Cranfield
UNIVERSITY

Evan James Hughes

Radar Cross Section Modelling Using Genetic Algorithms

Department of Aerospace, Power & Sensors

PhD Thesis

Royal Military College of Science
Shrivenham

Department of Aerospace, Power & Sensors

PhD Thesis

Academic Year 1998

Evan James Hughes

Radar Cross Section Modelling Using Genetic Algorithms

Supervisors
Professor B A White and M Leyland

May 1998

ejhughes@iee.org

Royal Military College of Science
Shrivenham

Abstract

In the design of new, more sophisticated missile systems, simulations need to be realistic and fast. Realistic target models are just as important as realistic models of the missile, but have often been overlooked in the past. Existing methods for creating realistic target models require considerable computational resources. This thesis addresses the problem of using limited resources to create realistic target models for simulating engagements with radar guided homing missiles.

A multiple genetic algorithm approach is presented for converting inverse synthetic aperture radar images of targets into scatterer models. The models produced are high fidelity and fast to process. Results are given that demonstrate the generation of a model from real data using a desktop computer.

Realistic models are used to investigate the effects of target fidelity on the missile performance. The results of the investigation allow the model complexity to be traded against the fidelity of the representation to optimise simulation speed.

Finally, a realistic target model is used in a feasibility study to investigate the potential use of glint for target manoeuvre detection. Target glint is considered as noise in conventional missile systems and filtered to reduce its effects on the tracking performance. The use of glint for target manoeuvre detection would provide a cheap and novel alternative to the optical techniques currently being developed. The feasibility study has shown that target manoeuvre detection using glint may be as fast as optical techniques and very reliable.

Keywords

Glint, Multiple Genetic Algorithms, Point Scatterer Model, Radar Cross Section, Target Manoeuvre Detection.

Contents

List of Figures

List of Tables

Glossary of Terms

1D	–	One Dimensional
2D	–	Two Dimensional
3D	–	Three Dimensional
BSP	–	Binary Space Partition
CAD	–	Computer Aided Design
dBm^2	–	Decibels WRT 1 square metre
DFT	–	Discrete Fourier Transform
FFT	–	Fast Fourier Transform
GA	–	Genetic Algorithm
GHz	–	Giga-Hertz
Hz	–	Hertz
$\Im(x)$	–	Imaginary Part of x
IFFT	–	Inverse Fast Fourier Transform
ISAR	–	Inverse Synthetic Aperture Radar
K–S	–	Kolmogorov–Smirnov
LOS	–	Line-of-Sight
PBIL	–	Population Based Incremental Learning
PC	–	Personal Computer
$\Re(x)$	–	Real Part of x
RCS	–	Radar Cross Section
RMS	–	Root Mean Square
WRT	–	With Respect To

1. Introduction

1.1. Introduction

Radar target tracking and radar guided missiles are an essential part of modern weapons systems. Aircraft technology is advancing with aeroplanes becoming faster, more manoeuvrable and more difficult to detect. The accuracy requirements of the tracking systems are being pushed to the limits. The technological problems are further aggravated by financial constraints on product development. Flight testing of missiles is very expensive.

Synthetic missiles may be flown in a virtual world against simulated targets. Simulation reduces the costs involved in researching into new homing and guidance techniques. These simulated engagements need to be realistic to derive any benefits. To be realistic, the parameters that affect the engagement must be modelled in a detailed manner. There are many different areas that need to be addressed in developing a full engagement model [1]. This thesis concentrates on the radar cross section aspects of the target modelling.

This chapter begins by defining the problem to be solved and then investigates the current approaches in the open literature. A package of work is defined and the approach to be taken and final objectives described. Finally, the original work and publications generated during the research are detailed.

1.2. Problem Definition

Target tracking algorithms perform best if the target manoeuvre is constant. Any changes mean the tracking algorithms must first identify that the target has altered course, and then update the track details [2, 3]. If indication could be given that a target is changing its track, the guidance algorithm could be improved.

Target glint (chapter 6) affects the missile's estimation of target acceleration and is usually treated as noise. I have proposed that it may be possible to use target glint to give indications of rapid target attitude changes, such as banking before turning. The missile could then be given advance warning of an evasive manoeuvre. In order to test this proposal, a simulation environment is required that allows the effects of glint on the missile to be observed. Target manoeuvre detection using glint is the motivation for the work covered in this thesis.

Most guidance models treat the target as an ideal source and add Gaussian noise or fixed offsets to the acceleration estimates ([4, 5] for example). More sophisticated models add coloured noise to the acceleration estimates (eg. [6, 7, 8]). These models are more realistic but the errors are not correlated with the missile and target motion and therefore do not create true glint effects.

The main problem addressed in this thesis is the production of target models that can be used to create realistic radar cross sections for any aspect angles. The following requirements must be satisfied by a model.

1. **Fast to process** – The model will be used thousands of times in each simulation. If genetic algorithms or artificial neural nets are used during the guidance system design process, many thousand trials will be performed.
2. **High fidelity** – In some situations it may be desirable for the target to represent a specific vehicle.
3. **High resolution** – Glint spikes have been observed that are less than $1/100^{\mathrm{th}}$ of a degree wide. The narrow spikes can become significant if the rate of relative angular rotation (between target and missile) is low. The model resolution should typically be at least $1/200^{\mathrm{th}}$ of a degree to model glint fluctuations accurately.
4. **Correlation** – Echo amplitude should be correlated with echo phase to give realistic glint effects in the seeker head. The echo signal must be correlated to relative motion.
5. **Low storage requirements** – It must be possible to contain the whole model in the machine's physical memory to prevent excessive disk use.

The processing and memory requirements should be suitable for a typical desktop PC. The following assumptions will be made.

1. **The target has a rigid body that does not flex in manoeuvre** – Any requirements for complicated structural analysis of the target are thus removed.
2. **Radar is continuous wave or uses pulses that are long compared to the target** – Each radar pulse illuminates the whole target, allowing average radar cross section measurements to be taken directly, rather than by integrating range profiles.

Once a realistic model has been created, trials must be performed to establish the effects of fidelity on the engagement, thus ensuring a realistic target. The glint tracking hypothesis may then be tested.

1.3. Existing Modelling Techniques

1.3.1. Introduction

A search of the open literature to find existing solutions to the modelling problem revealed a number of potential techniques. Broadly, the radar cross section of a target may be found by direct measurement or by calculation. Four main classes of solution have been identified and are discussed below:

1. **Real data** – Data are measured directly from the target or a scale model.

2. **Scatterer models** – Groups of ideal point scatterers are modelled to produce a radar cross section pattern. The scatterers may be positioned at random or placed so as to reproduce a specific cross section pattern.

3. **Statistical models** – Data are generated randomly but conform to typical probability density functions.

4. **Structural models** – The scattering calculations are based upon a model of the physical structure of the target.

Radar cross section is a function of the angular orientation and shape of the scattering body, the frequency, and the polarisation of the transmitter and receiver. The radar cross section of a target is defined as:

> A measure of the power that is returned or scattered in a given direction, normalised with respect to the power density of the incident field.
>
> $$\sigma = 4\pi \lim_{R\to\infty} R^2 \frac{|\vec{E}^s|^2}{|\vec{E}^i|^2} = 4\pi \lim_{R\to\infty} R^2 \frac{|\vec{H}^s|^2}{|\vec{H}^i|^2} \tag{1.1}$$
>
> Where $\vec{E}^s$, $\vec{H}^s$ are the scattered electric and magnetic fields respectively, and $\vec{E}^i$, $\vec{H}^i$ are the incident fields. [9, Chapter 3]

1.3.2. Real Data

Directly measured results are difficult to obtain in practice, especially with full scale targets ([10, 11, 12] for example). Scale models may be used but any discrepancies and small deviations will affect the measured results. Measurement noise may hide some small details and ultimately the resolution of the data has finite limits for any given measurement angle segment. Out-of-plane data are usually awkward to measure as the target orientation with respect to the ground has to be altered. This can create slinging and handling problems. Measuring a full scale target over 4π steradians to a useful resolution (ie. 200 points per degree) is often impractical.

Measured data though give a reliable representation of a *real* target; although small changes will occur in-flight due to structural distortions. These distortions can be minimised by suitable slinging and support techniques during measurement, introducing airframe stresses similar to those experienced whilst the target is airborne. Careful attention to supports, slinging wires and measurement chamber configuration is necessary to prevent artifacts being present in the measured data [9, Page 331–345][13].

1.3.3. Scatterer Models

Small numbers of randomly distributed, independent, isotropic point-scatterers are often used for theoretical radar cross section analysis ([14] for example). The cross section for a typical point scatterer model may be calculated using Equation 1.2.

$$\sigma_T = \left| \sum_{k=1}^{n} \sqrt{\sigma_k}\, e^{j\left(\frac{4\pi d_k}{\lambda}\right)} \right|^2 \tag{1.2}$$

The total radar cross section of the target, σ_T, is defined as the coherent sum of the echos from the n scatterers, each scatterer with its own cross section, σ_k [15, Page 23]. If the number of scatterers is large and the viewing angle segment is small, the model may be fitted to any arbitrary cross section pattern. Chapter 4 gives more details regarding radar cross section calculation of simple and extended point scatterer models.

Information about the scattering centres may be obtained through *Inverse Synthetic Aperture Radar (ISAR)* imaging (chapter 2) where two and three dimensional images may be created from radar cross section data obtained for both angle and frequency. ISAR images of point scatterer models exhibit the same spatial characteristics as real targets.

A strong argument may be made for considering targets to be collections of small numbers of point scatterers when the target is viewed at high frequency. Most papers obtained relating to target signature analysis, ie., identifying targets from their radar cross section, are based on the identification and comparison of the major scattering centres (eg. [16, 17]).

Point scatterer models have low storage requirements and radar cross section calculations are quick to perform. Unfortunately, although it should be possible to generate models to fit any arbitrary pattern, the fitting process can be difficult in practice.

1.3.4. Statistical Models

Few statistical radar cross section models appear to have been developed. Sandhu and Saylor [18] perform a rigorous analysis of glint and radar cross section statistics and compare these results to real data. Their methods assume that the target is in the far-field and that the phase front across a true phase-comparison missile seeker head is near linear (chapter 6). Thus these results may not apply to end-game scenarios where the target is engaged in the near-field. Further analysis of the effects on monopulse seekers is explored by Tullsson [19].

Daba and Bell [14] develop the statistics of small numbers of randomly placed scatterers and compare them to empirically determined probability density functions. Gordon [20] derives probability density equations for the radar cross section of simple convex bodies, ie., ovoids. Borden and Mumford [21] develop the statistics of a point scatterer model and use them to create a synthetic radar cross section and glint generator. The generator is based on filtering and combining Gaussian random processes. This process can be used to generate realistic cross section data but they will be uncorrelated to target motion.

1.3.5. Structural Models

Introduction

Calculating the *true* radar cross section of a target is not trivial. The methods of approaching the problem fall into three broad categories; dominant feature analysis, quasi-optical methods and element analysis. Many methods though use a combination of the techniques to obtain better cross section approximations.

Dominant feature analysis

Analysing the dominant features of a target is the classical technique for developing the radar cross section pattern. The method is usually performed by hand [22] but can be automated [23]. The target is assumed to have a radar cross section that is similar to a target constructed from basic primitive shapes such as cones, cylinders, wires, spheres, and flat plates. The procedure is to identify the parts of the target that are likely to be prominent in the radar signature; for example, the leading edges of wings, fuselage, cavities etc.. The features are modelled using primitive shapes and then the estimated contribution to the cross section for each of the identified features is plotted. A general pattern of radar cross section fluctuation with respect to angle is thus created. This method is concerned solely with the magnitude of the radar cross section, the effects of phase interactions are ignored. Effects caused by multiple bounces of the signal, such as the large echo from a trihedral corner reflector, are accounted for directly in this method.

Quasi-optical methods

Many computational packages have been developed based on quasi-optical methods. These methods rely on the approximation that the radar signal behaves in a similar way to light when the target features are greater than about ten wavelengths in size. This condition is termed the optical region of operation [9, Page 53]. The packages use three-dimensional CAD models to represent the targets. The targets are based on either solid modelling or surface representations. Surface representations range from simple faceted models through to complex spline surfaces. All of the packages are capable of generating good approximations to the true radar cross sections for the supplied models, at the expense of computation time [24, 25].

The models that represent the target surface as small triangular facets are some of the simplest [26, 27, 28, 29]. Unfortunately, faceted surfaces have inherent false discontinuities at the junctions between the faces and can generate spurious signals. The most accurate cross section predictions appear to be obtained through using structures built with solid models or curved patches. Near continuous surfaces can be produced with these methods that alleviate facet noise [30, 31, 32, 25].

Many of the systems use the physical optics [9, Pages 119–130] approximations to generate the basic radar cross section and then apply the method of moments [33] or method of equivalent currents [9, Pages 136–139] to account for effects caused by diffraction and discontinuities [30, 31, 27, 28]. Others achieve similar results through the physical theory of diffraction [9, Pages 140–144][25, 32].

Most of the computation time is expended in the ray-tracing elements of the processing. To counter this problem, most of the approaches use dedicated hardware-graphics-accelerator cards to produce images of the target. Hidden surfaces are removed and the image colours are used to represent the surface outward unit normals [25]. Cross section processing is then performed on the rendered image using look-up tables for speed. Accuracy is sacrificed as each surface normal is quantised into three 8-bit values. This approach allows the use of office computers with graphics accelerator cards to produce a reasonable approximation to the cross section for a specific angle and frequency in a few seconds. Indications of the processing requirements for generating a radar cross section pattern with conventional ray tracing are given by Turner [28]. Using a Cray 1S, a fairly simple tank or aircraft would take around 1 second per frame to render. Similar times are obtained using dedicated graphics hardware and a conventional PC.

Few of the methods take account of small gaps on the model surface [31, 26] or surface travelling waves. Kim and Ling [34] derive a ray technique for large inhomogeneous objects such as aircraft and missile plumes. This technique is also used for developing range profiles and ISAR images directly from the CAD models [35].

Element analysis

All the above methods are restricted to high frequency analysis where the bodies are electrically large and therefore mainly in the optical region. For smaller bodies that lie in the Raleigh and Mie regions, the computational approach is usually based on finite element analysis [36, 37]. These methods are also used to compute the radar cross sections of difficult subjects such as cavities [38] and targets coated with radar absorbent material [39].

Discussion

The data generated from surface models can usually be considered as realistic. The surface model used as the basis for the calculations must be modelled to very high levels of fine surface detail. It has been shown that minor discrepancies in the models can lead to major deviations in the radar cross section [32, 40, 28]. The cross section can be calculated to any required resolution and from any angle and range. As the model is virtual, no supports are necessary and so no support artifacts occur in the data. The only noise present in the data will be due to the precision of the calculations performed to generate the radar cross section. Unfortunately, calculating radar cross section in real-time is unlikely to be possible in the near future, even with very powerful computers.

Andersh et al. [32] concludes that improper modelling of small features and materials contribute to major errors in the predictions. He also states that the ultimate level of fidelity for CAD geometry is unknown. Turner [28] demonstrates this problem, showing radar cross section calculations for an unfitted and a fitted ship. The resulting cross section plots are dramatically different whereas the CAD drawing of the fitted ship appears to differ only in surface detail. The effects of fine surface detail are analysed by Williams [40] who indicates that the effects of the surface micro-geometry often dominate the radar cross section for large, complex bodies.

1.3.6. Conclusions

Lees and Davies [27] discuss the pros and cons of many prediction methods with respect to their accuracy at estimating the true radar cross section of a body. They conclude that good surface representations are required if the radar cross section pattern is to be predicted accurately. For this study, true cross section calculations are not the objective. Only signals that have similar characteristics to those seen by a missile, with a monopulse seeker head, flying at the target are required.

As the only true way to determine what the seeker head will see and do is to fly the missile at the target, a synthetic missile must be used in a virtual world scenario. Both radar-echo phase and magnitude must be modelled to allow the effects of glint to be determined. The model must be operated in a scenario where the full 4π steradians may be observed.

Table 1.1 shows the results of a decision analysis process. The techniques are scored against the requirements with

0	–	Bad
1	–	Average
2	–	Good.

The results are summarised in table 1.2.

	Requirement					
Technique	Speed	Fidelity	Resolution	Correlation	Storage	Total
Real Data	2	2	1	2	0	7
Scatterer Model	2	2	2	2	1	9
Statistical Model	2	0	2	0	2	6
Structural Model	1	2	2	2	1	8

Table 1.1: Decision analysis matrix for establishing best technique

Rank	Technique	Total Score
1	Scatterer Model	9
2	Structural Model	8
3	Real Data	7
4	Statistical Model	6

Table 1.2: Summary of decision analysis matrix results (table 1.1)

Of the different methods available for generating a synthetic radar cross section, the most practical appears to be the use of point scatterer models. Point scatterer models allow the radar cross section to be calculated quickly for any angle and any frequency. Any correlations between cross section and motion are inherent in the model. Complex cross section patterns may be represented easily with a moderate number of scatterers. Point scatterer models allow the effective resolution of the data to be increased by interpolating between measured sample points. The interpolation is non-linear and is related to the arrangement of the scatterers. The interpolated data therefore appear as a realistic radar cross section pattern.

If point scatterer models are used, they have to be generated from some known radar cross section data first. These data may be measured from a real target or could be synthetic data generated from a CAD surface model. There are two main requirements for the generation method:

1. **Coverage** – The point scatterer models need to give 4π steradian coverage for effective simulations.

2. **Fidelity**

The four existing methods below have been found for extracting point scatterer models from radar cross section data.

1. **Dominant Feature Analysis** – This technique uses detailed target models to identify the major scattering centres, such as corner reflectors and flat surfaces etc.. Three-dimensional models are generated but the radar cross section is only an approximation of the true radar cross section [23].

2. **ISAR** – Scatterer coordinates and amplitudes may be found directly from ISAR images. High fidelity models may be generated in both two and three dimensions from 2D/3D data [41, 42, 43, 44, 45, 46]

3. **Sinogram** – The range profiles of the target for an angle of revolution are laid side-by-side to generate the sinogram. Scatterers appear to create sinusoidal patterns in the sinogram that are related to the scatterers coordinates and amplitude. High fidelity models may be generated, but in practice, only in two dimensions [17].

4. **PTD/SBR** – The Physical Theory of Diffraction / Shooting and Bouncing ray technique allows point scatterer models to be generated directly from CAD models of targets. The fidelity of the models is restricted to the predictive capability of the Physical Theory of Diffraction [47].

Table 1.3 shows the results of a decision analysis process. It is clear that the best method for generating high fidelity point scatterer models is from ISAR images as this process allows both real and synthetic 3D data to be used. There are three main requirements for the model extraction process:

	Requirement		
Technique	Coverage	Fidelity	Total
Dominant Feature	2	1	3
ISAR	2	2	4
Sinogram	0	2	2
PTD/SBR	2	1	3

Table 1.3: Decision analysis matrix for establishing best model generation method

1. **Coverage**

2. **Complexity** – The method must be able to process satisfactorily images that require large numbers of scatterers to achieve high fidelity representations.

3. **Processing Speed** – The extraction process should be as fast to execute as possible.

The literature survey revealed the three existing approaches to extracting a model from an ISAR image listed below.

1. **Contour Processing** – Contour lines are drawn on the image. Scatterers appear as closed loops whose centres indicate the scatterers position. The processing is slow and only really viable for two-dimensional images [42].
2. **Least Squares / Prony's Method** – Scatterer positions and amplitudes are fitted using Non-Linear Least Squares or Prony's Method. These methods are only suitable for very small numbers of scatterers [41, 43, 44, 45].
3. **Iterative Peak Finding** – The image is searched to find the highest peak. A corresponding scatterer is generated and the peak is removed. The process is repeated until all the peaks of interest have been found. The method requires vast amounts of memory and is very slow for real data [46].

Table 1.4 shows the results of a decision analysis process.

	Requirement			
Technique	Coverage	Complexity	Speed	Total
Contour Processing	0	1	1	2
Least Squares / Prony	2	0	1	3
Iterative Peak Finding	2	2	0	4

Table 1.4: Decision analysis matrix for establishing best model extraction approach

Of the three methods, all of them are slow to process but the iterative peak finding approach is the most versatile, being able to process both two and three dimensional data and cope with large numbers of scatterers. The process suffers from the following problems:

- Requires high resolution data to accurately locate scatterers.
- Heavy processing requirements.
- Large data storage overhead.

These problems need to be overcome in any proposed solution. At present, the method is not practical with the computational resources available to this project.

1.4. Sequence of Work

The following areas of work were proposed.

1. Devise a method for automating model conversion and reducing the processing requirements. The following should be addressed:

1.1. Operate within the processing and memory restrictions of a desktop PC
1.2. Able to process low, medium or high resolution ISAR data
1.3. Minimise the number of scatterers used in the models.

2. Integrate point scatterer models into engagement scenario.

3. Establish level of model fidelity required in simulations.

4. Investigate potential of using glint for detecting target manoeuvres.

1.5. Approach and Objectives

The relationship between the work, the thesis chapter, and the flow of data, is illustrated graphically in figure 1.1. In the figure, boxes labelled as *external* relate to details outside the scope of this thesis. The approach taken in addressing each of the work items is as follows.

1.5.1. Item 1 – Automating model conversion

As described earlier in section 1.3.6, the model conversion problem is best tackled using an iterative approach to identify scatterers from three-dimensional ISAR images. Bhalla and Ling's approach [46] does not satisfy the requirements though. The requirements will be addressed as follows.

Item 1.1 – Operation with restricted resources

Bhalla and Ling's method identifies the largest scatterer in the image and generates a corresponding scatterer in the model. An approximation of the ISAR image of the single scatterer is generated and then subtracted from the original data. To calculate the image of the model, for an image of 640 pixels on each axis, $3 \times (640^2) = 1,228,800$ Fast Fourier Transforms (FFT) need to be performed. On a typical desktop PC, it would take around two hours to perform the transforms. To find the brightest spot in the image, around a Gigabyte of data must be accessed. To generate a model with 100 scatterers on a desktop PC using this method would take over a week.

The approach taken to reduce the processing time is to try to identify multiple scatterers in each iteration. A technique based on Genetic Algorithms (GA) [48] is used that allows multiple scatterers to be identified in each pass of the data. Genetic algorithms are very effective when applied to optimisation problems that are discrete, non-continuous or multi-modal ([49] gives review of genetic algorithm applications in electro-magnetics).

For each iteration, one image is produced of *all* the scatterers found in the pass, thus dramatically reducing the number of Fourier transforms required overall. An added benefit is the genetic algorithm only needs to access a fraction of the data to identify the same number of scatterers as the existing method, thus improving the speed further. The multi-species genetic algorithm will cut the processing time for a 100-scatterer model to around 1 day.

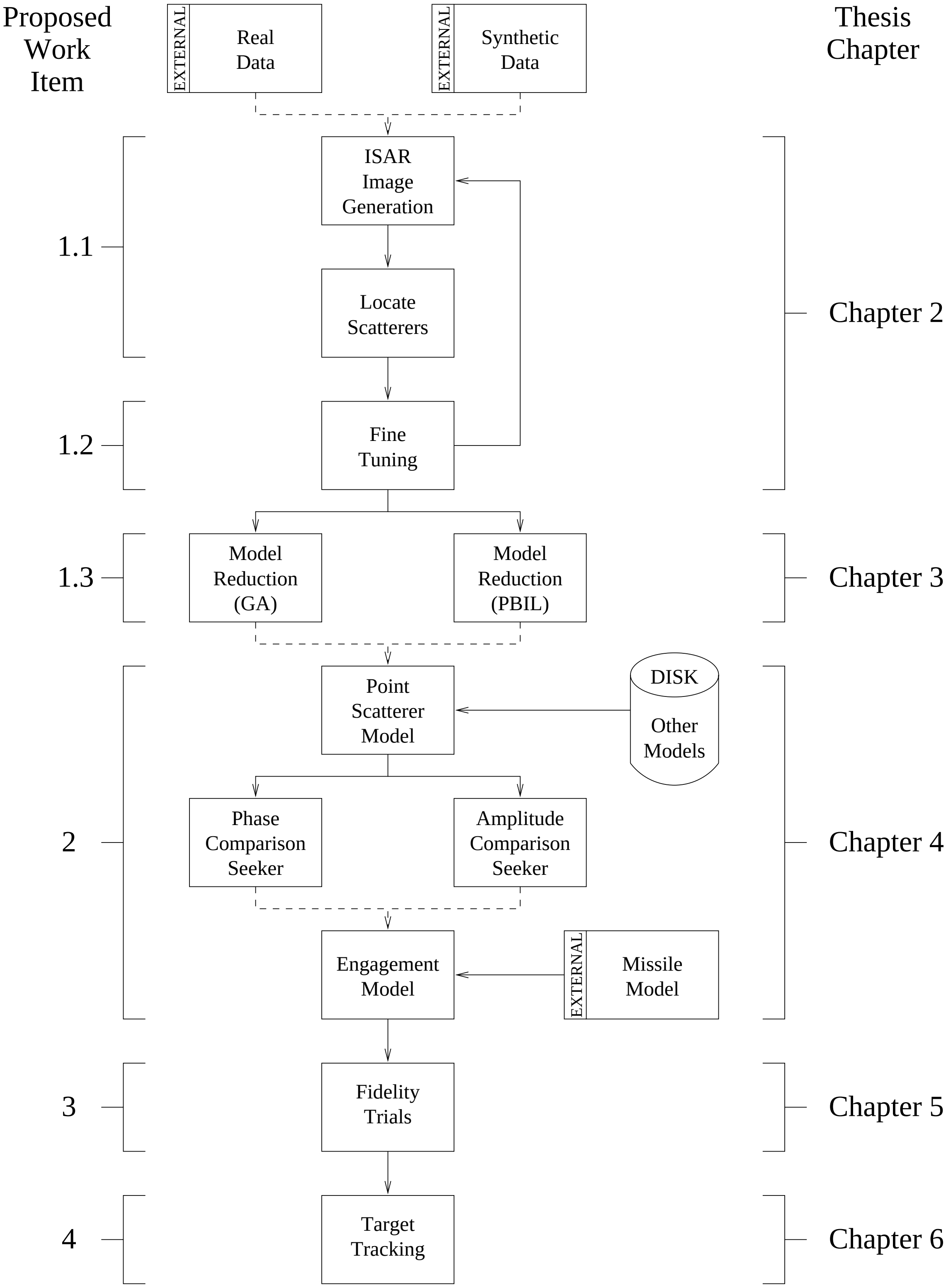


Figure 1.1: Outline of approach and thesis structure

Item 1.2 – Image resolution independence

As the image resolution is reduced, locating the scatterers accurately becomes impossible. A second genetic algorithm is used to adjust the raw position information in an attempt to minimise positioning errors. This fine-tuning process enables high, medium or low resolution data to be converted into models.

