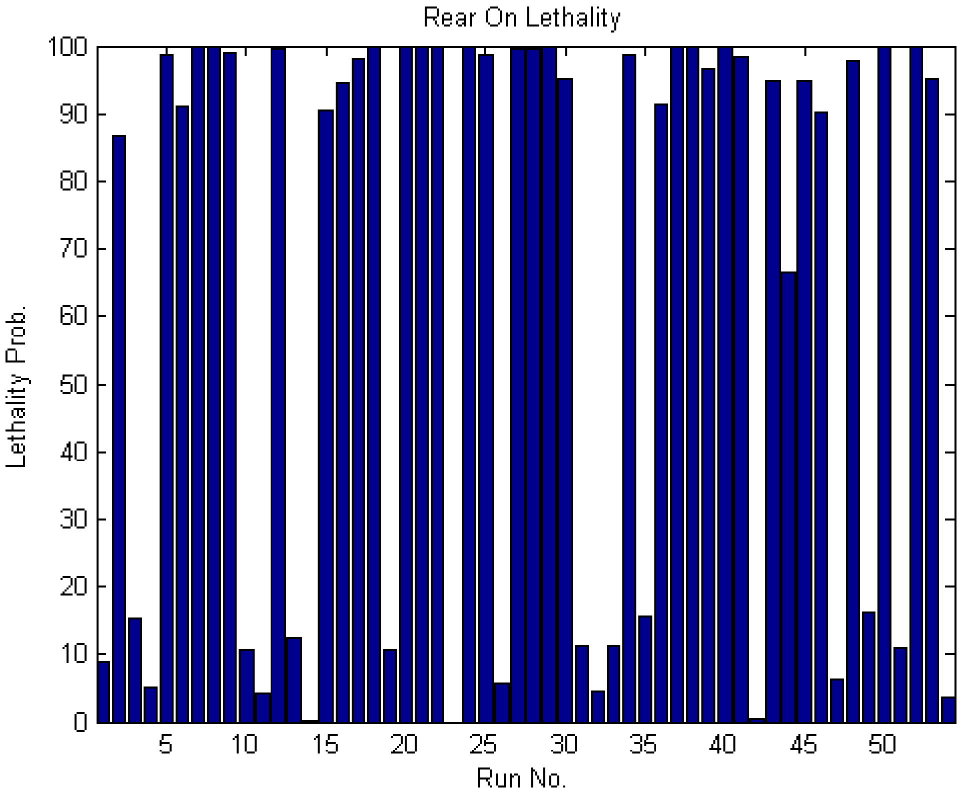


Figure 4.7: Rear On Endgame Lethality

It will be interesting to see how the missile approaches the target and to investigate whether it may have been better to fuse the missile at an earlier point, especially for those simulation runs that yield lower lethality probabilities. This will be examined in the next section

4.3. Study of Fly-Out Trajectory

The missile fly-out data allows for the analysis of the missile flight path in order to verify if the missile engages the target at an optimal point along its path. It is possible to step back along the missile trajectory to extract the GW372 parameters and find lethality probabilities along this path.

For each category the missile trajectory is analysed for the last second of flight. This final second is split into 25 intervals of 0.04s and a lethality calculation performed at each point. The results of this analysis are described below.

4.3.1. Front On Trajectory Analysis

Figure 4.8 shows the estimated lethality for each of the 71 valid endgames in the final second of flight. In the graph it can be seen that a range of conditions occur. For some of these runs the lethality reaches a maximum before the missile engages and as a result of this the lethality probability drops significantly at the end of the simulation. For some others the end lethality is high however it was greater at a point previous to engagement. For others the lethality values stayed high once this high value was attained. Similarly for a few their probability stays low and never increases to a significant level of lethality.

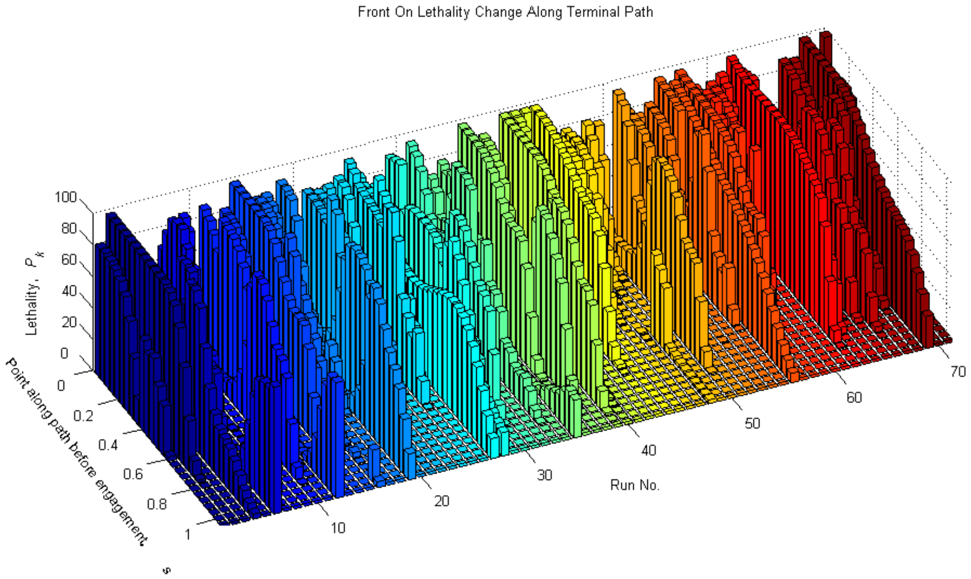


Figure 4.8: Front On Trajectory Analysis

Some of these runs showed higher lethality values prior to the end of the simulation. A sample of these will have their trajectories plotted with some runs that yield high endgames at the end of the fly-out. The trajectories of runs 33, 40, 50, 55, 62, 66, and 67, are plotted in Figure 4.9 with a close-up inset showing the last few points of the fly-out.

Table 4.1 shows the lethality probability data for points along the path. It can be seen from the Table 4.1 and Figure 4.9 that for each run the missile approaches from slightly different directions and each leads to a different lethality probability. The two worst cases of lethality from the subset, runs 50 and 66, both approach from a similar direction and so appear to be part of the same basin of attraction as their respective maximum lethality values occur at the same point along the missile path.

However run 62 has a similar path also but manages to finish with a high lethality. This may mean that there is a second basin very close by that yields higher lethality values, due to the missile fragments intersecting a different, more vulnerable part of the aircraft.

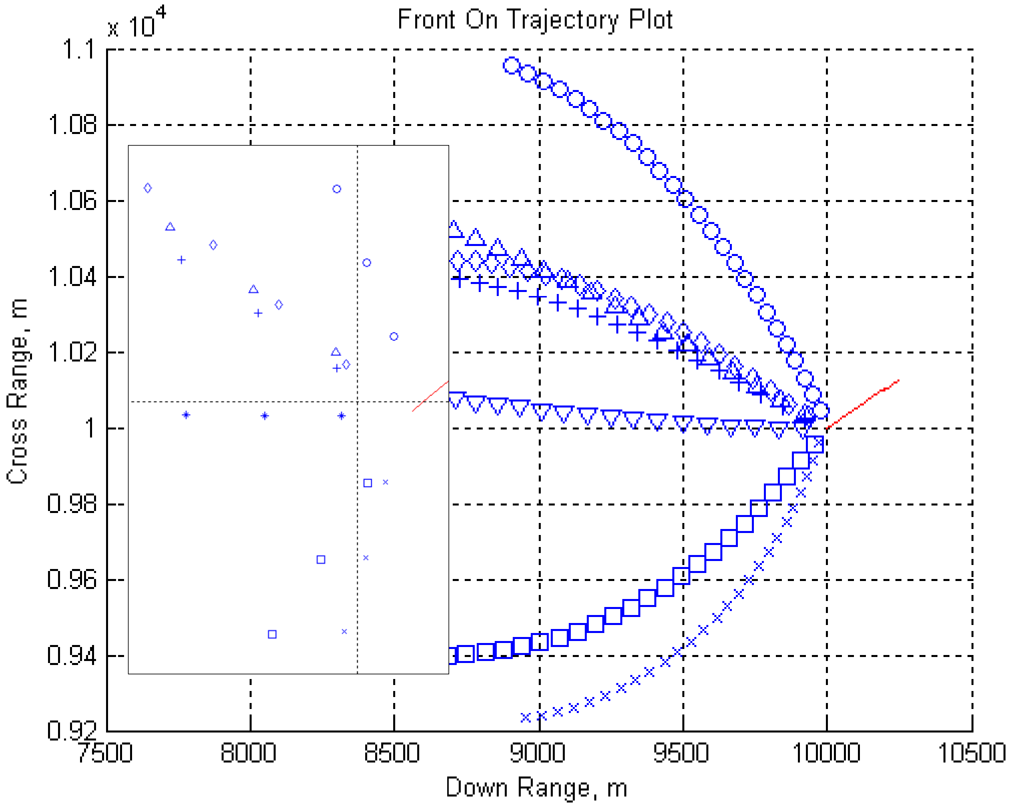


Figure 4.9: Trajectories for Selected Front On Cases.

Run	33	40	50	55	62	66	67
Symbol	x	o	+	□	Δ	◊	*
End Lethality, %	0.7266	1	0.1218	0.9784	0.9966	0.0767	0.8968
Max Lethality, %	0.9998	1	0.9690	0.9883	0.9966	0.9908	0.8968
Time before simulation end of maximum lethality, *s*	0.28	0	0.52	0.4	0	10.52	0

Table 4.1: Trajectory Data for Selected Front On Cases

Further investigation into the endgame conditions of these three runs (50, 62, and 66) determined that the key difference in the endgame parameters was the missile aim points. Both runs 50 and 66 had very similar trajectories and the missile aim points were also close to each other, which resulted in the warhead fragments dissecting the aircraft's wing, thus resulting in a lower lethality value.

It was established that the aim point for run 62 was directed more along the longitudinal, and thus provided a greater intersection of warhead fragments with the target's body, wing and cockpit, resulting in a higher lethality compared to the runs 50 and 66. The geometry of the endgame orientation is shown in Figure 4.10 for runs 50 and 62.

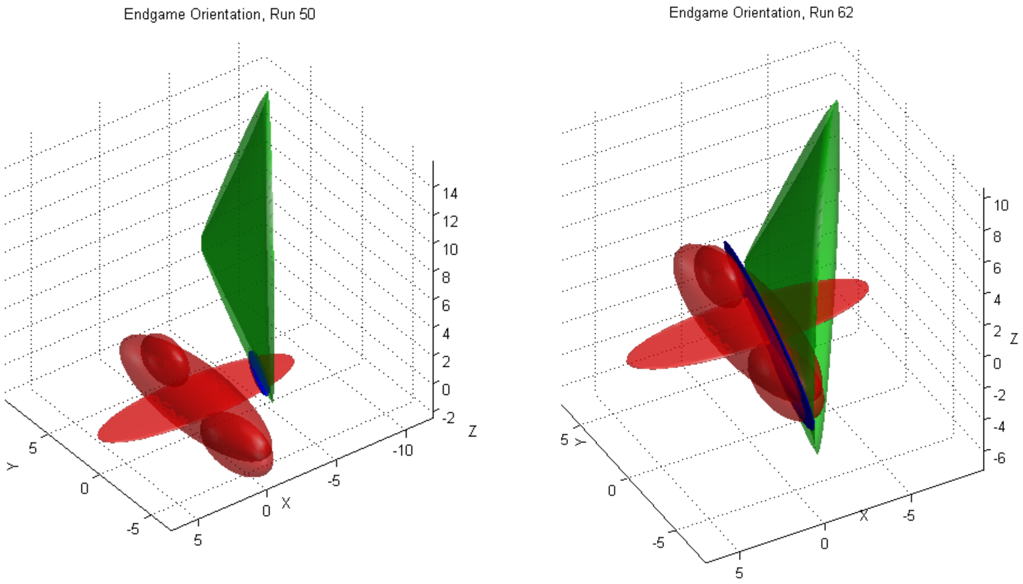


Figure 4.10: Front On Endgame Orientations

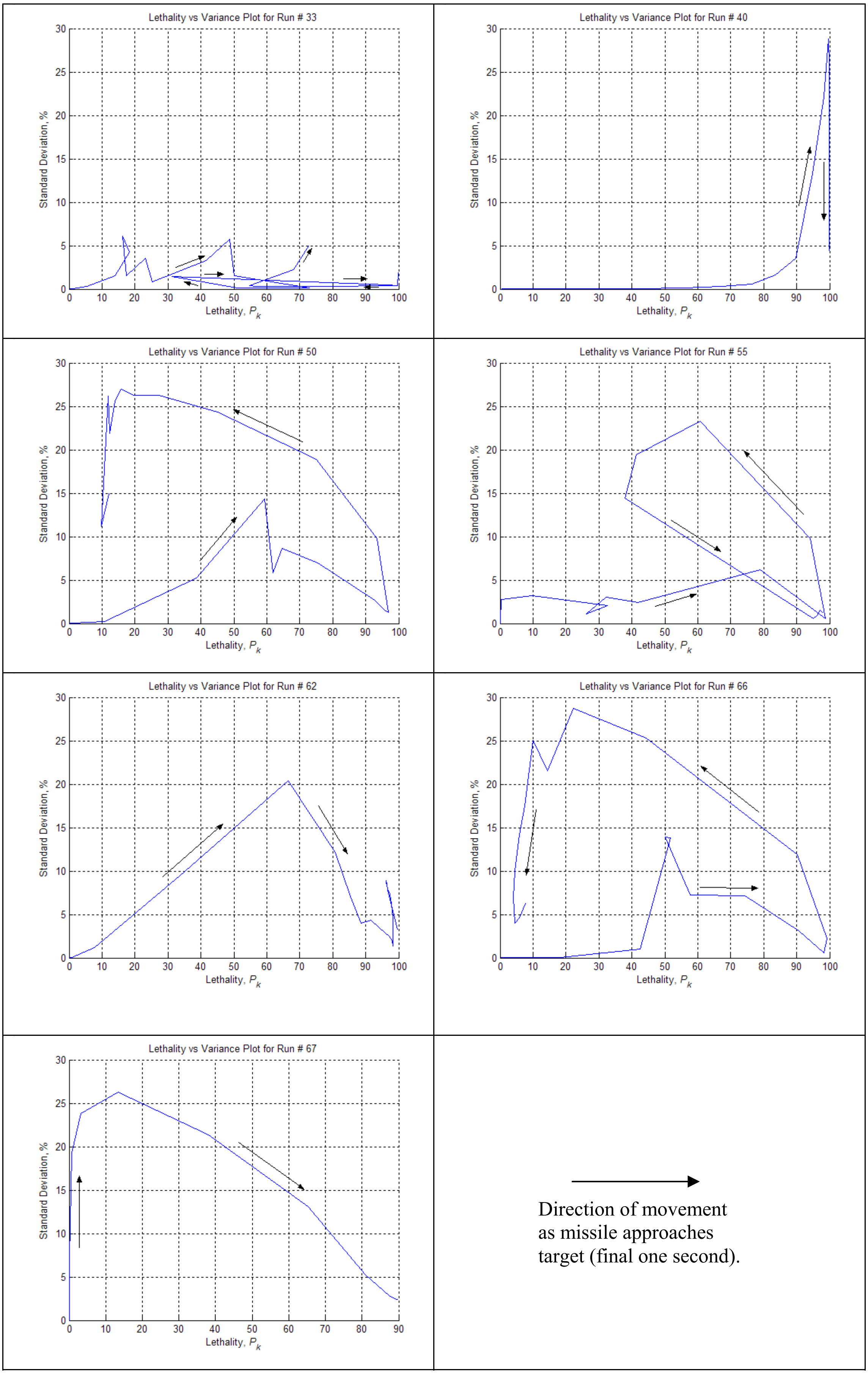
Sensitivity of these results to disturbances can also be measured with a standard deviation calculated by varying the missile aim point parameters to simulate uncertainty in the endgame scenario. Figure 4.11 is a set of scatter plots for the seven runs from above showing how the lethality probability and it associated standard deviation varies for differing points along the trajectory of the final second of fly-out of the missile.

For some cases, for example runs 33, 50 and 66, the lethality increases as one moves back along the trajectory before it falls off, and at this increased lethality point the sensitivity has lowered too. In contrast runs 40 and 62, which yielded an extremely high lethality with a relatively low sensitivity at fusing, maintains a high lethality as initially one moves back along the trajectory path, however the sensitivity increases until the lethality drops down by approximately 10% and the sensitivity is lowered again, before a big drop in lethality. Run 55 at fusing has the desirable high lethality and low sensitivity but loses both these traits before recovering them slightly before a large decrease in lethality. Run 67 fused again in a good position of high lethality and low sensitivity but in just one step the lethality is more than halved and the sensitivity has more than quadrupled, and in another time step the lethality has reduced to close to zero.

Generally, as the lethality decreases the standard deviation tends to increase. This is intuitive because the lethality decreases as it becomes more difficult to inflict sufficient damage to the target and therefore any perturbation of parameters results in the low lethality being diminished further.

For all cases, as the time along the trajectory increases to one second to fusing the lethality has decreased to zero in a region where no perturbation will increase the probability and the standard deviation is zero also.

A good engagement for front-on scenarios occurs when the missile approaches at an angle, allowing the fragments of the warhead to intercept as much of the aircraft as possible.



Direction of movement as missile approaches target (final one second).

Figure 4.11: Sensitivity Measure Along Front On Missile Trajectory

4.3.2. Side On Trajectory Analysis

The results for the 58 completed side on cases from Section 4.2.2 are shown in Figure 4.12. From this graph it can be seen there a small decrease in lethality for runs 10 and 41. Runs 3, 29 and 42 all maintain high lethality values along part of the trajectory, whereas runs 43 and 48 only reach midrange lethality probabilities.

These cases will be investigated further.

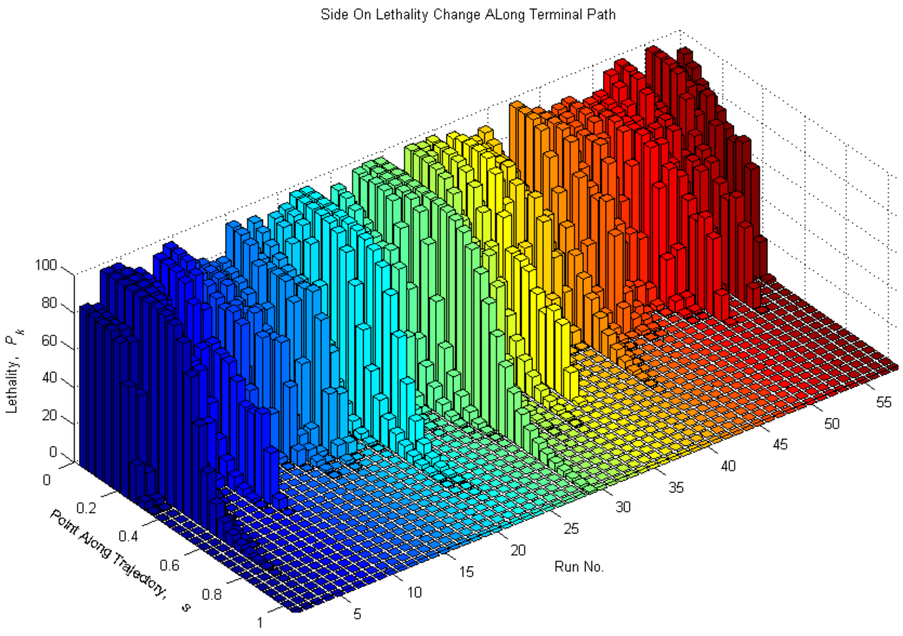


Figure 4.12: Side On Trajectory Analysis

Trajectory analysis on these seven cases is shown in Table 4.2 and Figure 4.13 with a close-up inset. From this data it can be seen that runs 10, 41 and 48 all approach from a similar direction and all result in lethality probabilities that are midrange, and that any further delay may possibly lead to the missile passing the target.

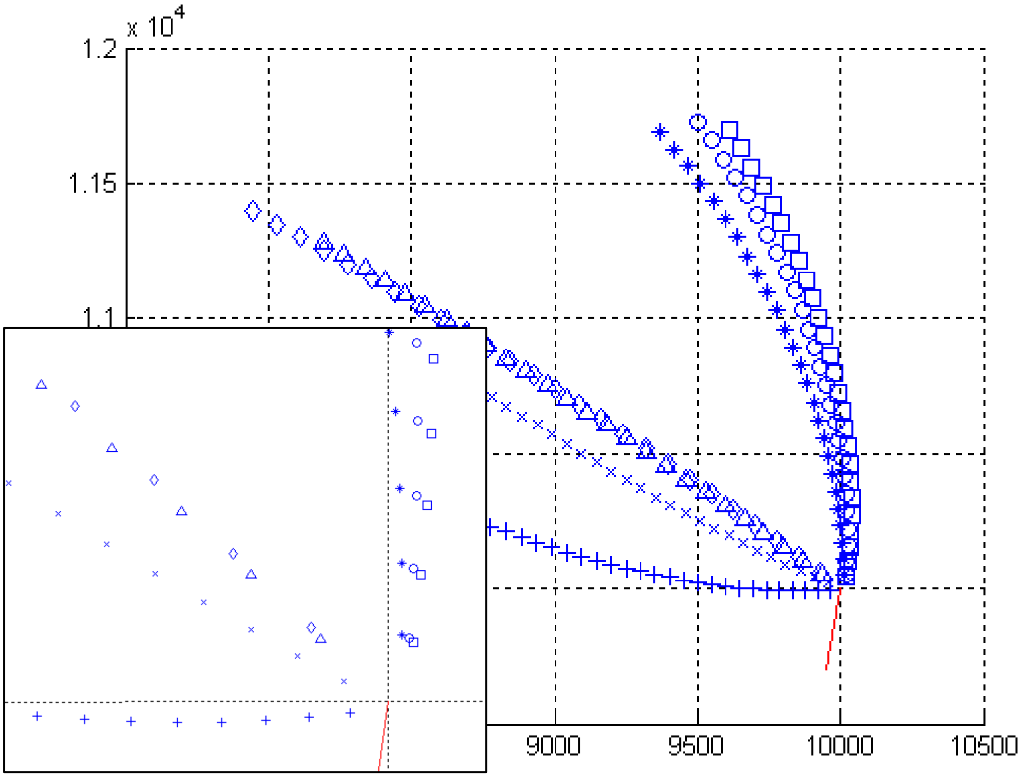


Figure 4.13: Trajectories of Selected Side On Cases

Run	3	10	29	41	42	43	48
Symbol	x	o	+	□	Δ	◊	*
End Lethality, %	1.0000	0.4349	0.9947	0.3709	1.0000	0.4108	0.5717
Max Lethality, %	1.0000	0.4663	0.9968	0.4418	1.0000	0.4108	0.5717
Time before simulation end of maximum lethality, *s*	0	0.25	0.16	0.24	0	0	0

Table 4.2: Trajectory Data for Selected Side On Cases

Of these selected cases, runs 10, 29 and 41 achieve a maximum lethality value prior to the fusing point. Simulation number 29 has a slightly different trajectory to the others but is still a good solution with high lethality.

Run 43 engages the target at a higher altitude, and as a result the warhead fragments impact the target on the wing, resulting in little damage. Its lethality is only 0.41 despite lying on a similar trajectory to runs 3 and 42, whose respective lethality probabilities are both 1. This is due to these scenarios yielding a warhead fragment spread that impacts along the length of the aircraft, thus damaging the fuselage, cockpit and engine. The endgame orientation of runs 42 and 43 are shown below in Figure 4.14.

Simulation number 29 has a slightly different trajectory to the others but is still a good solution with high lethality.

Of these selected cases, runs 10, 29 and 41 achieve a maximum lethality value prior to the fusing point.

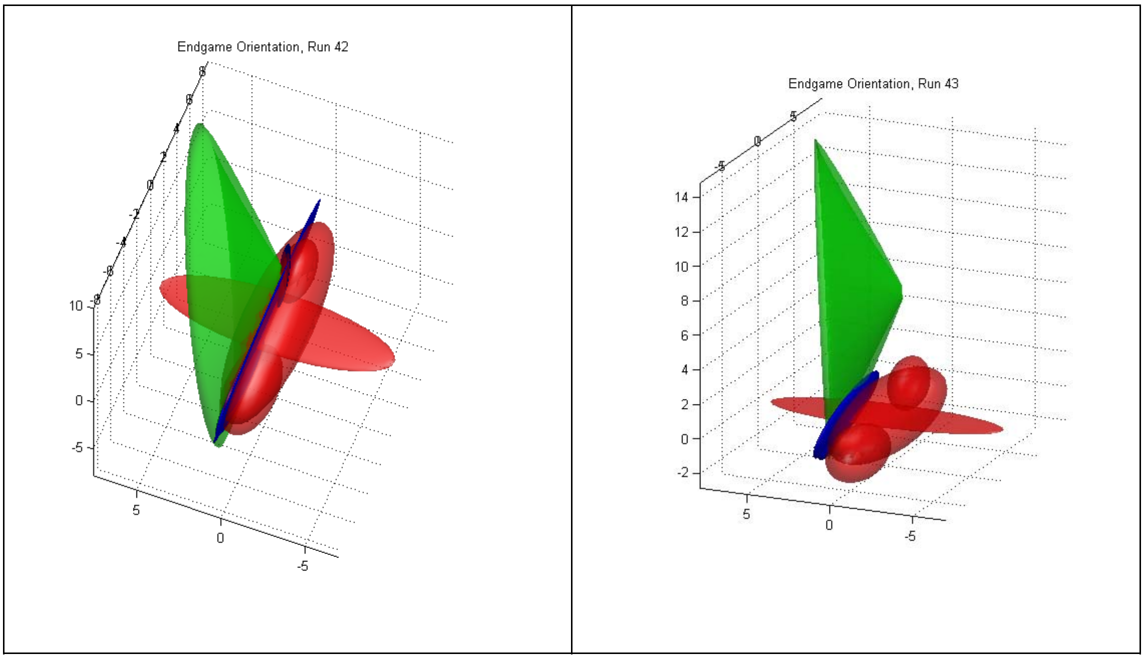
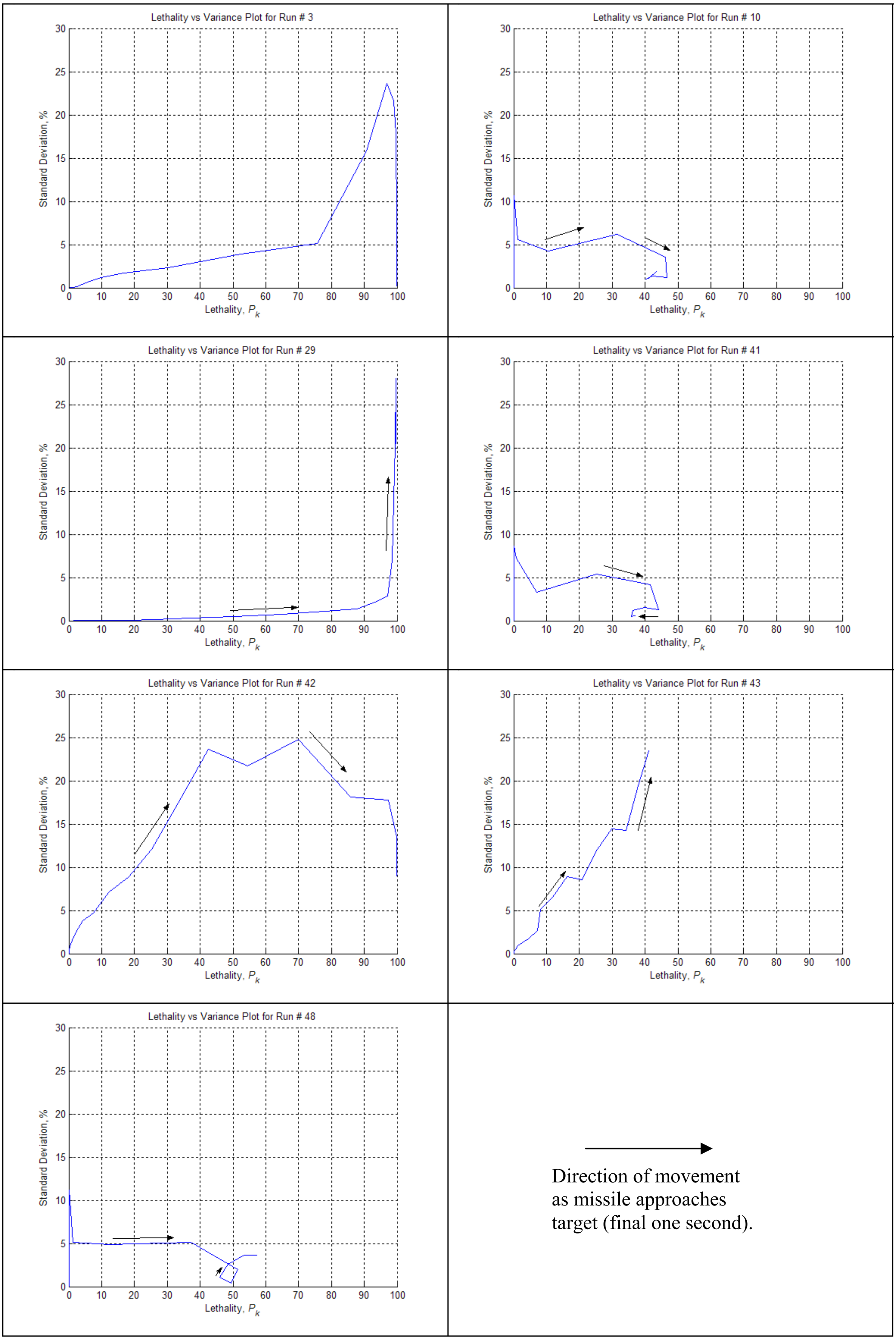


Figure 4.14 Side On Endgame Orientations

The standard deviation value was found for steps along the trajectory by perturbing the missile's parameters and calculating the corresponding lethality variances. Figure 4.15 is a set of scatter plots for the seven runs from above showing how the lethality probability and it associated standard deviation varies for differing points along the trajectory of the final second of fly-out of the missile.



Lethality vs Variance Plot for Run # 3

Lethality vs Variance Plot for Run # 10

Lethality vs Variance Plot for Run # 29

Lethality vs Variance Plot for Run # 41

Lethality vs Variance Plot for Run # 42

Lethality vs Variance Plot for Run # 43

Lethality vs Variance Plot for Run # 48

Direction of movement as missile approaches target (final one second).

Figure 4.15: Sensitivity Measure Along Side On Missile Trajectory

From the graphs it can be seen that for the cases where the lethality found was close to one there are two distinct types of endgame scenarios. Runs 3 and 42 show that as the missile approaches the target, both the lethality and the associated robustness increases until just before fusing at which point the standard deviation of the drops significantly. This suggests that the probability of destroying the target aircraft is virtually certain, and that variations of the missile's position and orientation would have little effect of the damage caused to the target.

Conversely, for run 29 as the missile approaches the target, the lethality increases but the robustness of the lethality decreases, suggesting that a deviation in orientation of the missile would lead to a large drop in damage caused to the target craft. However fusing at an earlier point would have resulted in a more robust solution as the standard deviation increased at the point at which fusing occurred during the fly-out when.

Runs 10 and 48, even though yield midrange lethality, are still fairly robust at fusing, however they drop off very quickly. Both runs 41 and 43 yielded middle range lethality values that drop off as we step back along the trajectory.

Thus, effective endgames, in terms of lethality probability, are those that are not pure side on but those at an angle of incidence similar to that of the front on cases. The more effective endgame lethalities are those that fuse at an angle, as the fragment cone of the warhead are more likely to hit multiple parts of the target, if the target is not moving perpendicular to the missiles trajectory.

4.3.3. Rear On Trajectory Analysis

The results of stepping back along the trajectory of the 54 completed runs of rear on engagement scenarios from Section 4.2.3 are shown in Figure 4.16. There are some simulation runs that yield higher lethality values prior to the final solution, for example runs 1, 6 and 46. Some runs maintain a low lethality, such as runs 10 and 26, and some maintain high lethality values like runs 20 and 50.

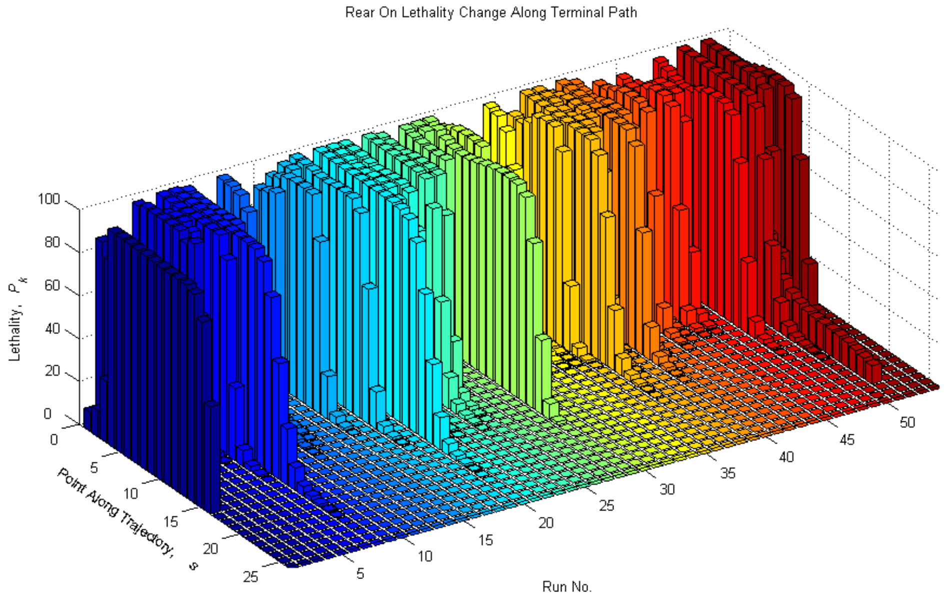


Figure 4.16: Rear On Trajectory Analysis

The trajectories of these seven runs are plotted in Figure 4.17 with a close-up inset, with their corresponding values of maximum and final lethality displayed in Table 4.3.

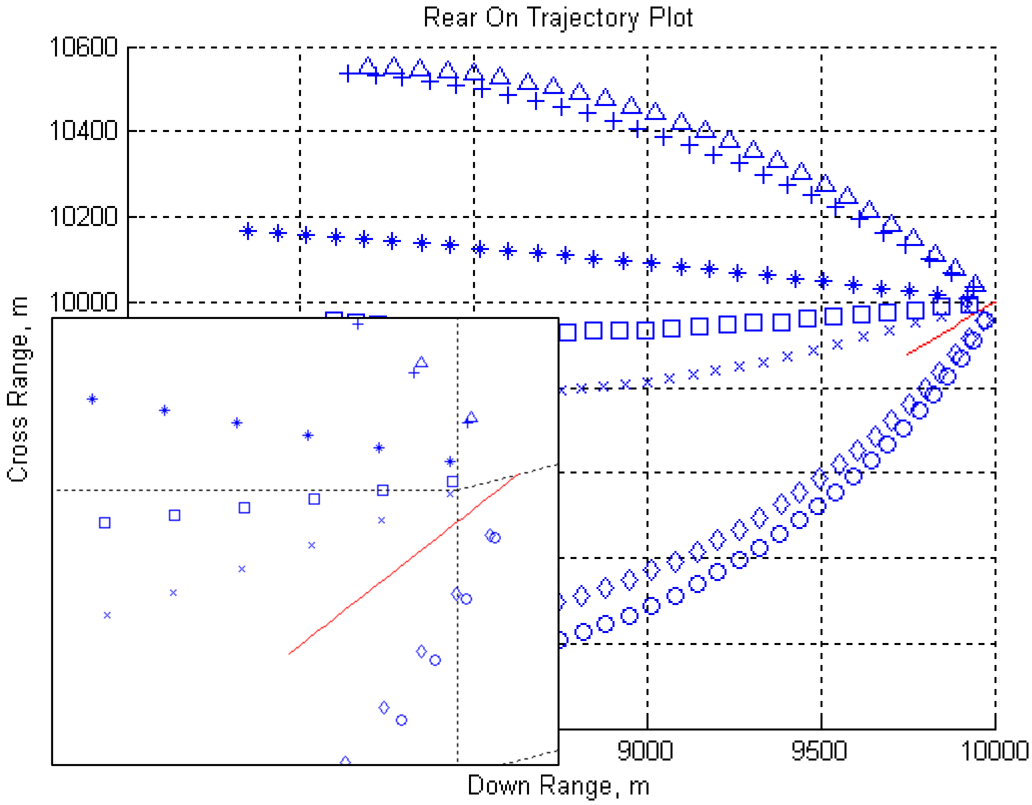


Figure 4.17: Trajectories of Selected Rear On Cases

Run	1	6	10	20	26	46	50
Symbol	x	o	+	□	Δ	◊	*
End Lethality, *%*	0.0888	0.9099	0.1080	1.0000	0.0585	0.9026	1.0000
Max Lethality, *%*	1.0000	0.9981	0.1080	1.0000	0.0585	0.9978	1.0000
Time before simulation end of maximum lethality, *s*	0.12	0.16	0	0	0	0.2	0

Table 4.3: Trajectory Data for Selected Rear On Cases

Runs 6 and 46 appear to be along a similar path and both lead to high lethality probabilities. These two cases also reached a maximum prior to fusing. Likewise runs 10 and 26 appear to approach the target from the same region and both share very low lethality values.

It can be seen that run number 1 yielded a lethality of 0.0888, despite it being calculated as 1 just 0.12s prior to the actual endpoint. The reason for the drop in lethality is that the warhead fragments impacted the nose of the aircraft and narrowly missed the cockpit. Contrary to that, runs 20 and 50 fuse the missile in a similar final position, however the lethality probability is calculated as 1, as the missiles warhead fragments impact the cockpit, a highly vulnerable area. The endgame orientations for run 1and 20 have been shown in Figure 4.18.

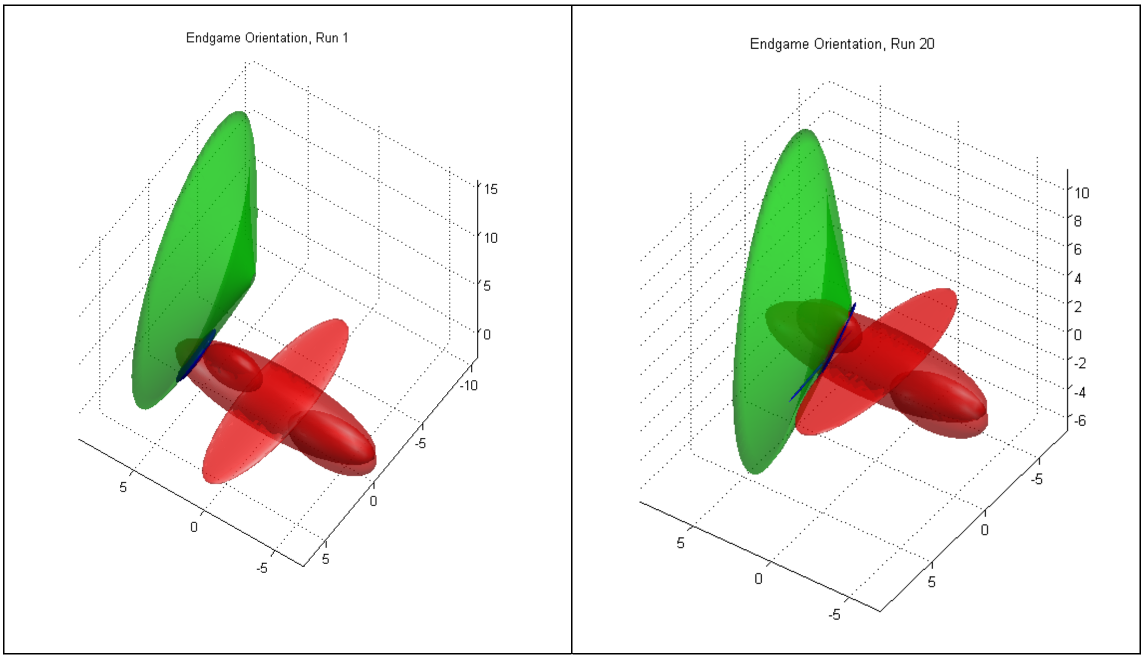


Figure 4.18: Rear On Endgame Orientations

For these seven cases, a sensitivity measure was calculated for points along the trajectory to see how sensitive to disturbances these cases are. These are shown in Figure 4.19. From the graphs it can be seen that both runs 20 and 50 follow similar patterns. At the final fusing point these cases show very high lethality values and very little standard deviation resulting in good solutions. However the standard deviation of these runs tends to increase as the trajectory is pulled back whilst maintaining a high lethality value. This shows that as the missile approached the target the certainty in the lethality increased.

Runs 6 and 46 also follow similar patterns; as the missile approaches the target the lethality increases and the variations increase and then decrease to provide robust endgames, however at the point of fusing the lethality drops by 10%.

It can be seen that for run 1 that fusing 0.12 seconds later resulted in a lethality drop of 0.088 from 1, where when the lethality was high the standard deviation was low. Runs 10 and 26 never appeared to increase lethality and as a result their standard deviations were higher. As the trajectory was pulled further back all cases resulted in near zero lethality probabilities, as would be expected for a tail chase scenario.

Thus, a good endgame from rear on engagements is more difficult to categorise. Missile orientation at fusing is an important factor as the target is moving away and as such the target offers less of a cross-section to the warhead fragments.

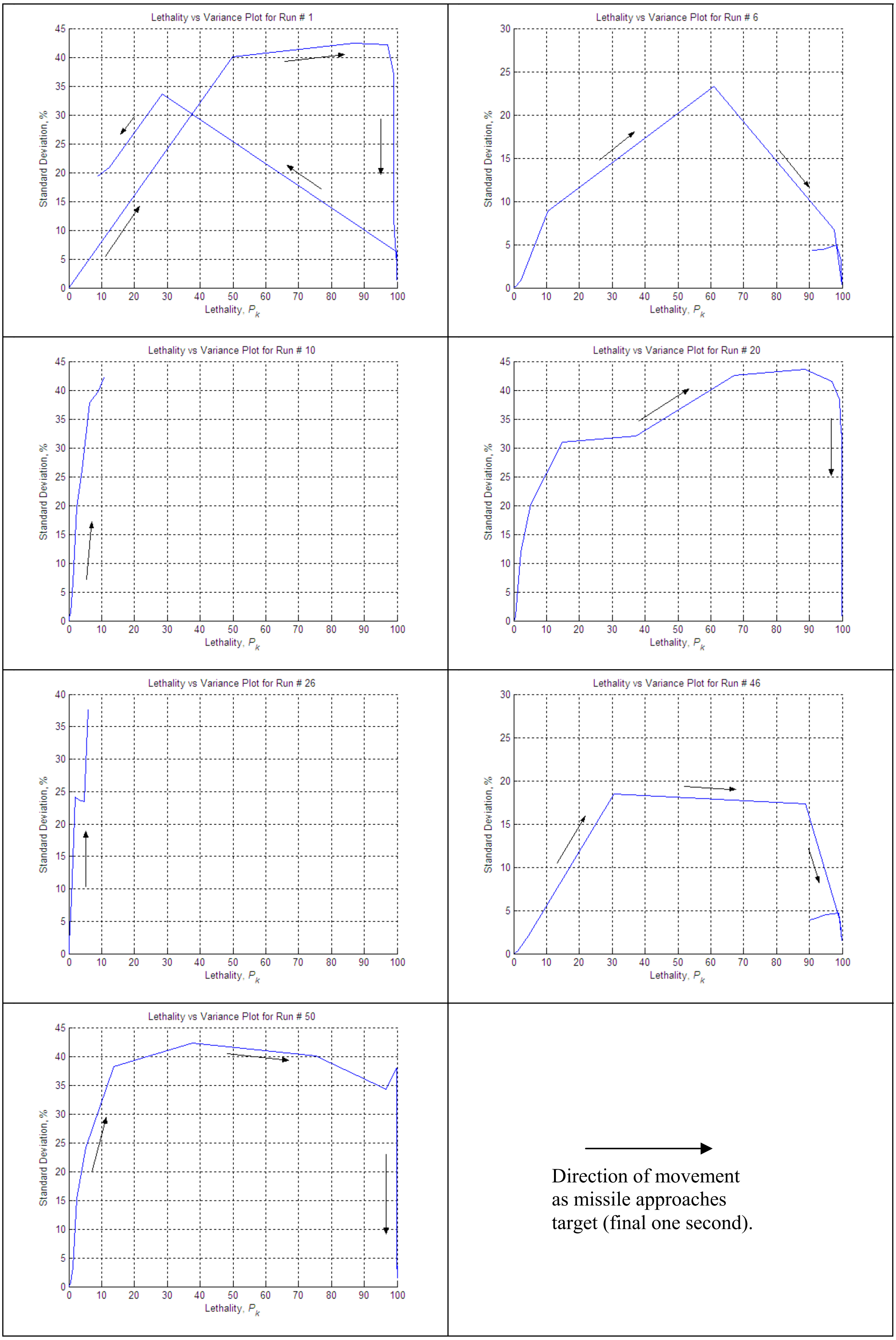


Figure 4.19: Sensitivity Measure Along Rear On Missile Trajectory

This section has described MSTARS, a simulation model for missile fly-out scenarios. It has described the basic components of the model and a set of initial fly-out scenarios have been assembled. These simulations have been compiled into three categories, defined by the direction from which the engagement takes place. It was noted that not all scenarios yielded high endgame lethality values and as a result of this the missile trajectory was looked at to see if it would have been beneficial to fuse the warhead prior to its eventual fusing point. This study found that for some cases fusing earlier would have resulted in much higher lethality values.

Summary

It has been seen that for some solutions it would have been beneficial to fuse the missile at an earlier point along the fly-out in order to produce a higher lethality than the eventual fusing point. It would therefore be beneficial to investigate whether it is possible to reclaim some of the 'lost' lethality by means of optimisation. The missile system has the ability to use warhead 'aim points'. These effectively orientate the warhead cone of fragments in a specified direction. This investigation will be undertaken in Chapter 5.

5. Endgame Optimisation Along Missile Trajectory

This chapter will examine the missile trajectory in more detail. Each point along the trajectory will be examined to see how the lethality varies along the trajectory prior to the fusing of the missile and whether the lethality value can be increased through optimisation. From these studies, potential fusing strategies will be assessed.

5.1. Motivation

Inspection of the fly-out process for the missile system described in Section 4 reveals that on some occasions it may be prudent to fuse the missile earlier in order to increase lethality as the final fusing point may yield a lower lethality value compared to that at a previous point along the missile's trajectory.

By optimising the missile's warhead aim points, which are three of the GW372 parameters defined in AGILE we can explore whether some of this 'lost' lethality can be reclaimed. Other variables will remain as they were during the simulation. Following this optimisation a perturbation analysis will be performed to find the sensitivity of these optimal solutions in order to assess how robust they are.

5.2. Optimisation of Points Along Missile Trajectory

The next three subsections will describe the optimisations performed on the data compiled from the cases that have been discussed previously in Section 4.3. The optimisation that takes place is of the missile warhead's aim points:

$$\max \rightarrow f(x_0, y_0, z_0) \tag{5.1}$$

For each point along the trajectory a calculation of lethality is performed using AGILE. In addition, an optimisation process will be performed on the lethality. The missile aim point parameters, [x_0, y_0, z_0], will be optimised within a limit of ±5m to establish if the lethality can be increased from the missile's current position.

5.2.1. Front On Cases

Taking the trajectories of the front on cases discussed earlier we can see from the graphs in Figure 5.1 that by optimising the missile aim point parameters it is possible to increase the lethality for points along the final second of fly-out of the missile's path. Overlaid on the graphs is a line plotting the miss distance of the missile to the target. The miss distance is the separation between the missile and the target for points along the missile fly-out.

In the graphs in Section 4.3, relating to the missile trajectory, the *x*-axis shows points along the missile's flight path for $x = 0$ to 1 seconds before fusing, moving backwards in 0.04s time steps. The *y*-axis shows the lethality probability as a percentage, and also the miss distance in metres. The actual lethality, as calculated using the missile fly-out data is shown in green, blue is used for the optimised lethality for each point, and the red line plots the miss distance of the missile as it travels towards the target craft.

These results for the font on cases can be generalised into the following groups:

- Cases where the miss distance is decreasing gradually during fly-out and lethality increases as the missile approaches the target, as shown by runs 40, 55, 62 and 67. In such cases optimisation will allow the missile to fuse earlier and still reach peak lethality values.

- Cases whereby the missile moves towards the target on a steeper gradient and then begins to move away, due to the angle at which the missile is flying the target, resulting in lower lethality at the burst point compared to if the missile had fused earlier, as seen in runs 33, 50 and 66. This loss in lethality can be restored number of ways:
    - Without any optimisation this can be achieved by fusing earlier, i.e. at lowest miss distance,
    - Or with an optimisation to regain some lethality that has been lost,
    - Or by utilising both an optimisation and earlier fusing if it appears that the lethality is decreasing rapidly.

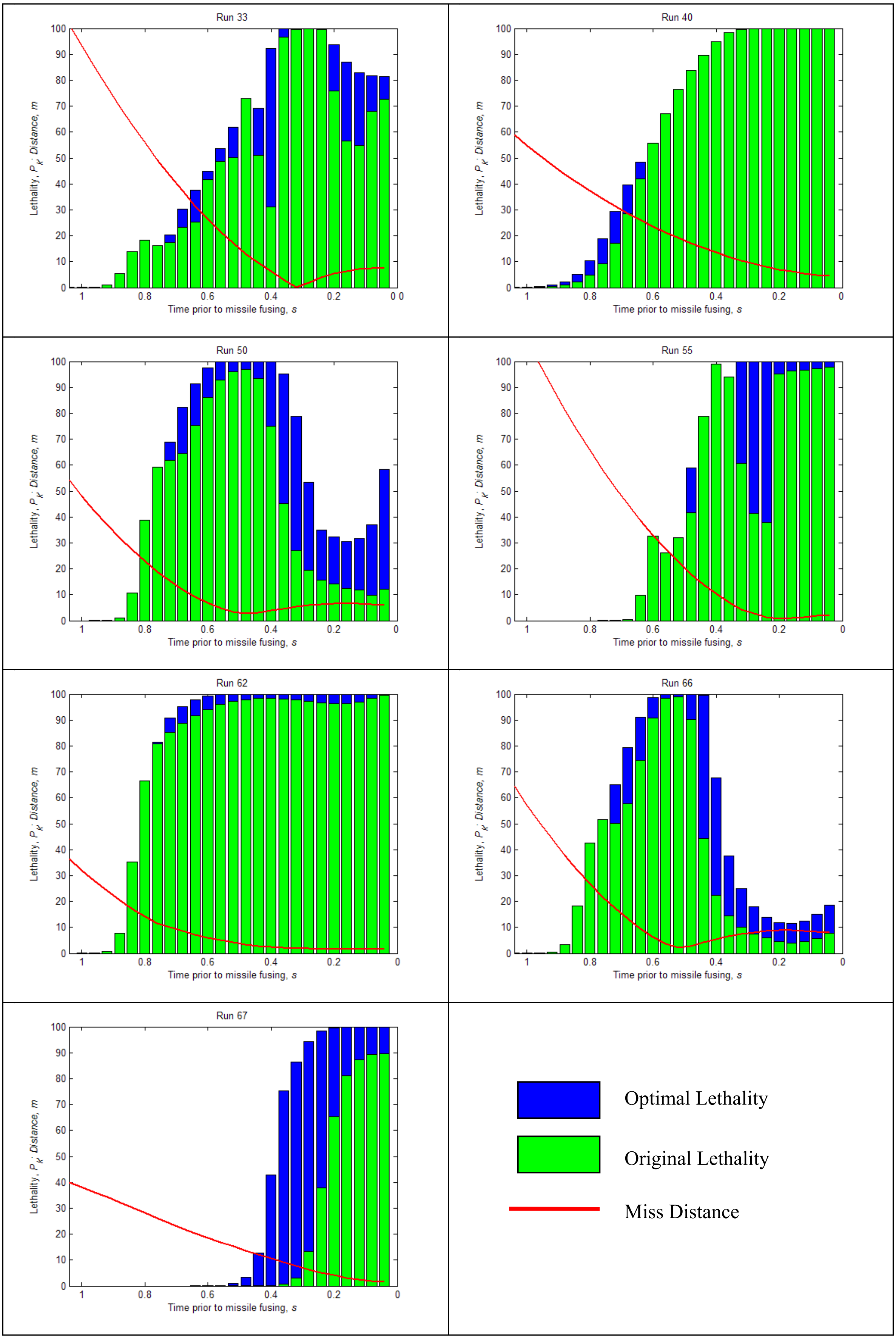


Figure 5.1: Optimisation of Trajectories of Front On Cases

The optimised solutions may not be robust in terms of how the lethality varies from changes in the lethality parameters, if the missile is not able to achieve the optimal orientation. An ideal solution would be insensitive to variations in the missiles orientation so that disturbances are not a factor in reducing lethality by a significant amount

The graphs in Figure 5.2 again show the actual and optimal lethality values and the miss distance, and overlaid on these are three lines depicting the robustness of the optimal solution found: one showing the mean lethality determined from perturbing the optimal parameters, and also two lines showing regions one and two standard deviation away from this mean, representing approximately 68% and 95% of the sampled deviations taken.

It can be seen in the graphs, that for cases such as run 50, the mean of perturbed variables about the optimal parameters is higher than the optimal. This is because the optimisation that takes place has bounds on the missile aim point variables and if the boundary occurs at a point where the lethality is increasing then the optimal parameter will lie on this boundary. Therefore sampling around this point will lead to some samples that are at points outside the initial optimisation boundary which will result in higher lethality values and thus yield a higher mean. In fact if the optimisation boundary was increased (or even unbounded) then the optimal solution would yield much higher lethality values than those encountered. However this is not practical in a physical sense, in that the aim is to improve the missile's performance in its current spatial position instead of looking for a new position for the missile to be in.

It is interesting to observe how the lower bounds (68% and 95%) vary for these scenarios. For the cases that have the minimum miss distance prior to the end fuse point (runs 33, 50, 66) it is also the most robust about this point.

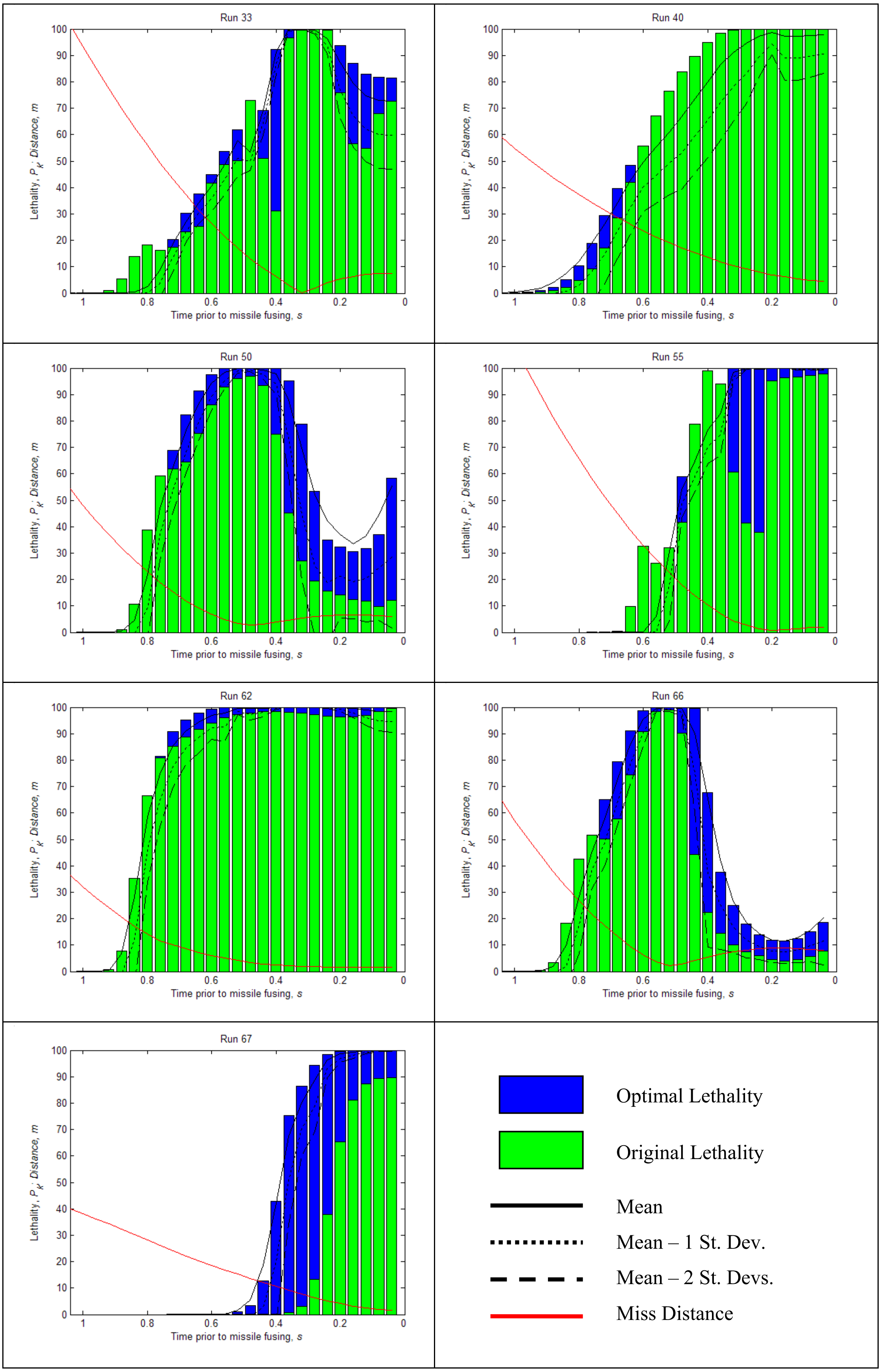


Figure 5.2: Robustness of Front On Optimised Cases

For runs 40 and 62, both of which have high lethality values and require little optimisation, the robustness values are more desirable a few steps before fusing. The robustness begins to increase even though the lethality remains the same. This suggests that again it would be better to fuse slightly earlier.

Runs 55 and 67 shows that even for large lethality gains by optimisation the results have low variations, and show that it is possible to fuse earlier.

It can be seen from the sample orientations in Figure 5.3, that for run 67, at a point 0.2 seconds prior to the missile's original fuse point, the variation in the lethality from the actual position (left) and optimal solution (right) is due to the warhead being aimed in a manner that results in a greater number of warhead fragments to impact the fuselage.

It can be seen that the best solution is the one that provides the most robust optimised solution as this would be the solution that would be able to yield the most damage to the target, even if the actual endgame parameters are not met. From the front on cases it can be seen that the beast case is thus run 62, as it the most robust.

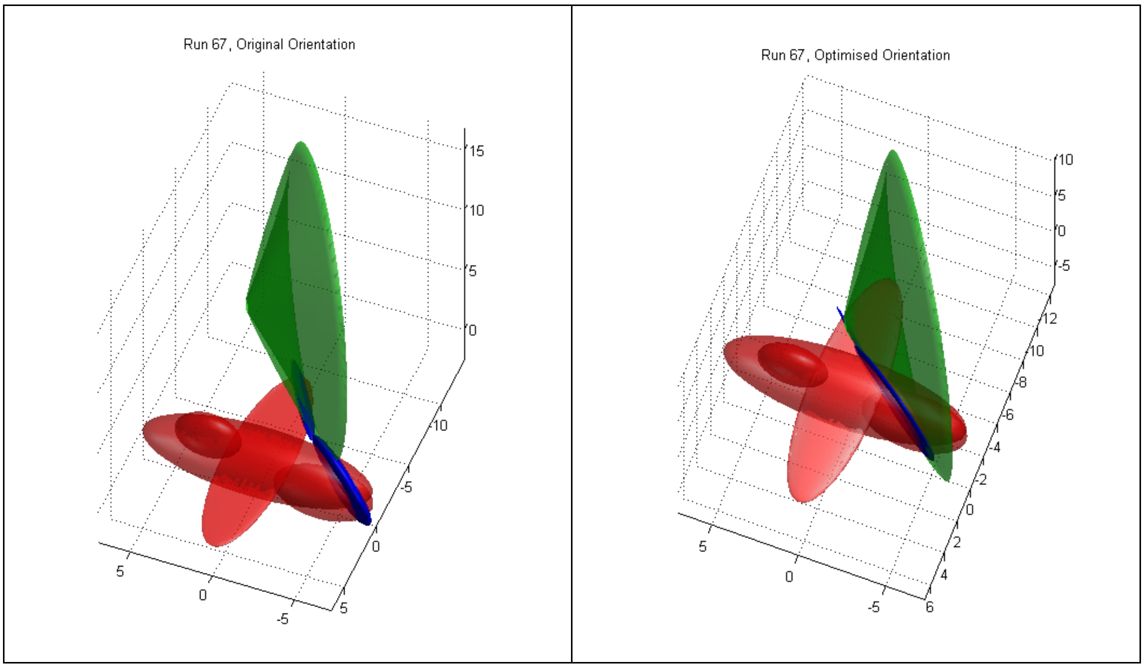


Figure 5.3: Original and Optimal Missile Orientation

5.2.2. Side On Cases

By optimising the missile aim points for the trajectories of the cases examined previously we can see if lethality can be improved, especially for cases where lethality is lost in the final few milliseconds before fusing. The results are shown below in Figure 5.4.

From these graphs for side on cases it can be seen that they all seem to follow a similar path in terms of the rate of change of miss distance. All of these cases show that the miss distance decreases continually and the missile does not move away from the target. However, the steepness of the approach and the final miss distance does affect the final lethality. For steeper approaches the lethality is not as high at fusing compared to the lethality for cases with a shallower approach, and those cases that fuse at a distance greater than 10m show a much lower lethality (runs 10, 41 and 48) compared to the cases that fuse much closer to the target (runs 3, 19, 42, and 43), as shown by the red lines on the plots.

These cases can be categorised into three groups,

- Fly-outs that once they have high lethality probabilities continued to stay high, like runs 3, 29, and 42,
- Cases that can provide mid-range lethality values but can be optimised to give good results, such as run 43,
- And those that yield low to mid-range results even after optimisation, such as runs 10, 41 and 48.

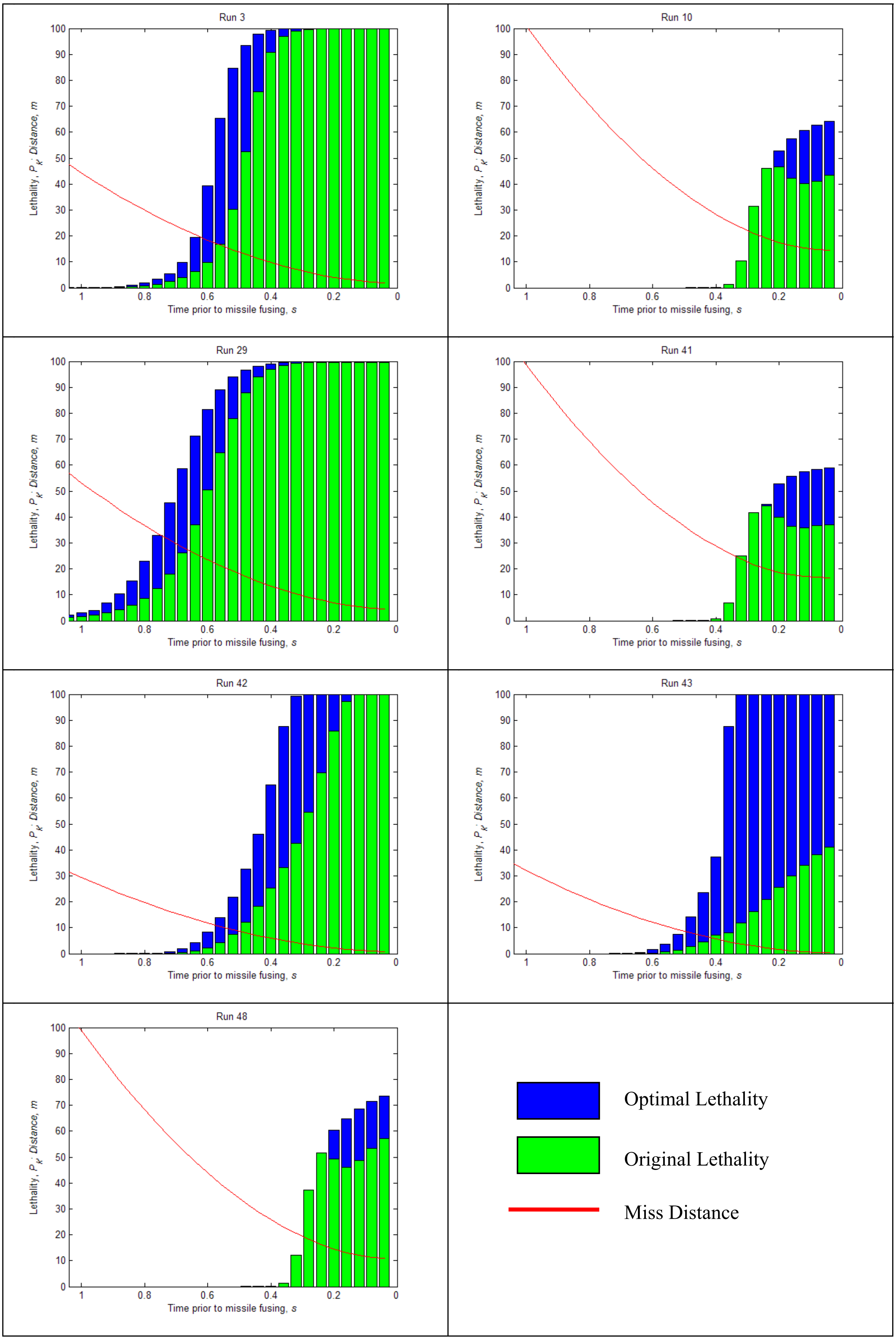


Figure 5.4: Optimisation of Trajectories of Side On Cases

The robustness associated these optimisations is shown in the plots in Figure 5.5. This measure will provide a good idea of when it is useful to fuse the missile earlier in terms of the confidence that the optimised lethality will not be affected by variations in the aim points of the missile. For example run 43 showed high gains in lethality but it may be the case that the increase is very sensitive to the missile achieving the exact optimal parameters and that any slight variation degrades the lethality back down to a lower value.

Run 3 provides the most robust solution at the fusing point as the lower bound of 2 standard deviations from the mean remains as high as the optimal. In addition to this case, runs 29 and 42 which both achieve very high lethality values, show that if the miss distance is decreasing then it is prudent to wait for fusing the warhead as this tends to increase the robustness of the solution.

Of these three cases run 42 has an interesting property when the optimal lethality is large. The confidence associated with the robustness of the optimised parameters shows that even when the lethality gain is large the robustness, even at two standard deviations below the mean, is above the initial lethality where the optimisation began.

In contrast for run 3, the two standard deviation bound lies close to the original lethality, and therefore using the optimal parameters may increase the probability of lethality, it will not hinder it by falling below the initial probability level.

For run 29 however, the two standard deviations bound falls below even the original lethality so would be a less ideal solution as there is potentially a much larger chance of lethality drop should the optimal condition not be met.

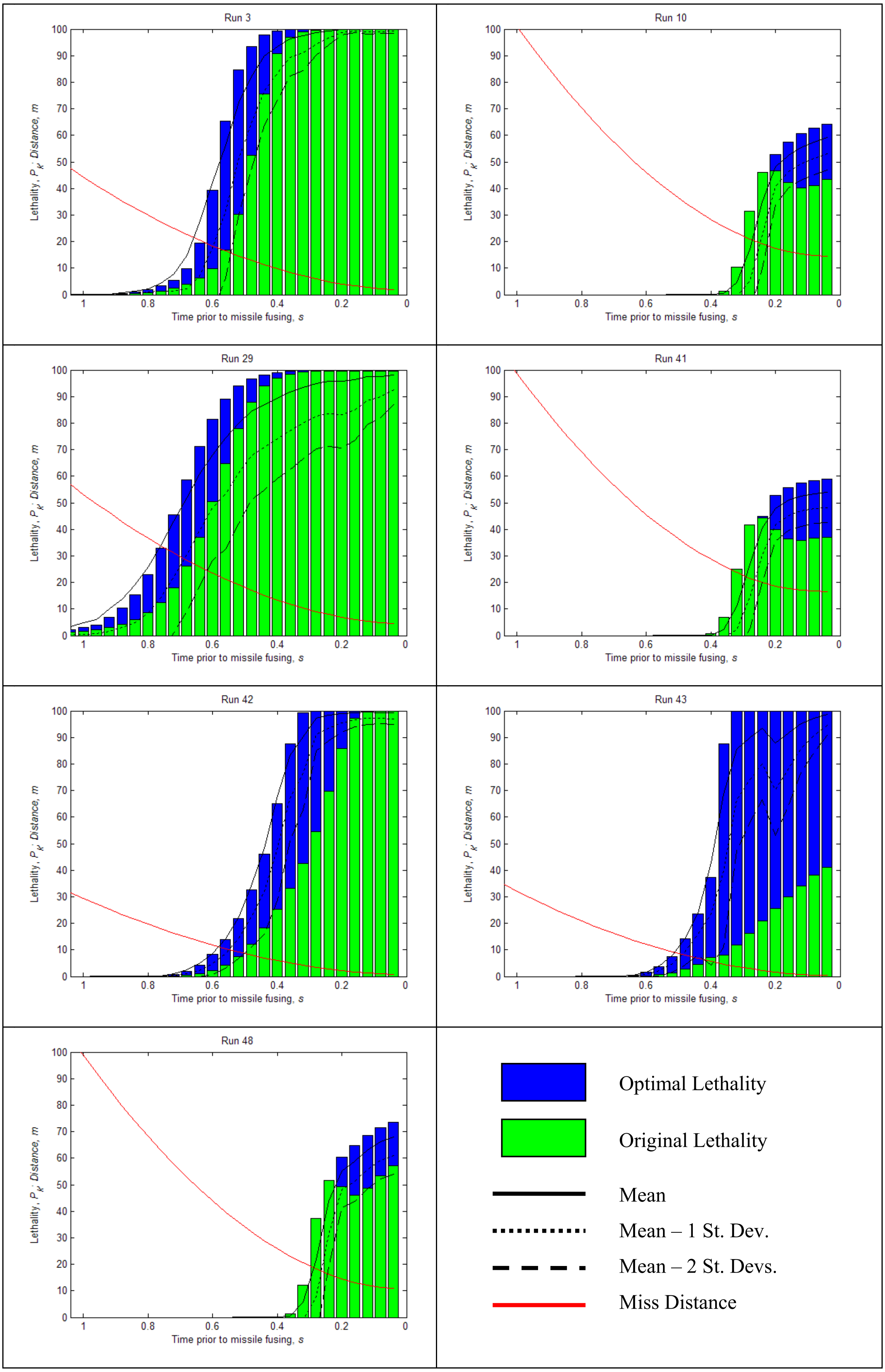


Figure 5.5: Robustness of Side On Optimised Cases

The case with the most improved lethality is run 43, which from 0.3s prior to fusing optimised to near perfect lethality from initial points of 12% up to 40% at actual fusing. The robustness varied greatly to during this phase of the fly-out, however at the end the optimal solution provide a robustness value similar to those of run 29, which has a more desirable fly-out in terms of lethality levels during the fly-out.