Item 1.3 – Model reduction

The fitting process can generate large numbers of scatterers for high fidelity models. If a model with a reduced level of fidelity is satisfactory, some of the scatterers may be discarded, resulting in a smaller model and therefore faster engagements. A third evolutionary algorithm is used to perform a combinatorial optimisation of the model.

Objectives for item 1

The two objectives for this section were as follows.

1. A small amount of low-resolution, two-dimensional, real data have been obtained. The data cover a ±20° azimuth sweep in 256 steps, looking nose-on to the target. The azimuth sweep is repeated over 256 frequencies in the range 2.5 to 3GHz. A 64-sample azimuth section has been extracted from the data that corresponds to approximately ±5°. The objective is to fit a two-dimensional model to this section of data. The radar cross section data for the azimuth sweep at the centre frequency and the frequency data at 0° azimuth should fit the model to an $\alpha = 0.9$ Kolmogorov-Smirnov (Appendix D) confidence level.

2. In the absence of real three-dimensional data, a synthetic model with fifty scatterers will be generated, based on the scatterer locations identified from the real 2D data. This 50-point model will be used in subsequent trials as a realistic 3D truth model. An ISAR image will be generated with 64-steps on each axis and over an angular range of ±1° in azimuth and elevation. The objective is to fit a three-dimensional model to the data. The radar cross section data for the azimuth and elevation sweep at the centre frequency and the frequency data at 0° azimuth should fit the model to an $\alpha = 0.9$ Kolmogorov-Smirnov confidence level.

1.5.2. Item 2 – Integration into the engagement scenario

The target is tracked by the missile using a monopulse radar seeker-head (section 4.3). Two different variants of the seeker-head, phase-comparison and amplitude-comparison, are used in current missiles. Models of the two types need to be evaluated to determine which may be most suited to target manoeuvre detection.

It is hypothesised that the miss distance distribution of trials flown from different directions against a stationary target should mainly be influenced by thermal noise, seeker head type, and the target's radar cross section.

Objectives for item 2

1. Evaluate the performance of a missile with each seeker against an ideal single point target with constant radar cross section to verify that the missile miss distance is near constant.

2. Repeat objective 1 against the 50-point truth target to generate miss-distance distributions and therefore validate the hypothesis.

1.5.3. Item 3 – Establishing fidelity

As the number of scatterers in the model directly influences the execution time for each engagement, the fewer scatterers in the model, the faster the trial. Reducing the number of scatterers in the model though may reduce the fidelity of the radar cross section representation.

The required radar cross section pattern can be fitted to a model with a smaller number of scatterers, but with a residual error level that increases as the model size is reduced. The way in which the missile is influenced by the reduction in fidelity must be investigated.

Objectives for item 3

1. Test the effect on the missile miss distance of reducing the number of scatterers in the 50-point truth model.

2. Repeat on a 100-point randomly generated model to allow the effects of model complexity and structure to be analysed.

3. Establish that the acceptance levels used in fitting the 2D and 3D ISAR data are valid.

4. Produce a high-fidelity reduced model for the target manoeuvre detection trials.

1.5.4. Item 4 – Target manoeuvre detection using glint

Any indication of a manoeuvre before the target deviates from its current track would be beneficial to the missile guidance system. Currently, optical methods exist for ground based tracking systems [2] but these methods require expensive extra hardware. The majority of radar guided missiles use monopulse seeker techniques and are therefore susceptible to glint. If manoeuvre information could be extracted from the seeker bore-sight error signals, no extra sensors would be required. The volume and cost of the seeker head could then be reduced.

Objectives for item 4

1. Investigate the noise on the monopulse bore-sight error signal.

2. Attempt to identify manoeuvres from glint errors. Use straight, level, flight with a single bank turn manoeuvre.

1.6. Original Work and Publications

1.6.1. Original Work

The following itemises the original work and ideas.

- **Multi-Species Genetic Algorithm** – Development of a novel multi-species genetic algorithm to locate multiple scatterers in ISAR images. This algorithm drastically reduces the processing overhead involved in existing model conversion techniques. The processing uses a novel species' statistics idea to increase execution speed (chapter 2).

- **Genetic Algorithm Based Fine-Tuning Method** – The fine-tuning process allows low and medium resolution ISAR images to be converted into point scatterer models. The existing iterative technique is restricted to processing only high resolution images. (chapter 2).

- **Evolutionary Algorithm Based Model Optimisation** – These techniques allow large models created from real data to be used efficiently in the simulation environment. The effects of the model reduction on the fidelity of the representation can be traded against simulation speed by the user (chapter 3).

- **Binary Space Partition Trees for Model Selection** – This novel approach gives a flexible generic structure for combining small models to allow the radar cross section coverage to be extended. Models derived from limited amounts of real data can be combined easily with synthetic structures to give 4π steradian coverage (chapter 4).

- **Analysis of Model Size and Fidelity** – The fidelity analysis using synthetic engagements provides models that are fast but sufficiently detailed to deceive the missile. The effects of fidelity have been established, allowing model complexity to be reduced while maintaining realism (chapter 5).

- **Target Manoeuvre Detection Using Glint** – This novel approach to target manoeuvre detection is cheap, very reliable, and utilises existing sensors. The research has also provided new insights into the structure requirements for realistic targets (chapter 6).

- **Optimal Tuning of PBIL Algorithms** – Two new parameters have been introduced to the algorithm, allowing performance criteria to be derived. Optimality requirements have been established for one parameter and empirical settings for two others have been proposed. The functions of all the parameters are now known, allowing Population Based Incremental Learning optimisation algorithms to be tuned easily (appendix E).

1.6.2. Publications

Author

Evan James Hughes, Transfer report: Radar cross section modelling., Royal Military College of Science, Cranfield University, February 1997.

Major Author

E. J. Hughes, M. Leyland, and B. A. White. "A multi-species genetic algorithm applied to radar scattering centre identification in three-dimensions.", In *GALESIA '97 Conference*, pages 472–477, Glasgow, UK, 1–4 September 1997. IEE Pub. No. 446. Won prize for best paper at conference.

E. J. Hughes and M. Leyland. "Radar cross section model optimisation using genetic algorithms.", In *RADAR '97 Conference*, Edinburgh, UK, 14–16 October 1997. IEE Pub. No. 449. Pages 458–462.

Joint Author

E.J. Hughes, P. Creaser, N.N. Jackson, M. Leyland, J.S. Dahele and B.A. White. Radar Target Augmentation Study. Technical Report for Agreement No. LSC/2004/115 with DRA, Malvern. November 1996

E.J.Hughes, P.Creaser. Engagement Model Software Defining Specification. Technical Report No. DAPS/EJH/17/97, Royal Military College of Science, Cranfield University, April 1998

2. Scatterer Location and Tuning

2.1. Introduction

This chapter covers the scatterer location and tuning phases of the model extraction process. First, an overview of the complete extraction process is given. ISAR theory is then covered in detail and then an introduction to genetic algorithms is given. The multi-species genetic algorithm for scatterer identification is described in detail. The genetic algorithm used for fine-tuning the scatterers is introduced, and non-dominated ranking is covered. Finally the fitting-cycle termination method is detailed.

2.2. Overview

Figure 2.1 shows a block diagram of the complete model extraction process. The radar cross section data need to be measured over a small range of azimuth and elevation angles and for a spread of frequencies. These data can then be used to form a three-dimensional inverse synthetic aperture radar image. This image is analogous to an optical hologram and allows the rough spatial locations of the major scattering centres to be identified [50]. A typical ISAR image is shown in figure 2.4.

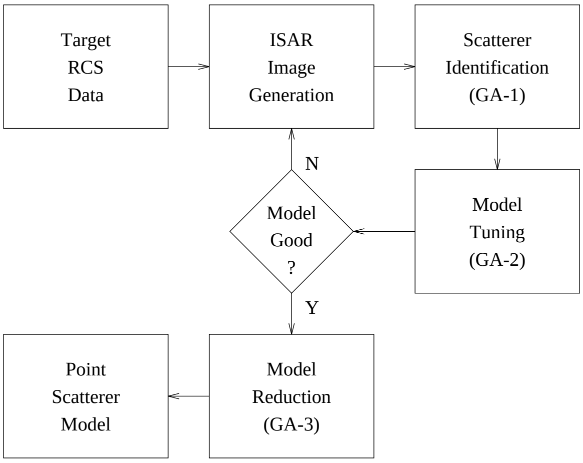


Figure 2.1: Block diagram of complete model extraction process

The positions of the scatterers are located using the first of the genetic algorithms. This genetic algorithm has a population split into multiple species and is capable of identifying multiple scatterers in each run. A model is generated with scatterers located at the rough positions identified from the image and the second genetic algorithm applied. This algorithm is designed to fine-tune the location of the scatterers to improve the accuracy of the model.

The radar cross section of the model is checked against the required target pattern. If the model does not fit, an ISAR image is generated of the model. This image is subtracted from the original image, revealing the scattering centres that have not yet been included. The identification / fine-tune cycle is repeated until the radar cross section of the model matches the required data satisfactorily. At this point, the model may have well in excess of one hundred scatterers.

Once satisfactory model elements have been generated when compared to the ISAR image and radar cross section data, the model must be reduced to a convenient size and fidelity. The third genetic algorithm performs a combinatorial search of different numbers of scatterers and configurations in an attempt to reduce the number of scatterers in the model whilst minimising the induced error. A *Pareto-optimal* [51, Pages 197–201] set of evaluated solutions is generated, allowing the designer to trade between final model size and the accuracy of the radar cross section to the original data. Final models suitable for high fidelity simulations often contain around one hundred scatterers.

2.3. ISAR Images

2.3.1. Introduction

Conventional radar processes often see the fluctuations in the radar cross section of a target as noise and attempt to remove them via signal processing. Inverse Synthetic Aperture Radar (ISAR) techniques exploit the variation of radar cross section with relative target motion to generate spatial images of the radar target. Coherent processing of the returned echoes allows the locations of the target scattering centres to be resolved both in *cross range* (perpendicular to line-of-sight) and *slant range* (parallel to line-of-sight).

The motion of the target relative to the radar is used to generate the diversity of information required about the target. It is assumed that the targets have dimensions that are small compared with the target range and images are obtained from observations made over small segments of viewing angle. These assumptions simplify the theoretical analysis. Data obtained under conditions that violate these assumptions can produce distorted images. In operational scenarios, long range imaging of non-cooperative aircraft and ship targets is possible using ISAR techniques.

2.3.2. Basic ISAR Theory

If a rotating target is observed with a radar that has a high range-resolution, n_s complex samples per range profile for each of N range profiles can be obtained during time T, while the target rotates through angle $\Delta\theta$. Each of the n_s samples form a single range cell. The size of these range cells determine the resolution of the range information. The Doppler frequency shift produced by a scatterer for small $\Delta\theta$ is proportional to the relative target angular rotation rate as well as cross range distance between the scatterer and centre of target rotation [15]. One Doppler spectral line will exist for each Doppler resolved scatterer, the magnitude being proportional to the reflectivity of the resolved scatterer. The targets reflectivity can be mapped therefore in both slant range and cross range. The cross range scale factor is dependent on the relative target angular rotation rate.

The orientation of the rotation axis relative to the radar Line-Of-Sight (LOS) establishes the orientation of the image plane. This plane always lies so as to define the slant range as being parallel to the LOS and the cross range direction perpendicular to both the line-of-sight and axis of rotation. Thus the best images are obtained when the LOS and axis of rotation are perpendicular. If they are parallel, no two dimensional images may be formed.

The relationship between a scatterer's position and the resulting Doppler frequency is shown in figure 2.2. If the target rotates at ω radians per second, a single scatterer at cross range distance r then has instantaneous velocity ωr towards the radar.

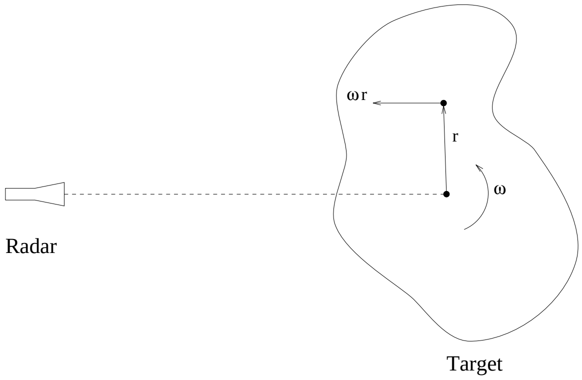


Figure 2.2: Relationship between scatterer and Doppler frequency

Equation 2.1 details the Doppler frequency shift produced by rotation over a small angle.

$$f_D = \frac{2}{c}\omega r_c \overline{f} \tag{2.1}$$

Where $\overline{f}$ is the centre frequency of the radar, c is the propagation velocity and r_c denotes cross-range.

If two scatterers are in the same slant range cell and are separated by a cross range distance of δr_c, then the received signals are separated by a frequency δf_D and therefore cross range separation may be defined as in equation 2.2

$$\delta r_c = \frac{c}{2\omega\overline{f}}\delta f_D \tag{2.2}$$

Doppler resolution may be related to the coherent integration time as $\Delta f_D \approx \frac{1}{T}$ [15, Page 278]. Cross range resolution for a small change in viewing angle, $\Delta\theta$, occurring during integration time T may be given by equation 2.3.

$$\Delta r_c = \frac{c}{2\omega T\overline{f}} = \frac{1}{2}\frac{\lambda}{\omega T} = \frac{1}{2}\frac{\lambda}{\Delta\theta}\bigg|_{\Delta f_D = \frac{1}{T}} \tag{2.3}$$

A discrete Fourier transform may be used to convert the time-history samples from the N range profiles collected over time T into a Doppler spectrum.

The slant range time history may alternatively be obtained using *stepped frequency* data. The target echo is measured at a series of discrete frequencies and the Inverse Discrete Fourier Transform taken, yielding the time response and therefore the synthetic range profile of the target. Range resolution can be defined as the range increment between any two adjacent discrete range positions. A set of n frequency steps spaced Δf apart, produces n equally spaced range increments within the unambiguous range length $\frac{c}{2\Delta f}$. Equation 2.4 details the slant range resolution Δr_s.

$$\Delta r_s = \frac{c}{2n\Delta f} \tag{2.4}$$

If the angular segment over which the radar cross section data are recorded, $\Delta\theta$, is large ($> 10°$ or so), the assumption that the Doppler frequency of a scatterer remains constant begins to break down. The effect is to blur the resultant image. If the segment is not too large, the blurring can be corrected by *focusing* the ISAR image. The raw radar echo data are measured in polar form. If the measurement angle segment is small, the data can be processed as if they are in a rectangular coordinate system. For large angles, the focusing process converts the raw data from a polar form to a true rectangular coordinate system [15, Page 311]. Figure 2.3 illustrates the re-sampling process. A uniformly spaced, rectangular grid is laid over the polar data. The echo data at the new sample points are obtained by interpolation. The spacing of the new grid does not necessarily have to be related to the original sampling scheme. The ability to change the sample points allows the final image resolution and scale to be altered, although significant errors can be introduced for major scale and resolution changes. The interpolation process will introduce errors into the image though. These errors are often small compared to errors arising from the assumption that the polar data can be treated as rectangular.

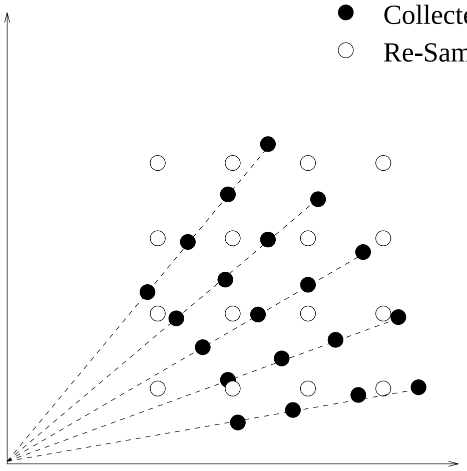


Figure 2.3: Re-sampling radar cross section data prior to imaging

The image is produced from stepped frequency data by first focusing and then performing an Inverse Discrete Fourier Transform (IDFT) to convert the frequency data to synthetic range profiles. If the range data are stored in the rows of a matrix, the next step is to perform a Discrete Fourier Transform (DFT) on the *columns* of the data to resolve the cross range Doppler information. The usual windowing requirements for Discrete Fourier Transforms should be met to reduce spectral spreading in the final image. Typically a *raised cosine* window is sufficient and is shown in equation 2.5. In practice, the Fast Fourier Transform (FFT) is used in the calculations.

$$W = \left.\frac{\cos(\theta)+1}{2}\right|_{-\pi\leq\theta\leq\pi} \tag{2.5}$$

2.3.3. Summary

The effective scaling of ISAR images in cross-range (up–down, left–right) and slant range (front–back) is determined by the sample step size and number of samples in azimuth, elevation and frequency. Equations 2.6 and 2.7 show how to calculate cross range and slant range resolution respectively.

$$\begin{aligned} r_c &= \frac{c}{2n_c\delta\theta\overline{f}} \\ w_c &= n_c r_c \end{aligned} \tag{2.6}$$

$$\begin{aligned} r_s &= \frac{c}{2n_s\delta f} \\ w_s &= n_s r_s \end{aligned} \tag{2.7}$$

where n_c is the number of steps in cross range, n_s is the number of steps in slant range , $\delta\theta$ is the angular step size in radians, δf is the frequency step in Hertz, c is the speed of propagation in metres/sec., and $\overline{f}$ is the mean frequency in Hertz. The cross range and slant range resolutions are denoted by r_c and r_s while the total range extent are w_c and w_s respectively. All range measurements are in metres. Care must be exercised in the choice of angular window ($n_c\delta\theta$) that the measurements are taken over. A window greater than 10° will begin to cause a blurring at the edges of the image. Focusing techniques can be used, reducing the effects of the wider angular coverage, but more scatterers are eventually required to fit accurately the radar cross section to the model. Image generation involves the application of Fourier transforms to the radar cross section data and therefore suffers from the spectral spreading problems inherent in this process. The genetic algorithms do not require the application of window shaping functions to reduce spreading effects. The inherent square window leads to sharp peaks with long tails extending into the image. The long tails aid the search abilities of the first genetic algorithm as they help indicate the locations of the peaks.

It has been proposed that inverted windows that enhance the size of the tails and make the main peak narrower may improve the genetic algorithm performance. Trials of the inverted raised-cosine window shown in equation 2.8 were performed. Five trials of each window were conducted and the number of peaks found in each trial was recorded. Using the Mann-Whitney test of means at $\alpha = 0.05$, no significant difference was found in the ability of the genetic algorithm to identify the peaks.

$$W = 1 - \left.\frac{\cos(\theta) + 1}{2}\right|_{-\pi \leq \theta \leq \pi} \tag{2.8}$$

2.3.4. Example Image

Figure 2.4 shows a typical ISAR image. The image was generated from real data over an azimuth sweep of −5.1° to 5.4° in 64 steps and for a frequency sweep of 2.5 to 3 GHz at 0° elevation in 256 steps. The data are un-focused and are subject to the inherent square window of the FFT process. The image has a resolution of 0.3 metres per sample in both cross range and slant range. The shape of the aircraft is clearly visible in the image. The engines and nose (upper part of image) are easily distinguished. The tail fin (towards the bottom)and wing tips (left of image) are less visible.

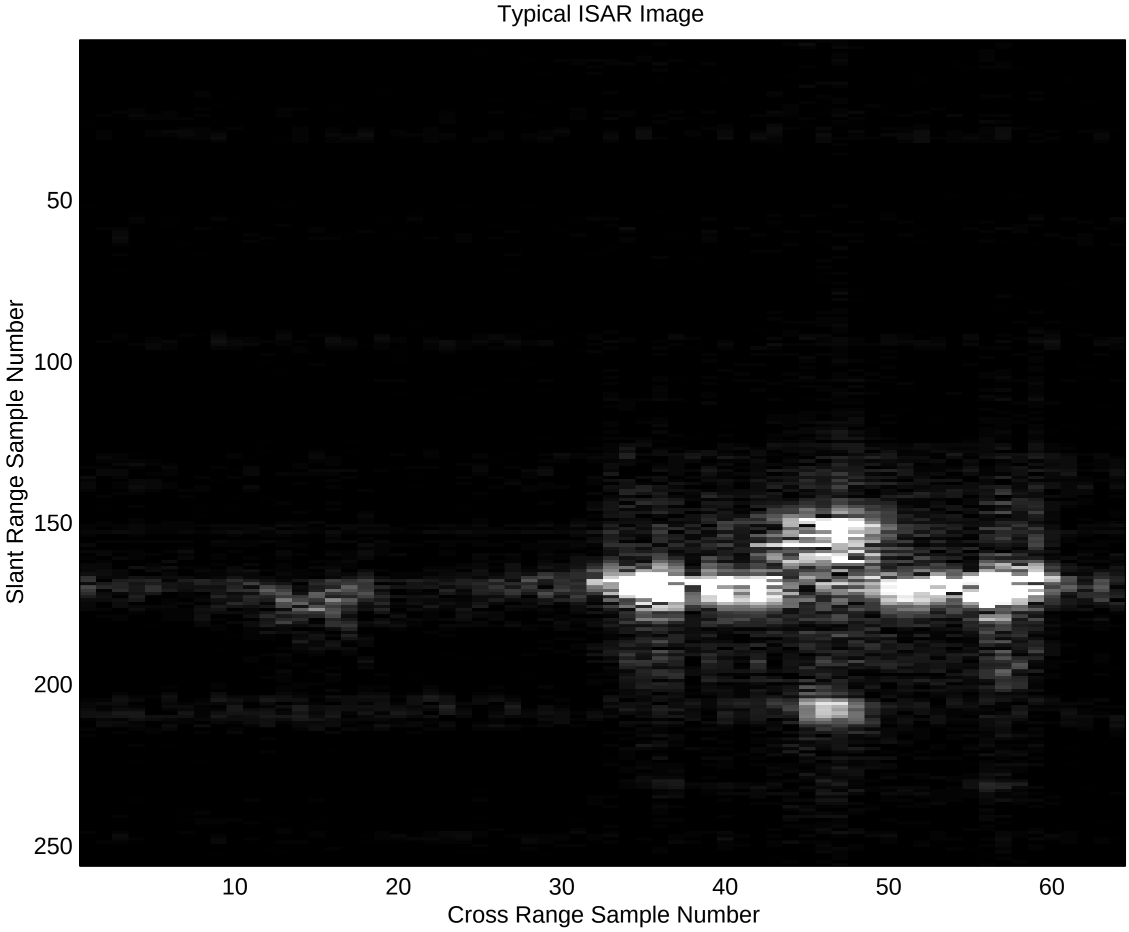


Figure 2.4: ISAR image example

2.4. Genetic Algorithms

Genetic Algorithms [48] are designed to mimic the natural selection process through evolution and survival of the fittest. A *population* of M independent individuals is maintained by the algorithm, each individual representing a potential solution to the problem. Each individual has one *chromosome*. The chromosome is the genetic description of the solution and may be broken into n sections called *genes*. Each gene represents a single parameter in the problem domain. Therefore, a problem that has five unknowns for example, would require a chromosome with five genes to describe it.

The three simple operations found in nature, natural selection, mating and mutation are used to generate new chromosomes and therefore new potential solutions. Each individual's chromosome is evaluated at every generation using an *objective function* that is able to distinguish good solutions from bad ones and to score their performance. With each new generation, some of the old individuals die to make room for the new, improved offspring. Over several generations, the majority of the solutions represented by the individuals in the population will tend to lie around an optimal solution for the given environment. The exact rate at which the population converges to a single solution is determined by the nature of the problem and the structure of the genetic algorithm.

When used to solve optimisation problems, genetic algorithms tend to search areas spread across the entire optimisation surface before converging on a maximum or minimum depending on the problem. Thus, despite being very simple to code, requiring no directional or derivative information from the objective function and being capable of handling large numbers of parameters simultaneously, genetic algorithms can achieve excellent results.

The method can be described by the following algorithm:

1. Create a population of M individuals, each having a chromosome with gene values chosen at random.

2. Assess the performance of each individual.

3. Rank individuals with respect to performance and assign a Fitness Value dependent on ranking.

4. Create a set of M parent individuals for breeding where the probability of being included in the set is proportional to fitness. The fitness consideration may lead to some individuals being chosen many times and others not at all.