Runs 10, 41 and 48 all follow a similar pattern and despite them providing an increase in lethality, it is not a great increase as seen in the other. This is mainly due to the miss distance; this missile appears to be arcing too soon and results in the minimum distance to too great. It can be seen that for run 48 at a distance of 10m the optimised lethality is 74% compared to run 42 which at the same distance away only has a lethality of 22%, however the latter closes the distance and as result finishes in better position.

It can be seen that the best solution is the one that provides the most robust optimised solution as this would be the solution that would be able to yield the most damage to the target, even if the actual endgame parameters are not met. From the side on cases it can be seen that the beast case is thus run 3, as it the most robust.

The orientation of the endgame for run 43 is shown in Figure 5.6. It can be seen that the reason for the optimised orientation providing a much higher lethality is because the warhead fragments impact the aircraft's fuselage as well as the wing, whereas in the original orientation only the wing is clipped.

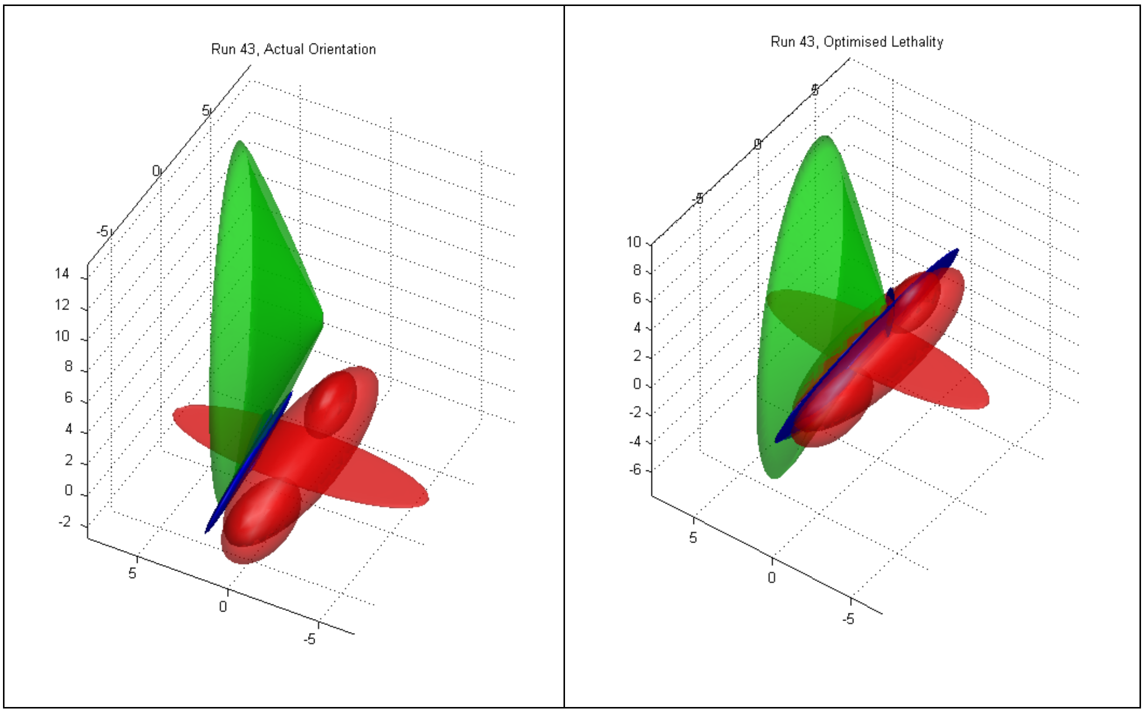


Figure 5.6: Side On Orientations

5.2.3. Rear On cases

The results for the optimisation of the trajectories for the rear on fly-out cases investigated in Section 4.3 are shown in Figure 5.7. The initial and optimal lethality probabilities are shown by the green and blue bars, and the missile miss distance by the red line.

As can be seen there are also some trends that are followed by these results that can be classified into three basic groups based on how the lethality varies during the last second of fly-out prior to the missile fusing. For all cases, as the missile approaches from the rear the miss distance is always decreasing and, as for the side on cases, the steepness of the approach affects the level of lethality.

- Lethality probability increases and stays high, and by optimising allows the fusing to take place earlier, for example runs 20 and 50,
- Lethality increases but then drops off but can be kept high by optimisation to recoup the loss or by fusing earlier, as shown in runs 1, 6, and 46,
- Lethality is very low at fusing point, but can be dramatically increased to very high levels by optimisation, seen for runs 10 and 26.

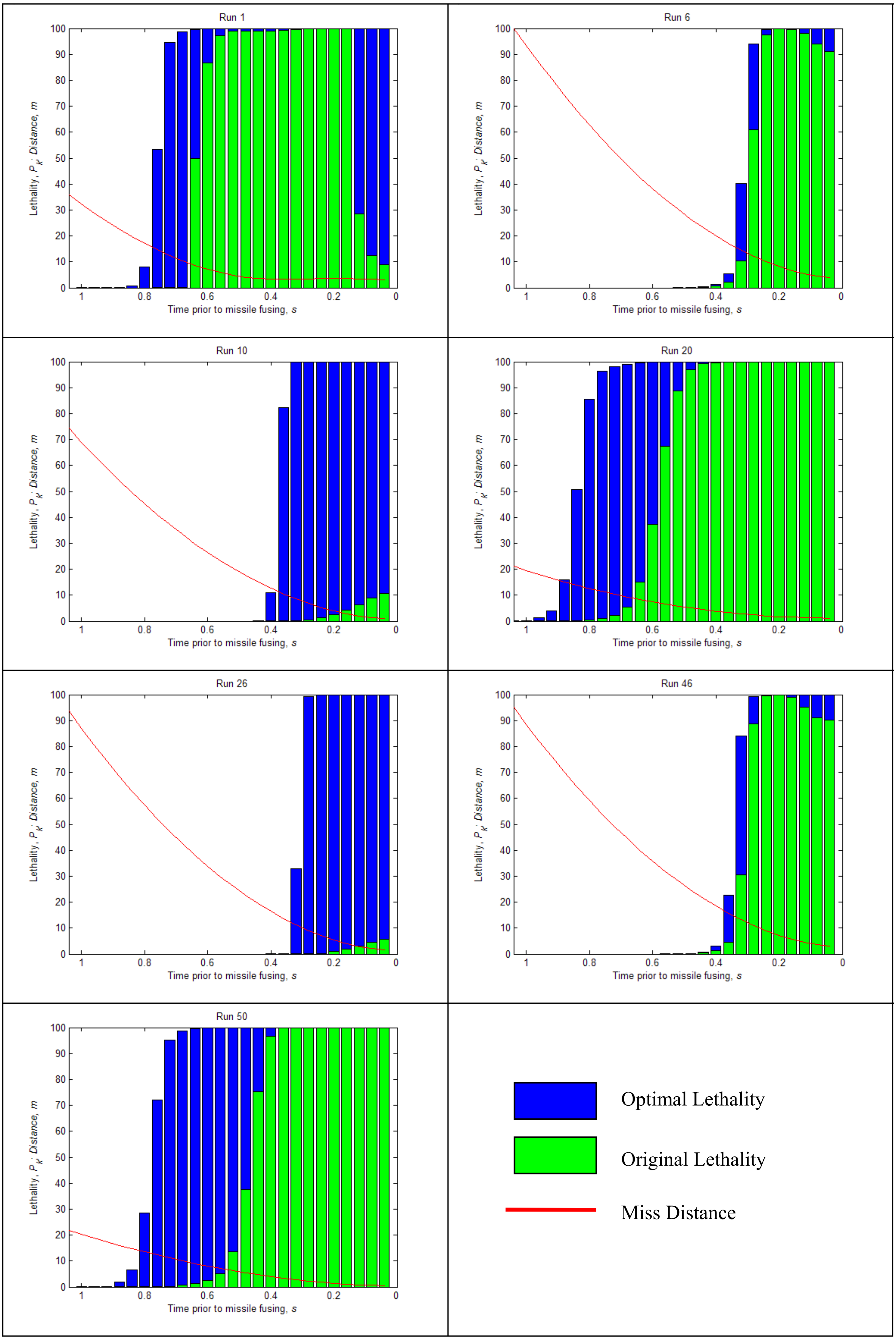


Figure 5.7: Optimisation of Trajectories of Rear On Cases

The robustness of these solutions was evaluated, in particular, the two cases that showed a large improvement in lethality. Also the means and 68% and 95% bounds vary along the final second of the fly-out was calculated in order to see if fusing could take place earlier.

The plots are shown in Figure 5.8, with the initial and optimal lethality values shown as bar graphs, the associated mean value plotted as a red line, and the lower bounds shown using black dotted (68%) and solid (95%) lines.

From the graphs it can be seen that for all the cases the mean and lower bounds remain very high for the optimal lethality found at the fusing point. It can also be seen for all cases that there is a point where the mean of the perturbed optimal parameters is higher than the optimal. This is because the optimisation that takes place has bounds on the missile aim point variables and if the boundary occurs at a point where the lethality is increasing then the optimal parameter will lie on this boundary.

Therefore sampling around this point will lead to some samples that are at points outside the initial optimisation boundary which will result in higher lethality values and thus yield a higher mean. In fact if the optimisation boundary was increased (or even unbounded) then the optimal solution would yield much higher lethality values than those encountered. However this is not practical in a physical sense, in that the aim is to improve the missile's performance in its current spatial position instead of looking for a new position for the missile to be in.

For the two cases that showed the most improvement, runs 10 and 26, they both have a slight decrease in the confidence of robustness at points along the fly-out, after reaching a high level, resulting in a potential drop in lethality of 10% and 25% respectively if the optimal parameters are not achieved. Run 50 shows a similar trait to these cases at 0.3s before the fusing point.

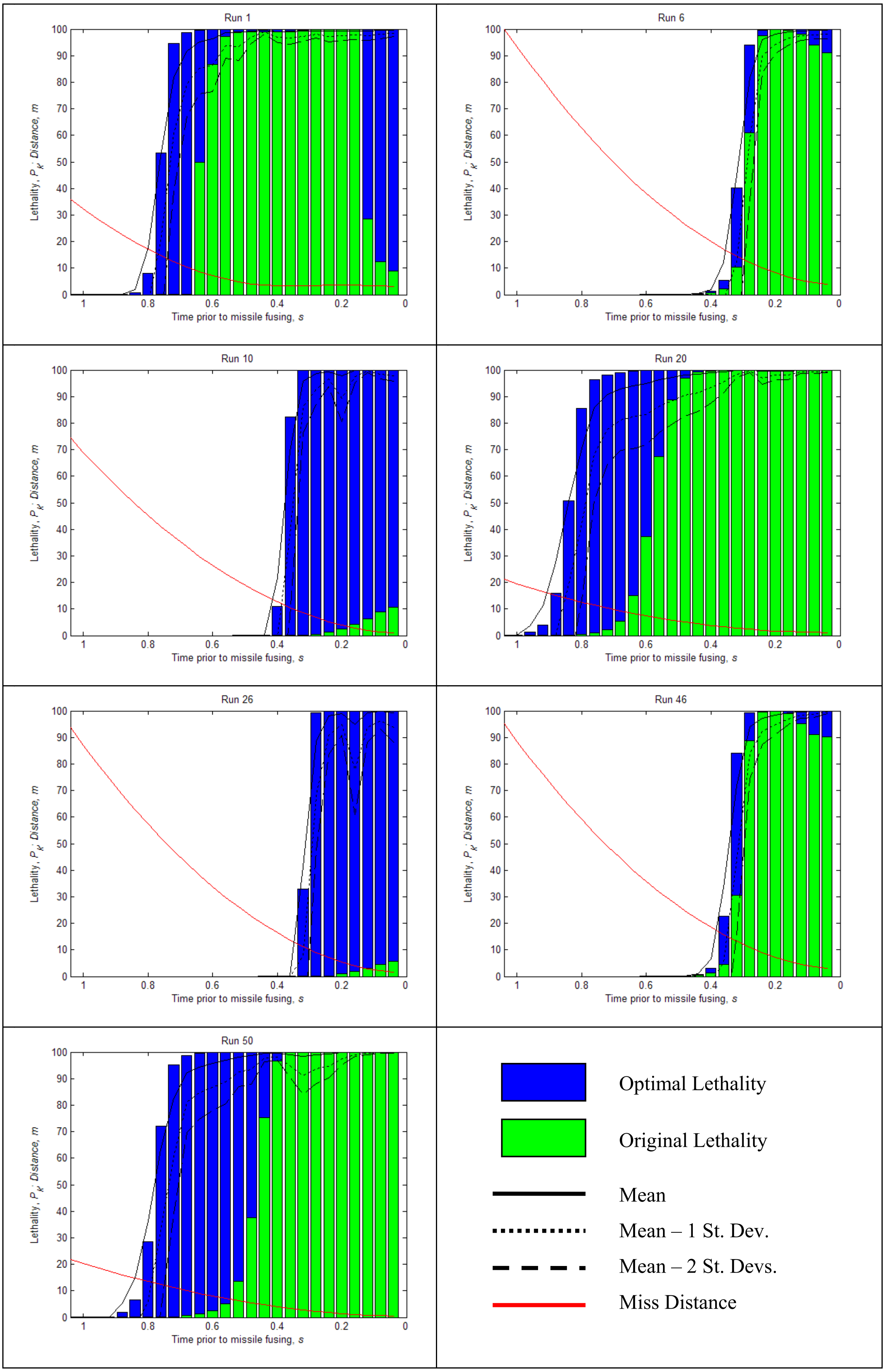


Figure 5.8: Robustness of Rear On Optimised Cases

For runs 1, 6 and 20 there are also similar points along the fly-out path where the lower bound drops off but in these cases the drop is minimal and suggest that not achieving the optimal parameters would result in a loss of less than 5%. In this respect run 46 does not show this characteristic and maintains the high levels once they are achieved. However, of the cases that required that are similar in pattern, for example runs 1 and 6 this high level is attained closer to the end of fly-out compared to the other cases which reach higher levels of lethality sooner.

It can be seen that the best solution is the one that provides the most robust optimised solution as this would be the solution that would be able to yield the most damage to the target, even if the actual endgame parameters are not met. From the rear on cases it can be seen that the beast case is thus runs 20 and 50, as they are the most robust.

The orientations for the actual and optimal solutions for run 10 are shown in Figure 5.9. It can be seen that the reason for the large increase in lethality is due to warhead fragments impacting the cockpit of the target aircraft which is the most vulnerable part of the vehicle.

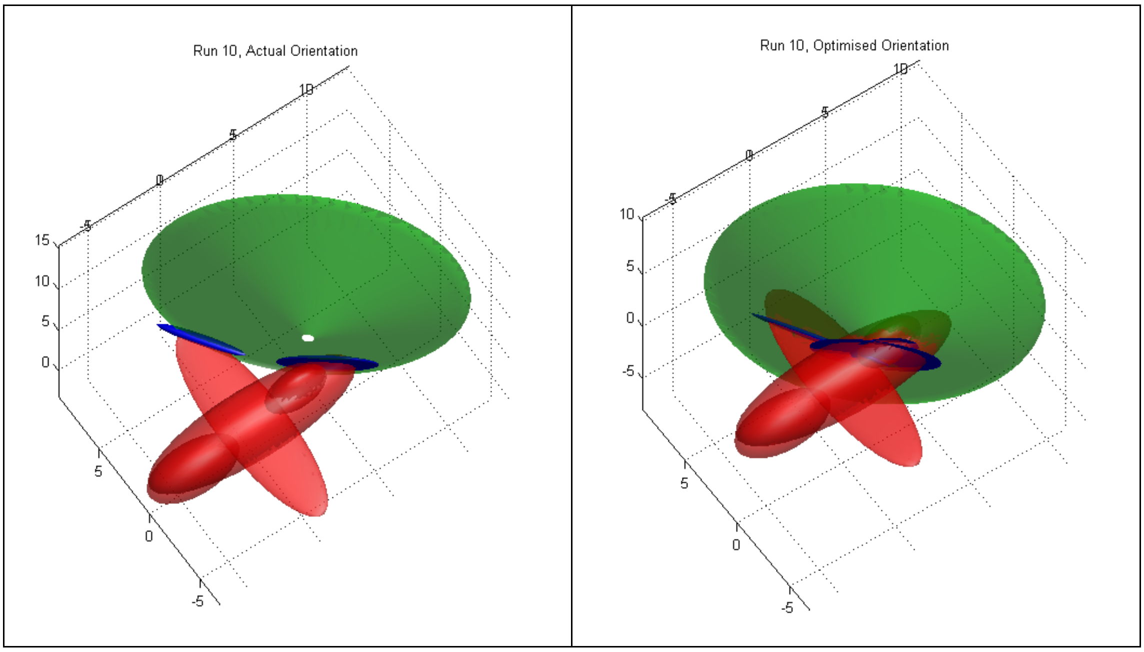


Figure 5.9: Rear On Orientations

5.2.4. Summary of Results

It could be seen that the best cases within each category were those that yielded the most robust, high lethality solution, such that each of these would be able to inflict the most damage even if the optimal condition is not physically realised.

It can be seen that for all three categories of fly-out there are similarities in the observations seen from the results. These can be generalised as follows:

- Cases that increase in lethality as the missile fly-out continues and the lethality stays high, and does not require optimisation.

- Cases in which the lethality peaks and then drops off before fusing and this loss can be regained via optimisation.

- Cases in which the lethality increases and then drops as before but can not be remedied by optimisation.

- Cases that have low to mid-range lethality values that can be optimised to yield desirable levels of lethality.

- Cases that show low to mid-range levels of lethality but can not yield higher values by optimisation.

From these findings it is possible to suggest the course of action that should be taken in order to achieve the highest possible lethality probability for a given starting condition for a fly-out.

It is possible to employ a decision process, especially for those scenarios whereby the lethality dropped by a large amount. This process would decide whether to fuse the missile warhead or to continue as is, and would be an ‘online’ process. There are many factors and combinations of measurements that can be used to decide how and when to fuse the missile as it approaches and sometimes even passes its target. There are many methods by which this decision can be made and these will be discussed in Chapter 6.

Summary

This section has further investigated the results of the fly-out scenarios simulated in Chapter 4. Missile trajectory data has been used to calculate lethality probability data for points along the missile trajectory in order to examine how lethality varied, and if it had been suitable to fuse the missile earlier for cases that displayed lower final lethality values. This study showed that for some cases the endgame lethality was indeed lower at the end of the simulation compared to some points along the missile trajectory.

Furthermore each point along the trajectory was then optimised for the missile aim points in order to see if lethality could be increased for cases that showed lower lethality values, and a robustness measure was also calculated to see how sensitive these optimal conditions were to variations in the optimised parameters. It could be seen that the best cases were those that yielded the most robust, high lethality solutions within each category.

From these studies it has been shown that a decision process could be utilised in order to obtain more desirable endgame conditions, especially for cases where by the missile has fused too late and resulted in lower lethality probabilities compared to previous points along the missile path.

6. Fusing Strategies

From the results above it can be seen that there exists a need for a decision to be made for when the missile system should fuse the warhead, and whether or not an optimisation is required to increase the lethality probability. This Chapter will describe differing methods by which a decision on whether the missile should fuse or not can be made. There are many strategies that can be employed, depending on the situation being considered, and the most relevant are now considered.

6.1. Current State of Fusing Strategies

The current state of fusing involves the use of radio frequency proximity fusing to trigger the warhead. A proximity fuse is designed to detonate the warhead charge when a received signal breaches a specified threshold value, usually based on the effectiveness of the warhead fragments field. The proximity is calculated by transmitting a radio signal out from the missile and 'listening' for a reflected signal to be received. The received signal is out of phase with the transmitted signal, due to the relative velocity of the missile and target. The interference pattern caused from combining the two signals can be amplified and used to activate the warhead trigger. The amplitude of combined signal is a function of the distance between the missile and the target, and as such this signal can be used as a trigger to detonate the warhead. This is achieved by tuning the gain on the amplifier and by setting a threshold (bias) that the amplitude of the combined signal has to reach for fusing to take place. A delay can also be employed.

Possible fusing strategies are discussed below. These will be split into three categories, simple decision processes, complex decision processes, and knowledge based decision processes. Each will be described and a framework for how each will be implemented will be shown.

6.2. Simple Strategies

6.2.1. Minimum Distance

The simplest strategy is based on the miss distance. The warhead is triggered when this miss distance is at a minimum. The missile can be configured such that as soon as the missile's approach to the target begins to move away from the target fusing takes place. The method does not perform any lethality calculation or optimisation and as can be seen in some of the front on results previously that this method, if employed, would have yielded higher lethality values than the final fusing point of the simulation. If an optimisation is performed during fly-out then this data could also be incorporated to maximise the lethality for a minimum miss distance fusing strategy.

The minimum distance fusing trigger is the simplest method of fusing the missile. The miss distance is an important parameter to consider in order to achieve high lethality probabilities in the endgame. The missile will continue to fly toward the target craft until the miss distance, S_r, begins to increase as shown in Figure 6.1 and described below:

$$if\ \frac{dS_r}{dt} \leq 0\ \ then\ activate\ the\ trigger \qquad (6.1)$$

At this point the missile will trigger the fuse and detonate the warhead. There are no lethality calculations that take place in this method.

The advantages of this method are that the process of detecting the distance is simple and quick as it uses on-board sensors, and as shown in some of the cases in Chapter 5 the minimum distance does yield a high value of lethality. The disadvantages are that there are no lethality calculations or optimisations that take place and as such fusing may result in sub optimal lethality values.

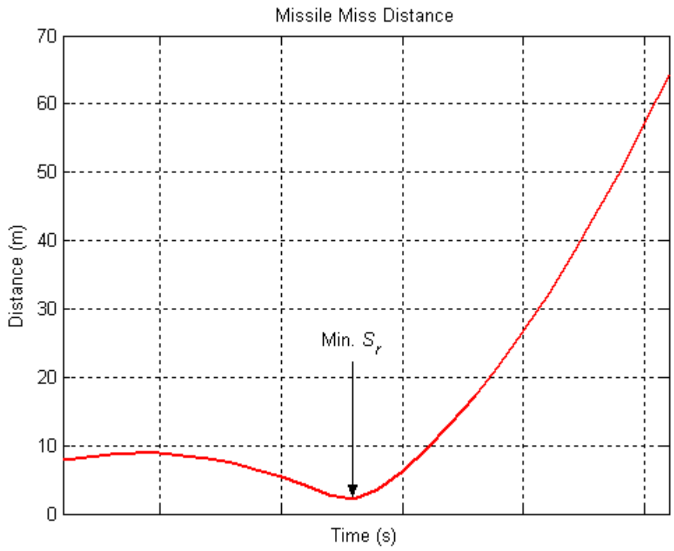


Figure 6.1: Minimum Distance Fusing

6.2.2. Threshold Level Fusing

Another simple method that does not necessarily require optimisation but would require lethality calculation is use a threshold lethality value. A lethality probability is calculated and if the probability is above a set value then the missile fusing takes place. This threshold value would ideally need to be set extremely high in order to inflict sufficient damage to the target. This method can also be employed using an optimised lethality to see if the optimal probability is above the threshold. Additionally the lower bound of the robustness could also be utilised to see if uncertainties in the optimal aim points will reduce the lethality below the threshold in which case fusing should not take place.

A threshold level fusing trigger will calculate the lethality, P_k, as the missile moves towards the target and fuse when the value of P_k passes above a set threshold value.

$$\textit{if } \boldsymbol{lethality \geq threshold\ value} \textit{ then activate the trigger} \tag{6.2}$$

This will ensure that the probability of damaging the target is high; however this will not be robust as no other calculation is carried out to ensure that any sudden changes in conditions between the check and fusing times will affect the lethality. An example with the threshold set to 95% is given in Figure 6.2.

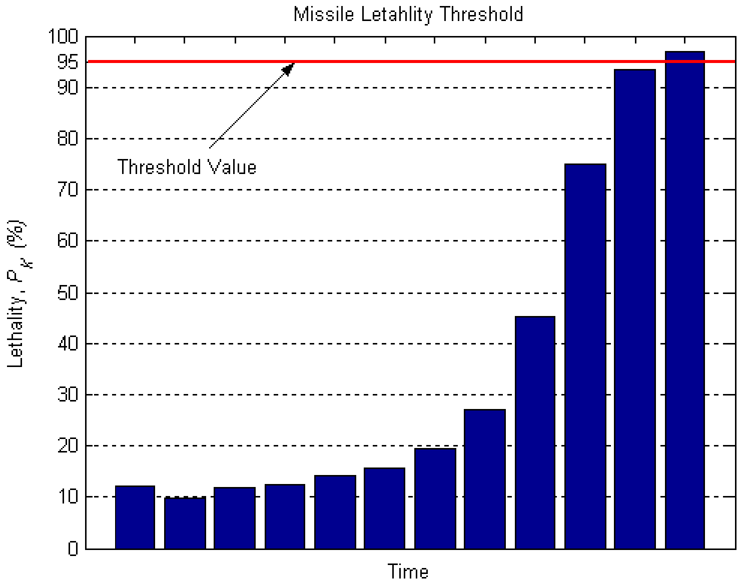


Figure 6.2: Threshold Level Fusing

As can be seen the missile at the start of the graph calculates a lethality of only 12%, which initially drops down to 10% before gradually increasing up until the point at which the threshold value is reached. At this point the trigger would be initiated and the target would be engaged.

The advantage of using this method for fusing is that as a calculation is performed on-board the missile will only detonate when the threshold is reached, and by setting this value very high it can be ensured that fusing will only take place in conditions that will yield a high lethality. This disadvantage of this method is that for some cases the missile may never reach a point in its trajectory that will yield a lethality probability that breaches the set threshold and hence would not fuse at all. Additionally, as no robustness measure is computed there is no way of knowing how sensitive the lethality probability will be if there are any errors in measurements used to calculate the probability.

A more advanced decision maker would take into account other factors such as robustness measures, and also look at a both the miss distance and lethality probability.

6.3. Complex Strategies

A more detailed approach would look at the lethality value at the previous point along the missile trajectory and compare that to a current calculation. If lethality is increasing then the missile continues to the next sample point. Another indication of this may be if the mean from the robustness measure is above lethality as shown in some cases. The robustness measure of the current lethality, if desirable, can be used to trigger the fuse of the missile if this robustness measure is beyond a threshold and the lethality is high (above its own threshold).

If the lethality drops and the previous measured lethality is higher than the current value, then an optimisation can be performed to see if lethality can be recovered. It may be prudent to trigger the missile fuse at the optimal if the drop in lethality is greater than a set amount. For a smaller drop in lethality if the optimal regains the loss of lethality then the missile can continue for another step. However it may be prudent to fuse the missile if there are consecutive drops in lethality and if the combined loss is above a set value.

It will also be useful to maintain a check on the rate of change of miss distance. If lethality is high and the miss distance is decreasing then again it would be better to continue the fly-out in order to get as close to the target a possible. Of course it will be necessary to ensure that lethality does not drop as a result of the missile trying to achieve a smaller miss distance.

For example it may be beneficial to use a closest point of approach method if the target begins to move away from the missile, as is the case in some front-on scenarios. As it would be difficult to find that exact point during a fly-out, it may be better to fuse just after this condition arises and by optimising the aim point to recover any loss in lethality incurred by the slight increase in miss distance resultant from the delay in fusing.

This section will describe more advanced forms of decision making that can be used by the missile to engage a target in an endgame. These advanced methods will attempt to incorporate strategies based on the observations made in Section 5.2.

6.3.1. Conditional Decision

A conditional decision process uses tests of statements or rules in order to reach a decision. Simple examples of these are the approaches of Section 6.2. Both test a condition in order to return a true or false answer. In the minimum distance case the condition being tested is:

> "Is the miss distance, S_r, at a minimum, or alternatively, is the rate of change of miss distance, δS_r, positive (i.e. moving away from the target)? If so, then activate the trigger."
>
> or
>
> "if $\delta S_r > 0$ then activate the trigger"

For this statement the missile will trigger the fuse when the missile begins to move away from the target.

Similarly for the lethality threshold fusing case the following statement can be tested:

> "Is the calculated lethality, P_k, above the set threshold value? If so then activate the trigger."
>
> or if the threshold is set to, say, 95%
>
> "if $P_k \geq 95\%$ then activate the trigger"

Again the missile will trigger the fuse when the lethality probability reaches the threshold value set on the missile.

It is possible to create a more complex decision making process by using a number of conditional statements together. This will create a set of rules for the missile to assess and make a decision with in order to decide whether to trigger the fuse of the missile or not.

Using the information obtained in Chapter 5 a set of rules can be developed to create a decision process. This rule base will examine various conditions that can lead to a more robust and desirable fusing point.

6.3.2. Rule Base Description

The following section will describe the set of statements that will be assessed and used for the missile fusing decision protocol. Unlike the simple fusing methods, which look at the current miss distance or lethality value individually, this more complex decision maker will use of combinations of values and also look at how lethality and miss distance are changing by looking at the current and previous measurements. These rules are based on observations made during the previous studies of Sections 3, 4 and 5.

Rule 1

> 'Is lethality above a threshold value?'

This rule is the same as in the previous section, whereby fusing should occur if lethality is sufficiently high. The next rule is related to the threshold value.

Rule 2

> 'Is miss distance, S_r, decreasing?'

This rule tests the condition of whether the missile is approaching the target or whether it is moving away from it. Generally if the missile is approaching the target then fusing should not take place.

Rule 3

> 'Is lethality, P_k, increasing?'

This rule tests if the calculated lethality has increased from the previous measurement. If lethality is increasing then fusing should not occur. The next rule is related to this as it tests the robustness of the calculation.

Rule 4

> 'Is the mean of perturbed solutions around the measured case higher than the measured case?'

From some of the cases shown in Section 5 it can be said that when the lethality is increasing the mean of perturbed cases is higher than the measured lethality. Hence, if the mean is higher then fusing should not occur as it is likely that the next measurement will yield a better outcome.

Rule 5

> 'Has lethality decreased?'

This rule looks at the amount the lethality has dropped by and if necessary will trigger the fuse if the drop is greater than a predefined amount.

Rule 6

> 'Is the lower bound of perturbed cases higher than the threshold value?

This rule will check if the lower bound of the perturbations, P_{kLB}**,** to the lethality still lies above the fusing threshold level. If so then this is an ideal point to trigger the fuse as the missile will be in a good robust position.

The rules are utilised to enable a decision to be made on whether the trigger should be activated or not. A fusing matrix, Table 6.1, shows the combination of conditions that will decide whether fusing will take place, depending on the conditions from the rules above. Each row is dependant on the row above, and so the conditions for each rule will be checked in order.

P_K Threshold	S_R Decreasing	P_K Increased	P_{KMEAN} > Pk	Is P_K Drop > Limit	Lower Bound > Threshold	Fuse?
Y	N	N/A	N/A	N/A	N	**Y**
Y	Y	N/A	N/A	N/A	Y	**Y**
Y	Y	N/A	N/A	N/A	N	**N**
N	N	Y	N	N/A	N/A	**Y**
N	Y	Y	N	N/A	N/A	**N**
N	Y	Y	Y	N/A	N/A	**N**
N	Y	N	Y	Y	N/A	**Y**
N	Y	N	Y	N	N/A	**N**
N	Y	N	N	Y	N/A	**Y**
N	Y	N	N	N	N/A	**N**
N	N	N	Y	Y	N/A	**Y**
N	N	N	Y	N	N/A	**N**
N	N	N	N	Y	N/A	**Y**
N	N	N	N	N	N/A	**Y**

Table 6.1: Fusing Matrix for Original Values of Lethality

These rules only look at the calculated lethality values, and no optimisation takes place. The same set of rules can be used for optimised lethality values along the fly-out of the missile.

These rules look at lethality and optimised lethality calculations and base decisions by looking back one step in order to decide whether to fuse or not. This is performed on an individual run-by-run basis and does not contain any kind of memory of simulations or data knowledge for prediction.

The next section will discuss methods for decision making for the trigger based on knowledge attained from previous fly-out cases.

6.4. Knowledge Based Decision Strategy

A decision maker algorithm is to be utilised that incorporates memory, or knowledge, to base the decision of when to trigger the missile's fuse. This is achieved by using data from the previous fly-out cases used in earlier sections. There are many methods by which a knowledge base can be initiated, including Case Based Reasoning [28], and the rule based methods described above.

6.4.1. Case Based Reasoning

Case Based Reasoning (CBR) has been used for many years in different subject areas. The basic concept behind CBR involves making decisions based on past experiences and the knowledge gained from these experiences. A case is a piece of knowledge that represents an experience, in the form of a lesson that can be applied to a set problem in a particular context. This process has been used in many diverse applications from legal firms, which can use CBR to contain various legal precedents, to engineering companies, which make use of CBR for fault diagnosis and repair of components in, for example, aircraft. There are essentially four main steps in the CBR process, as shown in Figure 6.3.

These steps are:

- **Retrieve**;

  A problem is matched against cases in the case base and those which are similar to the problem are *retrieved*.
- **Reuse**;

  Solutions suggested by retrieved cases are *reused* and tested for success.
- **Revise**;

  Retrieved cases that are not a close match to the problem will need to be *revised* and evaluated for its suitability to the problem.
- **Retain**;

  This new, suitable, revised case will then be *retained* in the case base.