5. Randomly pair parents and breed to form M offspring.

6. Randomly mutate some of the genes in the offspring chromosomes.

7. Offspring become new population, assess the performance of each individual.

8. Record best individual.

9. Repeat from step 3 for required number of generations.

2.5. Location Process

Bhalla and Ling's iterative method [46] operates by first finding the size and coordinates of the brightest spot in the image. A scatterer is placed in the corresponding position in the model. An ISAR image of this scatterer is produced and subtracted from the original image to remove the corresponding bright spot. The process is repeated until all the major bright spots have been removed.

The method works well but has one major drawback; it requires high resolution data in order to locate accurately the centre of each scatterer. A typical three dimensional image of 640 pixels on each axis will require $(640)^3$ elements and therefore two Giga-bytes of storage space. Finding the location of the maximum value necessitates searching the entire set of data for each scatterer that is resolved. Images often require 100 or more scatterers for accurate representation and therefore the equivalent of 200 GBytes of data must be retrieved from the storage media. On a small system, the data access and transfer times are significant. To remove each scatterer, $3(640^2) = 1,228,800$ Fourier transforms are required to create the ISAR image. For a model where 100 scatterers are identified, the processing would take longer than a week on a desktop PC. This is impractical.

A genetic algorithm may be used to locate multiple bright spots in one pass. These bright spots can then be formed into a model, and the model's effects subtracted from the original image as before. Further applications of the genetic algorithm will locate any smaller points remaining. Generating an ISAR image of one scatterer takes almost as long as generating an image of ten scatterers. Thus by processing multiple points in each pass of the data, vast savings can be made in image generation time. This multi-modal function approach can make model calculation viable on a small system.

2.5.1. Multi-Modal Optimisation and Sharing

Most genetic algorithms use a single population of a single species. The algorithms are designed so the solutions represented by the different individuals converge on the single optimum solution of the objective function. In multi-modal optimisation, the genetic algorithm is designed to converge with multiple solutions, each corresponding to a separate peak in the objective function.

There are a number of mechanisms that may be used to force a genetic algorithm to exhibit multi-modal behaviour.

1. **Iteration** – Many independent runs of the genetic algorithm are performed in an attempt to identify all the peaks. This method is very inefficent as the larger peaks will often be found many times [52, Page 176].

2. **Sharing** – The sharing system operates by modifying the objective value that is seen by each individual. If a number of individuals all occupy the same peak in the objective function they are made to share the objective value at that point [53]. This simple concept is enough to allow multiple stable populations to form.

3. **Crowding** – Crowding is a selective breeding technique where offspring are

inserted into the population by replacing individuals that are genetically similar [54]. The process allows multiple stable populations to form.

4. **Sequential Niching** – The process operates by iterating the genetic algorithm but maintains a record of the best solutions found. At each successive iteration of the genetic algorithm, the peaks that correspond to the solutions found in previous runs are suppressed. This method is essentially a sequential version of the sharing process described earlier [55].

In order to identify multiple scatterers in each pass of the genetic algorithm, either the sharing or crowding method must be used.

For the sharing process, a function that is related to the separation distance between two individuals (genotypic or phenotypic space) is used to control the modification of the objective function. Equation 2.9 defines the sharing function used, with $d(\chi(i), \chi(j))$ defined as the distance between the chromosomes $\chi(i)$ and $\chi(j)$, $s(i, j)$ is the sharing effect of i on individual j and α, β are factors for modifying the function shape. When $\alpha = 1$, this function produces a linear variation that moves from unity at zero distance to $1 - \beta$ at a distance of ν and zero thereafter. If $\alpha \neq 1$, the function has an exponential form. Using a value of β less than unity has a similar effect to using high values of α, but without the processing overhead of the exponential calculations.

$$\begin{aligned} s(i,j) &= \begin{cases} 1 - \left(\frac{d(\chi(i),\chi(j))}{\nu}\right)^{\alpha}\beta & d \leq \nu \\ 0 & d > \nu \end{cases} \\ \text{where} \quad d &= d(\chi(i),\chi(j)) \;=\; |\chi(i) - \chi(j)| \end{aligned} \tag{2.9}$$

For each individual, i, the distance is calculated from its chromosome to the chromosome of every other individual, j, in a population of N individuals and the values for each of the sharing functions are totalled (equation 2.10). The result is used to derate the image value at the point defined by the chromosome of i, $I(\chi(i))$ yielding a new objective value $\mathcal{O}(i)$. Equation 2.11 shows the objective calculation.

$$\mathcal{S}(i) = \sum_{j=1}^{N} s(i,j) \tag{2.10}$$

$$\mathcal{O}(i) = \frac{I(\chi(i))}{\mathcal{S}(i)} \tag{2.11}$$

These sharing functions work well but for the large and complex optimisation surfaces found in scattering centre identification, large populations are required. Thus the requirement for every individual to be compared to every other produces a significant processing overhead. In an attempt to reduce the processing requirements, the sharing function has been modified to operate using multiple species rather than individual members [56]. This process combines the niche forming properties of the sharing process with the selective breeding of the crowding algorithm.

The position and spread of a species may be defined by the mean of the species chromosomes and their standard deviation. Equations 2.12 and 2.13 define the position and spread respectively. Where n_k is the number of individuals in a species and $\chi(i, k)$ denotes the chromosome of individual i of species k. If it is assumed that the spread of individuals around the mean position is roughly Gaussian, a sphere with a two standard deviation radius from the mean will encompass the main bulk of the population. Thus $2\sigma_k$ may be defined as representing the spatial distribution of population k.

$$\begin{aligned}
\overline{\chi}_k &= \frac{1}{n_k}\sum_{i=1}^{n_k}\chi(i,k) && (2.12)\\
{\sigma_k}^2 &= \frac{1}{n_k}\sum_{i=1}^{n_k}\left|\chi(i,k)-\overline{\chi}_k\right|^2 \\
&= \frac{1}{n_k}\sum_{i=1}^{n_k} d(\chi(i,k),\overline{\chi}_k)^2 && (2.13)
\end{aligned}$$

Equation 2.14 defines the modified sharing function, where ν_k is a sharing distance that varies with the spread, σ, of the species. The spread of the species is still limited to a minimum distance of ν. This sharing function is then applied to all N_s species, except the members own, and the results summed (equation 2.15). The objective cost for the individual is then derated by one plus the share value to account for the individual itself and is shown in equation 2.16.

$$\begin{aligned}
s(\chi(i,j),\overline{\chi}_k) &= \begin{cases} 1-\left(\frac{d(\chi(i,j),\overline{\chi}_k)}{\nu_k}\right)^2\beta & d\leq\nu_k \\ 0 & d>\nu_k \end{cases} && (2.14)\\
\text{where}\quad \nu_k &= \begin{cases} \sigma_k & \sigma_k > \nu \\ \nu & \sigma_k \leq \nu \end{cases} \\
\mathcal{S}(\chi(i,j)) &= \left.\sum_{k=1}^{N_s}\frac{n_k}{2\sigma_k}s(\chi(i,j),\overline{\chi_k})\right|_{k\neq j} && (2.15)\\
\mathcal{O}(\chi(i,j)) &= \frac{I(\chi(i,j))}{1+\mathcal{S}(\chi(i,j))} && (2.16)
\end{aligned}$$

The sharing function defined in equation 2.14 is based on a squared law rather than a linear function, ie. with reference to equation 2.9, $\alpha = 2$. Individuals that are close to the species centre are affected more than those further away. An increase in calculation speed is gained by not requiring the square root of the magnitude of the distance to be taken. Unlike the fixed shape individual sharing functions used previously, the functions associated with each species are dynamic and vary with the geographical motion of the individuals within the species.

The ratio of the number of individuals in a species (n_k) to the specie's spread ($2\sigma_k$) has been included as a factor in equation 2.10 to give equation 2.15. This ratio makes the influence that each species has on other individuals change dynamically with the specie's spread. Each species is now referred to by its mean position. When a species population is widely dispersed ($\sigma \gg \nu$), the function has little effect on other individuals. As a specie's population converges ($\sigma \simeq \nu$), the range of the function decreases but its influence increases. This added influence forces different species to separate as their populations converge. A minimum distance, ν, for the spread of the function is used to prevent different species from converging too closely to one another. This minimum distance helps increase the diversity of the geographical spread of the species.

2.5.2. Algorithm Construction

A real valued chromosome with three parameters, or genes, has been used to define each individual. The parameters are defined as being the (x, y, z) coordinates of a location in the three-dimensional ISAR image. The genotypic level allows the parameters to have fractional components. The fractional component allows real-valued mutations to be applied. The genes are rounded to the nearest integer to obtain the phenotypic data for the picture element index. The raw objective value is defined as the image intensity at the indexed point.

The genetic algorithm follows the usual format of ranking, selection, crossover, mutation and evaluation but with each species being processed separately. The same number of offspring as parents are generated and a total replacement policy is used. The total replacement policy helps to reduce the rate of convergence and allows the species to relocate themselves to minimise problems caused by overcrowding.

The fitness value $F(x)$ is assigned according to rank position p_x of individual x. The individual with the lowest $\mathcal{O}(x)$ (least fit) being assigned a rank position of 1 and the best individual being assigned rank position M. Equation 2.17 details the calculation of $F(x)$.

$$F(x) = \frac{2s(p_x - 1)}{M - 1} + (1 - s)\Bigg|_{0<s\leq 1} \tag{2.17}$$

Where s is the *selective pressure* [52, Page 56] and may lie in the range $0 < s \leq 1$. The emphasis that is placed upon the selection process may be controlled by adjusting s. A value of zero is never used as any bias between good and bad individuals is prevented and therefore no natural selection occurs. A value of unity gives the maximum selection where the chances of selecting the worst individual are near zero. The effect of reducing the selective pressure is to slow the convergence of the genetic algorithm. In this algorithm, a selective pressure of 0.8 is used. This value has been selected empirically and allows the species' to search the entire image but still converge satisfactorily.

Stochastic Universal Sampling [48, Page 12] is used to select M individuals from the population, each individual having a probability of selection defined in equation 2.18.

$$\text{Prob}(x \text{ selected}) = \frac{F(x)}{\sum_{i=1}^{M} F(i)} \tag{2.18}$$

The individuals selected are randomly shuffled and then paired up for breeding. Uniform Crossover [52, Page 88] is used to generate two new offspring from each pair of parents. This operator swaps individual gene-pairs between the parents with a probability of 0.5. For example, if we take two parents, a and b, both with chromosomes containing two genes, uniform crossover can be used to generate two offspring. The first offspring may have its first gene from parent-b and its second from parent-a. Due to the spectral spreading that occurs with the Fourier transforms in the image generation, each peak has long tails that spread out in the axis directions. Although a real-valued chromosome is used, the uniform crossover is suited to searching the image as individuals often settle onto a tail emanating from a peak. If the two parents are each lying on different tails of the same peak, after crossover, the offspring may lie *exactly* on the peak. Other recombination techniques that create offspring by combining genes proportionally, are unlikely to score a direct hit on the peak.

Genes are mutated with a probability of 0.3. This probability will mutate, on average, approximately one gene per chromosome. The range of the mutation is governed by equation 2.19. This *Non-Uniform* mutation [52, Chapter 6] function is unity initially and progresses to zero at the final generation and is used to modify the maximum deviation from the current gene value, where G is the generation number and G_m is the maximum number of generations. Initially, the gene can mutate to any value within its range, but this range is reduced with time. The range modifier function forces the genetic algorithm to converge on a solution by confining the offspring of each subsequent generation to a diminishing region. In the final stages of the algorithm, the mutation range is limited to a very small locality, forcing the species to converge on the true local optimum. The function shape allows the genetic algorithm to perform a thorough search in the early generations but still retain the convergence properties of non-uniform mutation toward the end phase of the algorithm. The use of a real valued chromosome in this algorithm is due to the use of this specific non-uniform mutation operator. Figure 2.5 depicts the function shape graphically.

$$R(G) = 1 - \left(\frac{1 - \cos\left(\frac{(G-1)}{(G_m-1)} \pi \right)}{2} \right)^2 \tag{2.19}$$

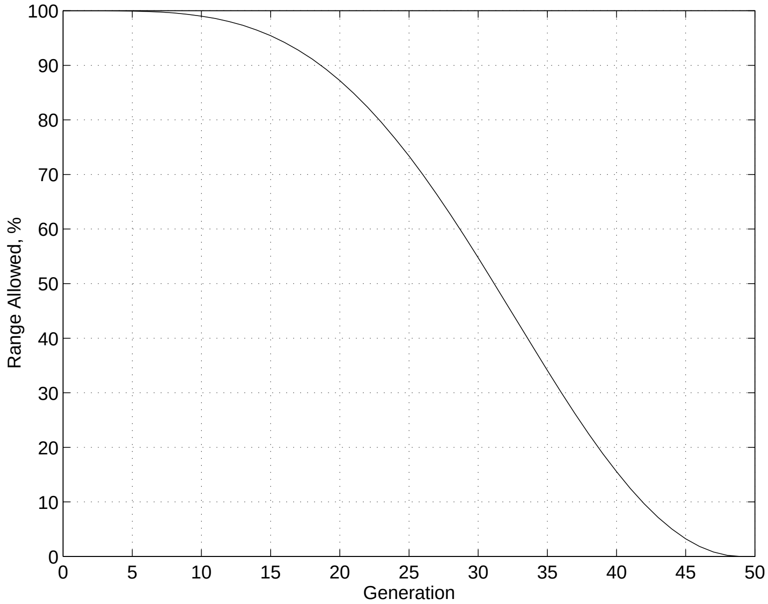


Figure 2.5: Mutation range modifier function

The objective function for the offspring is calculated as detailed previously, based on the statistics of the parent population. Policies other than total replacement have been tried but appear to offer little benefit as the objective function is effectively dynamic with the motion of the species. The use of dual chromosomes and a dominance mechanism (diploidy) has been tried to improve the search process by allowing species to develop a memory of good peaks they have occupied in the past. The effect was to slow the convergence of the algorithm but no improvements in the results were noticed, probably due to the low number of generations used in the algorithm.

The genetic algorithm is terminated after 50 generations and the best overall individual of each species is recorded as a peak location. The small number of generations combined with forced convergence has been chosen to give a consistent execution time. As the algorithm is being applied in an iterative fashion, the requirement is to identify any of the peaks in the image, rather than the highest set. Therefore the algorithm's ability to identify the global peak in each run is sacrificed for speed.

Once the algorithm has terminated, duplicate peaks and any that are within one spread distance ($\leq \nu$) are removed. A range of species is used, each with a different population size. This range of species sizes introduces a slight bias into the algorithm where the smaller species are able to move more rapidly than the larger species but have a weaker hold on any peaks they find. The larger species move slowly but are capable of evicting small species from peaks that are already colonised.

The individual parameters of genetic algorithms are notoriously difficult to tune. For parameters such as mutation rate and crossover rate, simple rule-of-thumb settings are often sufficient. Even large variations in these parameters often produce few noticeable changes in the performance of the algorithm. Typical tuning strategies for the three basic parameters are:

1. **Crossover rate** – Increasing the crossover rate reduces the rate of algorithm convergence in the early stages of the run. The chances of convergence on a local optima are increased. Typically, a crossover rate of unity is used.

2. **Mutation rate** – Increasing the mutation rate increases the rate of convergence in the latter generations of the algorithm. Increasing the mutation rate improves the chances of escaping from local optima, but reduces the ability of the algorithm to converge *exactly* on any optimum solution. A mutation rate that will, on average, mutate one gene per chromosome is often chosen.

3. **Selective pressure** – Reducing the selective pressure slows the convergence of the algorithm and therefore reduces the chances of premature convergence on a local optimum. A selective pressure of unity is often used.

4. **Population size** – Increasing the population size increases the diversity of genes in the population. The rate of algorithm convergence is reduced but it is less likely to converge on a local optima.

In the multi-species algorithms, the extra parameters are tuned as follows:

1. **Minimum spread distance (ν)** – This distance is related to the width of a typical peak in the image. If ν is too small, species may settle on the sides of high peaks. If ν is too large, small peaks that are very close to large peaks may be missed.

2. **Shaping Parameter (β)** – This parameter determines how brutal the effect of ν is. A value for β near unity will allow large species to exist at a distance

less than ν from smaller species. Reducing β will make it more difficult for large species to displace smaller species that have already converged on a peak.

3. **Number of species** – The number of species determines the number of peaks that can be identified in each run. Increasing the number of species will reduce the average number of individuals in each species. Smaller species are more likely to converge on a local optima, rather than search for the highest peaks. Experiments have shown that up to 25 species with population sizes ranging from 15 through to 100 provide good results over a range of different images.

Experiments using different population sizes have yielded some interesting results. By applying a genetic algorithm a number of times to an image in order to identify the highest peak, the probability of finding the peak can be established. It was noticed that as the population size was increased, the probability of finding the peak also increased. Trials with different images indicated that the probability was also related to the relative area that the peak occupied in the image, ie. the larger the footprint of the peak appeared to be, the easier it was to find. This relationship to area is quite intuitive as the larger the initial random population, the more chance there is of generating an individual that lies within the footprint of the peak. It was hypothesised that the probability of finding the peak is related to the area of the peak's footprint, relative to the total image area.

The population based incremental learning algorithm described in appendix E was used in an attempt to prove this hypothesis as the algorithm is very simple and has few parameters to tune. It has been established that for high values of learning rate, the probability of identifying the peak, P_H, follows a binomial distribution. This probability is shown in equation 2.20, where p is the size of the population, P_H is the probability of identifying the peak, and P_1 is the probability of identification for a population size of unity (impossible to do with PBIL, it is estimated using equation 2.20).

$$P_H = 1 - (1 - P_1)^p \tag{2.20}$$

Further experiments have shown that the probability P_1 is related to the footprint of the global peak. Doubling the area the peak covers appears to double P_1. This result allows the uncertainty that the best peak found is the global optima to be quantified approximately. The best result found will have an associated probability, P_1. Using equation 2.21, a population size may be calculated that should give a single run of the algorithm a probability, P_G, of hitting a global peak that has approximately one tenth of the area of the current best peak.

$$p = \frac{\log(1 - P_G)}{\log(1 - P_1/10)} \tag{2.21}$$

Thus if no other, better, solutions are found, it could be said that there is a probability, P_G, that there are no other solutions to the problem up to a tenth of the size of the best found.

Unfortunately, at low learning rates, P_1 does not remain constant with respect to population size. The relationship has not yet been identified. The research has not yet been applied to genetic algorithms but they should behave in a similar manner to PBIL. The work has, however, provided an insight into the operation of the PBIL algorithm and allowed major enhancements to be made.

Comparing the multi-species genetic algorithm to existing approaches that search for the highest peak, if we have an algorithm with 750 individuals and run it for 50 generations, it will require 37500 objective calculations. The algorithm can locate as many peaks as there are species, although a 70% identification rate is more realistic. For an algorithm with 14 species, if it is applied 10 times to identify 100 scatterers in a high resolution image, a total of 375,000 accesses are required to the image data. The conventional iterative model conversion approach accesses nearly 560,000 times as much image data and generates ten times as many ISAR images to achieve the same model resolution.

2.6. Fine-Tuning Scatterer Locations

2.6.1. Introduction

With low-resolution images, the scatterer locations generated by the first genetic algorithm may be a significant distance away from the optimum positions. Even with high resolution data, if two scatterers are very close, the image peak positions may not be truly aligned with the actual scatterer location [57]. By fine-tuning the model, these errors can be reduced. The fine-tuning process will ultimately result in the model requiring fewer scatterers to match the target data and so reduce the burden on the third genetic algorithm.

2.6.2. Tuning Process

The first stage in converting the scatterer image details into a point scatterer model is to register the image with the model. The image registration is achieved by placing a scatterer at the origin of the model and generating an ISAR image of it. The highest point in the image will correspond to the scatterer in the model, giving the image–model zero location, ±1 resolution cell. A rough amplitude scaling can be calculated from the brightness of the peak in the image. Knowing the true image resolution from equations 2.6 and 2.7, the model position of a point that is a fixed distance on each axis away from the centre is calculated. The distance chosen must be related to the position of the centre point so that the peak should not fall outside the image region. If it does lie outside, an aliased peak will be present, but at a false location. This second peak allows the image scaling to be be verified and any scale inversions identified. It is possible for increasing x in the model space to lead to decreasing x in the image etc.. This indicates that the data ordering has been reversed in some way and must be corrected. The image–model registration only needs to be performed once at the beginning of the conversion process.

Once the zero location and scale have been verified, the location of each identified scatterer in the image can be transformed directly into a point in the model, with an accuracy of ±1 resolution cell. If the images are high resolution, for example greater than 512 samples in each axis, the resolution induced error will typically be of the order of a few centimetres or so and therefore close enough for conversion purposes. The amplitude scale factor derived from the zero registration should also be accurate enough for direct conversion. An image of the model can then be subtracted from the original ISAR image to reveal the smaller scatterers that have not yet been located.

If the image is of low to medium resolution, ie. 32 to 512 samples, the error in the scatterer's location becomes progressively worse as the resolution decreases, and may ultimately be a few metres. Experience has shown that the amplitude scaling in these cases may be as much as 100% out. To cope with these images, the raw model positions must be fine-tuned in an attempt to reduce positional and amplitude errors. Figure 2.6 demonstrates the effects of a small number of samples on the image of a scatterer when the scatterer does not align with the sample position. When the peak of the image coincides with the sample location, the indicated amplitude is accurate. As the sample point is shifted left or right, the measurement error increases, with the indicated peak height being less than the actual peak.

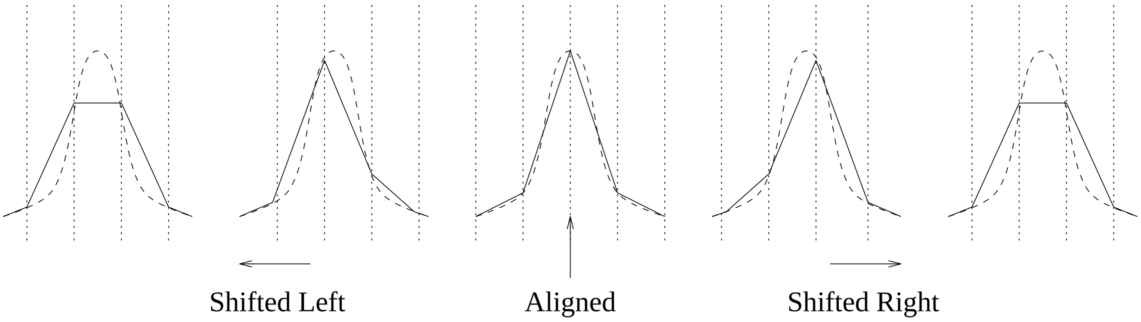


Figure 2.6: Effects of mis-alignment between scatterer and sample instant

The following genetic algorithm is designed to adjust the raw scatterer positions to improve the match to the required image. It should be noted that as the image resolution decreases, the work of this genetic algorithm increases and the load on the first genetic algorithm for scatterer location is reduced. This shift in processing load should be adjusted by the designer for each different image type that is to be processed.

To reduce processing overheads, instead of calculating and comparing full three-dimensional images, three one-dimensional images are used [46]. These images are formed by taking data from the three principle axes of the radar cross section pattern and using a Fourier transform to convert the radar cross section data to range information. Typically, if the radar cross section data in the region around the centre of the azimuth, elevation and frequency bands are of most interest, the radar cross section would be measured first at the mean azimuth and mean elevation and over the full frequency sweep; then at the mean elevation and mean frequency with a full azimuth sweep etc.. If we have a $64 \times 64 \times 64$ sample image, for full conversion, $12,288$ Fourier transforms are required. Only three are required if the one-dimensional approach is used. This major reduction in the processing overhead is offset by reduction in the fidelity of the error measurements. It has been observed that the fine-tune operation is not compromised by the use of a reduced set of data, although if very low resolution data are used, the processing of the full image is not too severe and its use in comparisons may be justified to increase the fine-tuning capabilities of the genetic algorithm.

If the tuning process is not perfect, position and amplitude errors in the model will lead to errors in the image. As the model image is subtracted from the required image, any peaks in the model that are smaller than they should be will leave a positive residue peak in the image after subtraction. This smaller scatterer will be identified in subsequent passes and reduced further. This mode of operation eventually leads to models with an excess of scatterers. If the scatterers in the model are larger than they should be or in the wrong position, a negative result is obtained in these areas of the image after the subtraction process. This error cannot be corrected in subsequent passes of the algorithm and causes bright spots in the ISAR image that are too large. The negative error is highly undesirable and is a problem with any technique that operates by an iterative scatterer subtraction process.

The problem is addressed in the genetic algorithm by calculating the amount of overshoot (negative error) and undershoot (positive error) of the fit separately. A multi-objective approach is used that allows the designer to trade between a slow fitting process that minimises the undesirable overshoot errors but leads to larger models, through to a less stringent fitting scheme that minimises undershoot and therefore uses less scatterers, at the expense of ISAR image accuracy. Alternatively a compromise can be drawn between the two objectives that attempts to minimise the negative error problems without creating a large model.

2.6.3. Non-dominated Ranking

A *Pareto Optimal Set* of results [51, Pages 197–201] may be formed where no one solution is better than any other in both objectives. These solutions are said to be *Non-Dominated* as no one solution can be chosen in preference to the others based on the two objectives alone. There exists a single Pareto optimal set of solutions to the problem. At any intermediate stage of optimisation, a Pareto set of results will have been identified. This set may or may not be the optimal set.