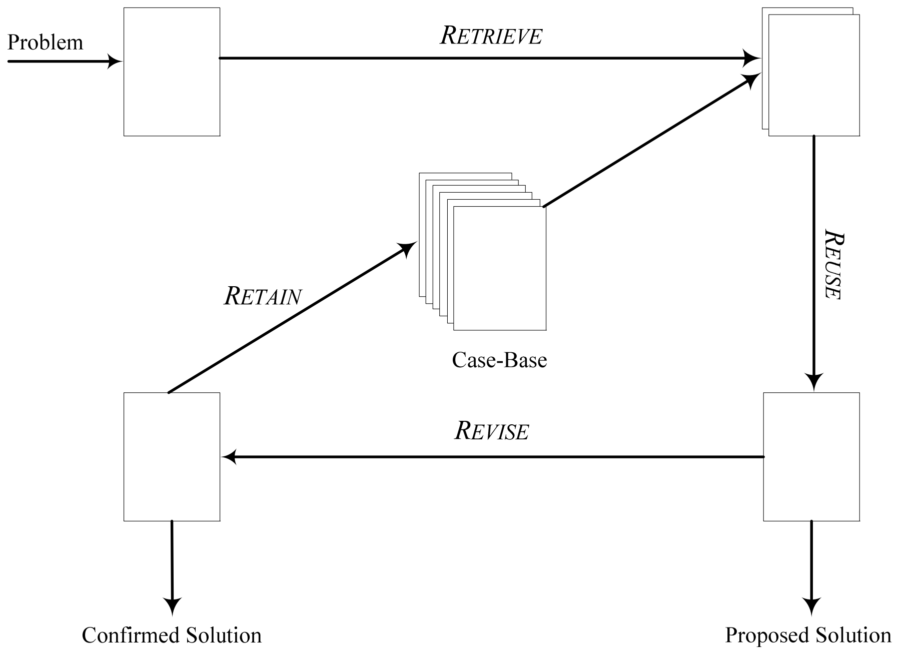


Figure 6.3: CBR Process

It is proposed that a variation of CBR be used in the decision process for the missile endgame fusing study, in which the data gathered previously will be used as a case base, integrated into two algorithmic solutions and applied to a sample of missile endgame fusing problems

6.4.2. Using Case Based Reasoning for Missile Fusing Algorithm

A case base will be used to compare missile readings for miss distance and calculated lethality and provide a basis for the fusing. The data gathered will be used in order to provide a probability of whether the lethality is likely to increase or decrease. This is achieved by categorising how the lethality varied in previous simulations using miss distance and the calculated current lethality as the two keys for indexing the data. The case base will therefore be indexed using these two properties, and contain information on how the lethality changed for each case. By categorising many cases that show similar circumstances and collating their outcomes a probability can be found to aid the fusing process.

The case base is to be generated by running and partitioning 5,000 simulations of engagement fly-outs and endgames using MSTARS and AGILE. The case base can be visualised as a 2D table of rows and columns as shown in Figure 6.4. Each row and column corresponds to a miss distance range and lethality range respectively. Within each cell, corresponding to a specific distance and lethality, a data structure will be defined to contain the following information:

- Whether the lethality at the next step increases, or decreases;
- An average of the increase or decrease in the lethality.

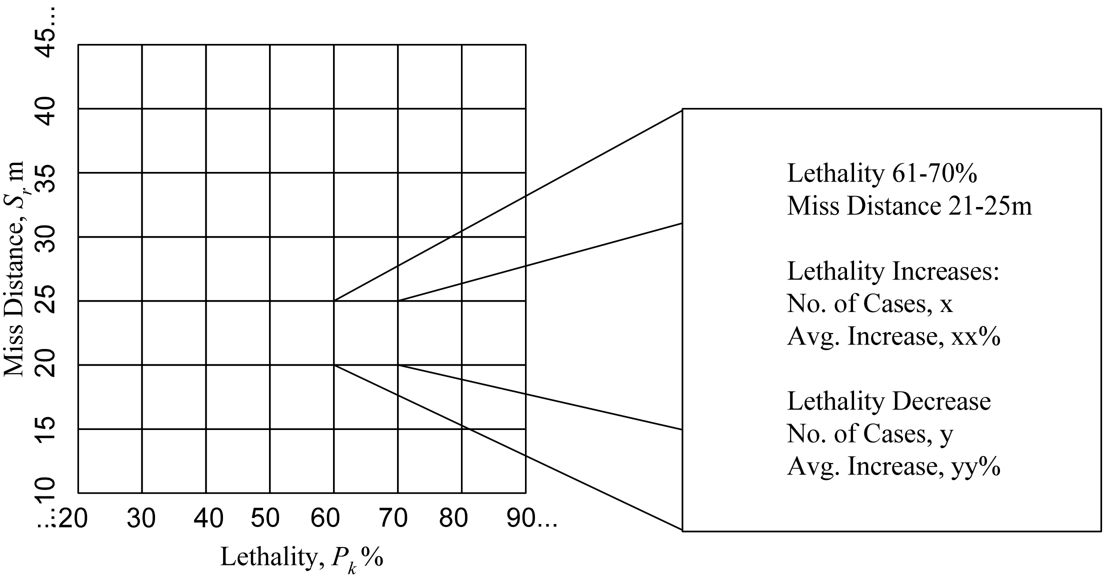


Figure 6.4: Case Base Table Design

For each fly-out, a lethality calculation will be performed for east step along the simulation and categorised using the lethality and miss distance as the defining indices. The lethality at the next point will be calculated and used to populate the case base data structure, thus, through this form of reinforcement learning [29], providing a probability of the nature of how the lethality changes for each 'cell' within the case base table, to will be used in the decision making process.

This probability will be used as a factor for the decision maker for the trigger of the missile. If the probability indicates that the chance of an increase in lethality is likely, then the missile will continue on its fly out. If, on the contrary, the probability is such that a drop in lethality is more likely, then the missile will fuse after performing an optimisation.

The algorithm will include elements from the rules stated previously in Section 6.3, for example fusing with calculated lethality levels above a threshold value if within a robustness level. In addition to the lethality threshold and robustness check, a missile approach check will be assessed prior to this to this probability of change being found from the case base. If the lethality calculated is above the threshold value and within the robustness bound then a check of whether the miss distance is decreasing is performed. If the missile is closing in on the target then fusing will not take place, else if the missile begins to move away then fusing will occur. If the threshold is achieved but the robustness is beyond the bound set then again if the missile is moving toward the target then fusing will not occur. If the miss distance is increasing then fusing will take place following an optimisation in order to attempt to improve the robustness for the case.

A flow chart outlining the steps described for the above algorithm is shown in Figure 6.5. This algorithm will be evaluated in Chapter 7.

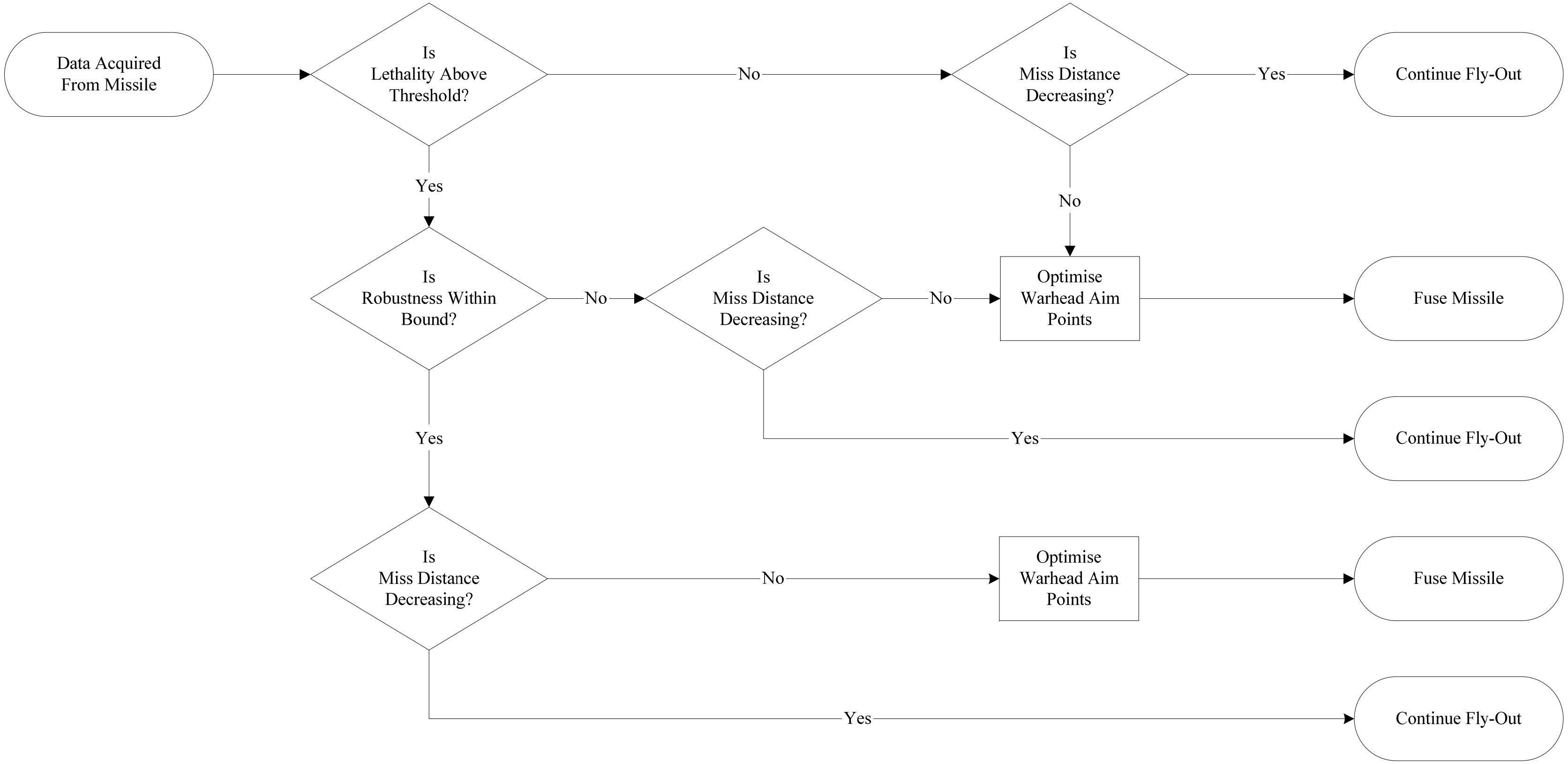

Data Acquired From Missile
Is Lethality Above Threshold?
No
Is Miss Distance Decreasing?
Yes
Continue Fly-Out
Yes
No
Is Robustness Within Bound?
No
Is Miss Distance Decreasing?
No
Optimise Warhead Aim Points
Fuse Missile
Yes
Yes
Continue Fly-Out
Is Miss Distance Decreasing?
No
Optimise Warhead Aim Points
Fuse Missile
Yes
Continue Fly-Out

Figure 6.5: Flow Chart for Simple Predictive Decision Algorithm

This is a relatively simple decision algorithm. A more advanced version of the algorithm can be used which examines the data in more detail in order to base the fusing decision. This is described next.

6.4.3. Advanced CBR Algorithm for Missile Endgame Fusing Problem

A more advanced case based algorithmic solution is proposed that will use fly-out properties other than just a probability from the case base data in order to invoke the fusing of the missile.

This advanced algorithm includes the elements from the previous algorithm and will add some more modified rules from Section 6.3. One such adaptation will involve rule six. Rule 6 stated that if the drop in lethality was to exceed beyond a certain predetermined amount then fusing would take place. This will be extended by adding the notion that if the lethality does drop but not by more than a set amount, then a flag will be raised for the next iteration of the algorithm, and so if lethality drops again then the missile will trigger the fuse. Therefore if the lethality drops it is given the chance to increase again before fusing.

The fusing algorithm is shown in Figure 6.6 as a flow chart. From this chart it can be seen how a decision is reached whether to fuse or not, and this algorithm will be evaluated in Section 7.

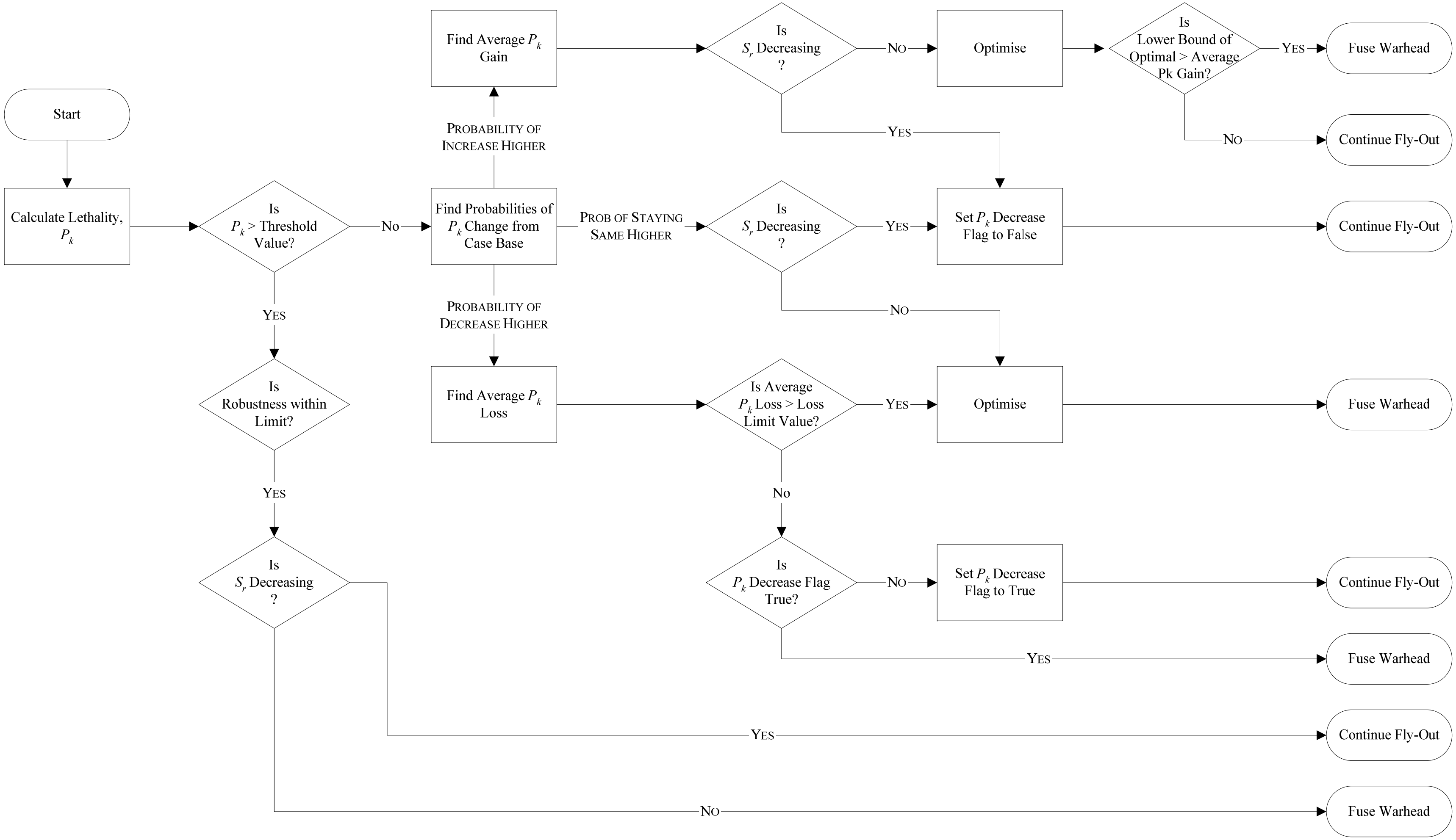

Start
Calculate Lethality, P_k
Is P_k > Threshold Value?
No
YES
Is Robustness within Limit?
YES
Is S_r Decreasing ?
YES
NO
Find Probabilities of P_k Change from Case Base
PROBABILITY OF INCREASE HIGHER
PROB OF STAYING SAME HIGHER
PROBABILITY OF DECREASE HIGHER
Find Average P_k Gain
Is S_r Decreasing ?
NO
YES
Optimise
Is Lower Bound of Optimal > Average Pk Gain?
YES
Fuse Warhead
NO
Continue Fly-Out
Is S_r Decreasing ?
YES
Set P_k Decrease Flag to False
Continue Fly-Out
NO
Find Average P_k Loss
Is Average P_k Loss > Loss Limit Value?
YES
Optimise
Fuse Warhead
No
Is P_k Decrease Flag True?
NO
Set P_k Decrease Flag to True
Continue Fly-Out
YES
Fuse Warhead
Continue Fly-Out
Fuse Warhead

Figure 6.6: Flow Chart for Advanced Fusing Algorithm

In total there will be four flags used in the algorithm for the following

- P_k Decrease – will flag true if the lethality decreases and if flag is already set to true then fusing will occur as this will indicate consecutive decreases in lethality;
- Robustness – will flag true if the calculated lethality is robust such that the lower bound is within a set limit from the lethality;
- Delta S_r - will flag whether the missile is approaching the target or moving away from it;
- Fusing – will flag false to continue and will change to true in order to detonate.

At the start of the iteration the algorithm has the two flags sent to it to indicate if the missile is closing in on the target, and if the lethality had decreased previously. From here the lethality and its associated robustness is calculated based on the endgame parameters. If this lethality is above the threshold then a check is performed to see if it is within the robustness limit and if so a second check, on whether the missile is approaching the target is performed. If both flags are true, i.e. yes it is robust and getting closer to the target, the algorithm goes to the next step as it is desirable to fuse closer to the target. If the delta S_r flag is false then fusing will take place.

Following the lethality threshold check the algorithm will look at the data from the case base of previous fly-outs and match the current lethality and miss distance to the data. From this data a probability will be found on the likelihood of the lethality increasing, decreasing or remaining in the same range.

If the probability of a decrease in lethality is highest, then the average loss is found from the data and compared to the limit imposed on lethality loss. If the predicted average loss is greater than the limit then fusing will take place following an optimisation. If the average loss is within the limit value then the P_k Decrease flag is set to true for the next iteration. If the P_k Decrease flag was already set to true then the fusing flag is set to true and the missile will detonate following an optimisation.

If the probability of change from the case base suggests that the lethality will remain in a similar range at the next time step then again the delta S_r flag is checked. If the missile is still approaching the target then the missile will continue as it is preferable to be closer to the target, and the P_k decrease flag is reset. If the missile is moving away from the target then the missile will fuse at optimal conditions.

If the probability from the case base retrieval suggests that the lethality will increase then the missile will continue if the miss distance is decreasing. If the missile is moving away then the lower bound of an optimisation on the current lethality is found and compared to the average gain from the probability. If the lower bound of the optimal is greater than the predicted lethality from the gain then the missile will fuse, else it will continue and the P_k decrease flag will be reset to false.

Summary

This section has described various methods by which fusing can be achieved. This included simple decision makers that look at just one parameter for example minimum distance or a threshold lethality value. Conditional or rule based methods were then described that are more complex than the simple methods and look at various conditional rules that need to be considered for the decision to fuse the missile to be made. Finally two case based reasoning algorithms have been developed. These will all be evaluated in the next chapter.

7. Analysis of Fusing Methods

Using the methods described in Chapter 6, a batch of 5,000 end game yielding scenarios will be evaluated, and the performance of each method compared. The strategies evaluated will include, minimum distance, lethality threshold level, fusing matrix with original and optimal lethality values, and advanced fusing algorithms.

7.1. Minimum Distance Fusing

Minimum distance fusing involves the measurement of the rate of change of the miss distance of the missile to the target, and fusing at the point that the missile begins to move away from the target.

Using a batch of 5,000 random start points as a reference, the lethality at the minimum miss distance was calculated. A histogram showing the distribution of lethality probabilities is shown below in Figure 7.1.

From the graph above it can be seen that for the 5000 scenarios used 74% of the samples values yielded extremely high probabilities of lethality, i.e. above 85% lethality. However there are also 233 runs yielding a lethality below 10%., which reinforces the fact that there is a need for more advance triggering mechanisms than the minimum distance fusing criteria.

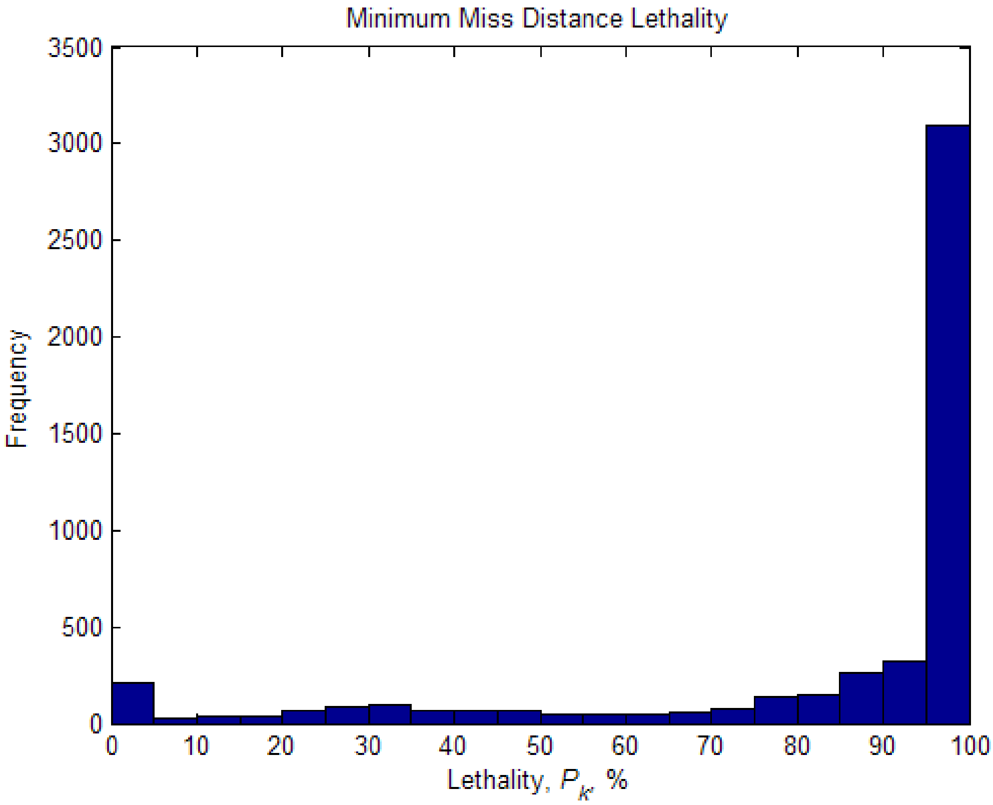


Figure 7.1: Minimum Miss Distance Lethality Distribution

It may seem that the distribution of lethality values is favourable; however many of these runs are not very robust. Figure 7.2 shows in a histogram how the distribution of lethality probabilities varies when the lower bound of the lethality is taken into account. The lower bound is calculated as the original value of lethality less two standard deviations based on varying the missile aim point parameters, as described in Chapter 3.4.

The number of scenarios that yielded a lethality probability above 85% fell to less than 2029 runs, a drop to 41% of the 5000 simulations. The number of scenarios in the range 50%-85% increased to almost half the scenarios. The number of scenarios in which the lethality probability fell to less than 10% increased by over 350% to 827 runs.

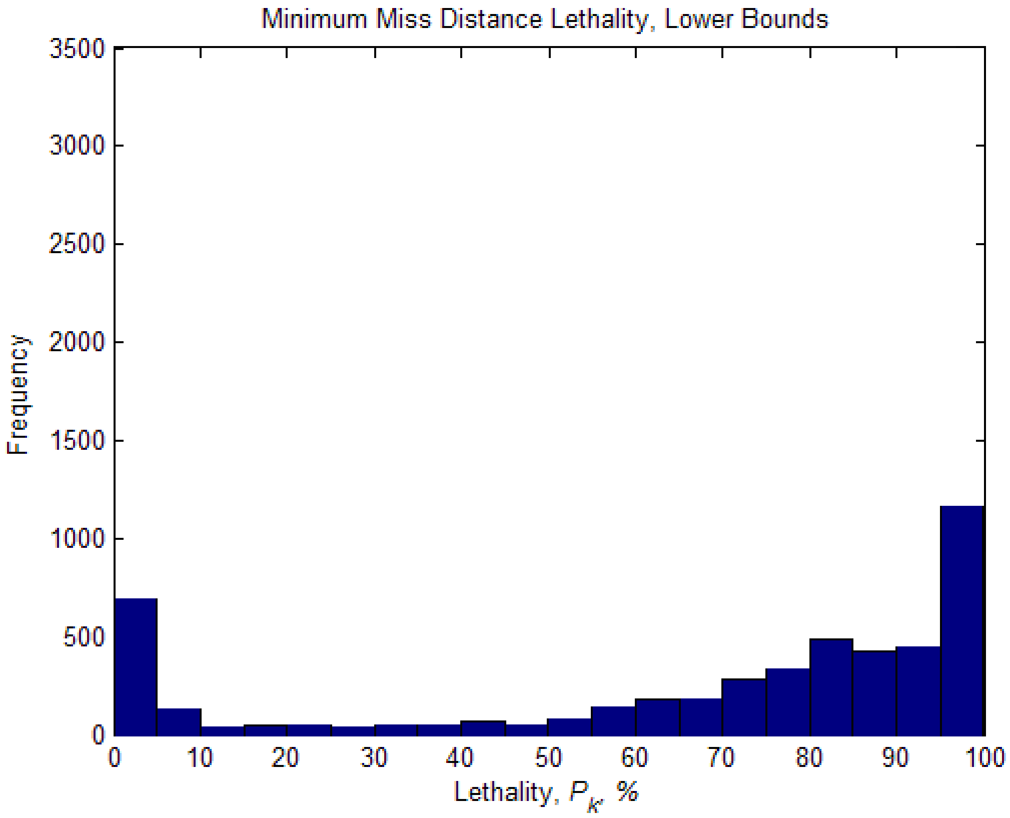


Figure 7.2: Distribution of Lower Bound Lethality at Minimum Miss Distance

From this it can be deduced that whilst the minimum distance of the missile from the target can be used as a strategy for fusing the missile to provide high lethality probabilities, these probabilities tend to be sensitive to variations of the missile aim points. Hence it may be better to use other strategies for fusing.

7.2. Threshold Level Fusing

Threshold fusing strategy comprises of choosing a fusing point based on a calculation of lethality and comparing that to see if it exceeds a threshold value. For the 5000 scenarios assessed in this section, the fuse point of the missile is taken as the point along the missile's trajectory that the lethality probability exceeds the threshold value set. For a lethality threshold set to 85%, 3831 runs reached the threshold value, whereas 1169, or 28% of the runs failed to reach the threshold value. Analysis of when the threshold value is met in the simulation is shown in Figure 7.3.

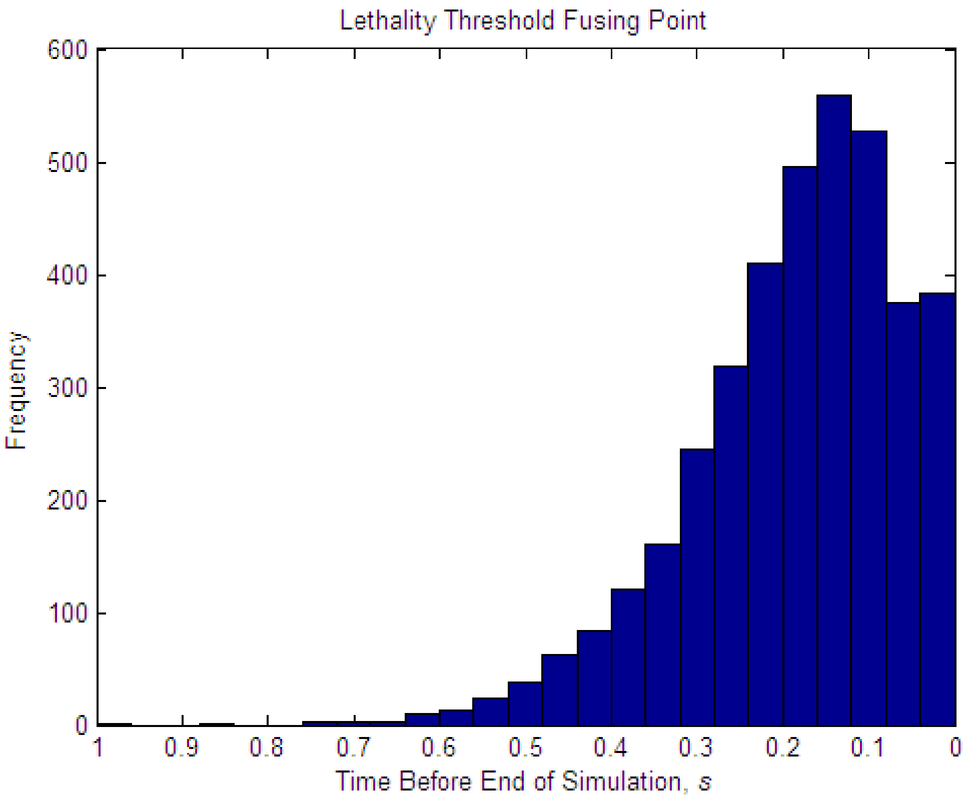


Figure 7.3: Time Before Simulation End of Fusing Point for Lethality Threshold Fusing

From the graph it is noted that the threshold is reached before the simulations finished, hence the missile could have fused earlier. However by examining the robustness of the lethality it can be seen that fusing the missile at the point at which the missile reaches the threshold level is not very good. Figure 7.4 shows the lower bound of the lethality for the point along the missile trajectory where the calculated lethality first reaches the threshold value of 85%.

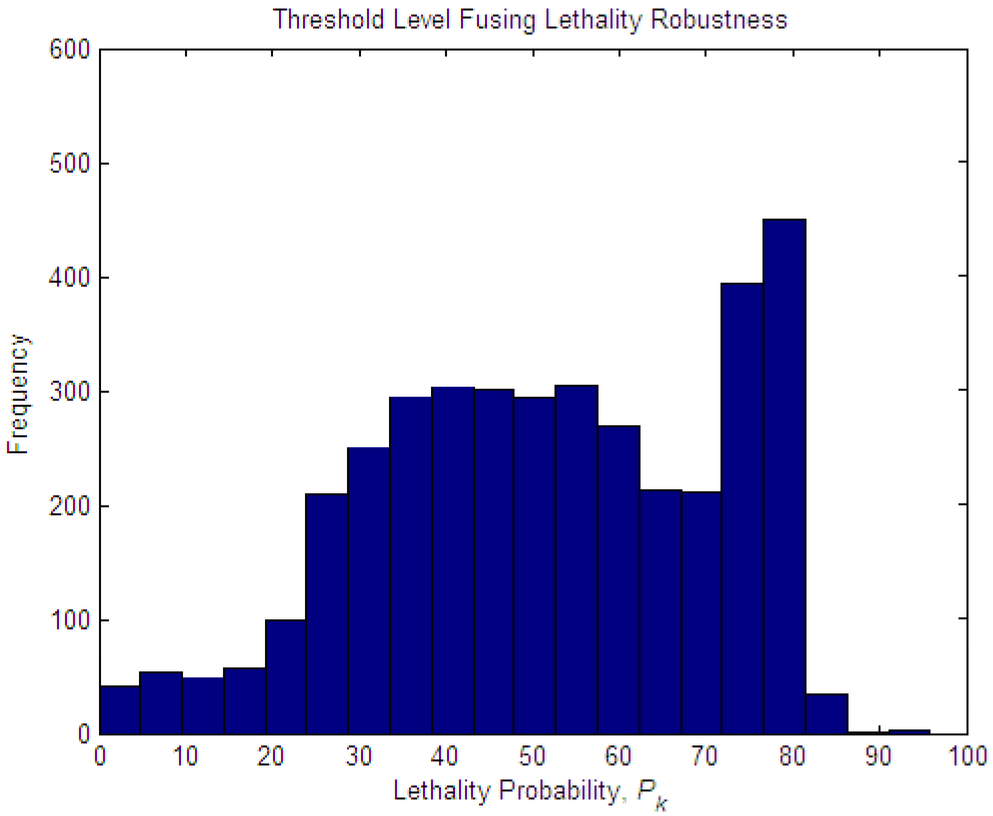


Figure 7.4: Robustness of Threshold Fusing Strategy Simulations

Only three of the 3831 runs that reached the threshold level were robust enough to maintain the lethality value above 85%. 47% of the runs were very sensitive to disturbances, resulting in a potential drop of lethality probability to below 50%.

From this it can be deduced that it may be beneficial to not fuse at the earliest point at which the lethality is calculated to be above the threshold value set, as that point may not be robust to potential variances in the missile's position. It would therefore be better to wait before fusing, for example, if the missile is still approaching the target.

7.3. Rule Based Fusing

7.3.1 Original Lethality Values

The rule based fusing matrix is shown in Section 6.3, and identifies the various conditions for fusing to occur based on missile parameters and lethality calculations. For the 5000 scenarios evaluated, the point of fusing has been plotted in Figure 7.5.

Using the fusing matrix a fusing point before the end of the simulation was found for 4188 of the simulations. Of these, only 145 of the simulations resulted in the fusing occurring at time zero, the time at which the missile simulation ended. The remaining 812 runs did not meet any of the condition described in the matrix; hence the simulation end point values were assessed.

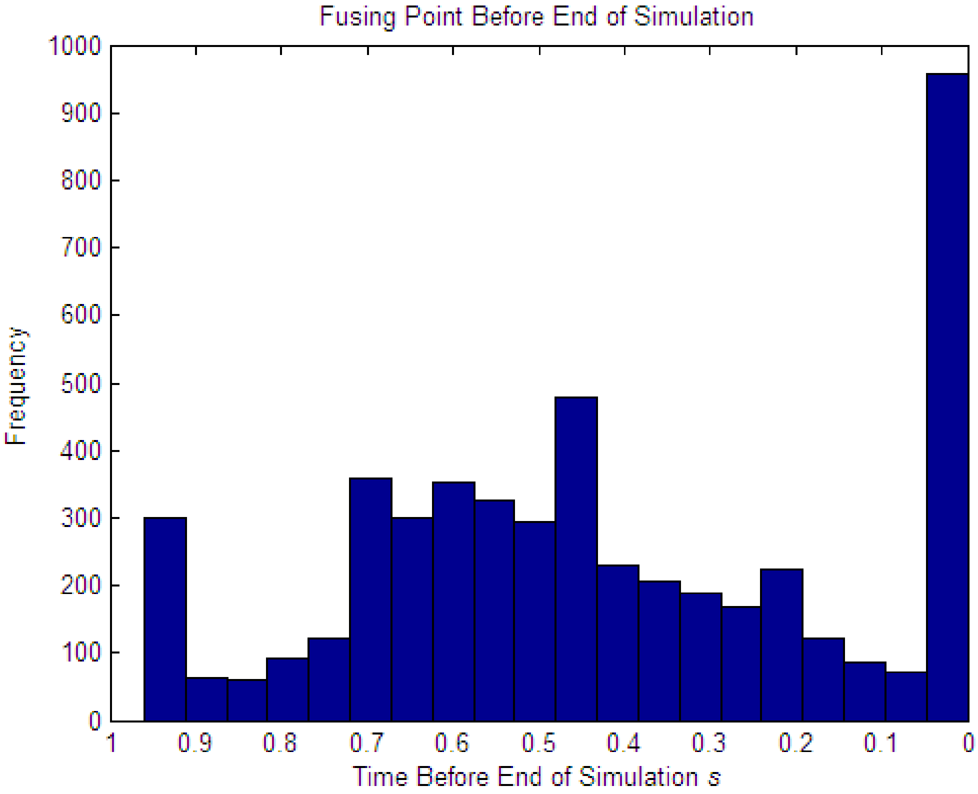


Figure 7.5: Fusing Points of Simulations Using Fusing Matrix

As shown in the figure above, fusing occurs at a variety of points before the end of the simulation. However, the fusing of the missile early results in significant reductions of lethality compared to fusing at the end of each simulation. Figure 7.6 shows that the lethality values are not as high as for the minimum distance fusing.