A non-dominated ranking method [58] is used in the genetic algorithm to generate and maintain a Pareto set of results. Conventional genetic algorithms often use a ranking method where the calculated objective values are sorted and assigned a rank that is dependent only upon their position in the list, rather than their objective value. The ranking operation helps to prevent premature convergence of the genetic algorithm. The non-dominated ranking system operates by first identifying the non-dominated solutions in the population and assigning them a rank of one. These solutions are removed from the population and the non-dominated solutions in the remaining set of individuals are then identified, this time assigning a rank of two etc.. The ranking process is continued until all of the individuals have been accounted for. Once all the individuals have been classified, a dummy value (1 in this algorithm) is assigned to all the solutions with rank one. The sharing process detailed in section 2.5.1 is applied to these individuals, reducing their assigned value if they have near neighbours (on a chromosome level). The sharing process ensures that a spread of solutions is obtained across the Pareto front. The minimum value assigned to the level one solutions is identified and then reduced slightly (by 1%). This reduced value is then used as the dummy value for the level two solutions and so on. The resulting objectives are intended to be used with a *maximisation* strategy.

The conventional ranking and selection processes are then applied as normal to the objective obtained by the non-dominated ranking and sharing operation. An elitist strategy is developed that preserves an entire Pareto front of P solutions from generation to generation. To maintain a working population of N individuals, the Pareto set from the previous generation is concatenated with the working population and then N offspring are generated from the $N+P$ parents. After evaluation, the offspring become the new working set of individuals. The new Pareto set is calculated from the population of solutions that results from concatenating the new working set of individuals and the old Pareto set, thus choosing the best from the new solutions and old Pareto front. The number of solutions that comprise the Pareto front, P, is dynamic. Kumar and Rockett [59] discuss procedures that may be used if assurances are required that the true Pareto optimal set of solutions has been identified.

2.6.4. Algorithm Construction

A real valued chromosome is used and is held in a matrix structure that has four columns corresponding to $[\Delta x \quad \Delta y \quad \Delta z \quad a]$, where Δx, Δy and Δz are offsets from the raw scatterer position and a is the amplitude. The chromosome matrix has the same number of rows as the number of scatterers identified in the scatterer location algorithm. The positional offsets are limited to ±1.25 resolution steps and the amplitude is allowed to range from zero to 50% larger than the largest identified scatterer. In the phenotypic space, the scatterers corresponding to each chromosome are concatenated to the previously identified model before the images are generated. For an image resolution of 64 samples on each axis, the algorithm would be run with typically 100 individuals and for 500 generations.

Equations 2.22 and 2.23 show the two objectives that are used in the fitting process, where $E(x,k)$, defined in equation 2.24, is the error between the required image, $I(k)$, at point k and the image of the model, $M(x,k)$, for solution x; N is the number of points in the image; $\mathcal{O}_1(x)$ is objective one and is a measure of mean squared overshoot; $\mathcal{O}_2(x)$ is objective two and describes undershoot. The objectives are both to be minimised to establish the Pareto front.

$$\mathcal{O}_1(x) = \frac{1}{N}\sum_{i=1}^{N}\begin{cases} 0 & E(x,i) > 0 \\ E(x,i)^2 & E(x,i) < 0 \end{cases} \tag{2.22}$$

$$\mathcal{O}_2(x) = \frac{1}{N}\sum_{i=1}^{N}\begin{cases} E(x,i)^2 & E(x,i) > 0 \\ 0 & E(x,i) < 0 \end{cases} \tag{2.23}$$

$$E(x,k) = I(k) - M(x,k)|_{k=1\ldots N} \tag{2.24}$$

The fitness function as defined in equation 2.17 is applied but with a selective pressure of $s=1$. This selective pressure gives the maximum bias towards the most fit solutions. Stochastic universal sampling is used to select N individuals from the $N+P$ set described previously. Uniform crossover is applied where parts of the paired chromosomes are exchanged. A fixed mutation rate of 0.25 is applied along with non-uniform mutation as described previously in section 2.5.2.

At the end of the algorithm, one solution is chosen from the Pareto set. Which solution is chosen is determined by the design strategy that is being employed. A slow but accurate method will chose the solution where $\mathcal{O}_1$ is the smallest (least overshoot) while a less stringent strategy will pick the solution that minimises $\mathcal{O}_2$ (least undershoot).

2.7. Fitting Cycle Termination

After each fine tuning phase, the model is tested to establish if enough scatterers have been identified to allow the fitting cycle to end and the reduction phase to begin. Scatterer amplitudes and phases are fitted using a Constrained Least Squares process (appendix C) in an attempt to match the required radar cross section pattern. The Kolmogorov–Smirnov statistical test (appendix D) is applied to establish the accuracy of the model. This test gives a measure of statistical similarity that is independent of the amount and mean amplitude of the radar cross section data used in the comparisons. If the radar cross section of the model does not fit the required data, an image is generated from the model and this image is subtracted from the original image. This process removes the scatterers that have been identified. The new image is then passed back to the first genetic algorithm to identify a new set of peaks.

3. Model Reduction

3.1. Introduction

The process of model fitting can yield models with large numbers of scatterers. This large amount of model data can create extended simulation times. If it is accepted that a measured or calculated radar cross section will never be a perfect representation of the real target [28, 40], small degradations in data fidelity are acceptable. Therefore, if some of the scatterers in an n-point model are removed, it should be possible to re-adjust the model to give an approximation to the desired radar cross section. As the model is used thousands of times in a typical engagement, *any* reduction in model size is beneficial. This chapter first describes and discusses the different approaches to reducing the number of scatterers in the model. It then proceeds to describe the methods using genetic algorithms and population based incremental learning in detail. The chapter concludes with the results of applying the complete extraction process to 2D and 3D ISAR data.

3.2. Methods

Four different approaches have been investigated.

1. **Exhaustive Search** – For a small model with twenty scatterers, there are $2^{20} = 1048576$ possible combinations to search to find the optimum solution. Finding the optimum is guaranteed, but an exhaustive search of all possible model combinations is often impractical. Increasing the model by just one scatterer doubles the search space. Exhaustive searching of a twenty-point model took two days of processing on a desktop PC. A fifty-point model would take nearly six million years to search. The exhaustive search process is thorough but extremely slow.

2. **Iterative Method** – The approach is to throw out the scatterer that has the least effect at each iteration. The method is very fast but unlikely to choose the best models as reduction progresses; the radar cross section is governed by scatterer interactions [60, Page 38][61] and a small scatterer may have little effect on its own but may be dominant when paired with another similar scatterer. For very small models (≤ 5 scatterers), the iterative method will produce satisfactory results.

3. **Multi-objective Stochastic Optimisation** – Stochastic combinatorial optimisation techniques such as genetic algorithms are used to find the best model for a given number of scatterers. The algorithms are allowed to generate models of all sizes for evaluation but eventually converge on models of the correct

size. Algorithms that track the current best models for all sizes may be used. The method is relatively slow but does not suffer from the exhaustive searches exponential increase in processing for increasing model size.

4. **Encoder Function with Stochastic Optimisation** – A function is used that generates model patterns with the correct number of scatterers. This function is used with a stochastic optimisation technique such as population based incremental learning or genetic algorithms. The process is repeated for each model size. The advantage with this technique is that processing time is not wasted in generating models with an inappropriate number of scatterers.

In all the methods, the scatterer locations in the trial model are kept fixed, and magnitudes and phases are fitted using the constrained least-squares method (appendix C). The constraint value used is the norm of the original model that is being reduced. The radar cross section pattern is fitted over a defined region of optimisation. This region may be a narrow-angle azimuth or elevation sweep, through to data measured from a set of random positions covering 4π steradians.

The cost performance of each trial model is calculated by fitting weightings to the selected scatterers and then generating the N radar echo data samples for the region of optimisation. The mean squared error of the radar echo from the trial model compared with the original required radar echo is calculated by applying equation 3.1, allowing the effects of the reduction to be monitored for the current region of optimisation. Where $r(x)_i$ is the radar echo of model x at point i, g_i is the required radar echo at point i, and N is the number of data samples.

$$\mathcal{O}_e(x) = \frac{1}{N}\sum_{i=1}^{N} |r(x)_i - g_i|^2 \tag{3.1}$$

For large models, the time taken to calculate the new scatterer weightings will be the dominant processing overhead. When the evolutionary algorithms are close to convergence, they tend to repeatedly generate the same small set of solutions at each generation. The speed of the evolutionary algorithm based methods may be improved by storing past scatterer patterns and their associated objective costs. These data may be maintained in a tree structure for fast retrieval (detailed in section E.3).

Table 3.1 demonstrates the processing requirements of the different reduction methods. For very small models (≤ 5 scatterers), the iterative approach will provide near-optimal results in the fastest time. The iterative method will not produce good results for larger models, although it can be used to produce an approximate set of results for seeding the genetic algorithm method. A model of fifteen scatterers or less is best reduced using an exhaustive search as the stochastic techniques are inefficient at small model sizes. For models containing up to two hundred scatterers, multi-objective genetic algorithms that can maintain a set of solutions are the most useful. For models with more than two hundred scatterers, the encoder function method may be used to generate a small subset of reduced models. The use of encoder function alleviates the need for all the model sizes to be evaluated.

	Number of Calculations		
Model Size	Iterative	Exhaustive	GA/PBIL
n	$n(n+1)/2$	2^n	–
5	15	32	$\approx 2,000$
15	120	32768	$\approx 40,000$
50	1275	1.3×10^{15}	$\approx 50,000$

Table 3.1: Comparison of different reduction methods

3.3. Reduction Using a Multi-objective Genetic Algorithm

A n-bit chromosome has been used to define the model structure, where n is the number of scatterers in the model. Each bit corresponds to a scatterer. If a bit is '1', the corresponding scatterer is present in the model, if it is '0', the scatterer is omitted. The number of active scatterers in the model is calculated using equation 3.2, where $\chi(x)_i$ is gene i in the chromosome of individual x. This number is used along with the radar echo error cost in equation 3.1 to generate a Pareto set of results where model size is traded against reproduction accuracy.

$$\mathcal{O}_s(x) = \sum_{i=1}^{n} \chi(x)_i \tag{3.2}$$

The non-dominated ranking methods described in section 2.6.3 are used to maintain the Pareto population. A selective pressure of $s = 1$ is used. Multi-point crossover [48, Page 13] is used to generate two new offspring from each pair of parents with a crossover rate of 0.8. The crossover rate value has been determined empirically to give good performance for a range of model sizes.

The genetic algorithm is terminated after 500 generations and the best overall individual is recorded as the solution. A population of 100 individuals has been used to reduce a 50 point model, therefore giving 50,000 objective calculations to generate a Pareto set of results.

3.4. Reduction Using Encoder Function and PBIL

The encoder function is designed to generate a series of models that all have the same number of scatterers. A recursive algorithm has been developed that allows scatterer patterns to be generated quickly. Using the encoder function, the optimisation algorithm optimises a pattern description rather than the model structure itself. For example, if we take a source model with six scatterers and we wish to find a model with three scatterers, there are twenty model combinations that have exactly three scatterers active. In the genetic algorithm approach described previously, the chromosome would contain six bits and would describe the model structure directly. Six bits give $2^6 = 64$ possible models, of various sizes, to search. With the encoder function, the chromosome would describe a pattern number in the range $1 \leq p \leq 20$. The pattern number would be converted to a unique model description that contained exactly three scatterers. Thus the optimisation surface is one third of the size. The number of possible patterns is given by equation 3.3, where the original model size is denoted by n, and the required number of scatterers by r.

$$\text{No. of possible patterns, } \binom{n}{r} = \frac{n!}{r!(n-r)!} \tag{3.3}$$

The algorithm functions by creating a default model description with all the set bits to the left and then adjusting the position of the rightmost set bit. As the rightmost bit is moved to the right, the remaining pattern to the left can be treated as a smaller sub-string. The number of combinations of the bits in the sub-string may be calculated using equation 3.3. A running total of generated model patterns is maintained. This total is continuously compared to the required pattern number. Each sub-string is evaluated to determine if the required pattern lies within it. If it does, the sub-string is processed, else the number of patterns in the sub-string is added to the running total and the sub-string skipped. If the substring is skipped, the rightmost bit of the current pattern is moved again to the right and the new sub-string to the left of the bit is processed. The patterns are processed recursively until the running total matches the required pattern number.

For example, if we have a six scatterer model ($n = 6$), and we require a model with three scatterers ($r = 3$) that corresponds to pattern number nine ($p = 9$) (where $1 \leq p \leq \binom{n}{r}$), we begin by setting up the default model pattern shown below. The initial model pattern starts with r set bits and $(n-r)$ zeros, all the set bits to the left and the zeros to the right.

$$\underbrace{1\ 1\ 1\ 0\ 0\ 0}_{n=6,r=3}$$

There is only one combination of the two bits to the left of the rightmost set bit. The total number of patterns expressed, T, is one ($T = 1$). The total is less than p so the pattern is skipped and the rightmost set bit is shifted to the right.

$$\underbrace{1\ 1\ 1\ 0\ 0\ 0}_{n=6,r=3} \longrightarrow \underbrace{1\ 1\ 0}_{n=3,r=2}1\ 0\ 0$$

The sub-string created to the left of the rightmost set bit has $\binom{3}{2} = 3$ combinations. The number of combinations would make the running total $T = 4$. The total is less than the required pattern number $p = 9$, so the sub-string is skipped and the rightmost set bit is shifted again. The running total is updated to $T = 4$.

$$1\ 1\ 0\ 1\ 0\ 0 \longrightarrow \underbrace{1\ 1\ 0\ 0}_{n=4,r=2}1\ 0$$

The sub-string created has $\binom{4}{2} = 6$ combinations. Skipping the sub-string would make the running total $T = 10$. As the required pattern number, $p = 9$, is less than $T = 10$, the sub-string should be evaluated. The running total is kept at $T = 4$.

There is only one combination of the one bit to the left of the rightmost set bit of the sub-string. This pattern will make $T = 5$. The total is less than the required pattern $p = 9$ and so the pattern is skipped and the rightmost set bit of the sub-string shifted.

$$1\;1\;0\;0\;\mathbf{1}\;\mathbf{0} \longrightarrow \underbrace{1\;0}_{n=2,r=1}\;\;1\;0\;\mathbf{1}\;\mathbf{0}$$

Here $T = 5$ and there are $\begin{pmatrix} 2 \\ 1 \end{pmatrix} = 2$ combinations. The pattern is not in this interval so the rightmost set bit of the sub-string is moved and T is updated to $T = 7$.

$$1\;0\;1\;0\;\mathbf{1}\;\mathbf{0} \longrightarrow \underbrace{1\;0\;0}_{n=3,r=1}\;1\;\mathbf{1}\;\mathbf{0}$$

Here the current total number of patterns skipped is $T = 7$ and there are $\begin{pmatrix} 3 \\ 1 \end{pmatrix} = 3$ combinations of the sub-string. The required pattern, $p = 9$, lies in this interval. As $r = 1$, ie. only one bit left, the bit may be positioned directly, giving the final pattern for $p = 9$.

$$0\;1\;0\;\mathbf{1}\;\mathbf{1}\;\mathbf{0}$$

Using the encoder function, all the models generated will have the same number of bits.

In the model reduction algorithm, the encoder function is used to convert each trial solution from a genotypic pattern number to a phenotypic model representation. Weightings are fitted to the scatterers in the trial model using the constrained least squares technique. The radar cross section pattern and the cost function, equation 3.1, are then evaluated for the trial solution. Care must be taken to ensure that the calculation of the number of combinations (equation 3.3) can be held to full precision in the internal representation of the computer. The maximum number of combinations occurs at $r = n/2$. The machines precision may limit the maximum model size unless precautions are taken to handle the large numbers properly.

Population Based Incremental Learning (appendix E) may be used to find the optimum pattern number. The algorithm is simple but very powerful and requires much less tuning than genetic algorithms. The algorithm operates by using the best pattern from a population of trial solutions to update a *prototype vector*, from which the next population of trial solutions is generated. The prototype vector has the same number of elements as there are bits in the genotypic representation. Each element represents the probability of generating a '1' at that bit position when the next population is generated. The elements of the prototype vector all begin at a value of 0.5 and are increased or decreased depending on whether the corresponding bit in the best solution was '1' or '0'. The algorithm terminates when all the elements of the prototype vector have converged to a level near zero or one.

Two main parameters are used to control the algorithm. The first, *learning rate*, lies in the range zero to one and determines the final accuracy of the solution. The lower the learning rate, the less likely it is that the algorithm will converge on a local optimum. With high learning rates, the algorithm will be less likely to do a comprehensive search of the optimisation surface. Low learning rates take far more function evaluations before convergence than high learning rates. Typical learning rates lie in the range $0.1 \leq l \leq 0.4$.

The second, and most important parameter, is *population size*. Population size determines the probability that the algorithm will find the global optimum. Increasing the population size will increase the chances of finding the optimum solution, though it will also increase the number of function evaluations required.

Trials for a small model have demonstrated that for a learning rate of 0.25 and a population size of twenty, the optimum combination of 15 scatterers out of a twenty scatterer model can be found with a hit rate of approximately 70% (average of 10 trials). An average of 1276 function evaluations were used. There are $15,504$ possible patterns, therefore the method is approximately 12 times faster than the exhaustive search in this experiment.

3.5. Model Extraction Results

3.5.1. Introduction

To satisfy the objectives detailed in section 1.5.1, two example trials are presented. The first trial was run on measured two-dimensional data and the second on a simulated three-dimensional image. The strategy used for fine-tuning in both trials was to try to find an average solution. The particular solution chosen is the one that minimises the sum of the normalised objective terms. For each objective, the minimum and maximum values are found from those individuals in the Pareto set. These limiting values are used to normalise the objective values to lie in the range zero to one. The normalised values are then summed for each individual. The individual which has the lowest sum is chosen. The non-dominated ranking genetic algorithm was used for both reduction phases. A limit of 300 scatterers was imposed in both trials. In both cases, the radar cross section data used in the fitting process were derived from the same source as the 1D ISAR images described previously in section 2.6.2. These azimuth, elevation (3D only) and frequency traces are concatenated to form a single pattern to match.

3.5.2. Two Dimensional Image

The two-dimensional data were measured from a real target at zero elevation and with the image conditions specified in table 3.2, where range resolution and total range are in metres. The results were obtained with the genetic algorithms operating under the conditions shown in table 3.3.

Range	Res.	Total	Steps	Sweep Range
Slant	0.2986m	76.44m	256	2.5GHz : 3GHz
Cross	0.2963m	18.96m	64	$-5.1°$: $5.4°$

Table 3.2: Conditions for ISAR image generation (2D)

The first fitting stage comprising GA-1 and GA-2 required 38 iterations to locate 174 scatterers with radar cross section K–S significance of $\alpha = 0.9$. Figure 3.1 shows the locations of the scatterers and figure 3.2 shows the original ISAR image. It

Parameter	GA-1	GA-2	GA-3
Maximum Generations	50	100	300
Total No. Individuals	100	20	25
No. of Species	7	1	1
Selective Pressure	0.8	1.0	1.0
Crossover Rate	1.0	1.0	0.8
Mutation Rate	0.3	0.25	$0.5/n$
Non-uniform Mutation	yes	yes	no
Minimum Share dist., ν	3	10	1
Share Shaping, β	0.5	1.0	1.0

Table 3.3: Operating conditions for genetic algorithms (2D)

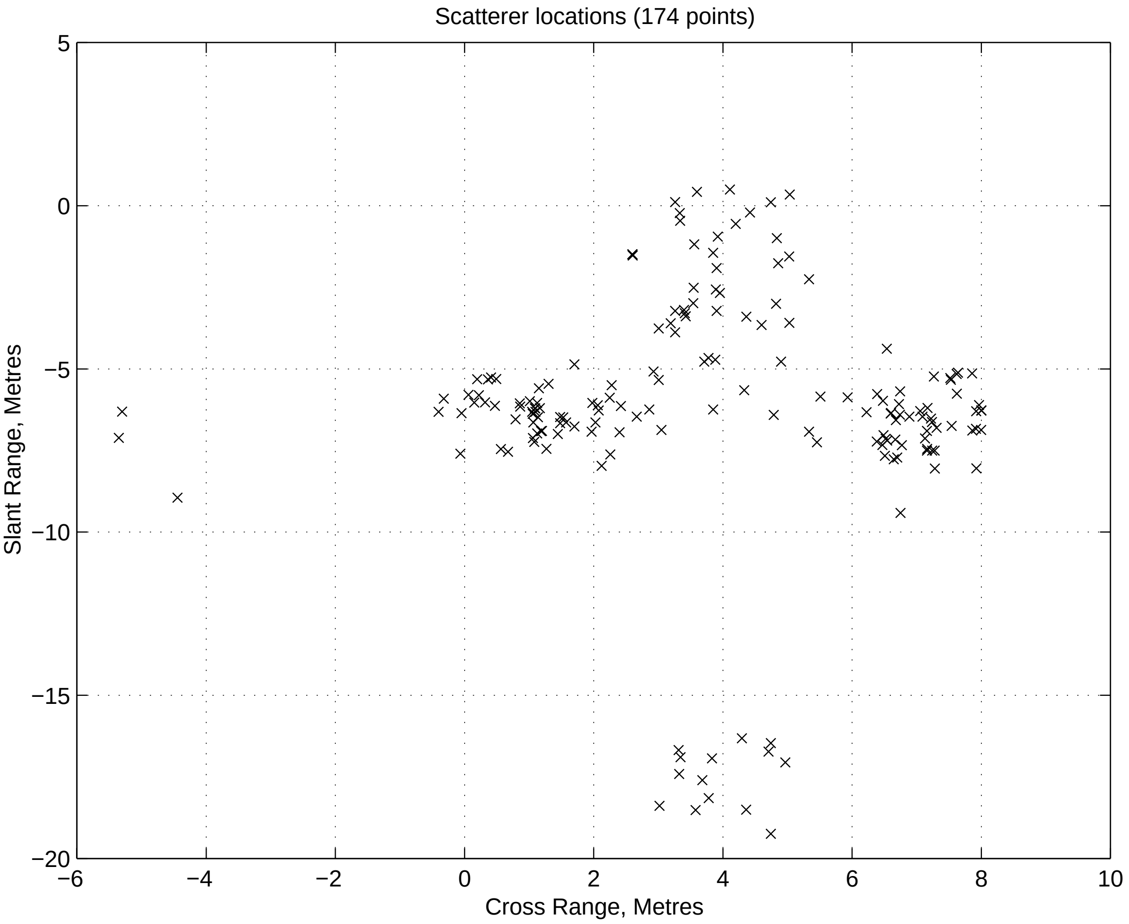


Figure 3.1: Scatterer locations (2D)

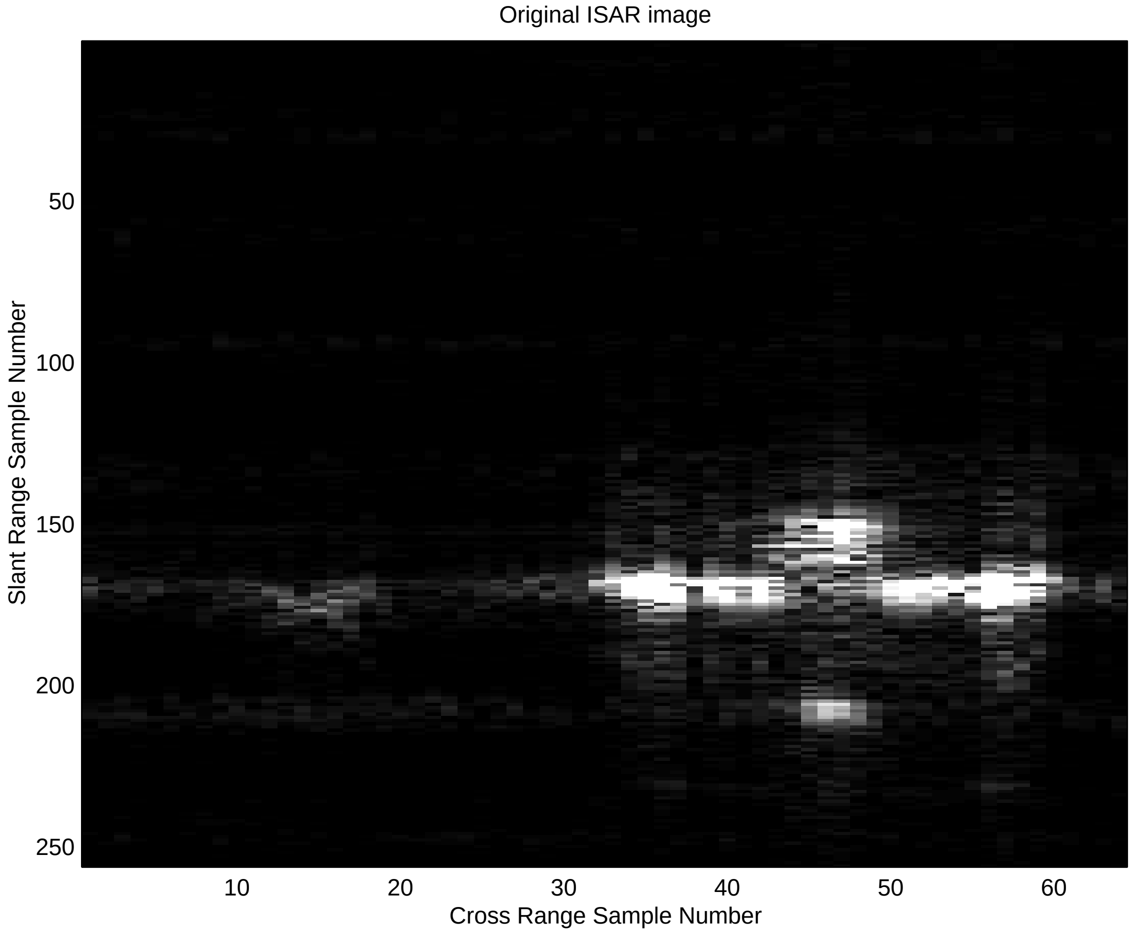


Figure 3.2: Original ISAR image (2D)

is quite apparent that the scatterer locations follow closely the form of the ISAR image. The image is presented in its full form with the area of interest confined to the lower half. The image scale and positioning is determined by the original target configuration and angle and frequency sweeps used to collect the data. As Fourier transforms are used to create the image from the cross section data, cropping the set of data will only alter the resolution of the image. The targets spatial extent and relative location in the image will remain unchanged.