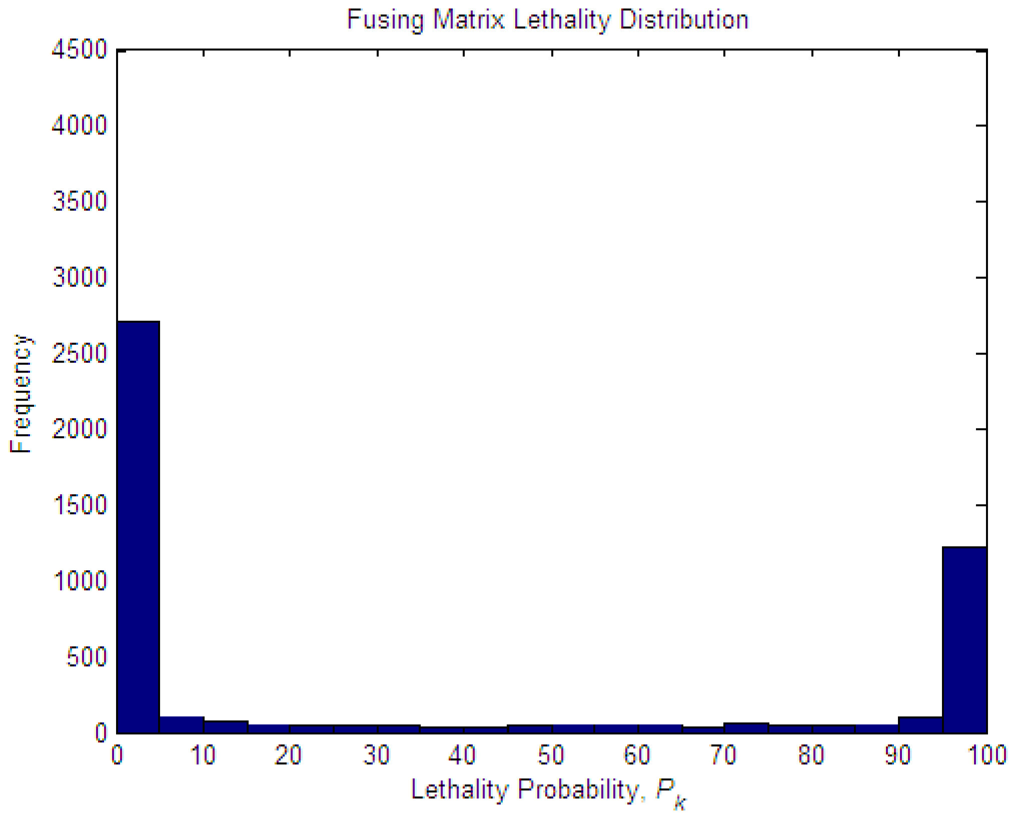


Figure 7.6: Original Values Fusing Matrix Lethality Distribution

The histogram shows that despite fusing earlier using the fusing matrix, the distribution of lethality values is not very good. 1390 of the simulations yielded lethality values above 85%, with the majority of fused points leading to lethality values of less than 10%.

In addition the scenarios which yielded higher probability values were not very robust. The lower bound of lethality values if shown in Figure 7.7. Of the 1390 higher lethality end game scenarios 773, or 56%, maintained a lethality probability above 85%. 3144 of the simulations had a robustness lower bound less than 10%.

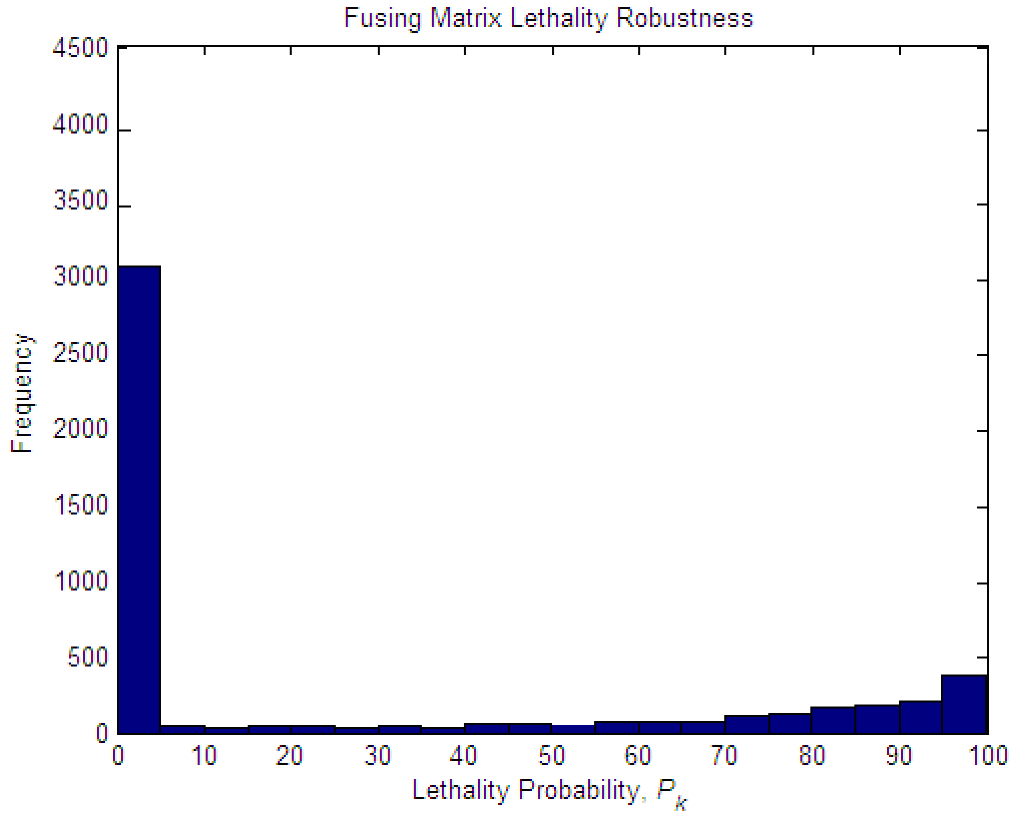


Figure 7.7: Robustness of Original Values Fusing Matrix Lethality

Again, this is not a very good return compared to previous method of minimum distance and threshold level fusing, however an improvement can be made if an optimisation is performed at each step, and the fusing matrix reapplied to the optimised values.

7.3.1 Optimised Lethality Values

Using the fusing matrix described in Section 6.3 and calculated optimised lethality values using the original fly out data, a fuse point was determined for the 5000 fly out scenarios. Of these, 4919 fly-out simulations resulted in triggering of the fuse, with the simulation end lethality values used for the remaining 81 runs.

The lethality distribution of this method of fusing is shown in Figure 7.8. Of the simulations, 1038 runs, or 21% of the runs yield lethality values above 85%. Over three quarters of the fly-out runs fused giving a lethality of less than 10%.

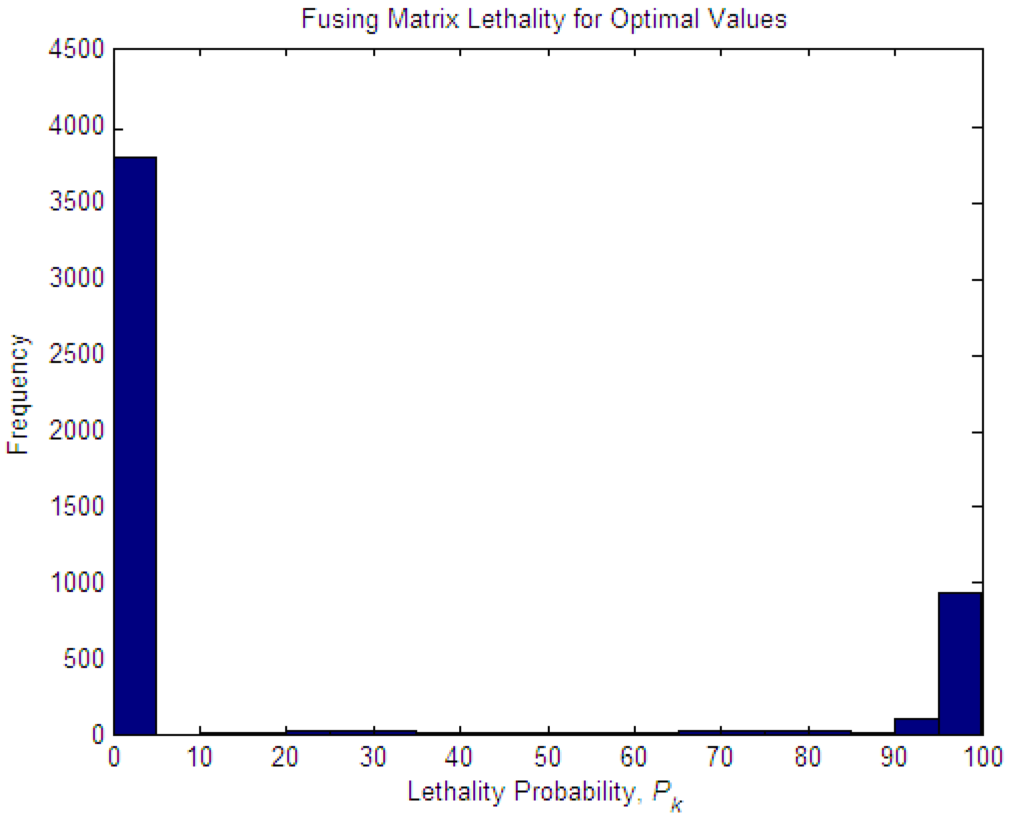


Figure 7.8: Optimal Values Fusing Matrix Lethality Distribution

The robustness of these simulations is shown in Figure 7.9. 95% of the scenarios that yielded high lethality maintained a lethality value above 85%, which is an improvement on the original value robustness seen in Figure 7.7 previously, where only 56% of the high lethality fly-outs maintained their lethality values. The number of scenarios that resulted in low endgame lethality values (i.e. <10%) increased by 13 to 3811.

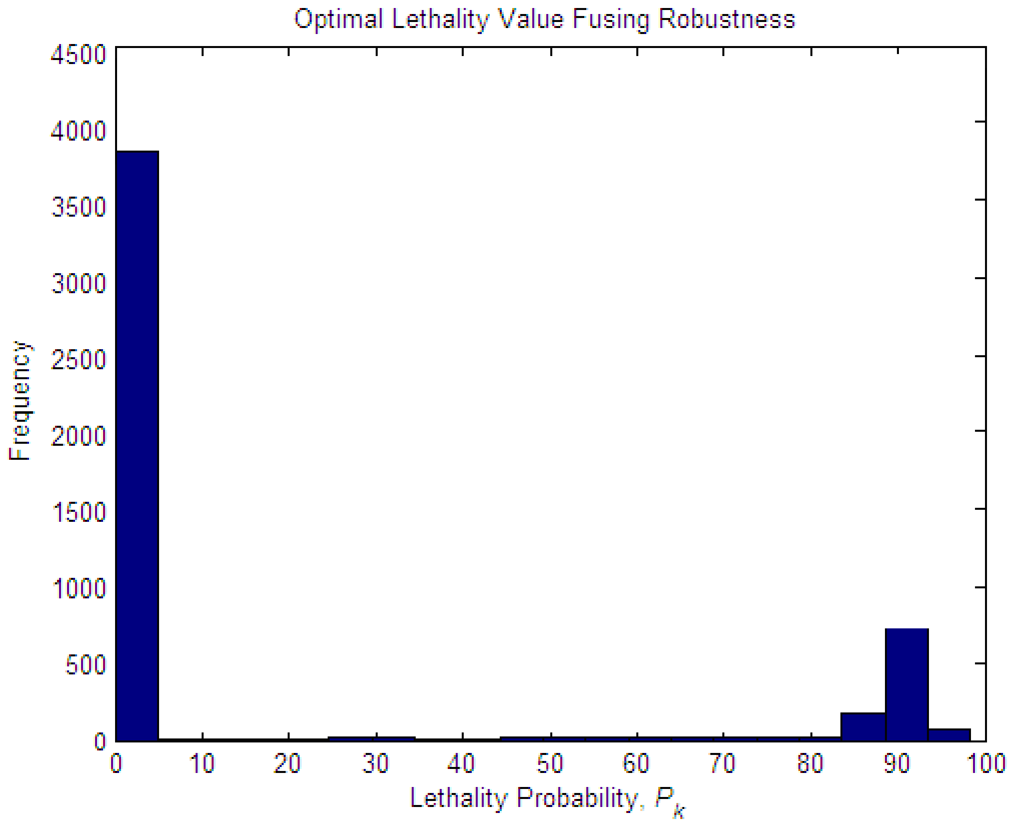


Figure 7.9: Robustness of Optimal Values Fusing Matrix Lethality

The fusing matrix method has produced some lethality values that yielded higher values of robustness, however only a small percentage of the fly-out scenarios resulted in high vales of lethality. The predictive fusing methods will be evaluated to assess if a better method of choosing when to fuse the missile is available using past fly out data.

7.4. Simple Predictive Fusing Method

The simple predictive algorithm is described in Section 6.4. An algorithm was produced which followed the flowchart shown in Figure 6.4. The lethality distribution of the 5000 fly-out scenarios is displayed in Figure 7.10. From these 5000 runs, 3308 of the fly-out scenarios yielded lethality values above 85%. This equates to 66% of the scenarios, however 1339 scenarios yielded lethality values below 10%. Although these scenarios did not yield acceptable endgame lethality values, this is never the less an improvement on the methods utilised previously, and as such suggests that a more advanced algorithm more improve lethality of these cases further.

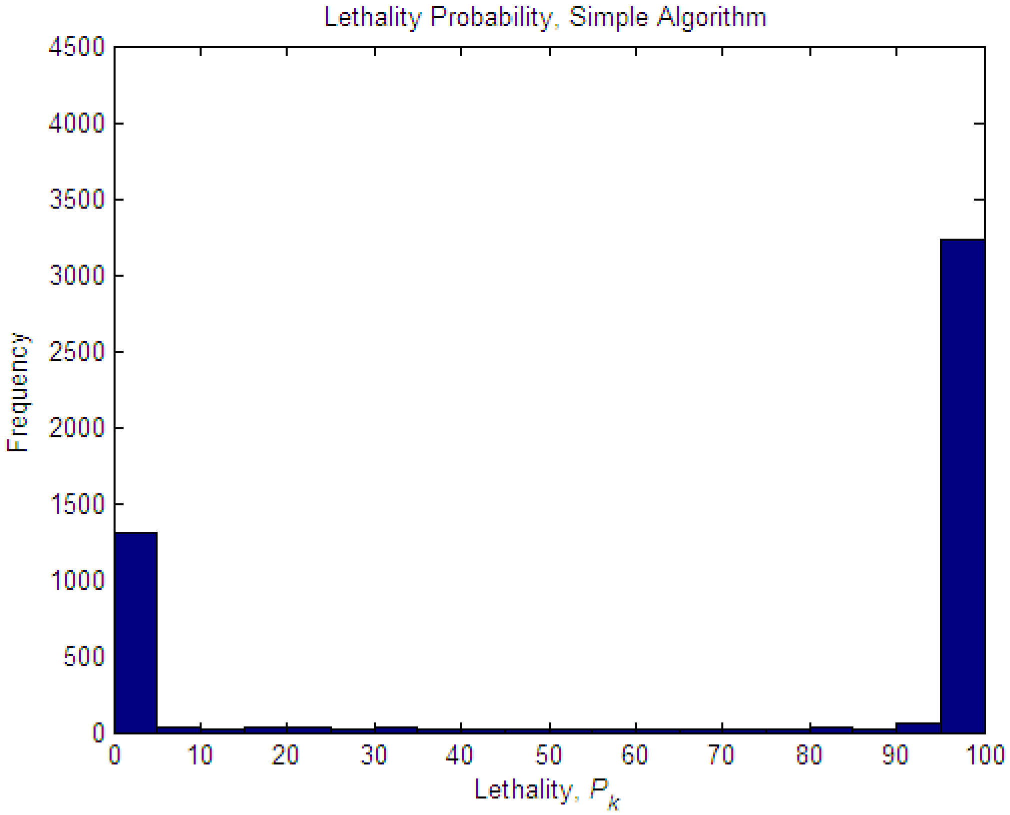


Figure 7.10: Simple Predictive Algorithm Lethality Distribution

The robustness of these solutions was also found in order to ascertain whether any uncertainty in the missile aim points would reduce the lethality by a significant amount. A histogram of the robustness is shown in Figure 7.11.

From the 3308 fly-out scenarios that yielded lethality above the 85% mark, 1702, or 34% of the fly-outs were robust enough to keep a high lethality above that level. The number of cases that gave robustness values below 10% increased to 1510.

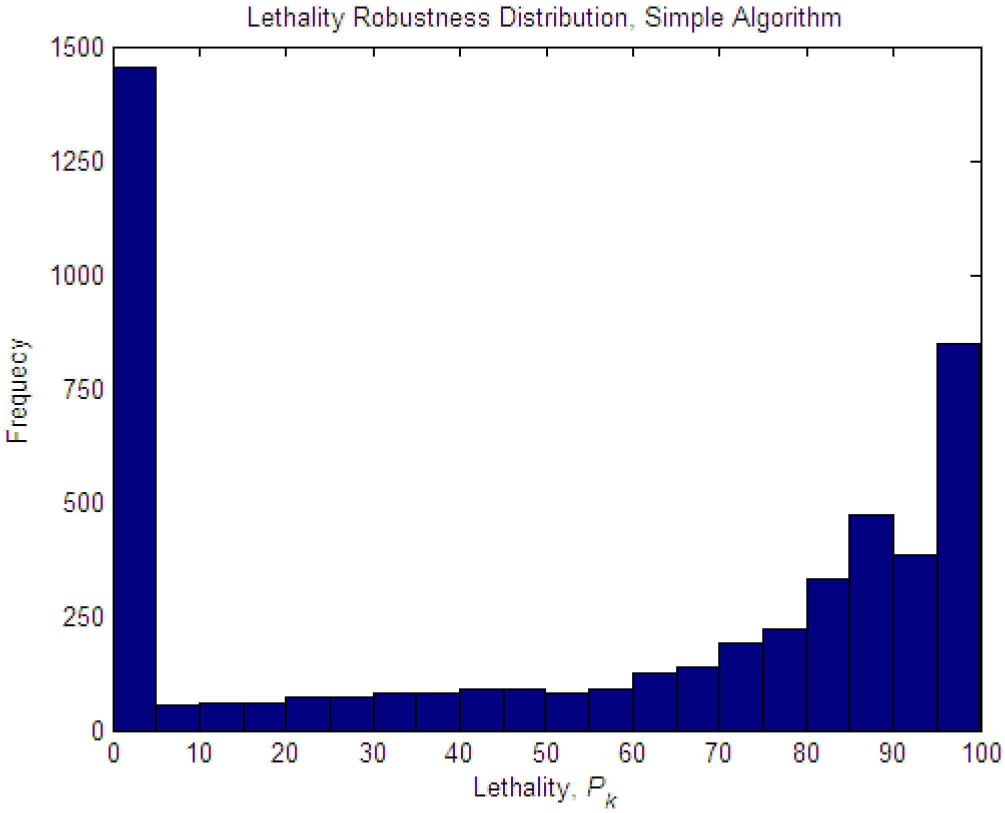


Figure 7.11: Robustness of Simple Algorithm Lethality Values

From these results it is can be seen that there is may be some potential for a more advanced algorithm to be employed that would raise the robustness at the point at which fusing occurs. This advanced algorithm will be evaluated in the next section.

7.5. Advanced Fusing Algorithm

The advanced fusing algorithm, described in Section 6.4, has been implemented and applied to the 5000 fly-outs. The distribution of lethality values that resulted from the algorithm has been plotted in Figure 7.12.

The number of fly-outs that yielded lethality values above 85% increased to 4611 using this algorithm. This represents 92% of the fly-outs. Only 66 of the runs yielded lethality values below 10%, which is an improvement of the strategies reviewed earlier.

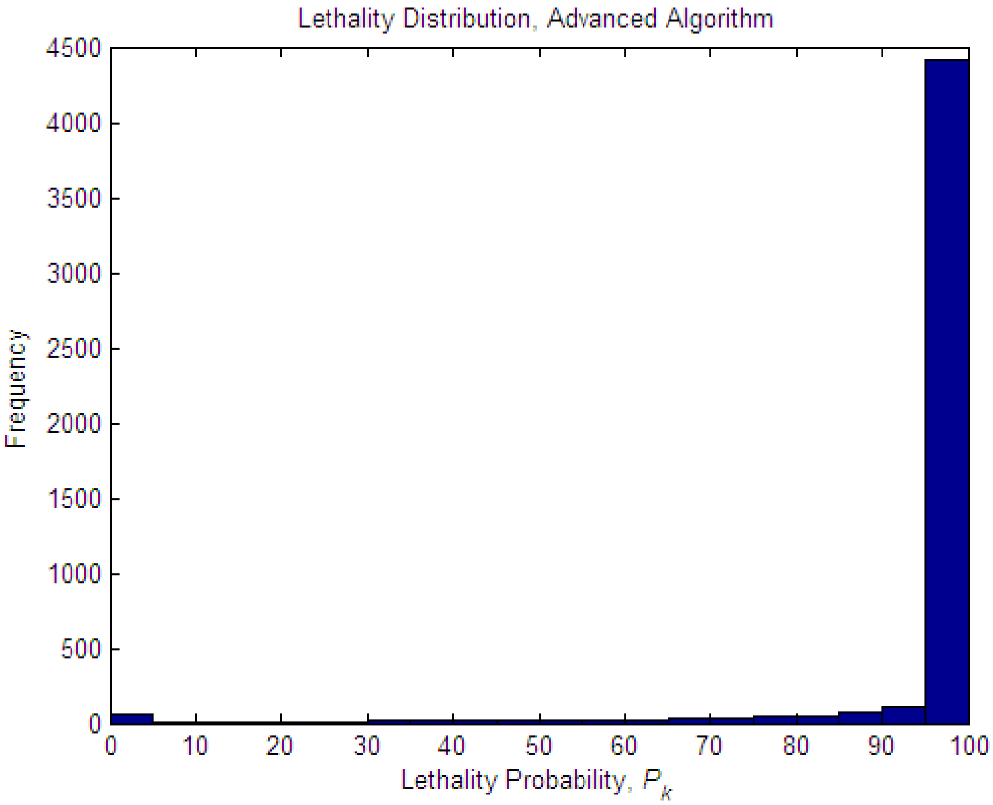


Figure 7.12: Lethality Distribution of Advanced Fusing Algorithm

The distribution of the associated robustness of the solutions found using the advanced fusing algorithm is shown in Figure 7.13. The number of solutions that maintain lethality above the 85% mark falls to 3946 fly-outs. This represents 86% of the high yielding simulations. The number of simulations in the lowest 10 percentile rises to 128 fly-outs.

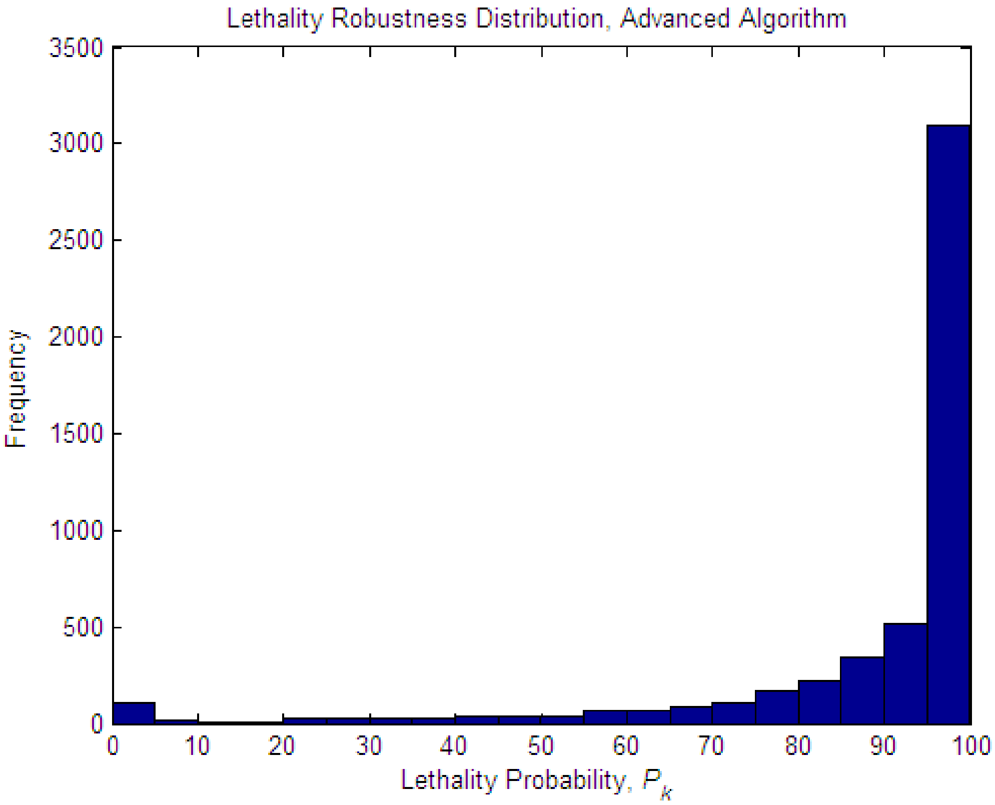


Figure 7.13: Robustness Distribution of Advanced Fusing Algorithm

From these results it can be seen that the use of the advanced algorithm is beneficial as it would allow the missile to trigger its fuse in such a manner that the lethality would be robust to potential variations in the aim point of the missile warhead, by performing on board calculations of lethality and robustness as well as on board optimisations of lethality that can be used to maximise the probability of inflicting the most damage to a target.

A sample endgame orientation for a robust endgame, found using the advanced algorithm is shown in Figure 7.14. It can be seen that the warhead fragment cones intercept the target cockpit, fuselage and wing, resulting in a robust solution as variation in the missile orientation will not reduce the lethality by a large amount.

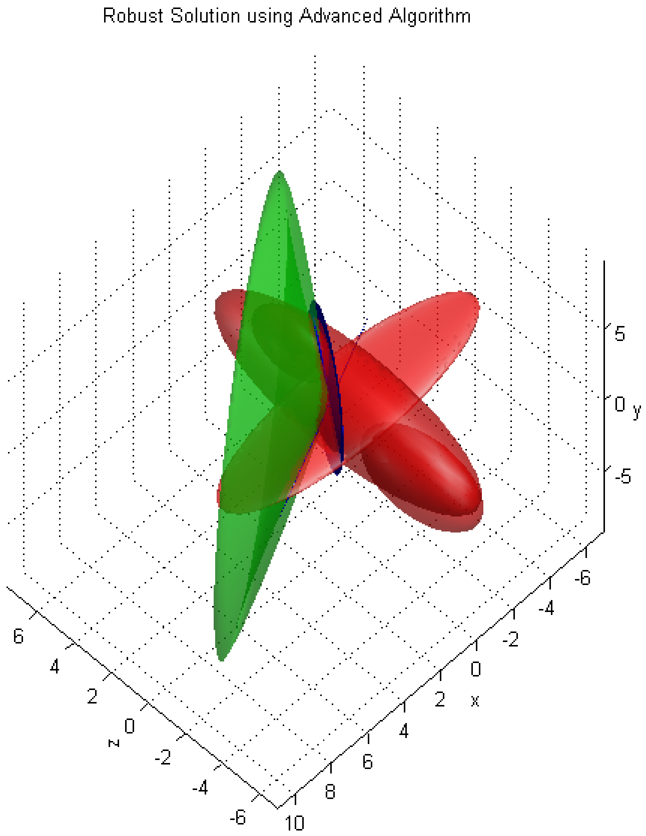


Figure 7.14: Example Orientation from Advanced Fusing Algorithm

Summary

A number of fusing strategies have been evaluated, both in terms of actual lethality and the associated robustness of lethality, to investigate how the lethality varies if the actual endgame scenario is not achieved. Strategies such as minimum distance and threshold level fusing provide a simple method of choosing when to fuse a missile; however they do not provide very robust solutions. Fusing matrix strategies based on original and optimal values whilst yielding high values of lethality again did not result in very robust solutions. The advanced algorithm strategies improved upon the fusing matrices and yielded solutions that not only provided high levels of end game lethality but also an increase in the robustness of the solutions.

8. Conclusions and Future Work

8.1. Conclusions

This thesis has investigated possible missile warhead fusing strategies which may be used in order to provide robust, high lethality probability engagement conditions for an air-to-air missile system.

The concept of endgame lethality and how it can be calculated was discussed. The key parameters of the engagement geometry were defined and illustrated. AGILE, the missile endgame lethality probability calculator, which uses Gaussian functions to represent the missile warhead and target aircraft, and performs operations on the Gaussian functions to calculate the level of damage inflicted on a target aircraft. A target aircraft was defined and a simple random search performed using the endgame parameters which highlighted how large and complex the search space was, and that optimisation may be an effective method with which to improve the level of lethality.

Following the initial introduction to lethality probability, a review of fast optimisation methods was performed, including a discussion on the aspects of sensitivity to disturbances and the resulting robustness of the prime solution found from optimisation. Optimisation of a representative set of endgame scenarios was performed within Matlab to establish if an optimal exists for each of the scenarios, and a corresponding robustness measure was found to assess how sensitive the optimal solution was to disturbances in the missile parameters.

A missile fly-out simulator, MSTARS, was described and its interface with AGILE discussed. Following this some example fly-out scenarios were analysed to see how lethality probability varied in the final stages of fly-out. It was seen that for some solutions it would have been beneficial to fuse the missile at an earlier point along the fly-out in order to produce a higher lethality than the eventual fusing point. It was proposed that optimisation could be used to reclaim some of the ‘lost’ lethality for some of the fly-out scenarios, and also that by optimising the missile’s aim points along the fly-out trajectory that fusing could occur earlier.

Missile trajectory data was used to calculate lethality probability data for points along the missile trajectory. This allowed the examination of how lethality varied, and to ascertain if it had been suitable to fuse the missile earlier for simulations that displayed lower final lethality values. This study showed that for some cases the endgame lethality was lower at the end of the simulation compared to some points along the missile trajectory.

Each point along the trajectory was then optimised for the missile aim points in order to see if lethality could be increased for cases that showed lower lethality values, and a robustness measure was also calculated to see how sensitive these optimal conditions were to variations in the optimised parameters

From these studies it was suggested that a decision process could be utilised to provide a strategy in order to obtain more desirable endgame conditions, especially for cases whereby the missile has fused too late and resulted in lower lethality probabilities compared to previous points along the missile path.

Each possible fusing strategy was described and a framework for how each would be implemented was documented. The strategies included:

- A simple decision maker that look at just one parameter (minimum distance or a threshold lethality value).

- Conditional or rule based methods were then described that were more complex than the simple methods and looked at various conditional rules in a fusing matrix that needed to be considered for the decision to fuse the missile to be made.

- Two advanced fusing algorithms were developed that utilised past simulation data to aid the fusing decision process.

The fusing strategies have been implemented and evaluated, both in terms of actual lethality and robustness of lethality. Strategies such as minimum distance and threshold level fusing provided a simple method of choosing when to fuse a missile; however they did not provide very robust solutions.

Fusing matrix strategies based on the original and optimal lethality values did yield high values of endgame lethality probability, but again did not result in very robust solutions.

The advanced algorithm strategies improved upon the fusing matrix strategies and yielded solutions that not only provided high levels of end game lethality but also an increase in the robustness of the solutions.

From these studies it can be seen that the use of the advanced algorithm would be beneficial as it would allow the missile to trigger its fuse in such a manner that the lethality would be robust to potential variations in the aim point of the missile warhead.

8.2. Contributions

A review of optimisation techniques has been performed (Section 3) which looked at the various methods of optimisation available within Matlab.

The development of various fusing strategies has been performed based on observations during the undertaking of this research (Sections 4, 5, & 6). This included the development of the rules used for the fusing matrix strategy and the advanced fusing algorithms that look at past lethality data to aid the decision process of when to fuse the missile.

Analysis of the developed missile fusing strategies has been undertaken, and it was established that an advanced knowledge based decision process using on board calculations and optimisation can enhance the lethality probability for a maille-target endgame scenario.

Initial multi-objective optimisation work has been performed which could be carried forward to develop a multi-objective optimisation based fusing algorithm. A paper has been published at the 16th IFAC World Congress in 2005 which highlights the multi-objective work initially performed.

8.3. Future Work

Possible work that could be performed in the future includes:

- Refinement of the fusing matrix and advanced fusing algorithm methods to further improve the number of solutions which would yield higher endgame lethality probability values, which are also robust.

- Further study of the fly-out of the missile to see if improvements can be made to the MSTARS model set up. This can involve the inclusion of evasive target manoeuvring concepts, which can be used to add to the knowledge based used for the warhead fuse trigger decision process.

- Further study of other optimisation techniques that could potentially be used as an 'on board' system to aid the fusing decision process of the missile warhead, possibly incorporating system noise into the parameter definitions.

- Continuation of initial work performed using multi-objective optimisation, which could lead to the finding of possible solutions or scenarios that yield both high lethality and highly robust fuse points.