The third genetic algorithm was then used to generate a set of smaller models. Figure 3.3 shows the set produced after 300 generations of GA-3. The cost function used is as defined in equation 3.1. Although the cost function is a good means of quantifying the error between the model and the required radar cross section, it is difficult to gauge the optimum model size to use. Figure 3.4 shows the the results of the K–S statistic when applied to the reduction set. It is clear that the best identified model has 128 scatterers as the $\alpha = 0.9$ significance level is maintained. Allowing GA-3 to run for more generations would eventually provide a smoother K–S curve but may not improve on the model size. The option to terminate the algorithm early is left to the designer.

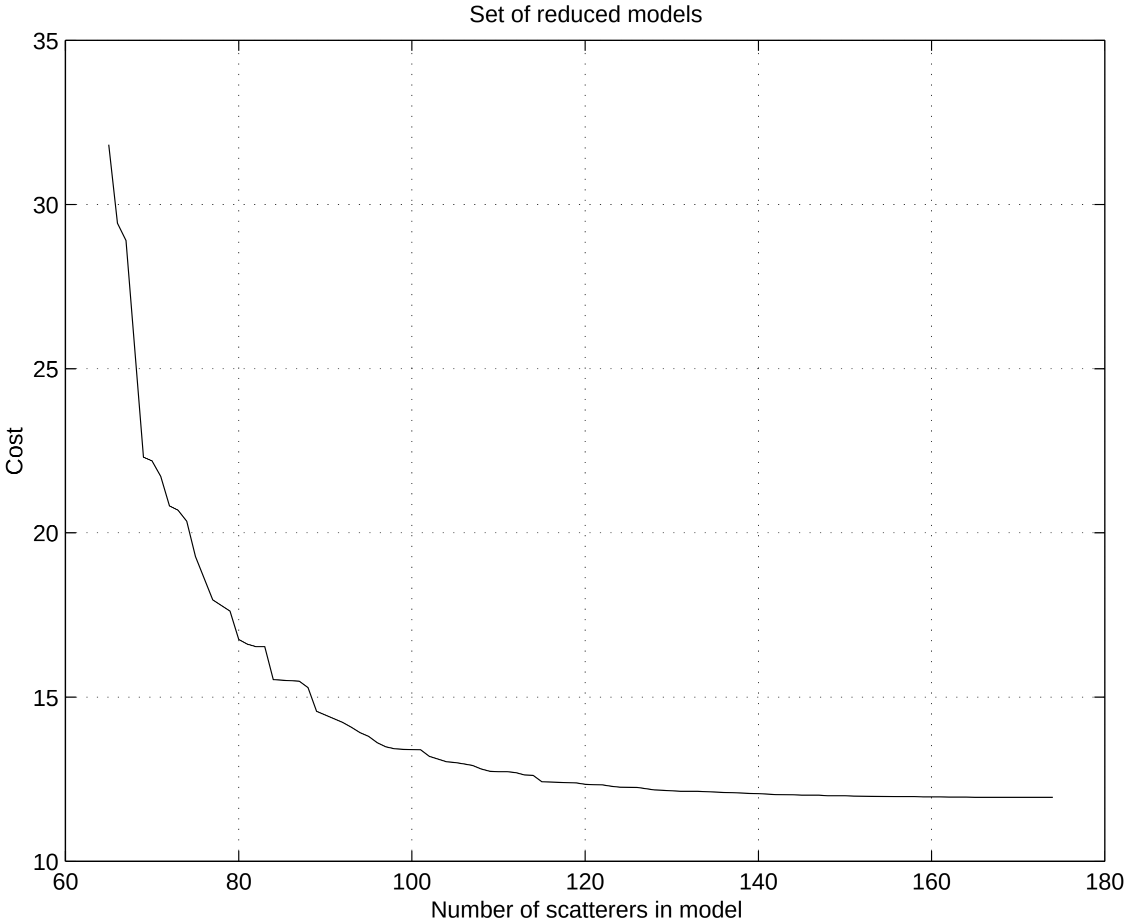


Figure 3.3: Set of reduced models (2D)

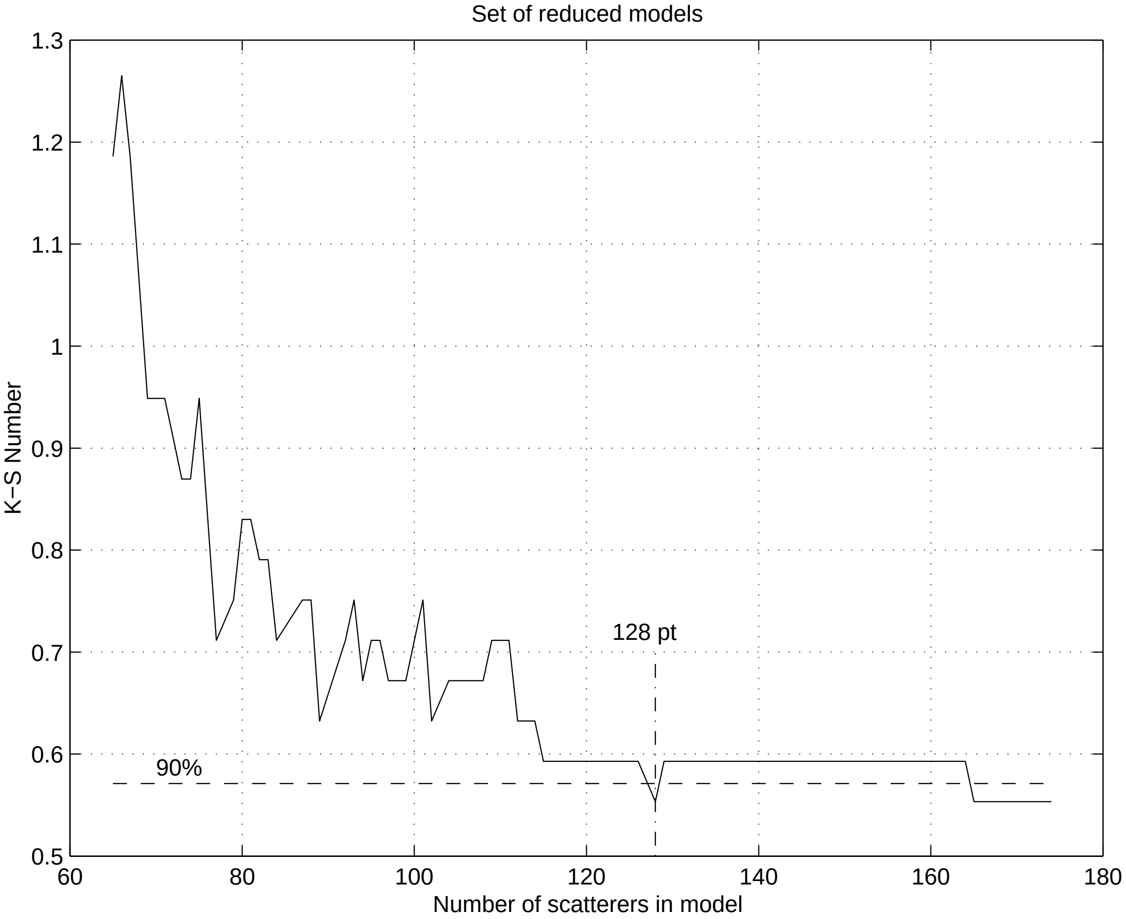


Figure 3.4: K–S significance of reduced models (2D)

Figure 3.5 shows the scatterer locations in the reduced model. The radar cross section of the model (solid) compared to the required cross section (dashed) is shown in figure 3.6. The ISAR image of the model is shown in figure 3.7.

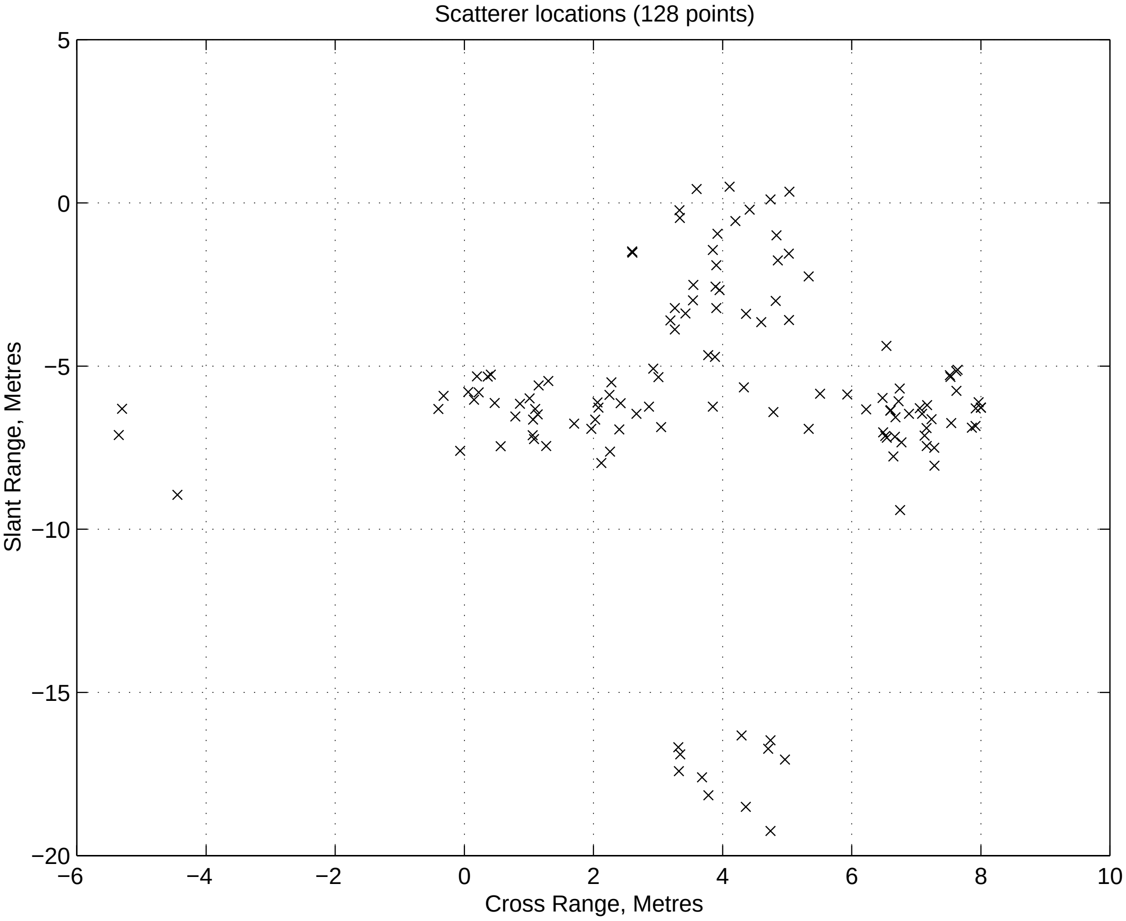


Figure 3.5: Scatterer locations of reduced model (2D)

Thus, despite having a resolution of approximately 30 centimetres, models can be generated that have a reasonable number of scatterers and still approximate the targets radar cross section.

3.5.3. Three-Dimensional Image

The three-dimensional test data were generated from a semi-random model consisting of fifty scatterers based on the scatterer locations of the model identified in section 3.5.2. This model is defined as the fifty-point truth model and is described in appendix B. The image was generated using the conditions specified in table 3.4, where range resolution and total range are in metres. The following results were obtained with the genetic algorithms operating under the conditions shown in table 3.5.

The first fitting cycle required 24 iterations to identify 173 scatterers that gave a fitted radar cross section to a K–S significance level of $\alpha = 0.9$. The third genetic

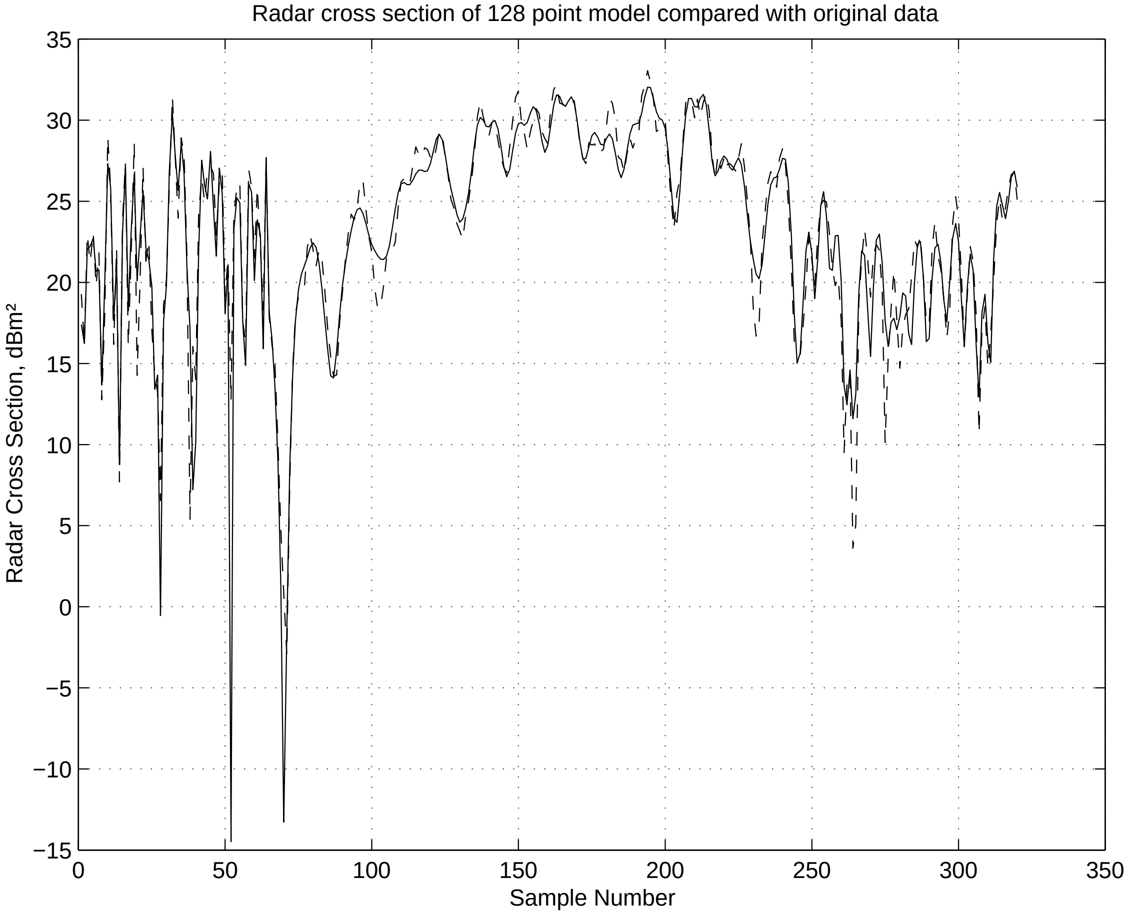


Figure 3.6: Radar cross section of reduced model (2D)

Range	Res.	Total	Steps	Sweep Range
Slant	0.3845	24.61	64	10.8 : 11.2 GHz
Cross, Az	0.3906	25.00	64	$-1°$: $1°$
Cross, El	0.3906	25.00	64	$-1°$: $1°$

Table 3.4: Conditions for ISAR image generation (3D)

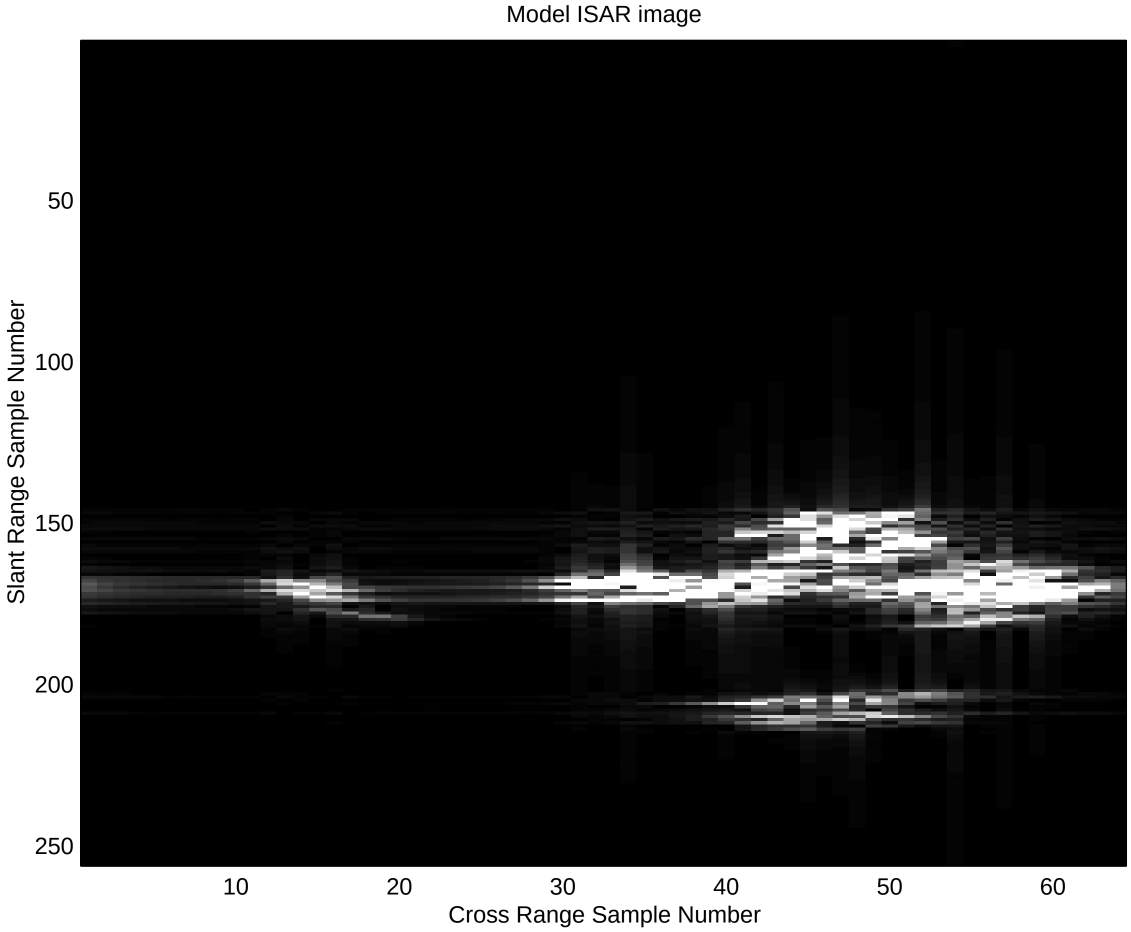


Figure 3.7: ISAR image of reduced model (2D)

Parameter	GA-1	GA-2	GA-3
Maximum Generations	50	500	500
Total No. Individuals	750	100	150
No. of Species	10	1	1
Selective Pressure	0.8	1.0	1.0
Crossover Rate	1.0	1.0	0.8
Mutation Rate	0.3	0.25	$0.5/n$
Non-uniform Mutation	yes	yes	no
Minimum Share dist., ν	3	10	1
Share Shaping, β	0.5	1.0	1.0

Table 3.5: Operating conditions for genetic algorithms (3D)

algorithm was then applied to obtain a set of reduced models. Figure 3.8 shows the set produced after 500 generations of GA-3. Figure 3.9 shows the the results of the K–S statistic when applied to the reduction set. It is clear that the best identified model has 88 scatterers as this model gives an acceptable $\alpha = 0.9$ significance level. Again it is up to the designer to decide how long the reduction algorithm should be run for. The smallest model of suitable fidelity should always be used to minimise the engagement simulation times. Figure 3.10 shows the scatterer locations in the reduced model. The radar cross section of the model (solid) compared to the required cross section (dashed) is shown in Figure 3.11.

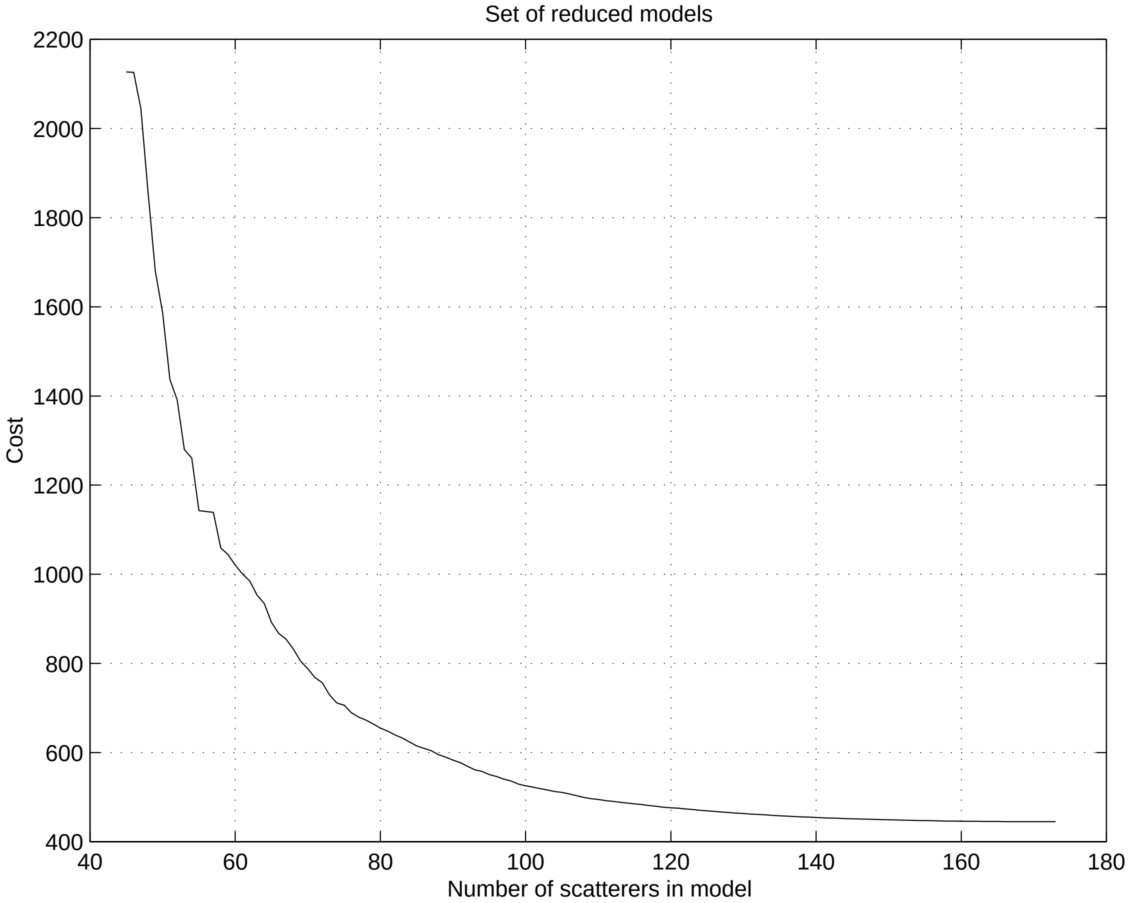


Figure 3.8: Set of reduced models (3D)

3.6. Conclusions

Attempts to solve this model identification problem with one large algorithm have proved fruitless. This lack of success suggests that sometimes the application of many small genetic algorithms may be preferable to using one large and complex one. The multiple algorithm approach is robust and will provide repeatedly a solution to the problem; even though some of the algorithms are forced to converge, thereby

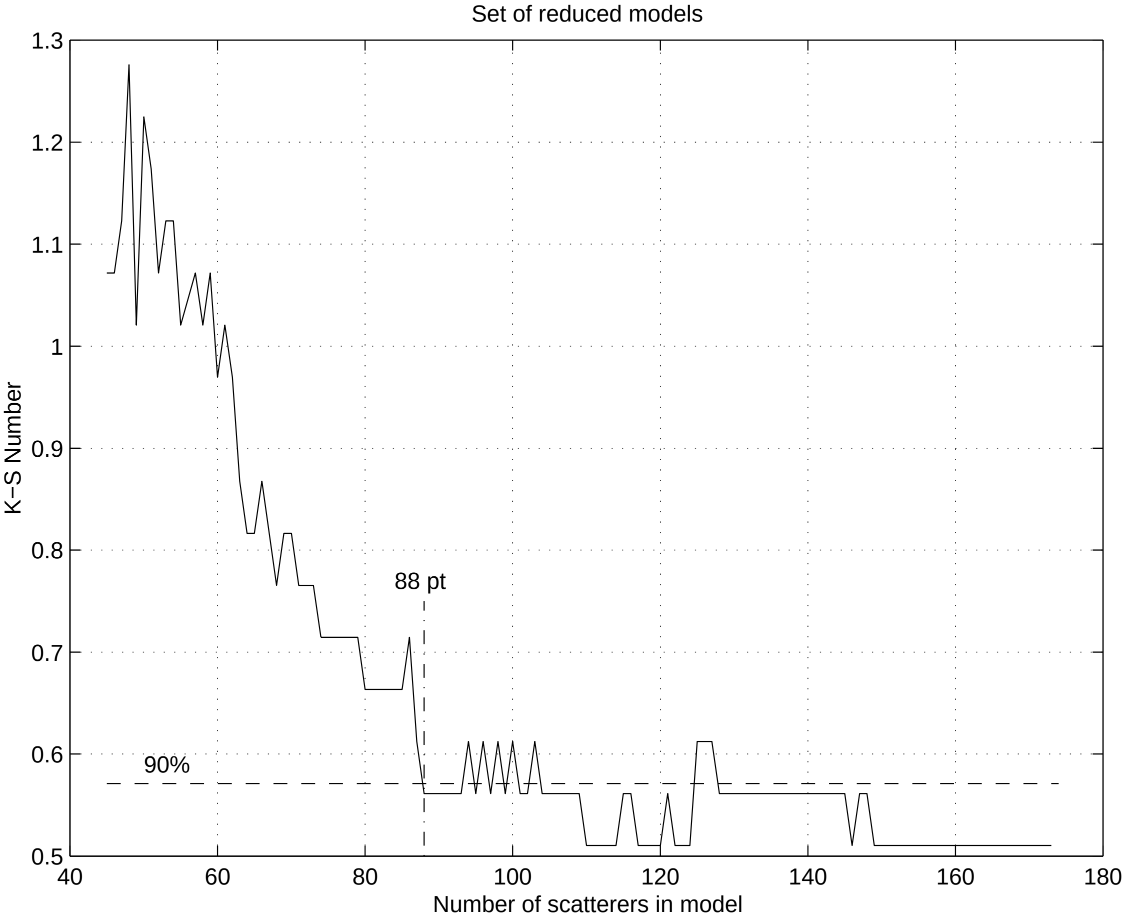


Figure 3.9: K–S significance of reduced models (3D)

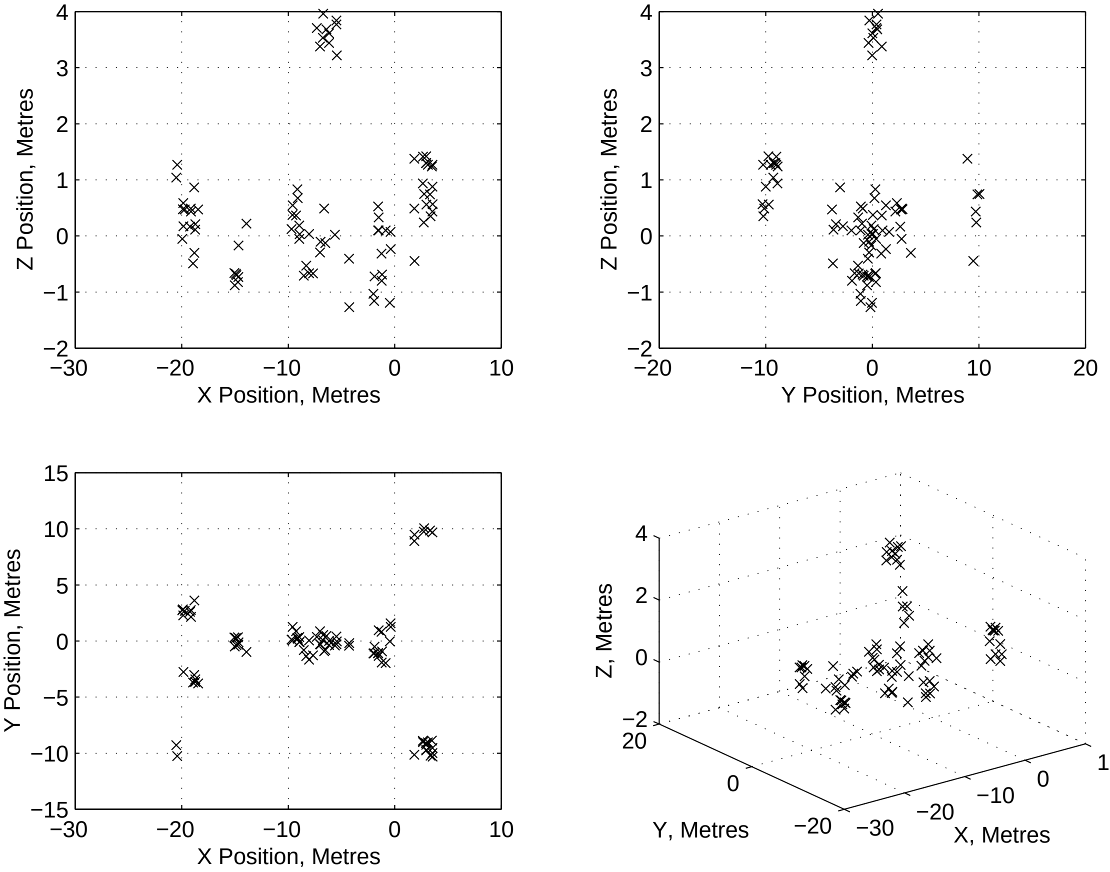


Figure 3.10: Scatterer locations of reduced model (3D)

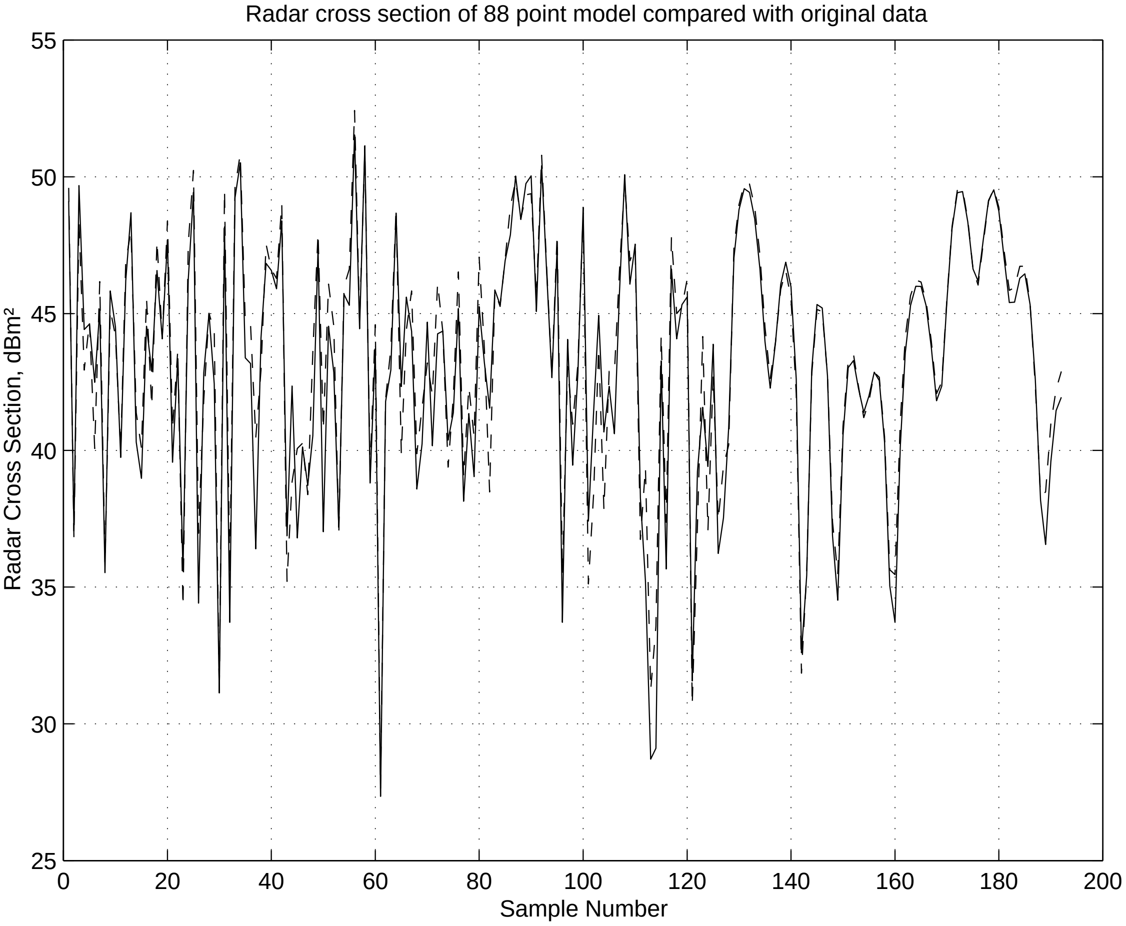


Figure 3.11: Radar cross section of reduced model (3D)

limiting their potential. Forced conversion gives more uniform results with respect to job execution times.

The fitting process is not exceptionally fast, especially on a small system. It does, though, allow results to be obtained that would otherwise not be achievable. When large problems are being tackled, there is no requirement for all the data to be rapidly accessible or even all stored on the same machine. The ability of the algorithms to operate on low and medium resolution data is a significant advantage over existing model extraction techniques.

4. Model Integration

4.1. Introduction

This chapter covers the integration of the point scatterer and seeker models into the engagement scenario. The construction of point scatterer models is described first. Monopulse principles and target glint are introduced and the seeker models defined. The engagement model and missile operation are covered and then miss distance and trial conditions are discussed. Finally, the results of the proving trials are presented.

4.2. Point Scatterer Models

4.2.1. Introduction

The radar cross section of a basic point scatterer model at wavelength λ may be defined as shown in equation 4.1.

$$\rho_T = \left| \sum_{k=1}^{n} \sqrt{\rho_k}\, e^{j\left(\frac{4\pi d_k}{\lambda}\right)} \right|^2 \tag{4.1}$$

The total radar cross section of the target, ρ_T, is defined as the square modulus of the coherent sum of the echos from the n scatterers, each scatterer with its own radar cross section, ρ_k and at a distance d_k from the observation point [15, Page 23]. The sum of the echos is a complex quantity with units of volts. The radar cross section is a scalar with units of square metres.

For high fidelity representation, many scatterer models are created, each one being valid for some small solid angle segment. The models are combined using a Binary Space Partition Tree structure [62, Pages 675–680][63] allowing the correct point scatterer model to be retrieved rapidly for any aspect angle. The structure allows the models generated for small aspect angles and frequency ranges to be combined to cover a larger region of interest. Azimuth and elevation are normally the main decision variables used to generate the tree but models that vary with frequency, range and polarisation for example, can be easily incorporated. Point scatterer models that do not all have the same coverage angle may be incorporated into the tree structure.

4.2.2. Binary Space Partition Trees

Space partition trees are designed to split an object into its component parts in a manner that makes them easily retrievable. If an object is split into N components, on average $\log_2(N)$ tests must be performed to establish the correct component part to be used. For example, if an object has 1000 component parts, an average of just under 10 tests must be performed. If we have the example object space shown in figure 4.1, the components *1*, *2* and *3* can be separated by the two lines a and b. The test procedure will be to determine whether the viewing position lies either to the front or reverse side of the partitioning line under consideration (arrows on figure indicate front of line). Although figure 4.1 shows the partitioning of a two dimensional space, the binary space partition tree technique will extend to n-dimensional space, where the sub-spaces are divided by a structure of n–1 dimensions. Therefore a three dimensional world will be partitioned by two dimensional structures, ie. planes.

The tree is constructed by recursively splitting the model into *sub-spaces* using partition lines defined by the junctions between the models. Components which lie on the front side of the partitioning line are placed in the *left* branch of the tree. The components in the remaining sub-space are placed in the *right* branch. Figure 4.2 shows the complete tree for the components in figure 4.1.

In a typical model, azimuth can be defined as running from $-180°$ to $+180°$ and elevation from $-90°$ to $+90°$. We may then have a partitioning line defined as being, say, 5° in azimuth. If the *front* side of the line is defined as being *increasing* angle, the models may be split about this line forming the first division of the tree. Each branch is then further sub-divided until there is only a single model at each node. During run-time, the first test made will be to check if the missile position relative to the target is greater than 5°. If it is, the left branch of the tree is traversed, else traverse the right side.

Care must be exercised when generating the models to pay particular attention to matching the radar cross section pattern at the edge of the model. If the edges of the models differ wildly where they overlap, discontinuities may appear in the data. Observations have shown that these discontinuities will often appear as small glint spikes in the engagement. ISAR images are unlikely to be affected noticeably by this problem.

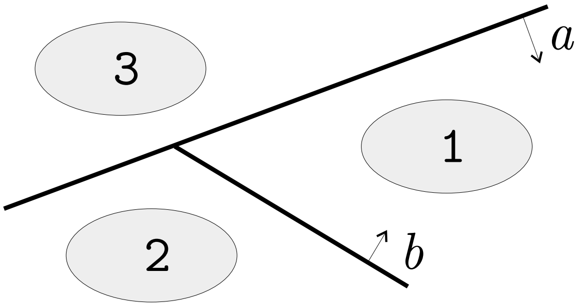


Figure 4.1: Partitioned model

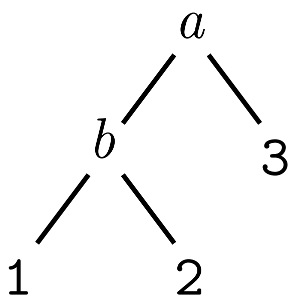


Figure 4.2: *BSP Tree* for figure 4.1

4.3. Monopulse Seeker Models

4.3.1. Introduction

Target tracking is not a simple problem. Many different methods have been devised to locate and track a target through space. In tracking radars, the antenna-beam axis, or bore-sight, is kept aligned with the target direction. If the target deviates from the bore-sight, or the antenna moves away, an error signal is generated whose magnitude is roughly proportional to the amount of deviation, and whose sign indicates the direction of the error. The error signal is then used to drive the antenna back towards the target.

The *Monopulse* technique generates error signals for each active radar burst, or will passively track a signal source. This passive mode of operation renders the monopulse seeker impervious to simple jamming techniques. The missile systems under consideration in this thesis are designed to use monopulse seekers.

Monopulse radars can be broadly classified as either amplitude comparison or phase comparison. Amplitude comparison involves comparing the received signal strength from four simultaneously generated squinted beams. The phase of the signals from each of the beams will ideally be the same. The angular error is formed by dividing the difference between a pair of beams by the sum of all the beams. Figure 4.3 shows a planar representation of two of the beams.

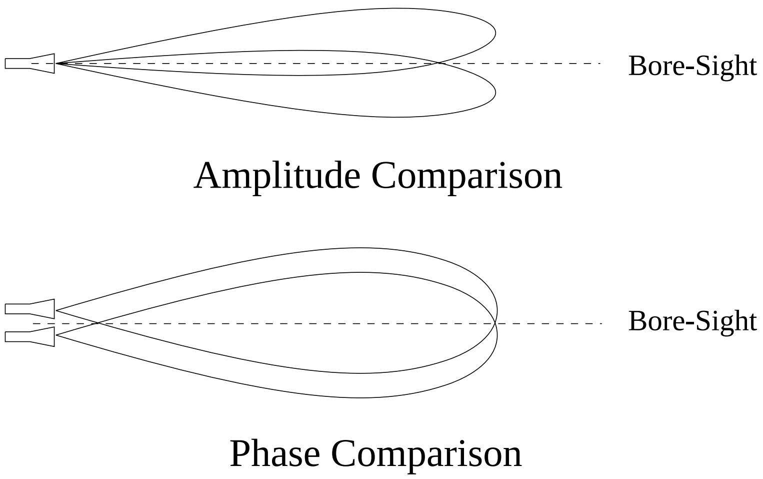
Phase comparison monopulse uses four overlapping beams that each have a slightly different phase centre, therefore each beam ideally receives signals of the same amplitude but differing in phase. The phase difference is then used to generate the bore-sight error.

In practice, for amplitude comparison, if the four antennas are arranged as

3	**1**
4	**2**

and ξ_n indicates the appropriate echo voltage, then equation 4.2 details the horizontal and vertical difference signals.

$$\begin{aligned} d_H &= (\xi_3 + \xi_4) - (\xi_1 + \xi_2) \quad \text{Horizontal Difference} \\ d_V &= (\xi_1 + \xi_3) - (\xi_2 + \xi_4) \quad \text{Vertical Difference} \end{aligned} \tag{4.2}$$



Amplitude Comparison

Phase Comparison

Figure 4.3: Monopulse beam patterns

The underlying principle behind phase-comparison monopulse tracking is based on detecting the incidence angle of the received phase front. If it is assumed that the target is a single-point, isotropic reflector, the normal to the received phase front local to the monopulse receiver will always point at the centre of the target. Differentiating the observed phase front will give the normal to the phase front and thus the target direction can be derived. Figure 4.4 illustrates the location process graphically.

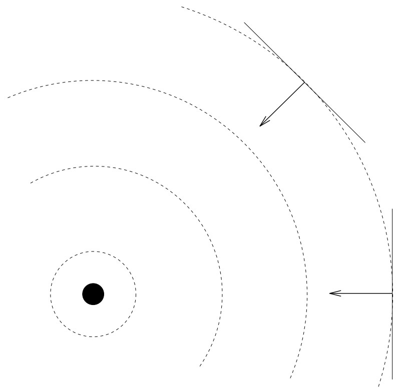


Figure 4.4: Locating target by observing phase front

4.3.2. Glint and Bore-Sight Error

The perfect target is a single isotropic scatterer. If the target is more complex, ie., two or more scattering centres, the phase front may not be spherical and may have discontinuities and variations. These variations are due to the interference of the wavefronts. The *Glint Distance* is the apparent shift of the target position due to the normal to the phase front not being directed towards the target centre. Figure 4.5 shows the glint distance, x, graphically.

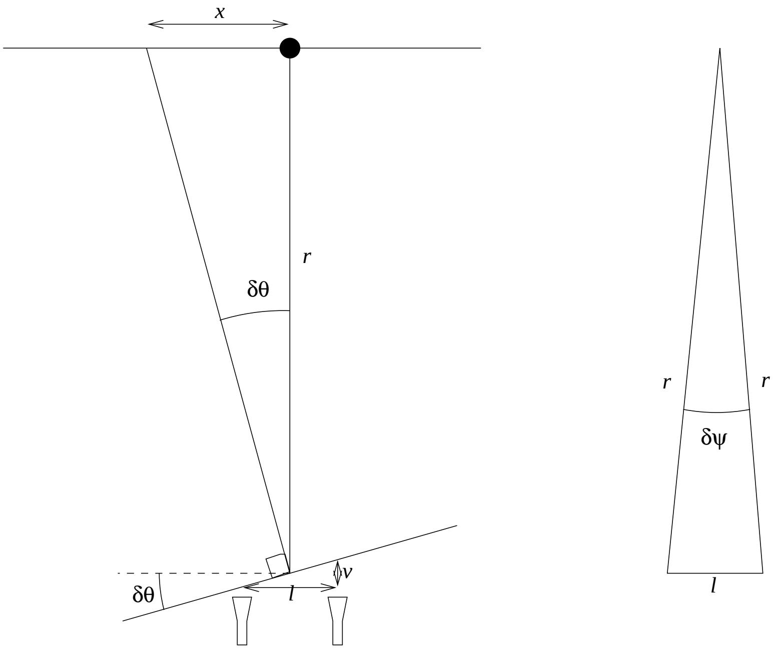


Figure 4.5: Graphical representation of glint distance

If we have a phase comparison monopulse seeker, with the two radar heads separated by a distance l as shown in figure 4.5, the phase difference $\delta\phi$ observed in the radar echo may be related to a physical offset distance v by equation 4.3. One wavelength, λ, corresponds to a phase change of 2π radians.

$$v = \frac{\delta\phi\lambda}{2\pi} \tag{4.3}$$

The apparent target offset, or *Glint Distance*, x, is described by the line-of-sight offset angle $\delta\theta$ and related to the seeker head and engagement geometry as shown by equation 4.4.

$$\delta\theta = \tan^{-1}\left(\frac{v}{l}\right) = \tan^{-1}\left(\frac{x}{r}\right) \tag{4.4}$$

Therefore, the glint distance x may be calculated as in equation 4.5.

$$x = \frac{rv}{l} = \frac{r\delta\phi\lambda}{2\pi l} \tag{4.5}$$

If the observation angle $\delta\psi$ as shown in figure 4.5 is small, then by the small angle assumption that $\sin(\psi) \approx \psi$, the separation, l, may be described as $l = r\delta\psi$ and therefore the glint distance may be defined as being proportional to the rate of change of phase with respect to view angle as shown in equation 4.6.

$$x = \frac{\lambda}{2\pi}\frac{\delta\phi}{\delta\psi} \simeq \frac{\lambda}{2\pi}\frac{d(\phi)}{d\psi} \tag{4.6}$$

In a practical monopulse seeker head, extracting phase difference information is difficult due to system noise and receiver imbalances. In phase comparison monopulse, it is assumed that the radar heads have near identical beam patterns but different phase centres. If we have two received complex signals, p and q, they may be described as in equation 4.7.

$$\begin{aligned} p &= me^{j\phi_p} \\ q &= me^{j\phi_q} \end{aligned} \tag{4.7}$$

where m is the signal magnitude and ϕ_p, ϕ_q are the two signal phases.

If the complex difference of the two signals is evaluated, and then divided by the complex sum, a definition for bore-sight error, b_e, is obtained. This definition is shown in equation 4.8. The result is purely imaginary and is governed solely by the phase difference between channels p and q.

$$\begin{aligned} b_e &= \frac{me^{j\phi_p} - me^{j\phi_q}}{me^{j\phi_p} + me^{j\phi_q}} \\ &= \frac{(\cos(\phi_p) - \cos(\phi_q)) + j(\sin(\phi_p) - \sin(\phi_q))}{(\cos(\phi_p) + \cos(\phi_q)) + j(\sin(\phi_p) + \sin(\phi_q))} \\ &= \frac{2j(\cos(\phi_q)\sin(\phi_p) - \cos(\phi_p)\sin(\phi_q))}{2 + 2(\cos(\phi_p)\cos(\phi_q) + \sin(\phi_p)\sin(\phi_q))} \\ b_e &= \frac{j\sin(\phi_p - \phi_q)}{1 + \cos(\phi_p - \phi_q)} \end{aligned} \tag{4.8}$$

Equation 4.9 shows a similar result derived for amplitude comparison monopulse systems. Here the squinted beams have a common phase centre giving a purely real bore-sight error.

$$b_e = \frac{m_p - m_q}{m_p + m_q} \tag{4.9}$$

It is possible to create the difference and sum signals within the waveguide section of the radar head and they are therefore less sensitive to imbalances and drift. The seeker head is consequently lighter and consumes less power with the passive processing of the signals [64, Pages 66–84]. Figure 4.6 shows bore-sight error signal with respect to angle for a phase comparison monopulse seeker with a 12cm separation between phase centres. The normal region of operation is usually along the near-linear section of ±2°. Figure 4.7 shows the bore-sight error signal with respect to angle for an amplitude comparison monopulse seeker with a beam-width of 3°.

In practical systems, the assumptions that the phase front is linear across the seeker head, that the signals from a phase comparison system have the same amplitude, and that amplitude comparison systems have a common phase centre, do not often hold. The breakdown of the assumptions leads to bore-sight error signals that are complex instead of being purely imaginary or purely real. The previous calculations may be performed for a situation where the received signals are defined as in equation 4.10

$$\begin{aligned} p &= m_p e^{j\phi_p} \\ q &= m_q e^{j\phi_q} \end{aligned} \tag{4.10}$$

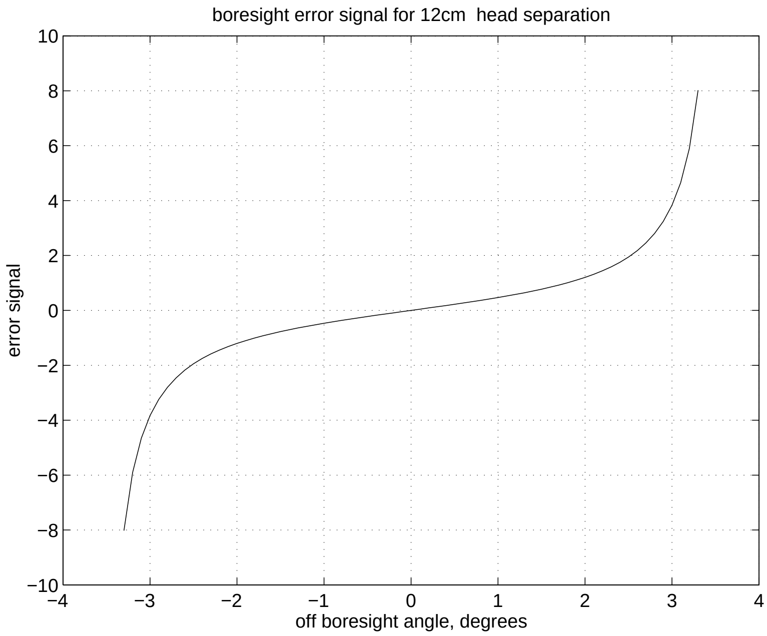


Figure 4.6: Bore-sight error signal with respect to angle (Ph. comp.)