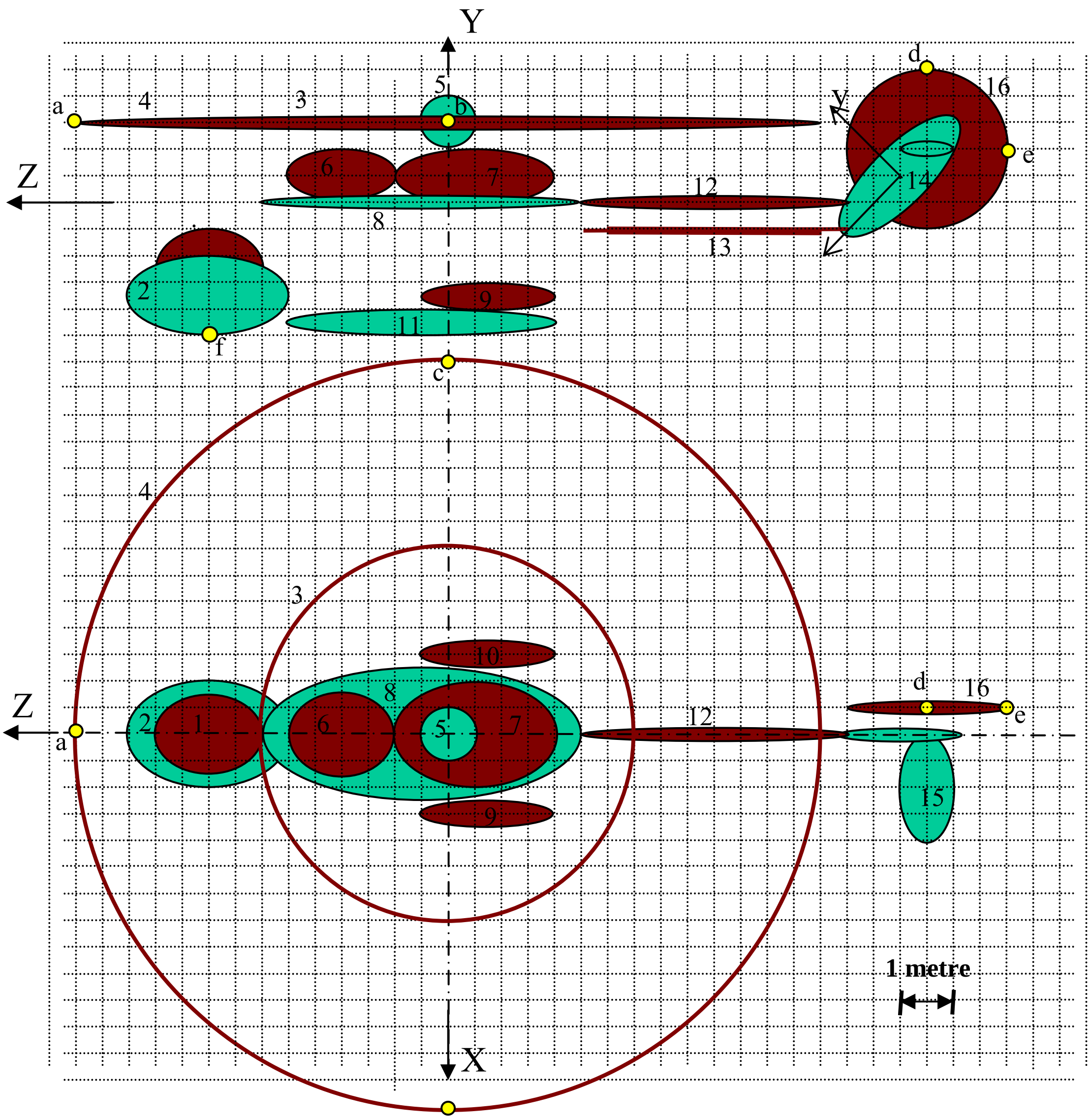
Appendix A. AGILE Target Definitions

A.1. Helicopter Definition

A helicopter used in Multi Objective Optimisation Studies, is defined [30] using the following data. Figure A.1 shows the graphical representation of the components of the helicopter.

```
% Crew
display_target 1
amp   0
mean  0 -1.25 4.5
stdev 0.265 0.265 0

% Cockpit Structure
display_target 2
amp   0
mean  0 -1.75 4.5
stdev 0.354 0.265 0.530

% Main Rotor Inner Ellipsoid
display_target 3
amp   0
mean  0 1.5 0
stdev 1.237 0.035 1.237

% Main Rotor Outer ellipsoid
display_target 4
amp   0
mean  0 1.5 0
stdev 2.475 0.035 2.475

% Main Rotor Hub
display_target 5
amp   0
mean  0 1.5 0
stdev 0.177 0.177 0.177

% Engines
display_target 6
amp   0
mean  0 0.5 2
stdev 0.265 0.177 0.354

% Hydraulics & Gearbox
display_target 7
amp   0
mean  0 0.5 -0.5
stdev 0.354 0.177 0.530

% Transmission platform
display_target 8
amp   0
mean  0 0 0.5
stdev 0.442 0.035 1.061
```

```
% Port Store
display_target 9
amp   0
mean  1.5 -1.75 -0.75
stdev 0.088 0.088 0.442

% Starboard Store
display_target 10
amp   0
mean  -1.5 -1.75 -0.075
stdev 0.088 0.088 0.442

% Fuel Tanks
display_target 11
amp   0
mean  0 -2.25 0.5
stdev 0.442 0.088 0.884

% Drive Shaft
display_target 12
amp   0
mean  0 0 -5
stdev 0.025 0.025 0.884

% Tail Controls
display_target 13
amp   0
mean  0 -0.5 -5
stdev 0.007 0.007 0.884

% Tail Fin
display_target 14
amp   0
mean  0 0.5 -8.5
stdev 0.035 0.177 0.530

% Stabiliser
display_target 15
amp   0
mean  1 1 -9
stdev 0.354 0.035 0.177

% Tail Rotor Disc
display_target 16
amp   0
mean  -0.5 1 -9
stdev 0.035 0.530 0.530
```



Description:

1. Crew
2. Cockpit structure
3. Main rotor inner ellipsoid
4. Main rotor outer ellipsoid
5. Main rotor hub
6. Engines
7. Hydraulics & gearbox
8. Transmission platform structure
9. &10. Stores
11. Fuel tanks
12. Drive shaft
13. Tail controls
14. Tail fin
15. Stabiliser
16. Tail rotor disc

Vulnerable components 🟥

Invulnerable components and/or shielding 🟩

Figure A.1: Helicopter Target Geometry

Appendix B. Multi-Objective Optimisation

Evolutionary algorithms are based on computational models of fundamental evolutionary processes such as selection, recombination and mutation. An overview of a general evolutionary algorithm is shown in Figure B.1. Individuals, or current approximations, are encoded as strings composed over some alphabet, i.e. binary, integer, real-valued, etc., and an initial population is produced by randomly sampling these strings. Once a population has been produced it may be evaluated using an objective function or functions that characterise an individual's performance in the problem domain. The objective function(s) is also used as the basis for selection and determines how well an individual performs in its environment. A fitness value is then derived from the raw performance measure given by the objective function(s) and is used to bias the selection process towards promising areas of the search space. Highly fit individuals will be assigned a higher probability of being selected for reproduction than individuals with a lower fitness value. Therefore, the average performance of individuals is expected to increase as fitter individual are more likely to be selected for reproduction and the lower fitness individuals get discarded. Individuals can be selected more than once at any generation (iteration) of the EA.

```
Procedure EA {
      t = 0;
      initialise P(t);
      evaluate P(t);
      while not finished do {
            t=t+1;
            select P(t) from P(t-1);
            reproduce pairs in P(t);
            mutate P(t);
            evaluate P(t);
      }
}
```

Figure B.1: An Evolutionary Algorithm

Selected individuals are the reproduced, usually in pairs, through the application of genetic operators. These operators are applied to pairs of individuals with a given probability and result in new offspring that contain material exchanged from their parents. The offspring from reproduction are further perturbed by mutation. These new individuals then make up the next generation. The processes of selection, reproduction and evaluation are then repeated until some terminal criteria are satisfied, e.g. a certain number of generations completed, a mean deviation in the performance in the population, or when a particular point in the search space is reached.

B.1. Multi-Objective Optimisation

The use of multi-objective optimisation (MO) in engineering design recognises that most practical problems involve a number of design criteria that need to be satisfied simultaneously, such that:

$$\min_{x \in \Omega} \quad G(x) \tag{B.1}$$

where $x = [x_1, x_2, \ldots, x_n]$ and Ω define the set of free variables, x, subject to any constraints and $\boldsymbol{G(x)} = [g_1(x), g_2(x), \ldots, g_n(x)]$ are the design objectives to be optimised.

For this set of functions, $\boldsymbol{G(x)}$, it can be seen that there is no one ideal optimal solution, but rather a set of solutions for which an improvement in one design objective will lead to a degradation in one or more of the other objectives. This set is known as the Pareto-optimal solution set. These solutions are also known as non-dominated solutions to the MO optimisation problem.

These solutions can be sought after using the NP methods discussed earlier by means of applying weighting and goal attainment functions for the objectives, however these approaches require precise expression of a usually not well understood set of weights and goals. In addition to this NP methods can not handle multimodality and discontinuities in the function space well, and so are likely to find local solutions only.

Evolutionary Algorithms (EA) on the other hand, do not require derivative information or a formal initial estimate of the solution region. Because of the stochastic nature of the search mechanism, genetic algorithms (GA) are capable of searching the entire solution space with more likelihood of finding the global optimal than conventional methods. Conventional methods usually require the objective function to be well behaved, whereas the generational nature of GAs can tolerate noisy, discontinuous and time-varying function evaluations. Furthermore EAs allow the use of mixed decision variables (binary, n-ary and real-values) that allows the parameterisation to closely match the nature of the problem.

It has been shown that EAs can offer an advantage over conventional methods in optimal design problems and the related field of performance seeking control [26].

B.2. Multi-Objective Genetic Algorithms

The idea of the fitness of an individual solution estimate and the associated objective function value are closely related in a single objective framework. The objective function characterises the problem domain and cannot be changed at will, whereas the fitness of an individual can change depending on the solutions ability to reproduce and as such can be treated as part of the GA search strategy. However with the multi-objective case, these two values cannot be linked so closely, and the distinction between them becomes more important. As described by Fleming and Fonseca [15], this distinction becomes important when performance is measured as a vector of the objectives, because the fitness value must remain a scalar. Individual are assigned a measure of utility dependant on whether they perform better, worse, or similar to others in the population.

B.2.1. Decision Strategies

In the absence of any information regarding the relative importance of design objectives, Pareto-dominance is the only method of determining the relative performance of solutions. Non-dominated individuals are all therefore considered to be the best performers and are thus assigned the same fitness, e.g. zero. However determining the fitness of dominated solutions is a more subjective matter. An approach that can be used is to assign a cost proportional to the number of individuals in a population that dominate a given individual, as illustrated in Figure B.2. In this instance non-dominated individual are treated as desirable.

If goal and/or priority information is available for the design objectives, then it may be possible to differentiate between some non-dominated solutions. For example, if degradation in an individual's objectives still allow those goals to be maintained but also allow the attainment of some goals in other non-satisfied objectives, then these degradations should be accepted. In cases where different priority levels are set for each objective then it is important to improve the high priority objective, such as hard constraints, after which the lower priority objectives may be improved.

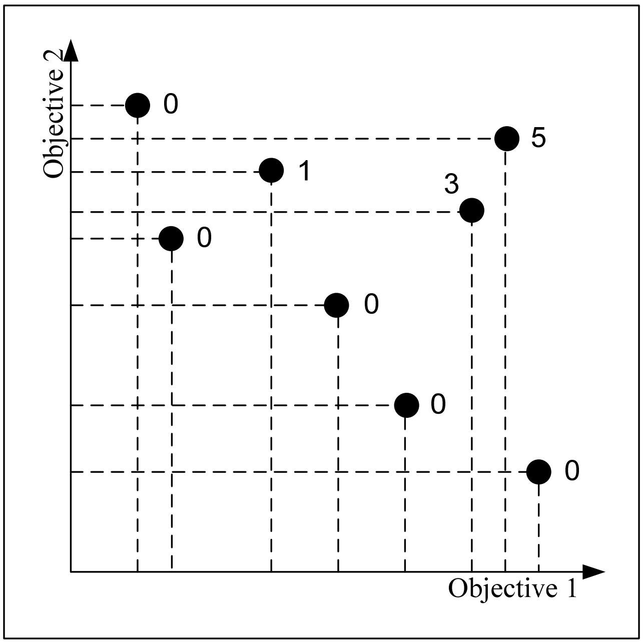


Figure B.2: Pareto Ranking

These considerations have been formalized in terms of a transitive relational operator, *preferability*, based on Pareto-dominance, which selectively excludes objectives according to priority and goal attainment [31]. For simplicity one level of priority is considered as follows. Consider two objective vectors **u** and **v** and their corresponding set of design goals, **g**. Let the smile $\breve{\mathbf{u}}$ denote the components of **u** that meet their goals and the frown $\widehat{\mathbf{u}}$ those that do not. Assuming minimisation, one may then write

$$\mathbf{u}^{\breve{\mathbf{u}}} \leq \mathbf{g}^{\breve{\mathbf{u}}} \quad \wedge \quad \mathbf{u}^{\widehat{\mathbf{u}}} > \mathbf{g}^{\widehat{\mathbf{u}}} \qquad \text{(B.2)}$$

where the inequalities apply component wise. This is equivalent to,

$$\forall i \in \breve{\mathbf{u}}, u_i \leq g_i \quad \wedge \quad \forall i \in \widehat{\mathbf{u}}, u_i > g_i \qquad \text{(B.3)}$$

where u_i and g_i represent the components of **u** and **g** respectively. Then, **u** is said to be preferable to **v** given **g** if and only if

$$(\mathbf{u}^{\hat{\mathbf{u}}} \prec \mathbf{v}^{\hat{\mathbf{v}}}) \vee \left\{(\mathbf{u}^{\hat{\mathbf{u}}} = \mathbf{v}^{\hat{\mathbf{v}}}) \wedge \left[(\mathbf{v}^{\breve{\mathbf{u}}} \leq \mathbf{g}^{\breve{\mathbf{u}}}) \vee (\mathbf{u}^{\breve{\mathbf{u}}} \prec \mathbf{v}^{\breve{\mathbf{u}}})\right]\right\} \tag{B.4}$$

where $\mathbf{a} \prec \mathbf{b}$ is used to denote that **a** dominates **b**. Hence **u** will be preferable to **v** if and only if one of the following is true:

The violating components of **u** dominate the corresponding components of **v**.

The violating components of **u** are the same as the corresponding components in **v**, but **v** violates at least on goal.

The violating components of **u** are equal to the corresponding components of **v**, but **u** dominates **v** as a whole.

B.2.2. Fitness Mapping and Selection

After a cost has been assigned to each individual, the selection of individuals can take place. There are many schemes that exist, including rank-based cost-to-fitness mapping [32] followed by stochastic universal sampling [33], or tournament selection, also based on cost, as described by Ritzel et al. [34].

Exponential rank-based fitness assignment is shown in Figure B.3. Individuals are sorted by their cost (in this case the values from Figure B.2 previously), and assigned fitness values according to an exponential rule in the first instance, shown by the narrow bars in Figure B.3. A single fitness value is then derived for each group of individuals sharing the same cost. Through averaging, and is shown in the figure by the wider bars.

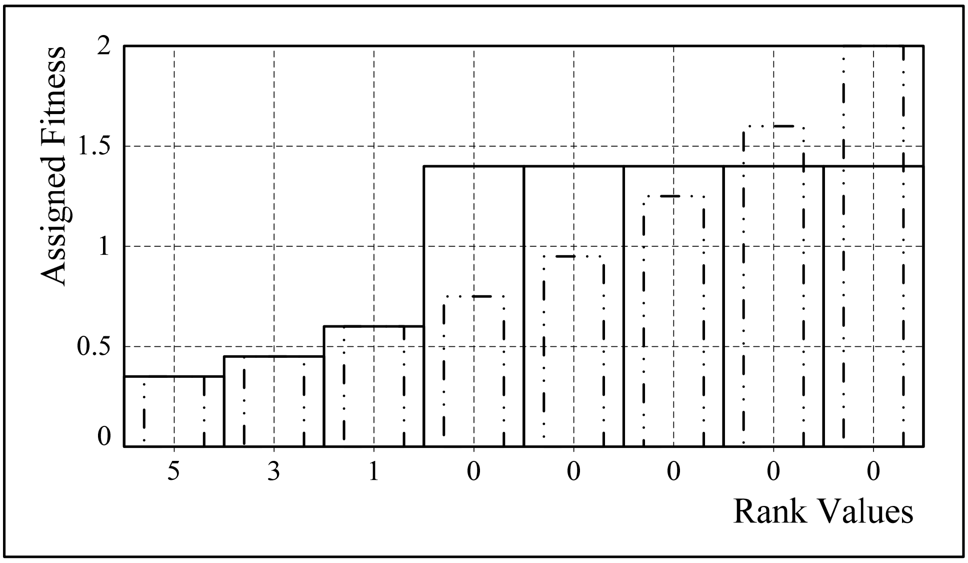


Figure B.3: Rank-Based Fitness Assignment

B.2.3. Fitness Sharing

Even though all preferred individuals in the population are assigned the same level fitness, the number of offspring they produce may differ due to the stochastic nature of EAs. Over generations, these imbalances may accumulate resulting in the population focussing on an arbitrary area of the trade-off surface, known as genetic drift [35]. Additionally, recombination and mutation may be less likely to produce individuals at certain area of the trade-off surface, e.g. the extremes, giving only a partial coverage of the trade-off surface.

Originally introduced as an approach to sampling multiple fitness peaks, fitness sharing [36] helps counteract the effects of genetic drift. This is done by penalising individuals according to the number of others in their neighbourhood. Each individual is assigned a niche count, initially set to zero, which is incremented by a certain amount for every individual in the population, including itself. A sharing determined the contribution or other individuals to the niche count as a function of their mutual distance in genotype, phenotype, or objective space. Raw fitness values are then weighted by the inverse of the niche count and normalised by the sum of the weights prior to selection. The total fitness of the population is redistributed, and thus shared, by the population. However a problem with the use of fitness sharing is the difficulty in determining the niche size, σ_{share}, i.e. how close individuals may be before degradation occurs.

An alternative, but analogous, approach to niche count computations are kernel density estimation methods [37] as used by statisticians. Instead of a niche size, a smoothing parameter, *h*, whose value is ultimately subjective, is used. However guidelines have been developed for suitable selection of the value of *h* for certain kernels, such as the standard normal probability density function and Epanechnikov kernels. The Epanechnikov kernel may be written as [38]

$$K_e(d/h) = \begin{cases} \frac{1}{2} c_n^{-1}(n+2)\left[1-(d/h)^2\right] & \text{if} \quad d/h < 1 \\ 0 & \text{otherwise} \end{cases} \tag{B.5}$$

where *n* is the number of decision variable, c_n is the volume of the unit *n*-dimensional sphere, and *d/h* is the normalised Euclidean distance between individuals.

Silverman [37] gives a smoothing factor that is approximately optimal in the least mean integrated square error sense when the population follows a multivariate normal distribution for the Epanechnikov kernel $K_e(d)$as

$$h = \left[8c_n^{-1}(n+4)\left(2\sqrt{\pi}\right)^n / N \right]^{1/(n+4)} \tag{B.6}$$

for a population with *N* individuals and identity covariance matrix. Where populations have an arbitrary sample covariance matrix , *S*, this may simple be 'sphered', or normalised, by multiplying each individual by a matrix *R* such that $RR^T=S^{-1}$. This means that the niche size, which depends on *S* and *h*, may be automatically and constantly updated, regardless of the cost function, to suit the population at each generation.

B.2.4. Mating Restriction

Mating restrictions can be employed to bias the way the in which individuals are paired for reproduction [38]. Recombining arbitrary individuals form along the trade-off surface may lead to the production of a large number of unfit offspring, known as *lethals*, which could adversely affect the performance of the search. To alleviate this potential problem, mating can be restricted, where feasible, to individuals from within a given distance of each other, σ_{mate}. A common practise is to set $\sigma_{mate}=\sigma_{share}$ so that individuals are allowed to mate with one another only if they lie within a distance *h* from each other in the 'sphered' space used for sharing [15].

B.2.5. Interactive Search and Optimisation

As the population of the MOGA evolves, trade off information will be acquired. In response to the optimisation so far, one may want to investigate a smaller region of the search space, or even move to a totally new region. This can be achieved by resetting the goals supplied to the MOGA which, in turn, affects the ranking of the population and modifies the fitness landscape concentrating the population on a different area of search space. The priority of design objectives may also be changed interactively using this scheme.

The introduction of a small number of random individuals at each generation has been shown to make the EA more responsive to sudden changes in the fitness landscape, as occurs when the optimisation is changed interactively [39].

B.3. Review of Current Multi-Objective Genetic Algorithms

The main differences between various Multi-Objective Genetic Algorithms are the methods by which the processes of fitness selection, recombination and mutation are used to maintain a set of solutions that are evenly distributed along the Pareto front.

B.3.1. VEGA

Schaffer, 1985

An early form of a multi-objective genetic algorithm, presented by Schaffer [40], is the Vector Evaluated Genetic Algorithm (VEGA). VEGA involved using sub-populations of the original population. Each sub-population is made by calculating one objective function at a time, rather than aggregating all objectives. Selection is performed by computational loops, whereby at each loop the fitness of an individual is evaluated using a single objective function. Members are selected for the next generation using stochastic selection methods. This selection process is repeated for each objective function. For example, a problem with *k* objectives, *k* sub-populations will be created, each with *N/k* individuals, where *N* is the total population size. The sub-populations are shuffled together to create a new generation. This is similar to using weighted sums for objectives.

This process is based on the notion that minimum of an objective is a unique Pareto optimal point, and as such these would define the vertices of the Pareto optimal set. Schaffer's method however, does not necessarily yield an evenly distributed set of Pareto optimal points, as solution tend to cluster about each individual objective's minimum. The resulting cluster is referred to as a *species*, which are groups of solutions that share common attributes.

Schaffer proposed two solutions, the cross-breeding of sub-populations (species), and that non-dominated solutions be given a selection preference.

B.3.2. Ranking

Some Alternatives to VEGA were described by Goldberg [41], Fonseca & Fleming [15], Srinivas & Deb [42], and Cheng & Li [43]. They proposed assigning a fitness ranking system to the population based on an individual's dominance within the population. All non-dominant members are assigned a rank of 1, and temporarily discarded. The next set of non-dominated solutions relative to this reduced population is then assigned rank 2, and so on. Therefore the fittest solutions have lowest rank value, i.e. Fitness is inversely proportional to rank. There are many methods described ([24], [15], [41], and Narayanan & Azarm [44]). Another method, Belegundu et al [45], suggests that high ranking members (i.e. those with low fitness values) should be discarded and replaced with new randomly created individuals.

B.3.3. Pareto-Set Filter

Sometimes it is possible that a Pareto optimal point does not survive to the next generation. To overcome this issue, Cheng & Li [46] suggest the use of a Pareto Set Filter.

At each generation, two sets of solutions are created, the current population of solutions, and the Filter which provides an approximation to the theoretical Pareto optimal set. This Filter is the set of all non-dominated solutions. At each generation new solutions with rank 1 are added to the filter and checked for non-dominance within the set. Any dominated solutions within the Filter are discarded. The size of the Filter is usually the size of the population. When the Filter reaches capacity, new solutions replace a solution that is situated close-by to another within the set. This helps maintain a distribution of points along the Pareto front, and would eventually converge to the true Pareto front.

B.3.4. Elitist Strategy

A similar approach to the Pareto-set filter was proposed by Ishibuchi & Murata [47] called Elitist Strategy, which functions independent of rank. As before two sets of solutions are created, the current population and a ‘tentative set of non-dominated solutions’, which is an approximate Pareto set. All points in the current set that are non-dominated by points in the tentative set are added to the tentative set. Then dominated solutions in the set are discarded. Following crossover and mutation a user specified number of solutions are reintroduced to the current population. These points are called elite points. Following up from this procedure Murata et al [48] suggest that *x* number of solutions with best values for each objective can be regarded as elite points and be kept for the next generation.

B.3.5. Tournament Selection

Another method for the selection process, Tournament Selection, developed by Horn et al [49], involves choosing two solutions from the population at random, called candidate points. These two will compete for survival into the next generation. A second set called the tournament set (or comparison set) is generated, again using random solution from the current population. The two candidate points are compared with each member of the tournament set for dominance. If one of the candidate points is non-dominant relative to the tournament set, it is selected for the next generation. If there is a tie, or no preference fitness sharing (see below) is used to select the appropriate candidate. Note that the size of the tournament set is crucial in this process. If the set is too small only a few Pareto optimal points will be found. Also if it is too large then the GA may converge prematurely. The size is usually a percentage of the total population size, and relates to the dominance pressure, or degree of difficulty for an individual survivability.

B.3.6. Niche Techniques

When a number of solutions group together it is known as a niche. Niche techniques (also known as niche schemes) are employed to ensure that the GA does not converge to a niche, i.e. a limited number of Pareto points. Therefore these techniques are used to yield an even spread of Pareto points. Multi-Objective GA's tend to cluster around or converge to a limited number of Pareto points. This process is known as genetic drift, and niche techniques aim to develop many of these niches whilst ensuring that each niche does not grow too much.

B.3.6.1. Fitness Sharing

One such niche technique is fitness sharing. It is achieved by penalising the fitness value of points that are located close to one another, effectively reducing the probability of selection for the group of solutions (Goldberg [41], Deb [50], Srinivas & Deb [42]). The fitness value is divided by a constant, *k*, where *k* is proportional to the number of points in the nearby space. Therefore the fitness of all points in this niche are 'shared' in some sense, hence the term fitness sharing.

In the context of tournament selection, if two individuals are dominated or non-dominated, the winner is the one with the fewer individuals close to it. This is known as 'equivalence class sharing'.

B.3.6.2. Preselection

Cheng & Li [43] suggest that if an offspring has a higher fitness value than its parent then it replaces its parent. Children have equivalent or superior characteristics to parents and remain close to parent positions, avoiding drift. This is known as preselection.

Following on from this method, Narayana & Azarm [44] present a method in which a limit is placed on the distance between parents. If they are close together then they are not selected for crossover. Also suggested is that only non-dominated solutions be evaluated for constraint violation and a fitness penalty is assigned to point that violate a constraint.

B.3.7. Non-Dominated Sorting Genetic Algorithm, NSGA-II

Deb, K, Pratap A, Agarwal S, and Meyarivan T, 2002 [51]

NSGA-II is an improved version of the NSGA proposed by Deb et al [42]. The idea behind the non-dominated sorting procedure is that a ranking selection method is used to highlight good points and a niche method is used to maintain stable subpopulations of good points. The algorithm was developed based on these concepts.

B.3.7.1. NSGA (I)

The NSGA varies from the simple genetic algorithm in the way that the selection operators work. The crossover and mutation operators remain as normal. Before the selection process is performed the population is ranked on the basis of an individual's non-dominance of other individuals. The non-dominated individuals present in the population are first identified from the current population. These individuals are then assumed to constitute the first non-dominated front in the population and assigned

a large dummy fitness value. The same fitness value is assigned to give an equal reproductive potential to all these non-dominated individuals. In order to maintain the diversity of the population these individuals are then shared with their dummy fitness values. Sharing is achieved by performing a selection operation using degraded fitness values which are obtained by dividing the original fitness value of an individual by a quantity proportional to the number of individuals around it. This allows multiple optimal points to co-exist in the population. After sharing these non-dominated individuals are ignored temporarily to process the rest of population in the same way to identify individuals for the second non-dominated front. These new set of points are then assigned a new dummy fitness value which is kept smaller than the minimum shared dummy fitness value of the previous group. This process is continued until the entire population is classified into several fronts. The population is then reproduced according to the dummy fitness values assigned. A stochastic remainder proportionate selection is used by the NSGA. Since individuals in the first front have the maximum fitness value they always get more copies than the rest of the population. This is in order to search for non-dominated regions or Pareto-Optimal fronts. This results in quick convergence of the population towards non-dominated regions and sharing helps to distribute it over this Pareto region. The efficiency of NSGA come from the way multiple objectives are reduced to dummy fitness functions using a non-dominated sorting procedure.

Figure B.4 shows a flow chart of this algorithm. The algorithm is similar to a simple GA except that the classification of non-dominated fronts and the sharing operators. The ranking classification is performed according to the non-dominance of the individuals in the population and a distribution of the non-dominated points is maintained using a niche formation technique. Both these aspects cause the distinct non-dominated points to be found in the population.

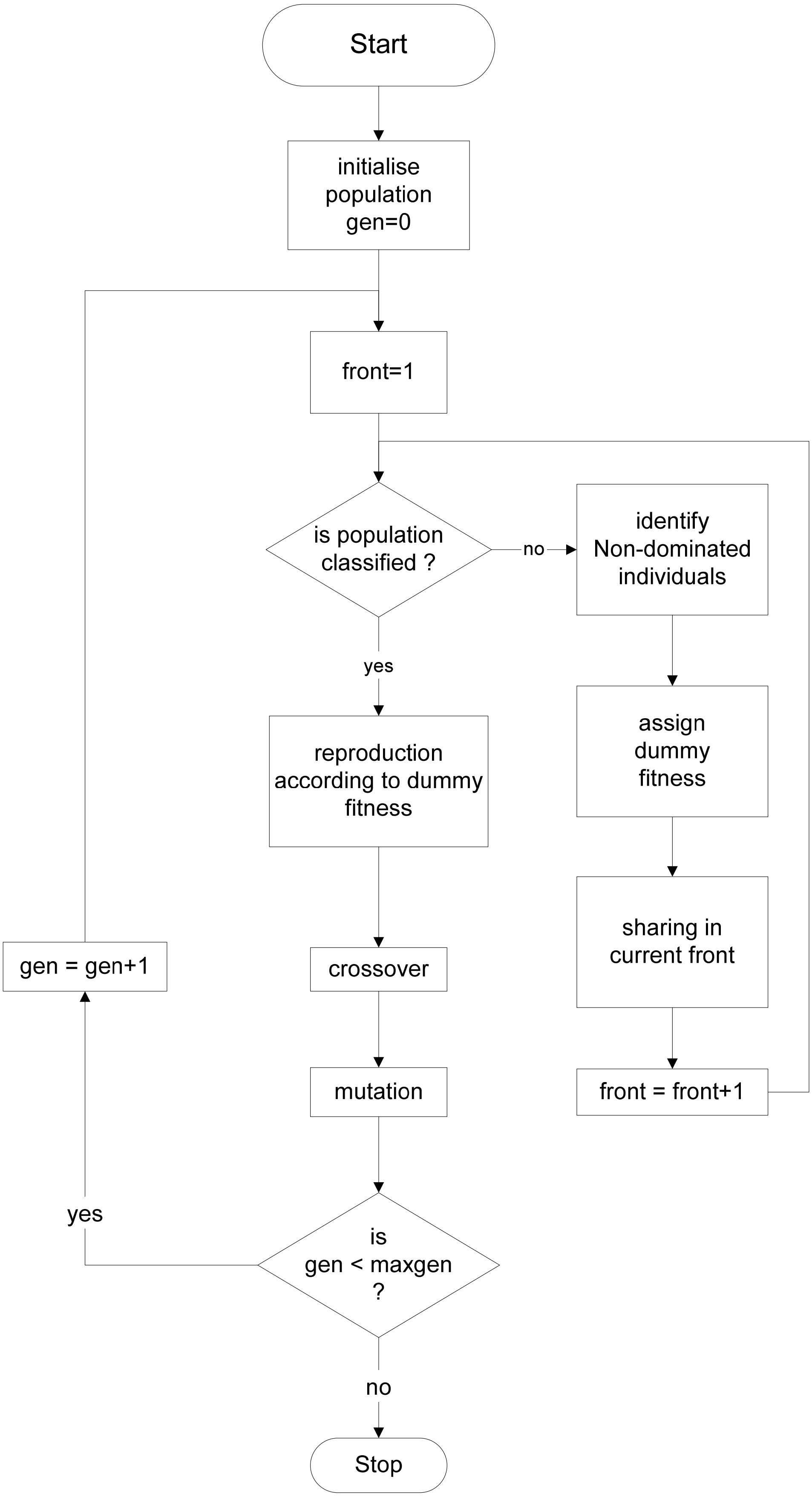


Figure B.4: Flow Chart for the NSGA

This NSGA is computationally expensive for large populations, due to the nature of non-dominated sorted procedure being complex which had to be performed at each generation. The NSGA also lacked elitism, which would result in the loss of some good individuals from the population

B.3.7.2. NSGA-II

The NSGA-II addresses these issues and incorporated techniques to overcome these problems. Using the following description, a fast non-dominated sorting approach is shown which requires fewer computations. Firstly, for each solution two values are calculated:

1. The domination count, which is the number of solutions that dominate the solution, and;
2. A set of solutions that the solution dominates.

All solutions in the first non-dominated front will have their domination count as zero. Then for each solution with a domination count of zero, the members of its set of dominated solutions have their domination counts reduced by one. If by doing so, any member of this set has its domination count become zero, it is put in a separate list, and these members belong to the second non-dominated front. Now, the above procedure is continued with each member of this list and the third front is identified. This process continues until all fronts are identified. Once a solution has a dominance count of zero, the solution is assigned a non-domination level and will never be visited again. Thus, the overall complexity of the procedure is reduced.

As well as convergence to the Pareto-optimal set, it is also desired that a GA maintains a good spread of solutions in the obtained set of solutions. The original NSGA used the well-known sharing function approach, which has been found to maintain sustainable diversity in a population with appropriate setting of its associated parameters. The sharing function method involves a sharing parameter, which sets the extent of sharing desired in a problem. This parameter is related to the distance metric chosen to calculate the proximity measure between two population members. The parameter denotes the largest value of that distance metric within which any two solutions share each other's fitness. This parameter is usually set by the user. There are two difficulties with this sharing function approach:

1. The performance of the sharing function method in maintaining a spread of solutions depends largely on the chosen value;
2. Since each solution must be compared with all other solutions in the population, the overall complexity of the sharing function approach is quite high.

In the NSGA-II, the sharing function approach is replaced with a crowded-comparison approach that eliminates both the above difficulties to some extent. The new approach did not require any user-defined parameter for maintaining diversity among population members. Also, the suggested approach had a better computational complexity. To describe this approach, we first define a density-estimation metric and then present the crowded-comparison operator.

B.3.7.2.1. Density Estimation:

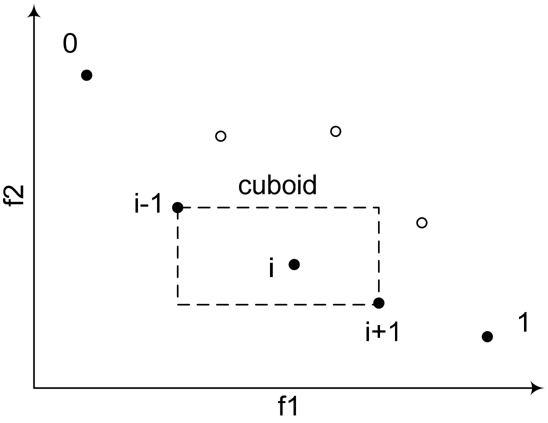


Figure B.5: Crowding Distance

To get an estimate of the density of solutions surrounding a particular solution in the population, we calculate the average distance of two points on either side of this point along each of the objectives. This quantity serves as an estimate of the perimeter of the cuboid formed by using the nearest neighbours as the vertices (call this the crowding distance). In Figure B.5 the crowding distance of the nth solution in its front (marked with solid circles) is the average side length of the cuboid (shown with a dashed box). The crowding distance computation requires sorting the population according to each objective function value in ascending order of magnitude. Then for each objective function, the boundary solutions (i.e. those with smallest and largest function values) are assigned an infinite distance value. All other intermediate solutions are assigned a distance value equal to the absolute normalized difference in the function values of two adjacent solutions. This calculation is continued with other objective functions. The overall crowding distance value is calculated as the sum of individual distance values corresponding to each objective. Each objective function is normalized before calculating the crowding distance.

After all population members in the set are assigned a distance metric, two solutions can be compared for the extent of their proximity with other solutions. A solution with a smaller value of this distance measure is, in some sense, more crowded by other solutions.. Although Figure B.5 illustrates the crowding distance computation for two objectives, the procedure is applicable to more than two objectives as well.

B.3.7.2.2. Crowded Comparison Operator:

The crowded comparison operator guides the selection process at the various stages of the algorithm toward a uniformly spread out Pareto-Optimal front. Assuming that every individual in the population has two attributes:

1. A non-domination rank;
2. A crowding distance.

An order is defined, that between two solutions with differing non-domination ranks, the solution with the lower (better) rank is preferred. Otherwise, if both solutions belong to the same front, then the solution that is located in a lesser crowded region is preferred.