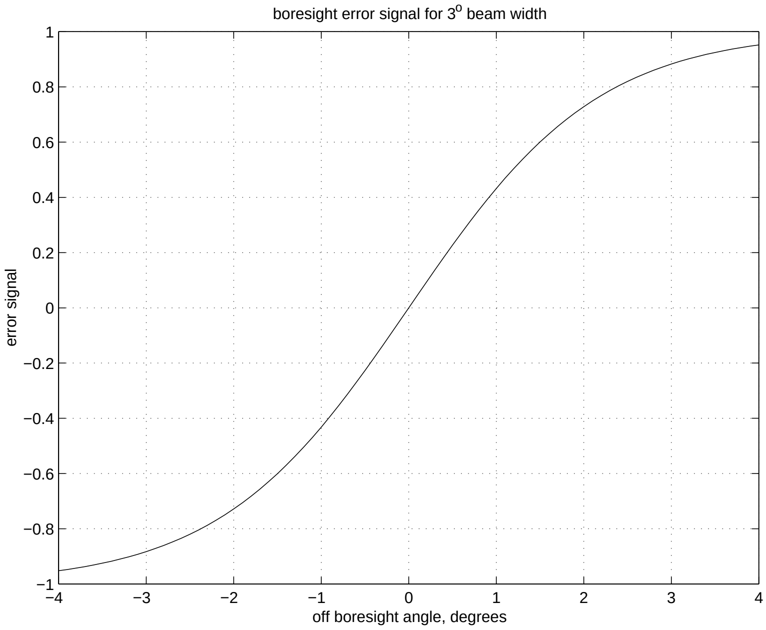


Figure 4.7: Bore-sight error signal with respect to angle (Amp. comp.)

where m_p, m_q are the signal magnitude and ϕ_p, ϕ_q are the two signal phases.

Equation 4.11 shows the actual bore-sight error signal.

$$\frac{d}{s} = \frac{m_p^2 - m_q^2}{m_p^2 + m_q^2 + 2m_p m_q \cos(\phi_p - \phi_q)} + \frac{j2m_p m_q \sin(\phi_p - \phi_q)}{m_p^2 + m_q^2 + 2m_p m_q \cos(\phi_p - \phi_q)} \tag{4.11}$$

Table 4.1 summarises the effects on the monopulse system.

Condition	Real Part	Imag. Part	Comment
$m_p = m_q$	0	$\frac{\sin(\phi_p - \phi_q)}{1+\cos(\phi_p - \phi_q)}$	Ideal Phase Comparison
$\phi_p = \phi_q$	$\frac{m_p - m_q}{m_p + m_q}$	0	Ideal Amplitude Comparison
$\Re\left(\frac{d}{s}\right) = 0$	0	$\frac{2m_p m_q \sin(\phi_p - \phi_q)}{m_p^2+m_q^2+2m_p m_q \cos(\phi_p - \phi_q)}$	Actual Phase Comparison
$\Im\left(\frac{d}{s}\right) = 0$	$\frac{m_p^2 - m_q^2}{m_p^2+m_q^2+2m_p m_q \cos(\phi_p - \phi_q)}$	0	Actual Amplitude Comparison

Table 4.1: Effect of non-ideal signals

Figure 4.8 shows typical phase-comparison bore-sight error plots with respect to both angle and range. The plots are derived from a twenty point model, viewed from broadside at 10GHz. Plot *a* shows the bore-sight error signal when the target rotates by 2° at 1Km. The plot is sampled at 400-points per degree. Plot *b* is the same rotation but at 100 metres. Figure 4.6 shows the relationship between bore-sight error signal and bore-sight error angle. A bore-sight error signal of unity is approximately equivalent to a 2° angle error, therefore, many of the glint spikes represent significant errors to the seeker head. The two plots demonstrate how the glint errors become worse as range decreases.

Plot *c* shows the horizontal bore-sight error signal as range decreases. The plot has a resolution of 1 metre and the target is stationary, with the missile approaching at 90°, ie. broadside. The model used has a twenty metre nose-to-tail span. The first glint spike is noticeable at a range of around 2Km for the horizontal bore-sight error. It is clear that the glint becomes much worse in the last 500 metres. The increased noise levels make guidance very difficult in the engagement end-game. Plot *d* shows the vertical bore-sight error signal as range decreases. The maximum vertical separation of target scatterers though is only four metres, leading to a reduced range over which the glint fluctuates and grows rapidly. These characteristics of glint are detailed further in section 6.4.

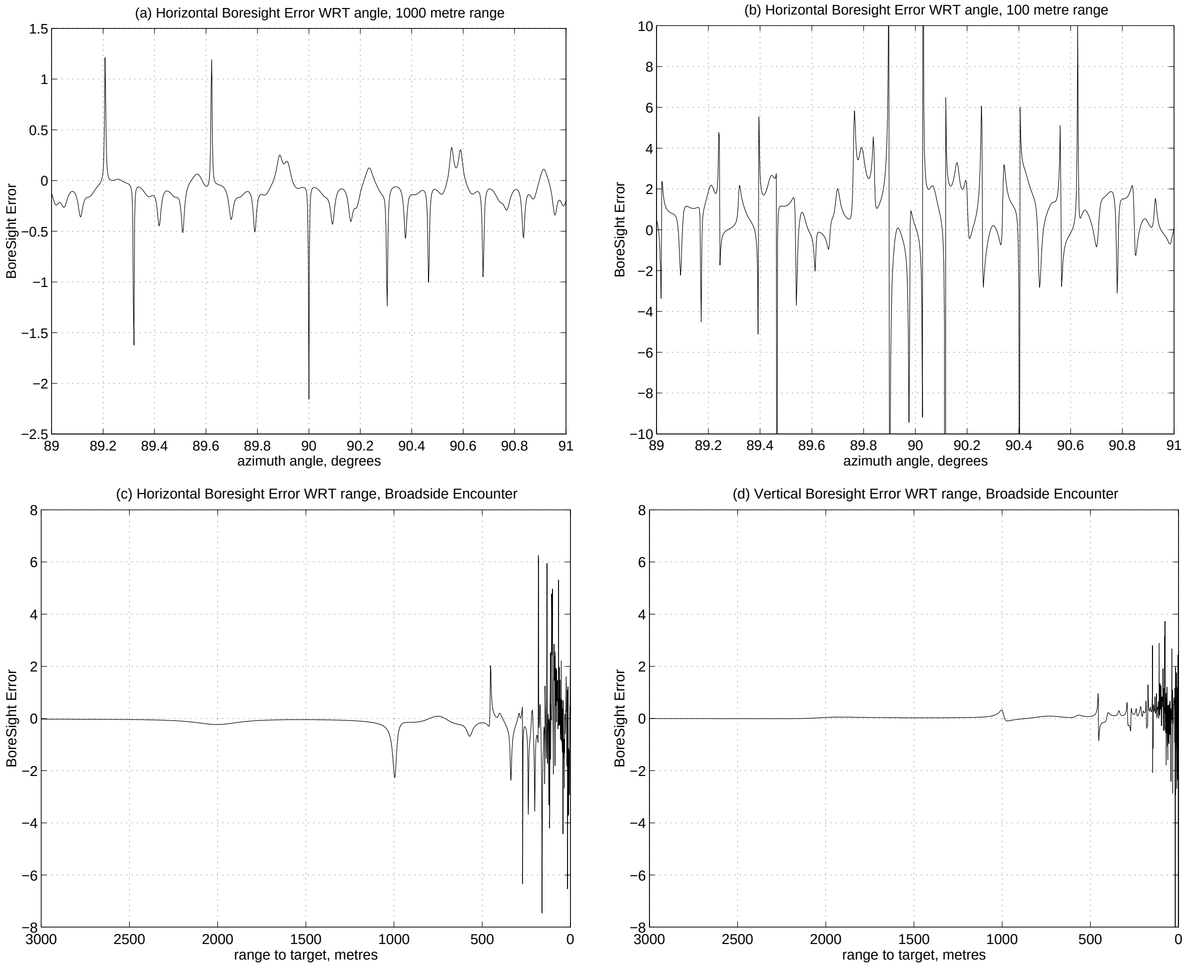


Figure 4.8: Typical bore-sight error signals

4.3.3. Seeker model Orientation

The seeker model takes target and missile positions and orientations along with relevant point scatterer details and produces the horizontal and vertical seeker error signals for the missile model. The target is defined as lying in target axes: the target wings lie in the x–y plane with the fuselage parallel to the x axis and tail upward. A right handed coordinate system is used with all objects being defined with the origin at their centre and increasing x running from tail to nose. The choice of the object centre is arbitrary but the object's geometric centre is often a convenient choice. If the object is viewed head-on, increasing y will be from left to right. Increasing z is defined as upward. The missile location and orientation is transformed from world to target axes, thereby preserving their relative alignment and allowing the radar cross section to be calculated without rotating the position of the target scatterers.

The rotations are performed using *Unit Quaternions* (see Appendix A) as they allow a rotation about an arbitrary axis in space to be defined easily. The target and missile position and orientation are passed as $[\ x \quad y \quad z\ ]$ coordinates and rotation angles in a vector:

$$\left[\begin{array}{c} \overbrace{[\ x \quad y \quad z\ ]\ \psi_T \quad \theta_T \quad \phi_T}^{\text{Target}}\ ,\ \overbrace{[\ x \quad y \quad z\ ]\ \psi_M \quad \theta_M \quad \phi_M}^{\text{Missile}}\ ,\ \overbrace{\psi_A \quad \theta_A}^{\text{Antenna}} \\ \\ \end{array}\right]$$

where ψ , θ and ϕ are as shown in figure 4.9.

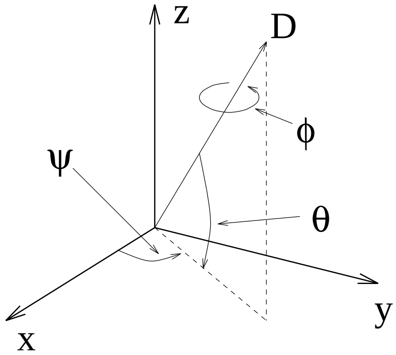


Figure 4.9: Coordinate system

All $[\ x \quad y \quad z\ ]$ coordinates and target and missile orientations (ψ_T , θ_T , ϕ_T and ψ_M , θ_M , ϕ_M) are defined relative to the world axes. Antenna orientation ψ_A and θ_A are defined relative to the missile.

A *Unit Quaternion* is first calculated that describes the orientation of the target relative to the world axes and the position of the missile relative to the target is then derived. Equation 4.12 details the process of converting the Euler angles defining the orientation into a quaternion.

$$
\begin{aligned}
\tilde{q}_a &= \left[\cos\left(\tfrac{\phi_{wt}}{2}\right) \ , \ \overline{[1 \ 0 \ 0]\sin\left(\tfrac{\phi_{wt}}{2}\right)}\right] \\
\tilde{q}_b &= \left[\cos\left(\tfrac{\theta_{wt}}{2}\right) \ , \ \overline{[0 \ 1 \ 0]\sin\left(\tfrac{\theta_{wt}}{2}\right)}\right] \\
\tilde{q}_c &= \left[\cos\left(\tfrac{\psi_{wt}}{2}\right) \ , \ \overline{[0 \ 0 \ 1]\sin\left(\tfrac{\psi_{wt}}{2}\right)}\right] \\
\tilde{q}_t &= \tilde{q}_c\tilde{q}_b\tilde{q}_a \\
\tilde{q}_t &= \tilde{q}_t/|\tilde{q}_t| \\
D_{tm} &= \overline{\tilde{q}_t^*[0 \ , \ \overline{(D_{wm} - D_{wt})}]\tilde{q}_t}
\end{aligned}
\tag{4.12}
$$

Equation 4.13 details the calculations to derive a unit quaternion that describes the transformation from seeker axes to target axes. Unit quaternions are used as they allow composite rotations to be easily normalised, ensuring orthonormality.

$$
\begin{aligned}
\tilde{q}_1 &= \left[\cos\left(\tfrac{\theta ma}{2}\right) \ , \ \overline{[0 \ 1 \ 0]\sin\left(\tfrac{\theta ma}{2}\right)}\right] \\
\tilde{q}_2 &= \left[\cos\left(\tfrac{\psi_{ma}}{2}\right) \ , \ \overline{[0 \ 0 \ 1]\sin\left(\tfrac{\psi_{ma}}{2}\right)}\right] \\
\tilde{q}_3 &= \left[\cos\left(\tfrac{\phi_{wm}}{2}\right) \ , \ \overline{[1 \ 0 \ 0]\sin\left(\tfrac{\phi_{wm}}{2}\right)}\right] \\
\tilde{q}_4 &= \left[\cos\left(\tfrac{\theta wm}{2}\right) \ , \ \overline{[0 \ 1 \ 0]\sin\left(\tfrac{\theta wm}{2}\right)}\right] \\
\tilde{q}_5 &= \left[\cos\left(\tfrac{\psi_{wm}}{2}\right) \ , \ \overline{[0 \ 0 \ 1]\sin\left(\tfrac{\psi_{wm}}{2}\right)}\right] \\
\tilde{q}_m &= \tilde{q}_t^*\tilde{q}_5\tilde{q}_4\tilde{q}_3\tilde{q}_2\tilde{q}_1 \\
\tilde{q}_m &= \tilde{q}_m/|\tilde{q}_m|
\end{aligned}
\tag{4.13}
$$

Where ψ_{wt} (yaw of target orientation), θ_{wt} (pitch of target orientation), ϕ_{wt} (target roll), ψ_{wm} (yaw of missile orientation), θ_{wm} (pitch of missile orientation), and ϕ_{wm} (missile roll) are all with respect to world axes. Yaw of seeker antenna (ψ_{ma}) and pitch of seeker antenna (θ_{ma}) are with respect to missile axes.

The radar cross section model consists of a binary space partition tree of point scatterer clusters (section 4.2), each cluster is a valid model for a specific solid angle segment. The appropriate cluster is retrieved by traversing the tree using the current azimuth and elevation of the missile with respect to the target. The azimuth (ψ), elevation (θ) and range (r) of the missile to the target may be found from equation 4.14.

$$
\begin{array}{rcl}
r & = & |D_{tm}| \\
[x\;y\;z] & = & \frac{D_{tm}}{r} \\
\psi & = & \tan^{-1}\left(\frac{y}{x}\right) \\
\theta & = & \sin^{-1}(z)
\end{array}
\tag{4.14}
$$

4.3.4. Phase Comparison Seeker

Once an appropriate point scatterer model has been retrieved, the seeker head output is calculated using equation 4.15.

$$
\begin{array}{rcll}
\theta_e & = & \Im\left(\frac{d_z}{s}\right) & \text{vertical error} \\
\psi_e & = & \Im\left(\frac{d_y}{s}\right) & \text{horizontal error}
\end{array}
\tag{4.15}
$$

Equation 4.16 details how the sum and difference signals in equation 4.15 are calculated from the simulated received echo from the target (equation 4.18).

$$
\begin{array}{rcl}
d_z & = & (\xi_2 + \xi_4) - (\xi_1 + \xi_3) \\
d_y & = & (\xi_1 + \xi_2) - (\xi_3 + \xi_4) \\
s & = & \xi_1 + \xi_2 + \xi_3 + \xi_4
\end{array}
\tag{4.16}
$$

Received echos are calculated for four independent phase centres located at the simulated missile position. The positions of the phase centres are calculated as shown in equation 4.17.

$$
\begin{array}{rcl}
F_1 & = & D_{tm} + \overline{\tilde{q}_m[0,\ \overline{[0 \quad x \quad x]}\ ]\tilde{q}_m^*} \\
F_2 & = & D_{tm} + \overline{\tilde{q}_m[0,\ \overline{[0 \quad x \quad -x]}\ ]\tilde{q}_m^*} \\
F_3 & = & D_{tm} + \overline{\tilde{q}_m[0,\ \overline{[0 \quad -x \quad x]}\ ]\tilde{q}_m^*} \\
F_4 & = & D_{tm} + \overline{\tilde{q}_m[0,\ \overline{[0 \quad -x \quad -x]}\ ]\tilde{q}_m^*}
\end{array}
\tag{4.17}
$$

Where x is the Phase centre offset (6cm typ.) and $F_{1\ldots4}$ denote the seeker head phase centres arranged $\begin{array}{|c|c|}\hline 3 & 1 \\ \hline 4 & 2 \\ \hline \end{array}$ when viewed from the front.

Equation 4.18 details the calculation of the returned echo. The calculations are performed with respect to the target axes set. Noise is added to the signal to give a maximum detection range of R_N metres against a $1m^2$ target. The noise level is set to give a unity signal to noise ratio of the received echo at the range R_N.

$$
\xi_n = \sum_{p=1}^{m} Z\left(\left(\frac{T_p - F_n}{|T_p - F_n|}\cdot\hat{\vec{d}}\right)^h\right)\frac{a_p\sqrt{\mathcal{S}}}{(\sqrt{4\pi}d_{n_p})^{\rho_p}}\,e^{j\left(\frac{2\pi\rho_p d_{n_p}}{\lambda}+\phi_p\right)}\Bigg|_{n=1\ldots4} + N(0,k^2)e^{jU(0,\pi)}
\tag{4.18}
$$

Where

$$Z = \begin{cases} x & x > 0 \\ 0 & x \leq 0 \end{cases}$$

and

$$h = \frac{\log\left(\sqrt{1/2}\right)}{\log\left(\cos\left(\frac{\pi\alpha}{360}\right)\right)}$$

and RMS noise level is

$$k = \frac{\sqrt{\mathcal{S}}}{4\pi R_N^2}$$

$\hat{\vec{d}}$ is the unit direction vector of the seeker head, $\hat{\vec{d}} = \overline{\tilde{q}_m[0,\ \overline{[1\quad 0\quad 0]}\ ]\tilde{q}_m^*}$
T_p are the coordinates of target scatterer p (Target Axes)
$\mathcal{S}$ is the radar source power
α is the half power beam-width of the seeker head (10° typ.)
$N(\mu, \sigma^2)$ denotes a Gaussian noise source with mean μ and standard deviation σ.
$U(a, b)$ denotes a random number generator that creates values uniformly distributed in the range $a \leq x \leq b$.

4.3.5. Amplitude Comparison Seeker

For amplitude comparison, the seeker head output is calculated using equation 4.19.

$$\begin{array}{rcll} \theta_e & = & \Re\left(\frac{d_z}{s}\right) & \text{vertical error} \\ \psi_e & = & \Re\left(\frac{d_y}{s}\right) & \text{horizontal error} \end{array} \tag{4.19}$$

Equation 4.20 details how the sum and difference signals in equation 4.19 are calculated from the simulated received echo from the target (equation 4.22).

$$\begin{array}{rcl} d_z & = & -((\xi_2 + \xi_4) - (\xi_1 + \xi_3)) \\ d_y & = & -((\xi_1 + \xi_2) - (\xi_3 + \xi_4)) \\ s & = & \xi_1 + \xi_2 + \xi_3 + \xi_4 \end{array} \tag{4.20}$$

Four received echos are calculated using four squint vectors that share a common phase centre located at the simulated missile position (D_{tm}). The squint vectors are calculated as shown in equation 4.21. The arrangement aligns the bore-sight along the half power regions of the beams.

$$\begin{array}{rcl} a & = & \cos\left(\dfrac{\pi\alpha}{360}\right) \\ b & = & \sin\left(\dfrac{\pi\alpha}{360}\right)/\sqrt{2} \\ \hat{\vec{V}}_1 & = & \overline{\tilde{q}_m[0,\ \overline{[a\quad b\quad b]}\ ]\tilde{q}_m^*} \\ \hat{\vec{V}}_2 & = & \overline{\tilde{q}_m[0,\ \overline{[a\quad b\quad -b]}\ ]\tilde{q}_m^*} \end{array}$$

$$\begin{aligned}\hat{\vec{V}}_3 &= \overline{\tilde{q}_m[0,\ \overline{[a\quad -b\quad b]}\ ]\tilde{q}_m^*} \\ \hat{\vec{V}}_4 &= \overline{\tilde{q}_m[0,\ \overline{[a\quad -b\quad -b]}\ ]\tilde{q}_m^*} \qquad (4.21)\end{aligned}$$

Where α is the half power beam-width and $\hat{\vec{V}}_{1...4}$ denote the unit seeker head squint vectors arranged $\begin{array}{|c|c|}\hline 3 & 1 \\ \hline 4 & 2 \\ \hline \end{array}$ when viewed from the front.

A beam-width of 3° is used in this thesis for the amplitude comparison seeker. The seeker head will then produce very similar bore-sight error signal levels to a phase comparison seeker with a 10° beam-width and 6cm head offset (over range ±1° of off-bore-sight angle).

Equation 4.22 details the calculation of the returned echo. The calculations are performed with respect to the target axes set.

$$\xi_n = \sum_{p=1}^{m} Z\left(\left(\frac{T_p - D_{tm}}{|T_p - D_{tm}|}\cdot \hat{\vec{V}}_n\right)^h\right) \frac{a_p\sqrt{\mathcal{S}}}{(\sqrt{4\pi}d_{n_p})^{\rho_p}}\, e^{j\left(\frac{2\pi\rho_p d_{n_p}}{\lambda}+\phi_p\right)}\Bigg|_{n=1...4} + N(0,k^2)e^{jU(0,\pi)} \qquad (4.22)$$

Each scatterer in the model is assumed to be independent of all other scatterers for simplicity. The seeker head functions return bore-sight error data and range to the missile model.

4.4. Engagement Model

4.4.1. Introduction

The engagement model consists of a homing guidance missile and a synthetic target and allows controlled missile–target engagements to be simulated. The engagements occur in a 3-dimensional world against targets that have a synthetic radar cross section which fluctuates realistically with respect to angle, range and frequency. The fluctuating radar cross section causes errors in the homing missile. These errors are correlated to the target's motion and usually prevents the missile exactly hitting the target. The extent of this *miss distance* gives an indication of the missile performance against the target.

4.4.2. Model Structure

The engagement model structure is shown in figure 4.10. Target and missile positions and orientations are used with the radar cross section data to generate heading error information for the missile model. The missile then manoeuvres in response and so alters its position and orientation, changing its viewpoint of the target and therefore the radar cross section that it sees. This process continues until the missile to target range begins to increase and the engagement is terminated. The software is written in a combination of MATLAB and 'C' with a modular design to allow different model configurations to be integrated into the engagement scenario. Details of the software may be found in [65]. Details of the homing missile may be found in [66].

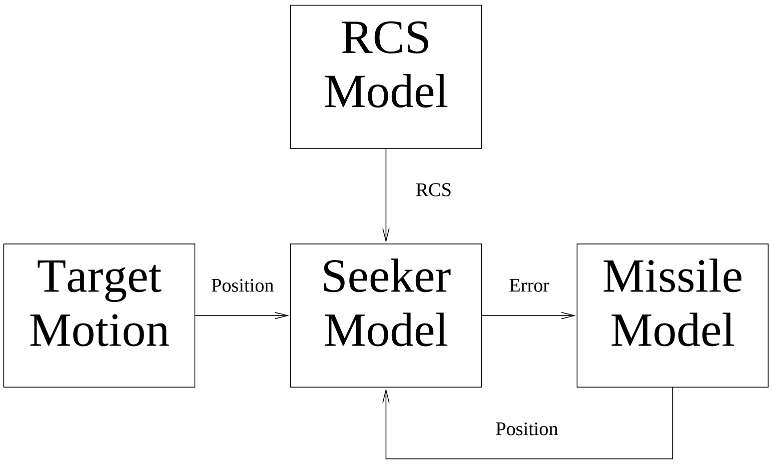


Figure 4.10: Block diagram of engagement model

4.4.3. Guidance Strategy

The missile uses a *proportional navigation* strategy to home onto the target. Before launch, a collision triangle is formed to determine the required missile heading. The target direction and velocity are used to triangulate an impact point. The missile is launched to fly toward the estimated impact point, rather than directly at the target. The seeker antenna is pointed towards the target. The missile lateral acceleration is controlled to keep the antenna angle, with respect to the missile body, constant. Figure 4.11 shows the geometry graphically. If the antenna angle θ is constant, the line-of-sight rate is zero. A zero rate implies that the missile and target are on a collision course.

The antenna is steered in response to the bore-sight error signals from the radar in the seeker head. Fluctuations in the target radar cross section will cause glint noise (chapter 6) and therefore upset the alignment of the seeker antenna. The the mis-alignment creates a noisy line-of-sight rate and therefore a *miss distance*.

4.4.4. Miss Distance Calculations

There are two definitions for miss distance. These are:

1. **Trial Miss Distance** – The minimum achieved distance between the missile seeker head centre and the target centre in each engagement.

2. **Missile Miss Distance** – The distance within which 39% of all *trial* miss distances fall. This corresponds to a half standard deviation radius from the target.