The NSGA-II is implemented using these new innovations (a fast non-dominated sorting procedure, a fast crowded distance estimation procedure,
and a simple crowded comparison operator).

B.3.8. Relational Multi-Objective Genetic Algorithm, RMOGA

Lee, S and Tsui, H, 2004 [52]

The Relation Multi-Objective Genetic Algorithm, RMOGA, proposed a new operator for a GA, inheritance.

B.3.8.1. Inheritance Operator

Inheritance is similar to the crossover operator; however it aims to exchange mathematical relationships, but not values, between two selected sub chromosomes (as opposed to the genes used in crossover). Such relationships can be time, temperature, spatial, hierarchical relationships etc. depending on the application and definition of the chromosome.

The process for this operator is as follows:

1 Find the Relationship function;
2. Calculate genetic relationship between pairs of sub-chromosomes;
3. Swap their relationship;
4. Calculate the values in solution space from the inverse of the relationship function.

To provide an example of this procedure, suppose a set of two chromosomes {C1, C2, C3} and {C4, C5, C6}. Traditional crossover operators would swap sub chromosomes C3 and C6 to give offspring {C1, C2, C6} and {C4, C5, C3}. The relationship between {C2, C3} and {C5, C6} is lost. The best relationship may not propagate to the next generation.

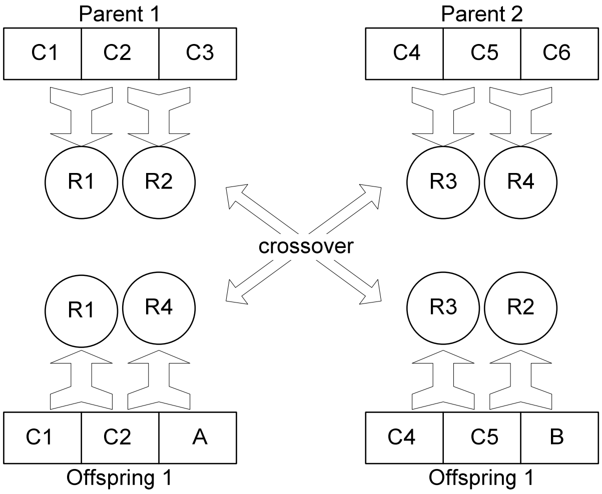


Figure B.6 Inheritance Operation

Figure B.6 illustrates this idea inheritance, which keeps relationships between sub chromosomes. Let R1, R2, R3, R4 be the relationship between {C1, C2}, {C2, C3}, {C4, C5} and {C5, C6} respectively. The relationships are swapped, yielding {R1, R4}, and {R3, R2}. The offspring chromosomes are {C1, C2, A} and (C4, C5, B} where A and B are calculated from the inverse of the relationship function such that the relation ship of {C2, A} equals that of {C5, C6}, and similarly, {C5, B} to {C2, C3}. These relationships are dependent on the problem and can be linear, or non-linear.

B.3.8.2. RMOGA

This can then be implemented within a MOGA framework, show below in Figure B.7, whereby the new offspring can be created from three modes: elitism (direct copy from parents), crossover with mutation (search for optimal at a micro level), and inheritance with mutation (optimal solution search at macro level).

In the selection stage the, the offspring, *N*, from each mode are selected to a given ratio, i.e. 2: (N-1)/2: (N-1)/2.

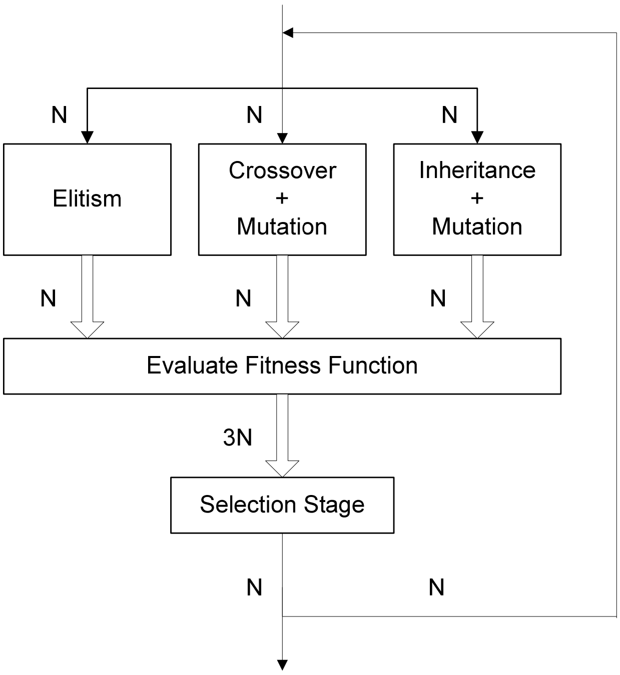
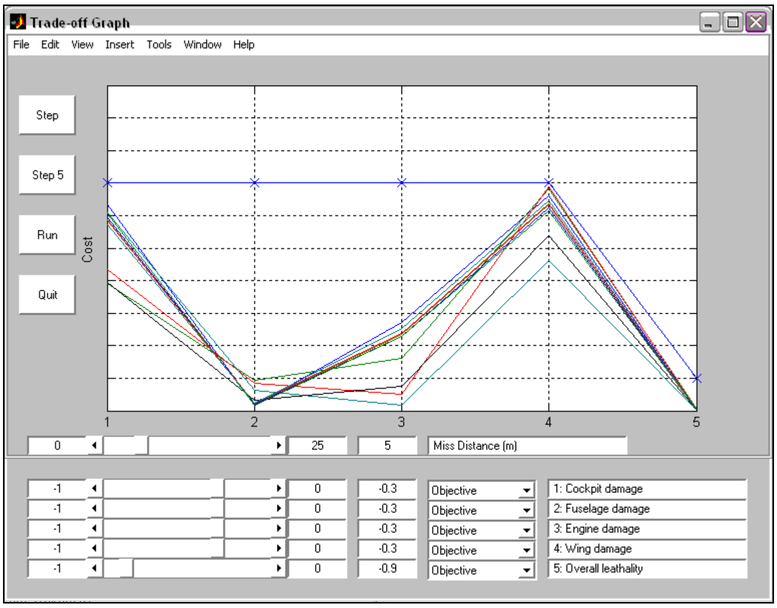


Figure B.7: RMOGA Structure

B.4. Multi Objective Studies

B.4.1. Initial MOGA Analysis

A multi objective optimiser, MOGA, was initially employed using each individual component's lethality probability as individual objectives (wings, engine fuselage, cockpit), with the overall lethality value as a fifth objective. A population of fifty individuals per generation was initialised, using the three parameter setup employed in the previous studies. The MOGA then generated a generation of solutions that provided a measure of how the individual components of P_k interacted with each other. This showed that if the cockpit P_k value was high for example, then the engine P_k was lower, due to its relative position on the aircraft itself. The output for the MOGA optimiser is shown below, in Figure B.8. As can be seen there are many solutions competing for optimality for this particular configuration, shown as the lines on the trade-off graph.



0	25	5	Miss Distance (m)

-1	0	-0.3	Objective	1: Cockpit damage
-1	0	-0.3	Objective	2: Fuselage damage
-1	0	-0.3	Objective	3: Engine damage
-1	0	-0.3	Objective	4: Wing damage
-1	0	-0.9	Objective	5: Overall leathality

Figure B.8: MOGA Optimiser Window

An example of such a case is shown in Figure B.9. the MOGA attempts to optimise maximum damage to all components of the target aircraft and as a result, the cone of fragments is a flat disc shape that hits the aircraft diagonally along the length of the craft in order to inflict damage upon all four components.

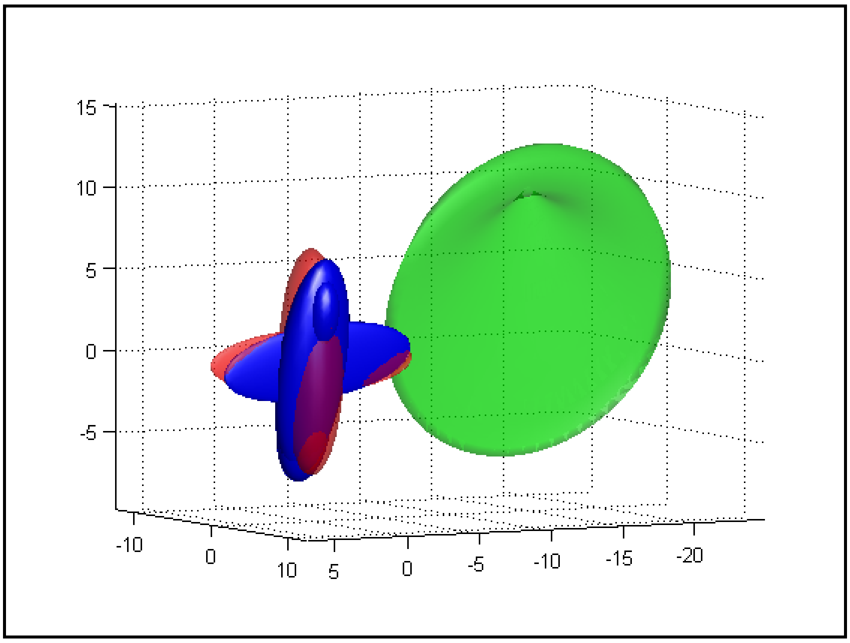


Figure B.9: Example MOGA Optimised Endgame Geometry

B.4.2. Robust MOGA Optimisation

The next stage for the MOGA software was to set the objectives as total P_k and the robustness value from the previous work with the standard Matlab optimisers. This configuration was set such that for each suitable individual (solution) a routine was run that sampled ten deviant (from optimal parameters) solutions as for the Matlab optimiser and the worst case was used as the sensitivity measure. This configuration is shown in Figure B.10.

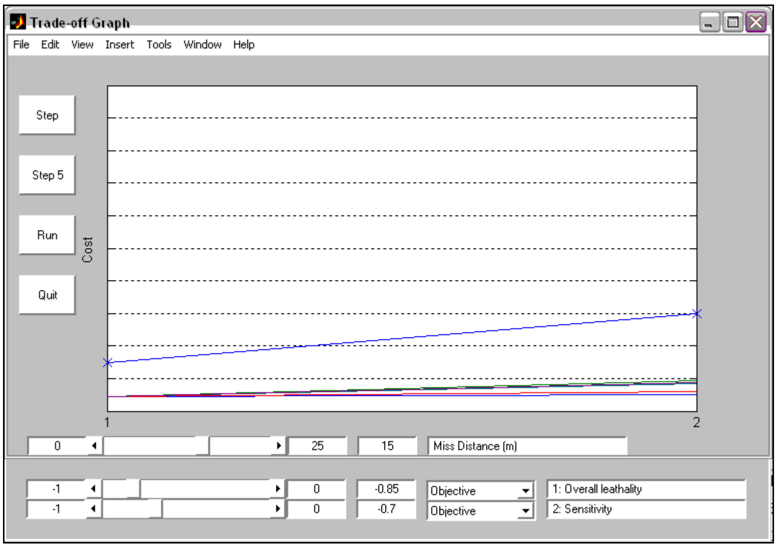
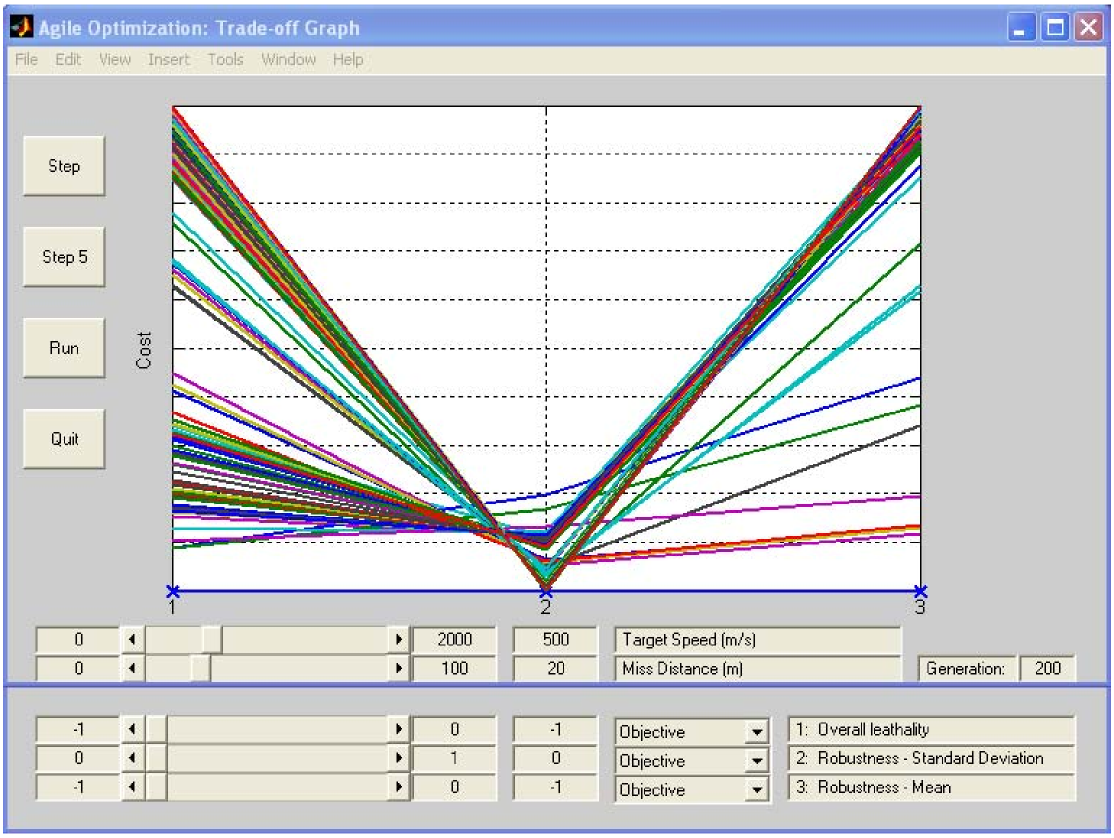


Figure B.10: MOGA Robustness Trade-Off Window

For this setup the minimum value of perturbed samples was used. However, a more suitable measure for this sensitivity is a standard deviation of the perturbed samples. Another implementation of MOGA explored this measure of sensitivity as the second objective, and also a third objective of maximising the mean was implemented, although this is closely connected to the standard deviation, it gives a slightly easier visual of the performance of individuals in the population.

All the endgame parameters were considered for this implementation. The Trade-Off window is coded so that selecting an individual's line would display the corresponding engagement geometry using the AGILE GUI.

Figure B.11 shows a MOGA Trade Off window for this setup. Three runs were undertaken, for front on, side on and rear on scenarios using engagement angle constraints (-45 < η < 45 for rear on, 45 < η < 135 for side on, and 135 < η < 225 for front on), using 50 individual per generation, for 200 generations, and for each individual, 50 perturbed samples are taken to establish the sensitivity measure of standard deviation and mean. The sensitivity measure is calculated by perturbing only those variables that are controllable, i.e. the missile parameters, δ, ε, x_0, y_0, z_0, Z_{delay}.


Agile Optimization: Trade-off Graph
File Edit View Insert Tools Window Help
Step
Step 5
Run
Quit
Cost
1
2
3

0 2000 500 Target Speed (m/s)
0 100 20 Miss Distance (m)
Generation: 200

-1 0 -1 Objective 1: Overall leathality
0 1 0 Objective 2: Robustness - Standard Deviation
-1 0 -1 Objective 3: Robustness - Mean

Figure B.11: Objective MOGA Trade-Off Graph

As can be seen there are many competing solutions present that offer high lethality probabilities which are also robust to perturbations in the missile parameters. These solutions are shown in Figure B.12. The scatter plots of the Pareto Solutions (left), accompanied by the scatter plot of all solution found in 200 generations (right), are shown as overall P_k (nominal) vs. standard deviation.

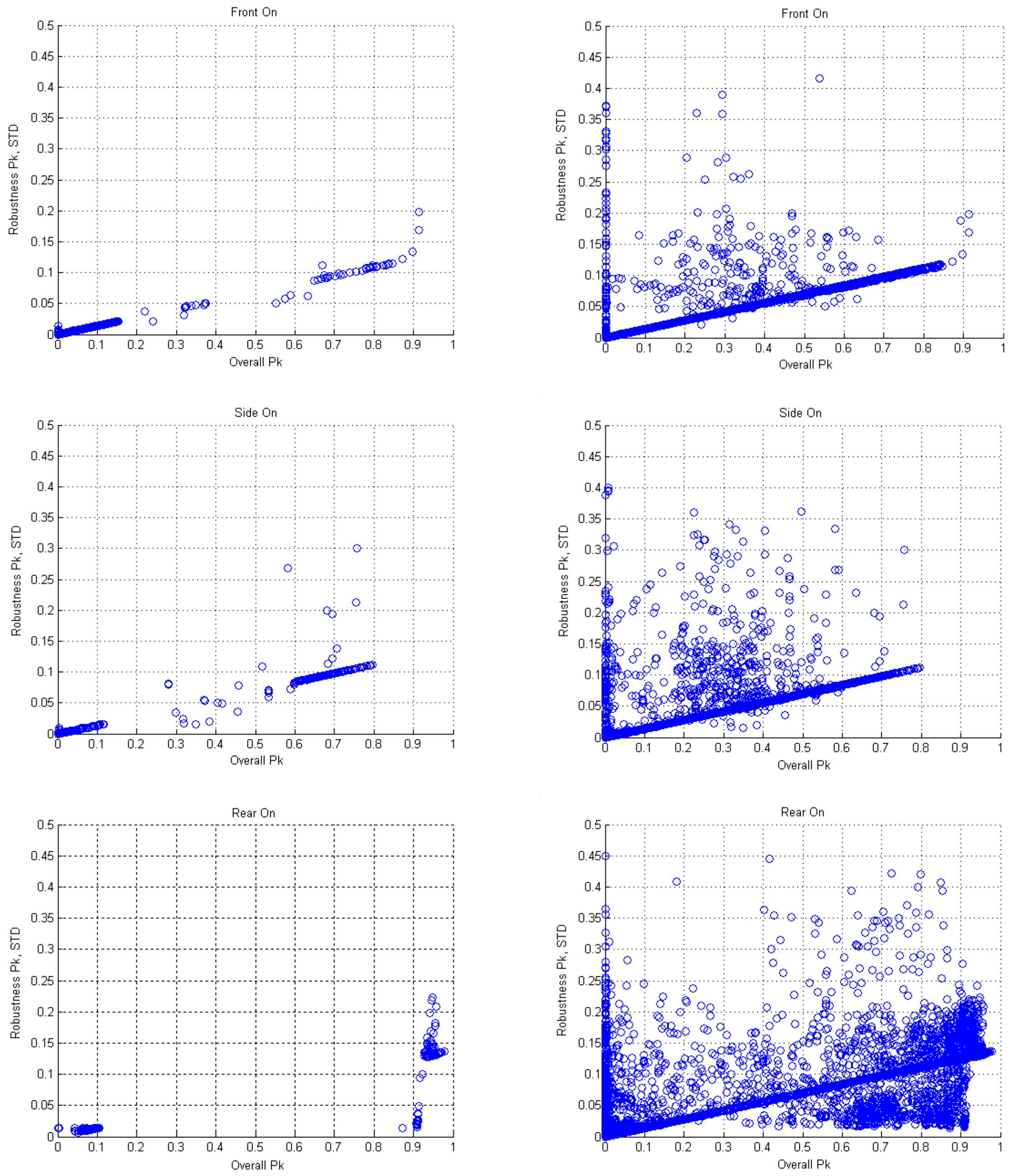
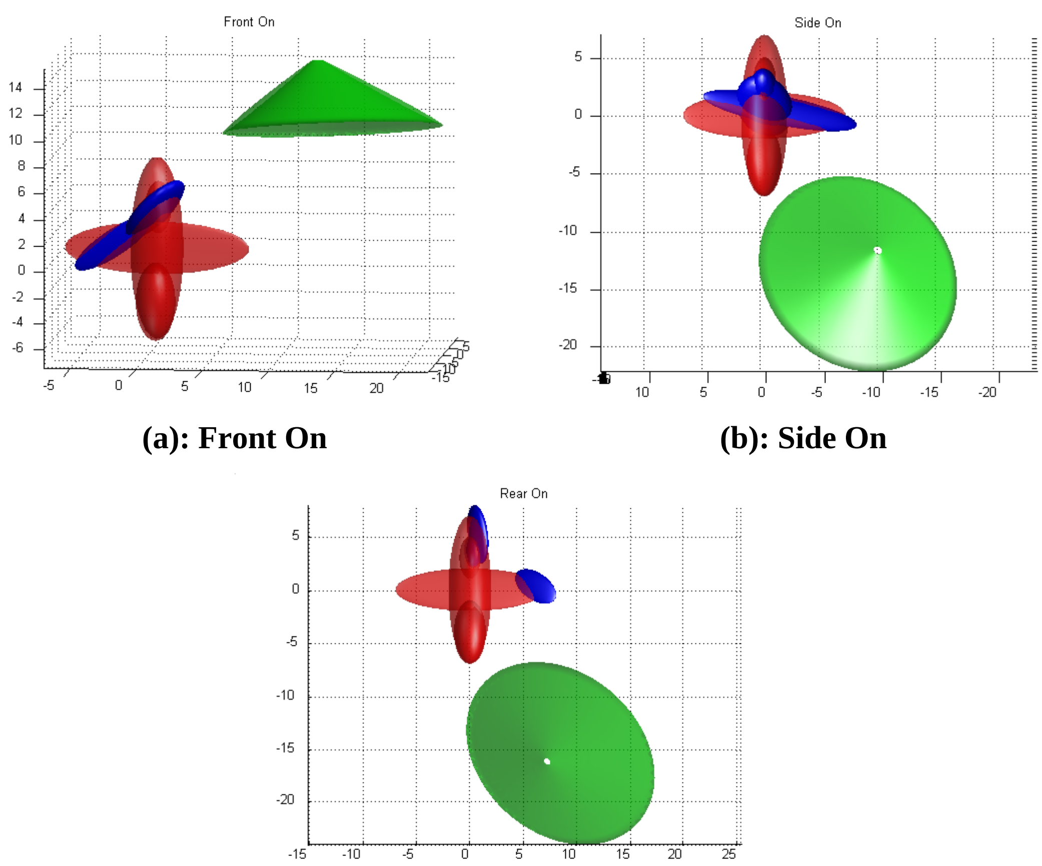


Figure B.12: MOGA Optimisations for 200 Generations

As can be seen many solutions exist, however most are dominated by the Pareto set, and for each case the middle region of P_k yields sparser solutions. Looking at the graphs on the right, for all solutions, a definite trend can be seen showing the increase in standard deviation as overall optimal probability increases, however there do exist some solutions that can provide a good robustness measure, and it is these that show on the Pareto Front.

Some example endgame geometries from the studies above are shown below.



(a): Front On

(b): Side On

(c): Rear On

Figure B.13: Example of Engagement Geometry

In Figure B.13(a) the damage is inflicted on the aircraft across the cockpit and fuselage, with some damage to the wing. This provides a good likelihood of kill as the cockpit is deemed highly vulnerable to damage. The side on engagement, (b), again aims to line up the majority of fragment to hit the wing structure and the cockpit, and this is also fairly robust as slight changes will still inflict quite a lot of damage. The rear on case (c), however shows that although fragments hit the edge of the wing and the cockpit regions it could easily miss the target entirely if the parameters deviate slightly in the wrong direction.

B.4.3. NSGA-II Implementation For AGILE

The above optimisations performed on MOGA were duplicated using the NSGA-II algorithm for comparison. The following results were yielded following a preliminary run on the NSGA-II for 50 generations. In Figure B.14, the upper section shows the nominal (*) and mean (o) values. As can be seen only a few solutions are found compared to the MOGA implementations.

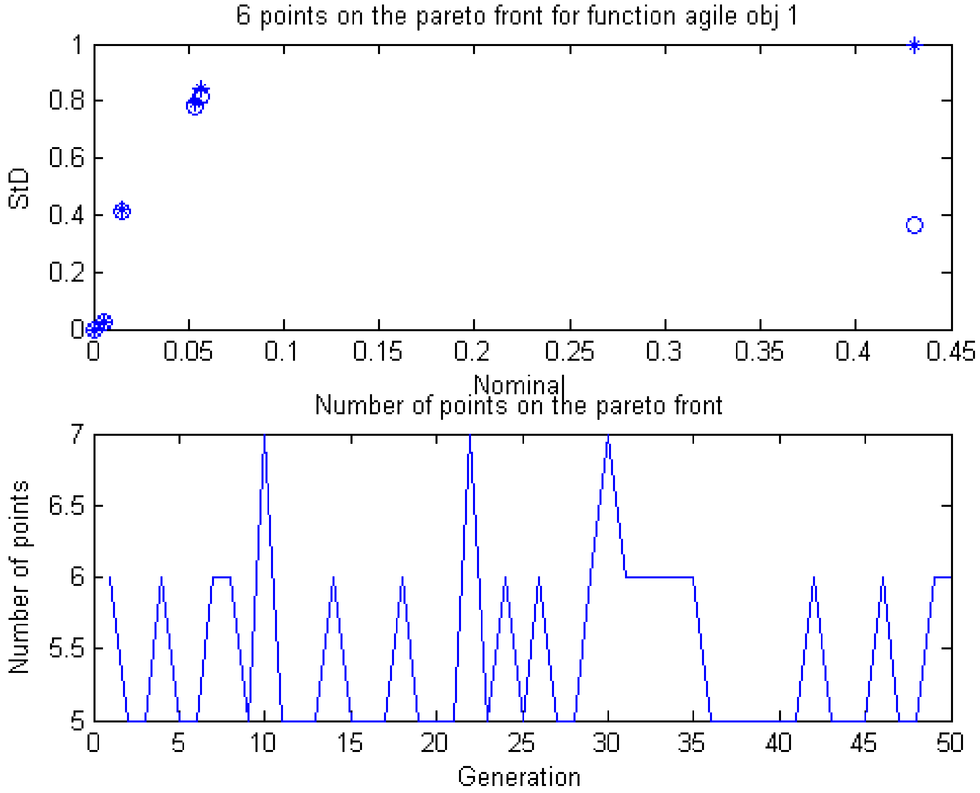


Figure B.14: Initial NSGA-II Optimisation on Simple Aircraft Model

However this result will need to be investigated further to ensure that the NSGA-II has been setup correctly, and this process is currently under way.

B.4.4. Initial Helicopter Model Work

Following the work carried out using the simple fixed wing aircraft model, the helicopter model described in Appendix A has been implemented using the MOGA setup used previously. Three optimisations for front on, side on and rear on engagements were initialised using 50 individuals per generation, 50 perturbations per individual, and for 100 generations. The results are plotted in the graphs in Figure B.15, showing the competing Pareto optimal solutions (left), and all solutions found by the optimiser (right).

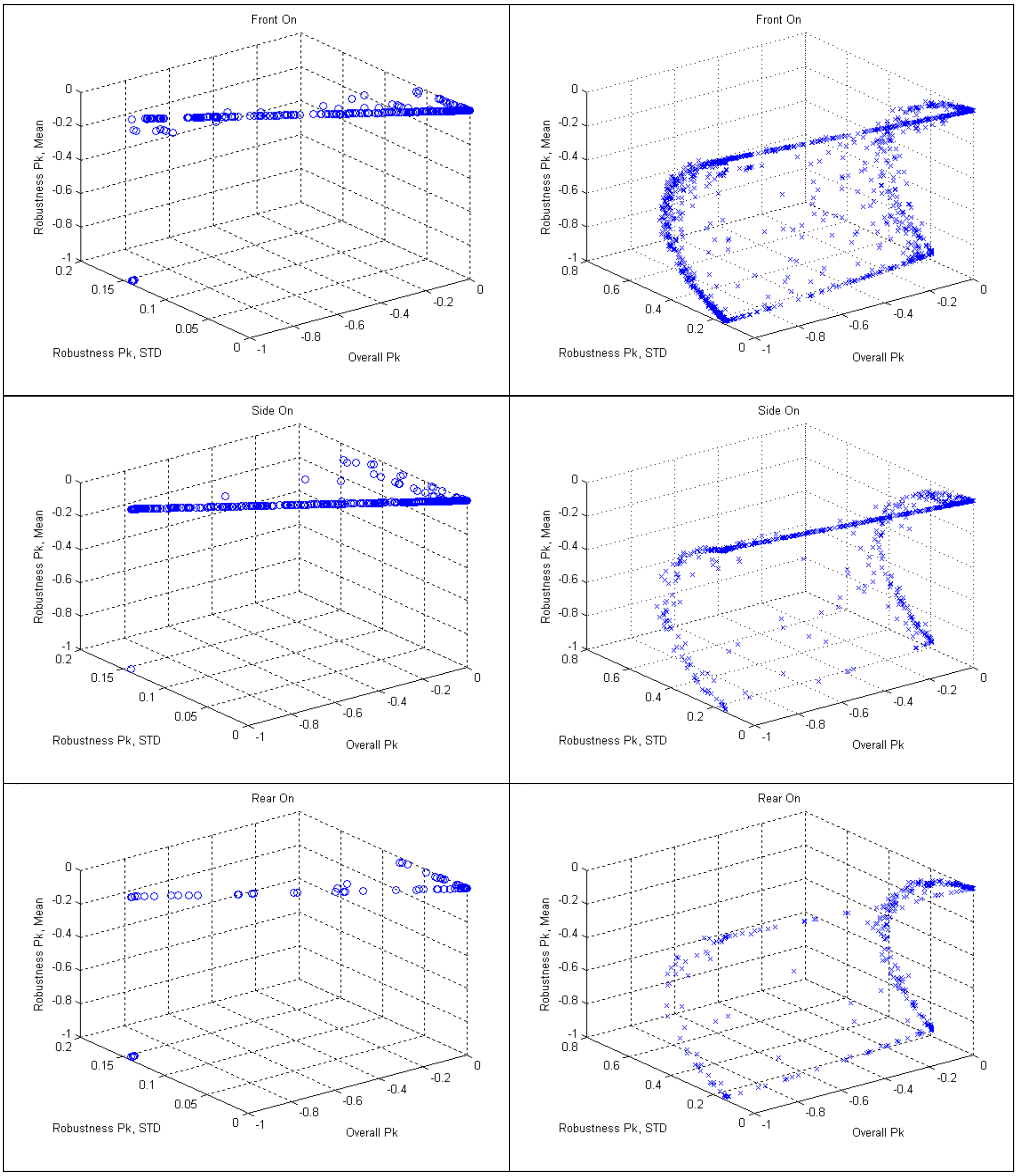
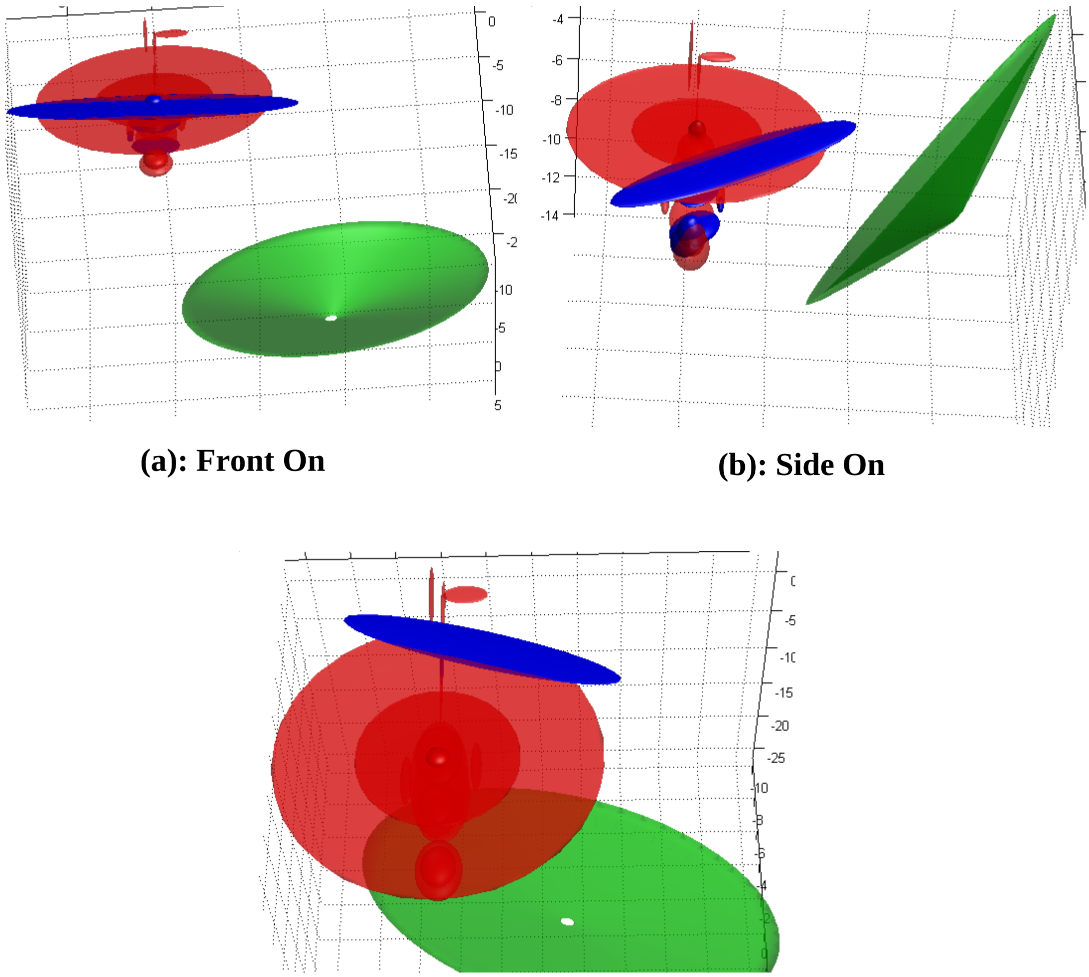


Figure B.15: MOGA Optimisations for Helicopter Model

As can be seen a Pareto front exists containing many solutions, and that overall a concave surface of solutions exist, showing that for some engagements of high lethality and low lethality there is very little deviation in perturbations of controllable parameters. However for inter mediate lethality values (0.3 < lethality < 0.7) there appears to be some deviation. This area of the search space warrants further study. Examples of endgame geometry from one of the solutions for each category are shown in Figures B.16.



(a): Front On

(b): Side On

(c): Rear On

Figure B.16: Endgame Geometry for Helicopter MOGA Optimisation

From the front on scenario (a), it can be seen that the missile fragments are hitting the helicopter target across the main rotor and cockpit structures, providing a fairly robust solution as the rotor is a highly vulnerable component and slight deviation will still result in a high kill probability. Similarly for the side on engagement, (b), the missile is aiming to hit the cockpit and main rotor structures. For the rear on engagement (c), however, it is much harder to hit the cockpit as it is located at the front of the target, therefore in this solution the drive shaft to the tail has been hit from below and to the rear as this is also a vulnerable area of the target.

Appendix C. IFAC World Congress Publication

MISSILE ENDGAME ANALYSIS VIA MULTIOBJECTIVE OPTIMIZATION

N. Patel, A. J. Chipperfield and A. J. Keane

*School of Engineering Sciences, University of Southampton,
Highfield, Southampton, Hampshire, England SO17 1BJ*

Abstract: The traditional approach to improving the lethality of a missile has been to concentrate efforts in the guidance and control systems to improve accuracy and agility. In this paper, we consider how optimizing the endgame, the final few milliseconds before detonation, can yield improvements in overall lethality. As there is likely to be uncertainty in both the target parameters and missile coordinates, a multiobjective problem is developed so that the robustness of a solution can be traded against its efficacy. The ability to quickly determine promising endgames is likely to be of benefit when exploiting modern control schemes, such as MPC, that offer improved accuracy and agility. *Copyright © 2005 IFAC*

Keywords: Endgame optimization, guidance and control, aerospace systems, Gaussian functions.

1. INTRODUCTION

The present generation of missile systems are likely to be sub-optimal in many engagement scenarios currently considered. Examples of engagements include both anti-air and ground attack domains and these have to allow for an increased use of stealth, more effective countermeasures and the use of redundant subsystems for increased mission survivability. Traditionally, improvements in missile lethality have been sought through improved guidance and control laws, for example, to optimize guidance for a specific control law and engagement conditions (Gurfil, 2001) or by solving receding horizon optimizations to achieve fast and realisable online target tracking (Kim *et al*, 2001). In this paper, the focus is on optimization of the endgame, i.e., the reachable set of outcomes in terms of engagement geometry, rather than the guidance and control laws that result in such a state being achieved.

Flyout is the portion of flight from release to immediately before detonation. During flyout the missile has to engage the target and deliver the warhead to within a close distance of the target. The engagement geometry at the start of the endgame is critical to the lethality and is the state at the end of flyout. The next section describes how the endgame can be modelled and a programme (AGILE) for achieving that is briefly described. The use of optimization to enhance the lethality of endgames is then considered and further developed with multiobjective formulations to find endgames that have a high probability of kill as well as robustness to variations in the parameters of the problem.

2. ENGAGEMENT MODELLING

The trajectories and orientations of the missile and target in the final milliseconds before detonation are collectively known as the endgame geometry. Consider the missile-target engagement shown in Fig. 1. Using GW372 notation (Payne, 1995) the relationship between the Cartesian frames of reference for the missile and the target can be defined where the *x*, *y* and *z* axes are usually aligned in both frames as follows:

- The *x*-axis is to the left (e.g., along the port wing of a fixed-wing aircraft);
- The *y*-axis is up (in level flight, the direction of the pilot's torso);
- The *z*-axis is ahead, along the centre line of the aircraft or missile (i.e., in direction of flight with zero incidence).

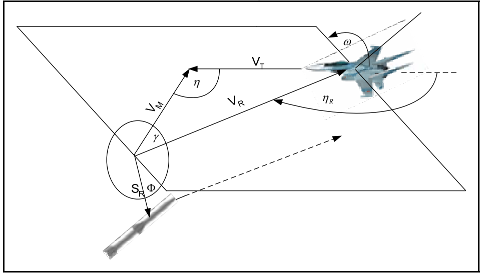


Fig. 1: Engagement geometry in GW372 notation.

The GW372 coordinates only specify relative position, velocity and orientation. Higher derivatives, e.g., acceleration and rotation rates, are not required as lethality is not usually sensitive to them due to the very short time periods involved in damage mechanisms. All angles in GW372 are specified in degrees and from Fig. 1 the following are identified:

- V_m and V_t are missile and target speeds (m/s).
- η is the engagement angle that is subtended between the missile and target velocity vectors ($\eta = 0 \rightarrow$ tail chase, $\eta = 180° \rightarrow$ head-on).
- ω is target roll.
- δ, ε, and ψ define missile yaw, pitch and roll.
- ϕ, S_r and z define the missile burst points. ϕ is known as the dartboard angle, S_r is the dartboard radius and z specifies the position of the burst point along the trajectory.
- Additional parameters, x_0, y_0 and z_0, define a missile aim point in the target's frame of reference. This point defines the warhead detonation point as the cylindrical polar coordinate system (ϕ, S_r and z) where the x-axis is aligned with missile velocity.

The important feature of this system is the use of a 'Common Velocity' (CV) plane as a datum for measuring many of the angles in the system. The CV plane is defined as the plane containing the missile and target velocity vectors (or parallel vectors), and passing through the target origin. The CV plane can have any orientation in space. In reality the missile and target both move along their respective velocity vectors; however it is easier to think of the target as stationary with the missile moving along a vector V_R towards it. It is usually assumed that as the missile approaches the target along V_R all the other parameters remain constant (no manoeuvre takes place). This assumption is justified because all the fusing and lethality events take place over a few milliseconds and within a very short distance (a few meters) of the trajectory length. The GW372 system therefore has the advantage that the primary parameters can be changed independently of each other, and each has a clear physical meaning.

A lethality prediction tool, Analytic Gaussian Intersection for Lethality Engagement (AGILE), allows engagements defined using GW372 to be evaluated and a value (probability) of engagement uncertainty, or 'kill probability', P_k determined (Watson, 2003). AGILE can evaluate an endgame geometry in milliseconds, including: prediction of damage inflicted by warhead fragments on the target or target components; a close-burst model incorporating blast effects; direct impact model; and a simple fuzing model.

The principal method of representing the above model features is by using 3-dimensional Gaussian functions. A Gaussian function f has the following form:

$$f(x) = a \exp\left[-\frac{1}{2}(x-b)^T C^{-1}(x-b)\right]$$

where x is a spatial position vector (with three Cartesian components), a is the maximum value of f, b is the position vector where f is maximal and C is a 3 × 3 positive-definite symmetric matrix representing the shape and orientation of level sets (surface contours) of f. The level sets of a Gaussian are ellipsoids, so the Gaussian itself can be thought of as a fuzzy ellipsoid; the value of f decays smoothly from a to zero as the distance from the centre b of the ellipsoids increases.

The following objects are represented by sums of Gaussian functions in AGILE:

- Target vulnerability to warhead fragment damage. Regions of high vulnerability are close to the centre of one of more Gaussians, whilst regions of low or zero vulnerability are typically further away from the centres.
- Warhead fragment cluster density. This is not the density or mass of individual fragments, but their average number per unit volume, or 'population density'. Where the target vulnerability and warhead fragment density are both high, the level of damage (probability of target kill or component failure) will be high.
- Close-burst lethality and warhead blast damage. A set of ellipsoids and cylinders is used to define a neighbourhood of the target for which a 'kill' is certain. This neighbourhood is the set of all points inside one of more of these objects; the latter are derived from level sets (contour surfaces) of Gaussian functions.
- Target shape, which is used by both the fuzing and impact models. In the fuzing model Gaussians are used to define the external shape of the target and its reflectivity to the radiation used by the fuzing sensor. In the impact model Gaussians are used to define the shape of both the missile and target, so that the severity of a collision can be calculated.
- Missile shape. Used by the impact model.
- Radiation pattern of the fuzing sensor. This information is used in conjunction with the shape and reflectivity of the

target to predict the moment when the fuze is triggered.

Gaussian components are used in AGILE for the following reasons:

- Their intersections can be computed very efficiently using an analytical formula, hence the acronym Analytic Gaussian Intersection for Lethality Engagement.
- Uncertainty in the endgame geometry can be represented directly by Gaussian components, reducing or avoiding the need for Monte-Carlo methods.

The reason for AGILE's speed is its ability to represent many warhead fragments as a single entity; instead of computing the intersection of each fragment with the target, a single calculation can applied to hundreds of fragments as an ensemble. Fig. 2 shows an example of an endgame for a simple fixed-wing target. Here, the engagement angle $\eta = 46°$, represents a rear, side-on impact at a miss-distance of 15m. From the fragment trajectories, it can be seen that for this endgame geometry, the port wing is vulnerable to fragment damage while the rest of the aircraft remains unshaved. AGILE evaluates kill probabilities from the Gaussian components described above assigning an overall probability of kill, P_k, and individual probabilities for a kill arising from cockpit, fuselage, engine and wing damage. Clearly, in Fig. 2 the majority of the P_k arises from fragments damaging the wing and its components.

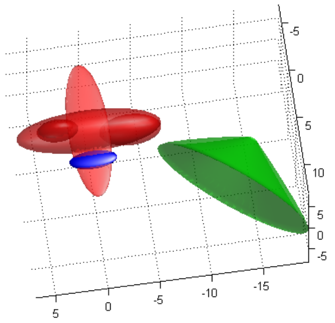


Fig. 2: Fragment vulnerability for simple fixed-wing aircraft.

The parameters listed in Table 1 can be varied over the ranges shown in an AGILE endgame evaluation. Fig. 3 shows the result of exercising AGILE with 1000 input sets where the values for the parameters are chosen randomly over these ranges. It is clearly unlikely that a random endgame will yield a high value of P_k. By optimizing the endgame geometry to achieve high and/or robust P_k, the missile flyout endpoint is determined and a suitable guidance law can be developed using conventional approaches such as Shinar & Vladimir (2003) or intelligent ones such as Leng (1996).

In the next section, a series of optimizations are employed to determine good engagement geometries. The engagement space is first sampled and direct optimization of the probability of kill considered through a restricted parameter set. However, a requirement of a good endgame is that the probability of kill should be robust to uncertainty in the parameters. Thus, multiobjective optimization is used to identify such solutions and their properties assessed.

Table 1: Agile engagement parameters

Parameter	Min	Max	Nominal
V_M	0	2000	-
V_T	0	2000	-
η	0	180	-
S_R	0	100	15
ω	0	360	0
δ	0	180	0
ε	0	180	0
ψ	0	360	0
x_0	-5	5	0
y_0	-5	5	0
z_0	-5	5	0
Z_{delay}	-10	10	0

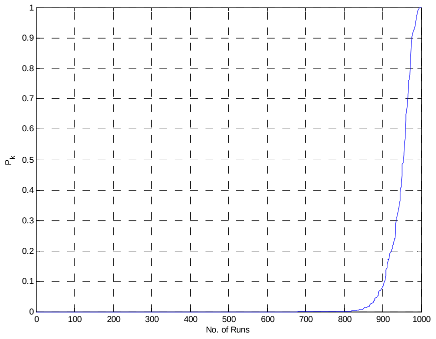


Fig. 3: Random search P_k distribution.

3. ENDGAME OPTIMIZATION

The endgame is the final few milliseconds of flight before detonation of the warhead. In order to maximize the probability of a kill, the missile guidance system must ensure that the missile parameters approach those of a suitable endgame. Alternatively, achieving the maximum P_k given a limited deviation from a nominal endgame might be a suitable goal for a model-based predictive controller used in the guidance loop. The following four problems explore the use of AGILE as a tool for determining and maximizing endgame lethality.

Problem 1: max P_k

The three most significant parameters affecting the endgame are missile speed, V_M, target speed, V_T, and the missile-target engagement angle, η. The single objective considered is min ($-P_k$) and Table 2 shows ten examples of achievable P_k given the starting point $\{V_{Mi}, V_{Ti}, \eta_i\}$. These optimizations where performed using the SQP algorithm in the MATLAB Optimization Toolbox with the remaining parameters set to the nominal values of Table 1. The initial sets are not always sensible, but demonstrate how the engagement geometry should be modified to improve potential lethality. For example the initial set {1400, 1600, 0} represents a tail-chasing missile travelling slower than its target. However, given that it is detonated 15 m from the tail of the target, the low probability of kill, 0.366, arises mostly from fragment damage to the engine. By slowing the target to 1465 m s^{-1}, increasing the missile speed to 1538 m s^{-1} and engaging at a slight incidence of 1.2°, P_k increases to 0.967.

Table 2: Three parameter engagement geometry optimizations

V_{Mi}	V_{Ti}	η_i	P_{ki}	V_M	V_T	η	P_k
750	500	90	0.619	767	300	66.8	0.912
1000	500	90	0.418	800	322	65.7	0.906
1000	600	90	0.441	899	400	63.3	0.885
1300	900	60	0.494	1243	700	55	0.797
1700	1100	30	0.45	1681	1119	24.4	0.509
1400	1600	0	0.366	1538	1465	1.2	0.967
1800	1200	15	0.281	1798	1201	4.4	0.922
1000	500	0	0.607	1000	500	5.2	0.937
1000	500	75	0.546	800	322	65.7	0.906
1800	1300	50	0.288	1651	1100	46.9	0.681

The first three endgames in Table 2 represent side-on engagement. In all three cases, increasing the difference in speed between the missile and target and engaging more towards tail-chase significantly improves P_k. Fig. 4 shows the variation in P_k with V_M and η about the optimized set $\{V_M, V_T, \eta\}$ from the first row of Table 2 for fixed V_T = 300 m s^{-1}. Similarly, Fig. 5 shows how P_k varies with V_T and V_M for a fixed η = 66.8°. These two figures confirm what would be expected during an engagement, namely that maximum lethality will occur at an angle and missile-target speed ratio such that fragment damage is focused on the more vulnerable areas.

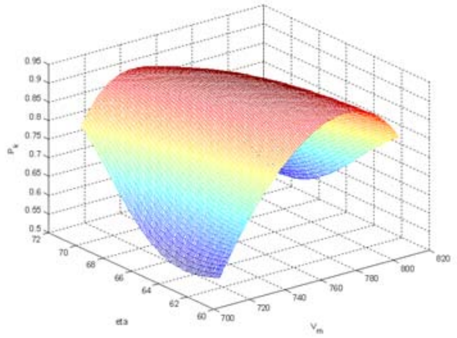


Fig. 4: Variation in P_k for fixed V_T = 300 m s^{-1}.

Figs. 6 and 7 show lethality plots for the engagement of the eighth row of Table 2. The plots are in quite a different area of the permissible engagement space than those of Figs 4 and 5 although the plots show similar characteristics.

Note though that while the engagement of Fig. 4 is relatively insensitive to angle, that of Fig. 6 is very sensitive to variation in engagement angle. Thus a small error in engagement angle in the first case will result in only a small reduction in P_k, in the second case the same small change in η could result in P_k of less than 0.4.

In realistic engagement problems, the target is not completely known and the feedback measurements will be imperfect. The engagement will be also subject to exogenous disturbances. Although these unknowns can be accommodated to some degree in the Gaussians modelling the engagement, it is also important to understand the sensitivity of solutions to parameter uncertainty. In practise this can be achieved by sampling around an 'optimal' solution by, say, taking 100 samples uniformly distributed at random by perturbing the parameters within a percentage of full-scale as depicted in Fig. 8.

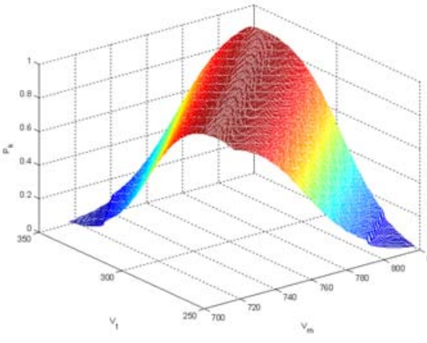


Fig. 5: Variation in P_k for fixed η = 66.8°.

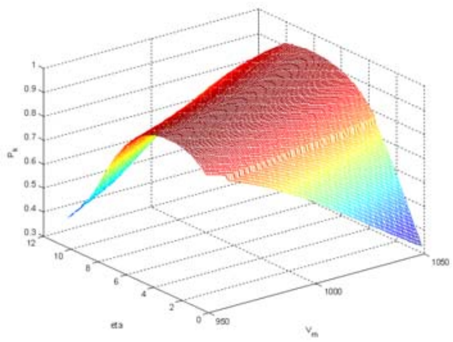


Fig. 6: Variation in P_k for fixed $V_T = 500$ m s^{-1}.

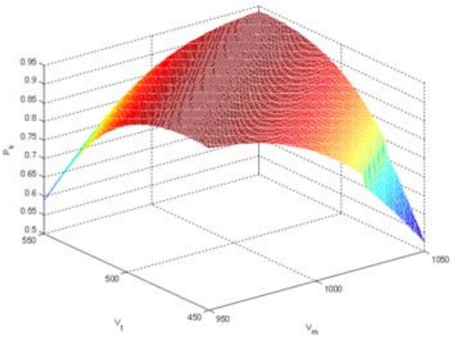


Fig. 7: Variation in P_k for fixed $\eta = 5.2°$.

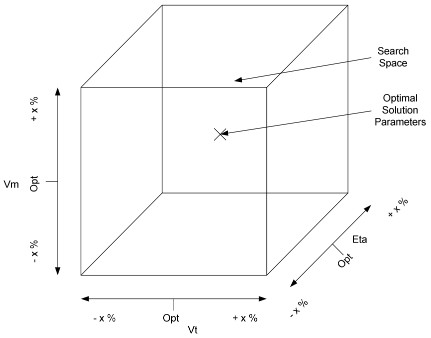


Fig. 8: Perturbation for robustness trade.

Problem 2: max P_k and check for robustness

In practise, a realistic endgame, and therefore flyout, is unlikely to be achievable using only $\{V_M, V_T, \eta\}$, not least because the target velocity is unlikely to be under the control of the missile. In this example all parameters in Table 1 are used and the missile's controllable parameters, i.e. δ, ε, x_0, y_0,z_0, and Z_{delay}, are optimized to determine suitable endgames for engaging targets grouped in either head-on, side-on or tail-chase categories, based on engagement angle. The same SQP used in Problem 1 is kept, and 500 scenarios were calculated for each engagement category. For each scenario the optimised parameters are then perturbed 1000 times and the resultant standard deviation of P_k is recorded.

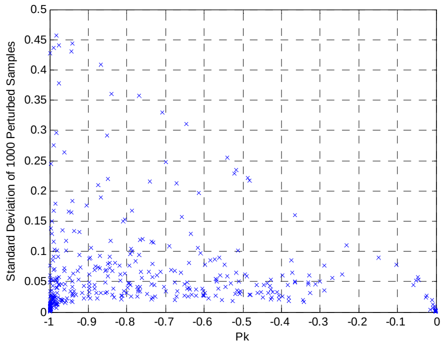


Fig. 9: Scatter plot of head-on engagements and variance (standard deviation) with 10% uncertainty.

As can be seen in Fig. 9, a pair of scenarios with similar P_k's can have varying robustness values. As well as maximizing P_k, it is also desirable to maximize the robustness of the solution to uncertainty in the parameters Minimizing the standard deviation in the sample is the equivalent of minimizing loss in P_k due to parameter variations. Attempting to maximize P_k while simultaneously minimizing the standard deviation should result in endgames that have both a high probability of kill and a high-degree of robustness to parameter uncertainty.

Problem 3: max P_k min $s(P_k)$ using novel methods

This problem was addressed with a multiobjective genetic algorithm, as described by Fonseca and Fleming (1998), to determine fitness on the basis of non-dominance of the individuals. A MOGA was attractive as the population-based nature of the search allows many endgames to be evaluated at each generation. The objectives used to assess the performance being (i) overall P_k as used in problem 1, and (ii) robustness of P_k calculated as described above. In the example presented here, a ±10% uncertainty is assumed on the free parameters. In Fig. 10 individual endgames are plotted with their P_k against the standard deviation in 20 samples around that point in $\{V_M, V_T, \eta\}$. Clearly, a fairly large number of high P_k solutions appear to offer robust endgame performance.

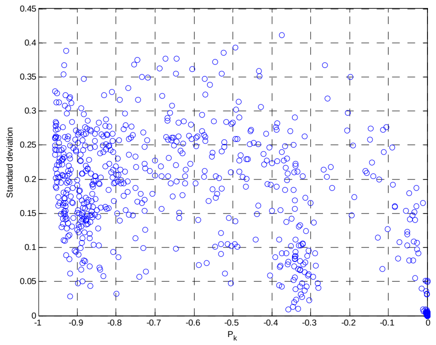


Fig. 10: Scatter plot showing P_k and variance in P_k (standard deviation) with 10% uncertainty.

The trade-off between robustness and lethality is shown in Fig. 11 and for the lowest variance sample at $P_k = 0.9189$, $s = 0.0298$ the endgame is illustrated in Fig. 12. Improving P_k to 0.9569 results in an increase in variance to $s = 0.25$. A choice of which was the best P_k would have to be made on a number of factors including: time to endgame; precision of missile; and target vulnerability. The flyout to arrive at an endgame will also have uncertainties arising from the usual modelling considerations, but may also account for target manoeuvring.

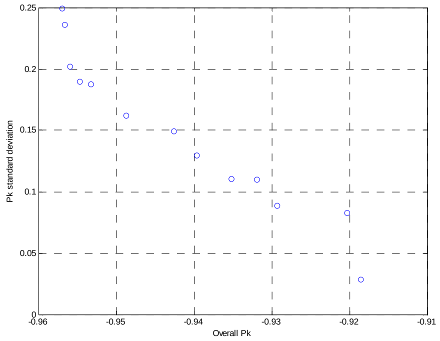


Fig. 11: Trade-off between P_k and endgame robustness.

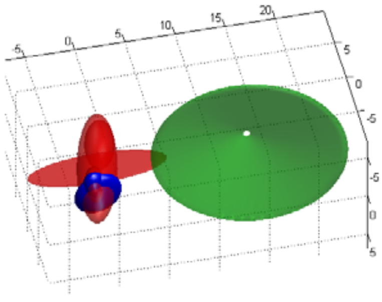


Fig. 12: Robust endgame with $P_k = 0.9189$.

The engagement shown in Fig. 12 has $V_M = 1214$, $V_T = 705$, $\eta = 2.7$ and achieves probabilities of kill of 0.9048 for cockpit damage and 0.1447 for the fuselage. No engine or wing damage is predicted by AGILE. The reason that this is robust to variations in $\{V_M, V_T, \eta\}$ is the relatively high vulnerability of the cockpit area and the coverage of fragments from the warhead. Such an endgame therefore exploits the characteristics of the missile and the target.

Problem 4: max P_k min $s(P_k)$,

$$P_k = f(V_M, V_T, \eta, S_R, \omega, \delta, \varepsilon, \psi, x_0, y_0, z_0, Z_d)$$

The same MOGA formulation employed in Problem 3 is retained, and the uncertainty is assumed over all the parameters and the corresponding number of samples at each nominal geometry is increased to 50.

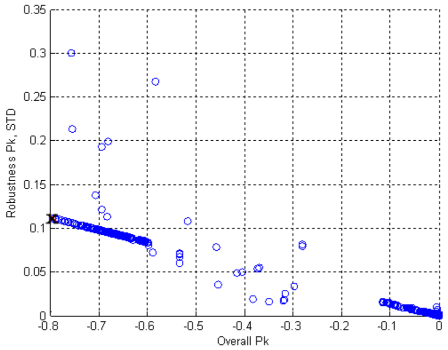


Fig. 13: P_k vs. robustness trade-off, Problem 4.

The Pareto optimal solutions for side-on scenarios found after 200 generations of 50 individuals are shown in the trade-off of Fig. 13. While similar characteristics can be observed to that of Fig. 11 (Problem 2), in this case the search space is now much larger and hence the greater spread in the solutions. The cross in Fig 13 identifies the endgame shown in Fig 14.

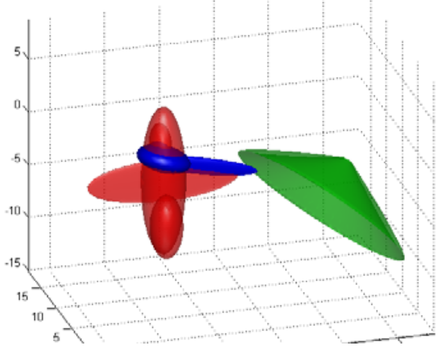


Fig. 14: Engagement with good robustness.

This figure shows an engagement where the missile is approaching fast from towards the aircraft side (V_M = 749.91 m s^{-1}, η = 108°) and the missile is oriented $\{\delta, \varepsilon, \psi\}$ = { 27°, 209°, -8°} with a burst point (fuze delay) -2 m along the missile trajectory. The overall P_k = 0.8 with 0.7192 cockpit, 0.08 fuselage, and zero engine and wing probabilities of kill. Although a very different endgame to that presented in Fig. 12, the endgame of Fig. 14 is robust in that the fragment damage to the cockpit is achieved when the missile is detonated within a large region of the 'optimal' point identified in Fig. 13.

4. CONCLUDING REMARKS

This paper has demonstrated how missile endgame conditions with a high probability of kill can be identified using optimization techniques against various performance criteria. There is not one single 'optimal' engagement for a missile-target rather there are families of solutions that trade-off overall lethality with robustness to parameter uncertainty at a number of different condition, for example target speed or engagement angle. Having a better understanding of the location and sensitivity of potential engagement conditions can be readily used in the guidance system to enhance the overall efficacy of the missile which is essential if projected future threats are to be dealt with effectively. The final choice of a suitable endgame will inevitably be a compromise over the criteria and will be determined to some degree by flyout considerations. However, an acceptably accurate simulation, AGILE, can readily and rapidly be used to determine suitable engagement geometries.

9. ACKNOWLEDGEMENTS

The first author wishes to acknowledge the financial and technical support of DSTL throughout this work.

REFERENCES

Fonseca, C. M. and Fleming, P. J. (1998). Multiobjective Optimisation and Multiple Constraint Handling with Evolutionary Algorithms – Part 1: A Unified Formulation. *IEEE Trans. Sys., Man. & Cyber.*, **28**, 26-37.

Gurfil, P. (2001). Synthesis of zero miss distance guidance via solution of an optimal tuning problem. *Control Engineering Practice* (**9**) 1117 - 1130.

Kim, K.B., T.-W. Yoon and W.H. Kwon (2001). Receding horizon guidance laws for constrained missiles with autopilot lags. *Control Engineering Practice* (**9**) 1107 - 1115.

Leng, G. (1996). Missile guidance algorithm design using inverse kinematics and fuzzy logic. *Fuzzy Sets & Systems*, (**79**), 287-295.

Payne, G.D, (1995). A standard format for the interchange of pass-path geometry data for guided weapon fuzing and lethality assessments. DRA/WSS/WD5/WP94/29/1.2, UK.

Shinar, J. and T. Vladimir (2003). What happens when certainty equivalence is not valid? Is there an optimal estimator for terminal guidance? *Annual Reviews in Control* (**27**), 119-130.

Watson, G.H. (2003). Agile user guide, QinetiQ, UK

References

[1] Siouris, G.M., 2004, “Missile Guidance and Control Systems”, Springer

[2] Brown, L., 1993, “The Proximity Fuse”, *Aerospace and Electronic Systems Magazine, Vol 8 Issue 7,* IEEE

[3] Creaser, P. A., Stacey, B. A., & White, B. A., 1998, “Fuzzy Missile Guidance”, *Proceedings of AIAA Conference*, Boston, USA, 10–12 August

[4] Gurfil, P., 2001, “Synthesis of zero miss distance guidance via solution of an optimal tuning problem”, *Control Engineering Practice* (**9**) p1117 – 1130

[5] Kim, K.B., T.-W. Yoon and W.H. Kwon, 2001, “Receding horizon guidance laws for constrained missiles with autopilot lags”, *Control Engineering Practice* (**9**) p1107 – 1115

[6] Shinar, J. and T. Vladimir, 2003, “What happens when certainty equivalence is not valid? Is there an optimal estimator for terminal guidance?”, *Annual Reviews in Control* (**27**), p119-130

[7] Leng, G., 1996, “Missile guidance algorithm design using inverse kinematics and fuzzy logic”, *Fuzzy Sets & Systems*, (**79**), p287-295.

[8] DSTO/WSD Australia, DSTL, 2002, MSTARS v2.1

[9] Watson, G.H., 2003, AGILE User Guide, QinetiQ

[10] Clayton, J.K.S. and Ruston, G.C.A, 1955, “Methods Used at R.A.E for the Assessment of the Lethality of Fragmenting AA Munitions”.

[11] Payne, G.D, 1995, “A Standard Format for the Interchange of Pass-Path Geometry Data for Guided Weapon Fuzing and Lethality Assesments.”, DSTL

[12] Burton, A, 2002, “Matlab Based Anti-Air Endgame Simulation”, DSTL

[13] Andrieu, C. et al, 2003, "An Introduction to MCMC for Machine Learning”, Machine Learning, vol. 50, p5-43

[14] The Mathworks, 2000, “Optimisation Toolbox for use with Matlab: Users Guide”, The Mathworks

[15] Fonseca, C.M. and Fleming, P.J., 1993, “Genetic Algorithm for Multiobjective Optimisation: Formulation, Discussion and Generalisation”, The Fifth International Conference on Genetic Algorithms, Morgan Kauffman, pp416-423.

[16] Schrijver, A., 1998, “Theory of Linear and Integer Programming.”, John Wiley & sons

[17] Nocedal, J & Wright, S.J., 2006, “Numerical Optimization (2nd ed.)”, Springer-Verlag

[18] Nelder, J.A. and Mead, R, 1965, “A Simplex Method for Function Minimisation”, The Computer Journal, Vol. 7, pp. 308-313.

[19] Broyden, C.G., 1970, “The Convergence of a Class of Double-rank Minimization Algorithms,” J. Inst. Maths. Applics., Vol. 6, pp 76-90.

[20] Fletcher, R., 1970, “A New Approach to Variable Metric Algorithms,” Computer Journal, Vol. 13, pp 317-322.

[21] Goldfarb, D., 1970, “A Family of Variable Metric Updates Derived by Variational Means,” Mathematics of Computing, Vol. 24, pp 23-26.

[22] Shanno, D.F., 1970, “Conditioning of Quasi-Newton Methods for Function Minimization”, Mathematics of Computing, Vol. 24, pp 647-656

[23] Davidon, W.C., 1959, “Variable Metric Method for Minimization,” A.E.C. Research and Development Report, ANL-5990.

[24] Fletcher, R. and M.J.D. Powell, 1963, “A Rapidly Convergent Descent Method for Minimization,” Computer Journal, Vol. 6, pp 163-168.

[25] Kuhn. H.W. and Tucker, A.W., 1951, “Nonlinear Programming”, Proc. 2nd Berkeley Symposium on Mathematical Statistics and Probabilistics, pp. 481-492.

[26] Gage, P. and Kroo, I., 1993, “A Role for Gentic Algorithms in a Preliminary Design Environment”, AIAA 93-3933, AIAA Aircraft Design, Systems and Operations Meeting, Monterey, CA.

[27] Price, N.J, 2003, Cartesian to GW372 Conversion Algorithm, DSTL.

[28] Kolodner, J., 1993, “Case-Based Reasoning (Morgan Kaufmann Series in Representation & Reasoning)”, Morgan Kauffmann

[29] Sutton, Richard S. and Barto, Andrew G., 1998, “Reinforcement Learning: An Introduction.”, MIT Press

[30] Trufitt, M., 2005, “Agile Helicopter Target Ellipsoid definitions”, DSTL/Qinetiq.

[31] Fonseca, C.M., 1995, “Multi-Objective Genetic Algorithms with Application to Control Engineering Problems”, Ph.D Thesis, University of Sheffield.

[32] Baker, J.E., 1985, "Adaptive Selection Methods for Genetic Algorithms", Proc. 1st Int. Conf. on Genetic Algorithms, J.J, Grefenstette (Ed.), Lawrence Erlbaum, pp. 101-111.

[33] Baker, J.E., 1987, "Reducing Bias and Inefficiency in the Selection Algorithm", Proc. 2nd Int. Conf. on Genetic Algorithms, J.J, Grefenstette (Ed.), Lawrence Erlbaum, pp. 14-21.

[34] Ritzel, B.J., Eheart, J.W. and Ranjithan, S., 1994, "Using Genetic Algorithms to Solve a Multiple Objective Groundwater Pollution Containment Problem", Water Resources Research, Vol. 30, pp. 1589-1603.

[35] Goldberg, D.E. and Segrest, P., 1987, "Finite Markov Chain Analysis of Genetic Algorithms", Proc. 2nd Int. Conf. on Genetic Algorithms, J.J, Grefenstette (Ed.), Lawrence Erlbaum, pp. 1-8.

[36] Goldberg, D.E. and Richardson, J., 1987, "Genetic Algorithm with Sharing for Multimodal function Optimization", Proc. 2nd Int. Conf. on Genetic Algorithms, J.J, Grefenstette (Ed.), Lawrence Erlbaum, pp. 41-49.

[37] Silveman, B.W., 1986, "Density Estimation for Statistics and Data Analysis", Chapman and Hall.

[38] Deb, K. and Goldberg, D.E., 1989, "An Investigation of Niche and Species Formation in Genetic function Optimization", Proc. 3rd Int. Conf. on Genetic Algorithms, J.D. Schaffer (Ed.), Morgan Kaufmann, pp. 42-50.

[39] Grefenstette, J.J., 1992, "Genetic Algorithm for Changing Environments", Parallel Problem Solving from Nature 2, R. Männer and B. Manderick (Eds.), North-Holland, pp. 137-144.

[40] Schaffer, J.D., 1985, "Multiple Objective Optimisation with Vector Evaluated Genetic Algorithms", The First International Conference on Genetic Algorithms and their Applications, Lawrence Erlbaum Associates, pp 93-100.

[41] Goldberg, D.E., 1989, "Genetic Algorithms in Search, Optimisation and Machine Learning", Addison-Wesley.

[42] Srinivas, N. and Deb, K., 1995, "Multiobjective Optimization using Non-Dominated Sorting in Genetic Algorithms", Evolutionary Computing Vol. 2, pp221-248.

[43] Cheng, F.Y. and Li, X.S., 1998, "Genetic Algorithm Development for Multiobjective Optimisation of Structures", AIAA vol 36, pp1105-1112.

[44] Narayana, S. and Azarm, S., 1999, “On Improving Multiobjective Gnetic Algorithms for Design Optimisation”, Structural Optimisation Vol. 18, pp146-155.

[45] Belegundu, A.D, Murthy, D.V., Salagame, R.R., and Contans, E.W., 1994, “Multi-Objective Optimisation of Laminated Ceramic Composites Using Genetic Algorithms”, 5th AIAA/UUSAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimisation, pp1055-1022, American Institute of aeronautics and Astronautics.

[46] Cheng, F.Y. and Li, X.S., 1997, “Multiobjective Optimisation Design with Pareto Genetic Algorithm”, Journal of Structural Engineering September, pp1252-1261.

[47] Ishibuchi, H., and Murata, T., 1996, “Multi-Objective Genetic Local Search Algorithm”, IEEE International conference on Evolutionary Computation, IEEE, pp119-124.

[48] Murata, T., Ishibuchi, H., and Tanaka, H., 1996, “Multi-Objective Genetic Algorithm and its Applications to Flowshop Scheduling”, Computating in Industrial Engineering Vol. 30, pp957-968.

[49] Horn, J., Nafpliotis, N., and Goldberg, D.E., 1994, “A Niched Pareto Genetic Algorithm for Multiobjective Optimisation”, 1st IEEE Conference on Evolutionary Computation, pp82-87.

[50] Deb, K., 1989, “Genetic Algorithms in Multimodal Function Optimisation”, MS Thesis, TCGA Report 89002, University of Alabama, Tuscaloosa, AL.

[51] Deb, K., Pratap A., Agarwal S., and Meyarivan T., 2002, “A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II”, IEEE Transactions on Evolutionary Computation Vol. 6, No. 2, pp182-197.

[52] Lee, S. and Tsui, H., 2004, “A Relational Multi-Objective Genetic Algorithm”, Proceeding of 2004 IEEE International Joint Conference on Neural Networks Vol.1, pp222-225.