The *trial* miss distance is calculated at the end of each engagement. The *missile* miss distance is calculated from a batch of *trial* miss distances. The *missile* miss distance gives a good indication of overall missile performance. Real missiles have a proximity fuse that detonates the warhead when the missile is within an optimum range. The warhead in most missiles is effective up to three or four metres from

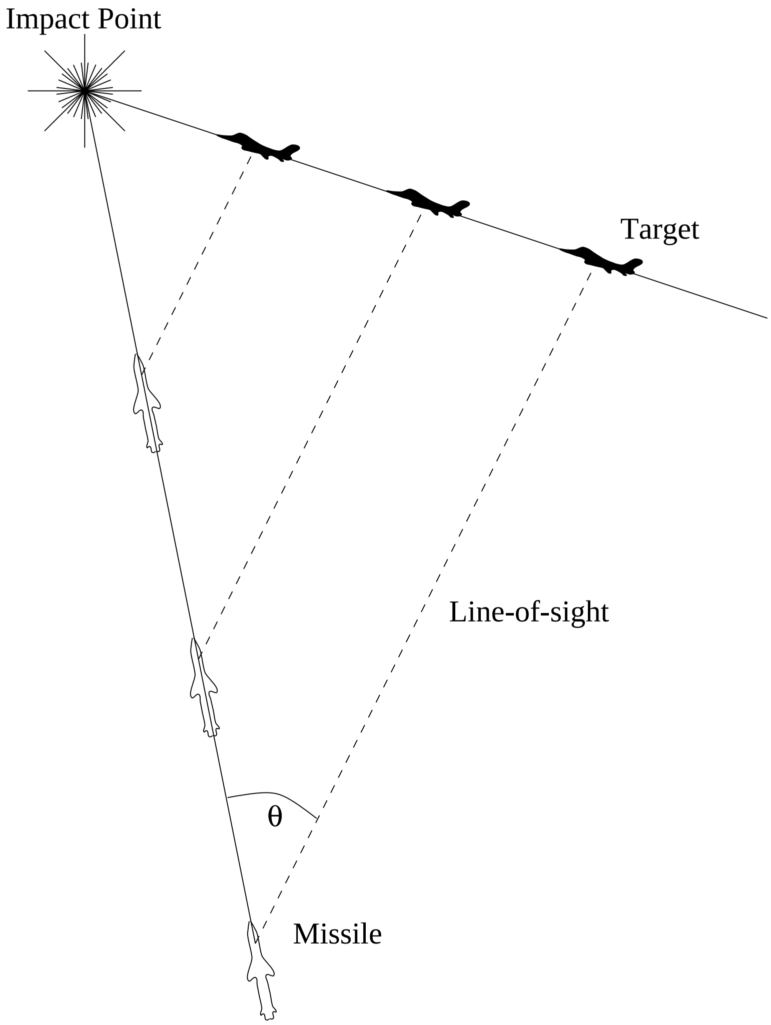


Figure 4.11: Engagement geometry

the target hull. In the target models used in this thesis, the closest scatterer to the target centre is 1.6 metres away. Therefore, any miss-distances of 1.6 metres or less may be classed as a direct impact. With an effective range of 4 metres for the warhead, distances of 5.6 metres or less are a hit, ignoring the chances of hitting the aircraft's wings.

As the engagement is conducted in discrete time, the smallest range measurement in the engagement may not be an accurate representation of the miss distance. Figure 4.12 shows the missile and target tracks around the point of interception, with the indicated locations being separated by the seeker sample time ($\approx$ 2 milli-seconds). As the missile has a speed of around 600 m/sec, in a 2 milli-second frame it travels around 1.2 metres. Thus the assumption may be made that over the sample interval in which the interception occurs, the target and missile are travelling in straight lines.

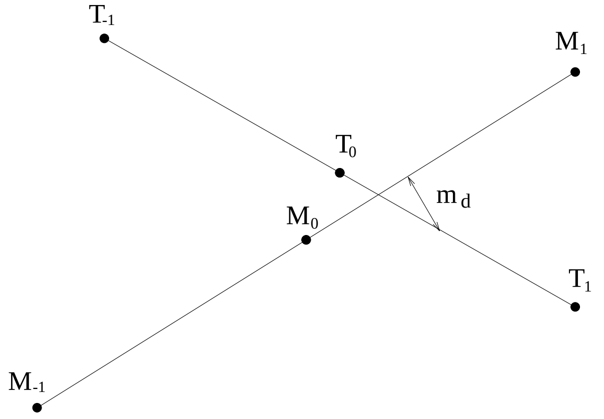


Figure 4.12: Missile and target tracks at interception

If the target and missile tracks are described as the vector representation of the lines shown in equation 4.23, then the trial miss distance may be defined as shown in equation 4.24, with $0 \leq e \leq 1$.

$$\begin{aligned} T_I &= T_0 + e(T_1 - T_0) \\ M_I &= M_0 + e(M_1 - M_0) \end{aligned} \tag{4.23}$$

$$\begin{aligned} \gamma &= |M_I - T_I| \;=\; |\vec{g}| \;=\; \sqrt{\vec{g}\cdot\vec{g}} \\ m_d &= \min(\gamma) \end{aligned} \tag{4.24}$$

The minimum miss distance occurs when

$$\frac{d(\gamma)}{de} = 0 \tag{4.25}$$

Therefore if

$$\vec{g} = M_I - T_I = M_0 - T_0 + e(M_1 - M_0 - T_1 + T_0)$$

Then

$$\frac{d(\gamma)}{de} = (\vec{g} \cdot \vec{g})^{-1/2} \left(\vec{g} \cdot \frac{d(\vec{g})}{de} \right) = \frac{\vec{g} \cdot (M_1 - M_0 - T_1 + T_0)}{\sqrt{\vec{g} \cdot \vec{g}}} \tag{4.26}$$

As $\sqrt{\vec{g} \cdot \vec{g}}$ on the denominator of equation 4.26 is γ, for equation 4.25 to be satisfied,

$$\begin{aligned} \vec{g} \cdot (M_1 - M_0 - T_1 + T_0) &= 0 \\ (M_0 - T_0 + e(M_1 - M_0 - T_1 + T_0)) \cdot (M_1 - M_0 - T_1 + T_0) &= 0 \end{aligned} \tag{4.27}$$

If we select $a = M_0 - T_0$ and $b = (M_1 - M_0 - T_1 + T_0)$ then equation 4.27 becomes

$$\begin{aligned} (a + eb) \cdot b &= 0 \\ a \cdot b + e(b \cdot b) &= 0 \\ e &= -\frac{a \cdot b}{b \cdot b} \end{aligned} \tag{4.28}$$

and

$$\gamma = |a + eb| \tag{4.29}$$

Finally, inserting equation 4.28 into equation 4.29 gives

$$m_d = \left| a - \frac{a \cdot b}{b \cdot b} b \right| \tag{4.30}$$

If the interception point is near to M_1 and T_1, the engagement may sometimes have one too many sample frames to interception. The stop criteria of the range increasing can be fooled into not terminating at the correct sample instant. In this scenario, a negative value for e will be obtained when equation 4.28 is evaluated. In this situation, the correct result may be obtained for the miss distance by repeating the calculations with $T_1 = T_0$, $M_1 = M_0$, $T_0 = T_{-1}$ and $M_0 = M_{-1}$.

4.5. Engagement Model Proving Trials

There are two main disturbance mechanisms that occur during the engagement, antenna noise and target glint. Antenna noise is Gaussian noise that is generated within the antenna and processing circuits and remains at a constant level throughout the engagement. When the bore-sight error signal is generated, the antenna noise gets divided by the received echo strength and so appears to decrease with reducing range. Target glint noise is caused by fluctuations in the radar cross section inter-reacting with the mono-pulse processing in the seeker head. Target glint errors are related to the relative rate of rotation between the target and missile, and also to range. Short ranges are most affected by target glint, whereas long ranges are most affected by antenna noise. Details of the noise characteristics are covered in chapter 6.

Two initial experiments with 1000 trials in each were performed. The ideal single-point model was used to give no glint effects, and each experiment used a different seeker head, but with no simulated antenna noise. Every trial had a zero miss distance. Therefore, with no antenna noise and no glint, the missile miss distance is zero. These trials give the benchmark results for testing a hypothesis.

The hypothesis that the miss-distance distribution is dependent on antenna noise, target radar cross section, and seeker head type must be tested. Two objectives are defined in section 1.5.2. The first requires the effects of the two seeker heads on an ideal target to be evaluated. The second requires that the effects of a complex target on the two seeker types is investigated. The models used are detailed in appendix B. Four experiments are required to be performed. They are detailed in table 4.2.

	Target Type	
Seeker Type	Single Point	Fifty Point
Phase Comparison	Experiment *a*	Experiment *c*
Amplitude Comparison	Experiment *b*	Experiment *d*

Table 4.2: Model proving experiments

In each of the experiments, 1000 missile trials are performed. A set of 1000 launch positions are generated at random. The launch positions are generated using equations 4.31 and 4.32. These equations generate points with a uniform distribution across the surface of a 5Km radius sphere. In the equations, x_1 and x_2 are uniformly distributed random variables with $-1 \leq x \leq 1$, ψ is azimuth in radians, θ is elevation in radians, and $[x\,y\,z]$ is a position in metres referenced to world axes. Each experiment uses the same set of launch positions for consistency.

$$\begin{aligned} \psi &= \pi x_1 \\ \theta &= \sin^{-1}(x_2) \end{aligned} \tag{4.31}$$

$$\begin{aligned} x &= 5000\cos(\psi)\cos(\theta) \\ y &= 5000\sin(\psi)\cos(\theta) \\ z &= 5000\sin(\theta) \end{aligned} \tag{4.32}$$

4.5.1. Experiment *a*

The engagements were run with the parameters in table 4.3.

Figure 4.13 shows the cumulative probability distribution for the 1000 miss distances. The missile miss distance equates to the distance within which 39% of the shots fall and is marked on the graph. All the trials have a miss distance of less than 1.6 metres and are therefore all direct hits. The small spread of miss distances are due to the noise on the bore-sight error signals from the antenna. This result shows that the miss distance is affected by simulated antenna noise, but at a low level.

4.5.2. Experiment *b*

The engagements were run with the parameters in table 4.4.

Figure 4.14 shows the cumulative probability distribution for the 1000 miss distances. The spread of miss distances are again small and due to the antenna noise

Parameter	Value
Source power, $\mathcal{S}$	100W
Target detection range, R_N	10Km
Seeker type	Phase Comparison
Frequency	10 Ghz
Beam-width, α	10°
Head Offset, x	0.06m
Model	Single Point

Table 4.3: Experiment *a* configuration

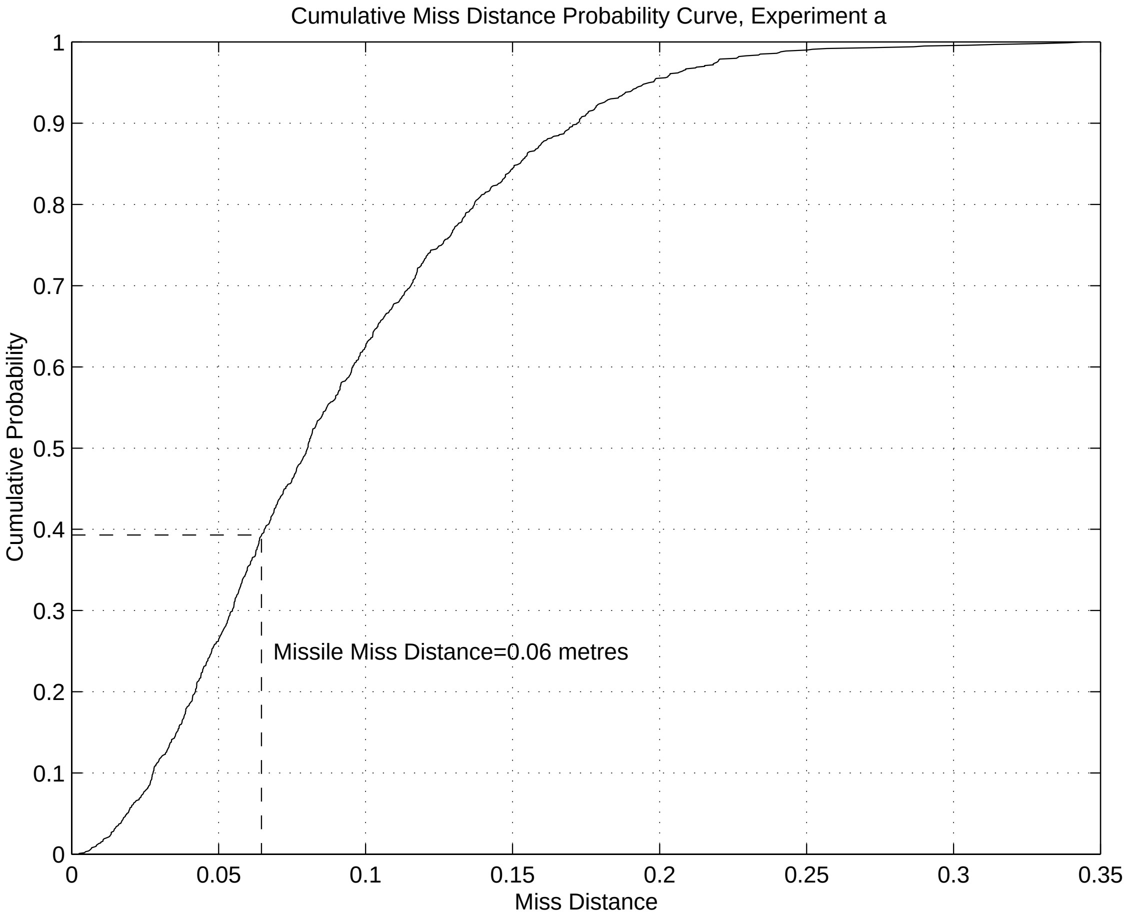


Figure 4.13: Cumulative miss distance probability – Experiment *a*

Parameter	Value
Source power, $\mathcal{S}$	100W
Target detection range, R_N	10Km
Seeker type	Amplitude Comparison
Frequency	10 Ghz
Beam-width, α	3°
Model	Single Point

Table 4.4: Experiment *b* configuration

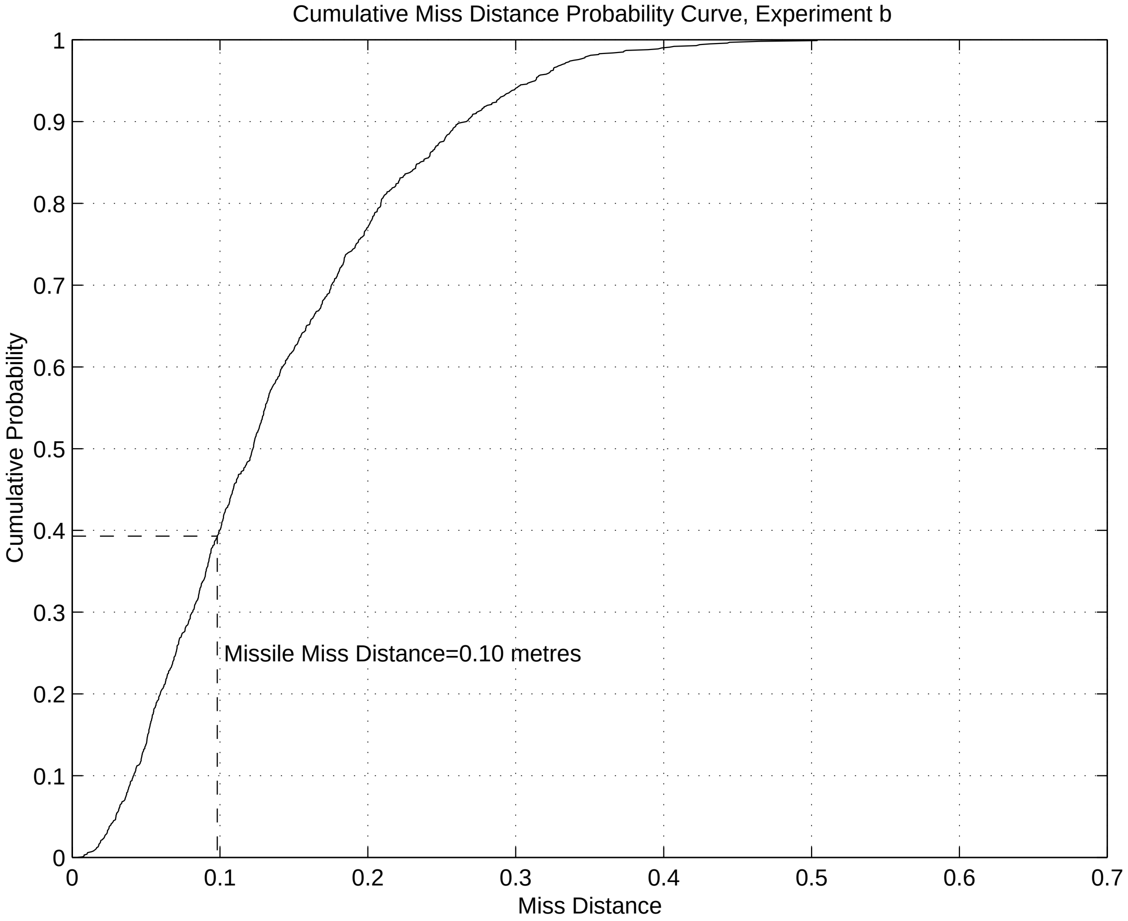


Figure 4.14: Cumulative miss distance probability – Experiment *b*

affecting the bore-sight error signals. The performance of the missile is slightly worse with the amplitude comparison seeker head. This degradation of performance is probably due to the missile's Kalman Filter configuration being non-optimal. The filter configuration was tuned for the phase comparison seeker head. The deviation from the optimal filter configuration is small, but significant enough to show a difference in these results. All the trials may be classified as direct hits.

4.5.3. Experiment *c*

The engagements were run with the parameters in table 4.5.

Parameter	Value
Source power, $\mathcal{S}$	100W
Target detection range, R_N	10Km
Seeker type	Phase Comparison
Frequency	10 Ghz
Beam-width, α	10°
Head Offset, x	0.06m
Model	Fifty Point

Table 4.5: Experiment *c* configuration

Figure 4.15 shows the cumulative probability distribution for the 1000 miss distances. Only 83% of the trials have a miss distance of less than 5.6 metres and are therefore within warhead range. The missile miss distance is calculated as 2.26 metres. This distance is quite small but it is apparent from the distribution that there are a small number of very large miss distances of up to five times the target length. This result shows that the miss distance is affected by target glint. The effects on the missile can be drastic in a small number of cases.

4.5.4. Experiment *d*

The engagements were run with the parameters in table 4.6.

Parameter	Value
Source power, $\mathcal{S}$	100W
Target detection range, R_N	10Km
Seeker type	Amplitude Comparison
Frequency	10 Ghz
Beam-width, α	3°
Model	Fifty Point

Table 4.6: Experiment *d* configuration

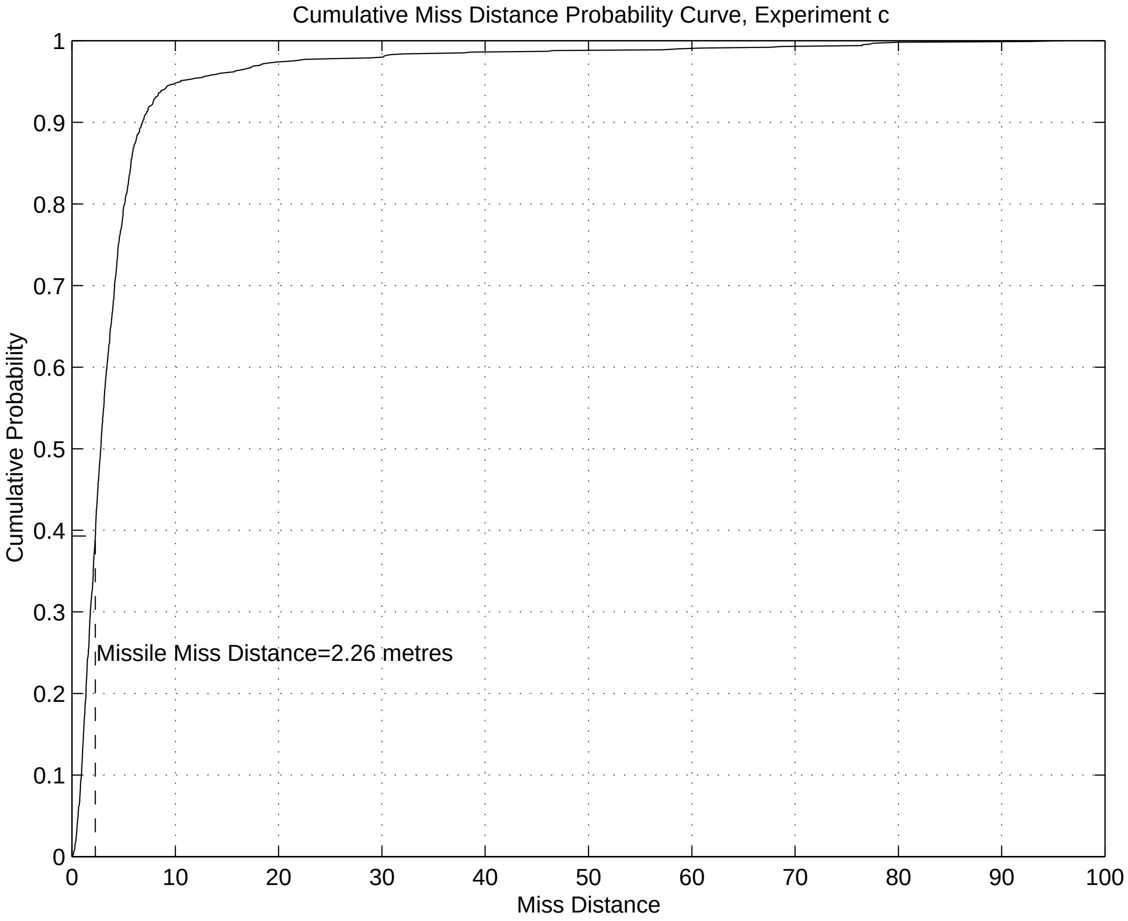


Figure 4.15: Cumulative miss distance probability – Experiment *c*

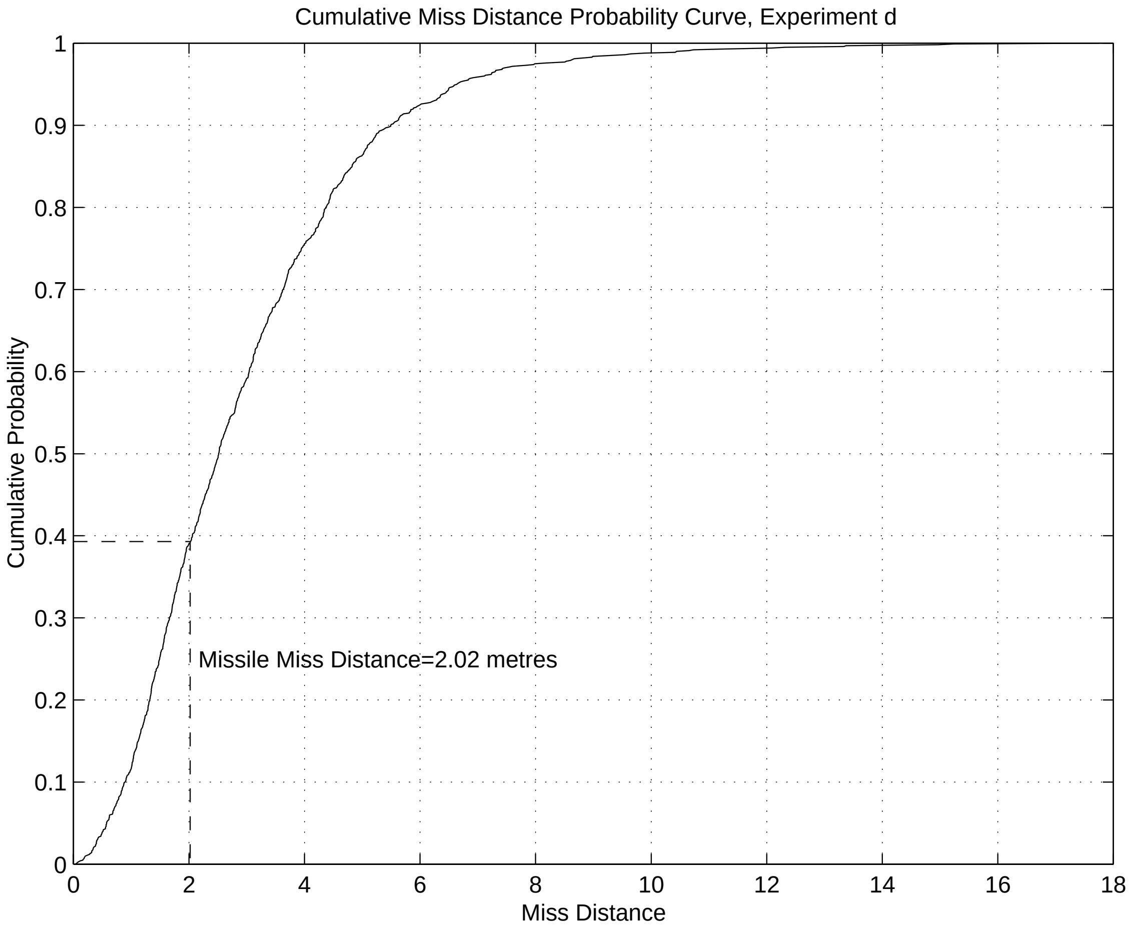


Figure 4.16: Cumulative miss distance probability – Experiment *d*

Figure 4.16 shows the cumulative probability distribution for the 1000 miss distances. Only 91% of the trials have a miss distance of less than 5.6 metres and are therefore within warhead range. The missile miss distance is calculated as 2.02 metres. This distance is smaller than for the phase comparison seeker and the distribution shows that there are far fewer large miss distances. The largest distance is around the same size as the target length. This result shows that the amplitude comparison seeker is not affected as badly by target glint. The improved performance is probably due to the bore-sight error function being better behaved at extreme error angles than for a phase comparison seeker (see figures 4.6 and4.7).

4.5.5. Conclusions

The results of the experiments are summarised in table 4.7.

Experiment	Missile Miss Distance, metres	Hit rate,% (< 5.6m)
a	0.06	100
b	0.10	100
c	2.26	83
d	2.02	91

Table 4.7: Summary of experimental results

Miss distance is influenced by thermal noise. At long ranges where the reflected energy from the target is small, the effects of thermal noise may be large. At close range, the levels of noise are likely to be small compared to the echo strength. Target glint has a major influence on the miss distances. The phase comparison seeker head is effected more than the amplitude comparison seeker with the possibility of a small percentage of extreme misses. These results suggest that the fidelity trials should be performed with the amplitude comparison seeker. This seeker appears to have a more compact distribution compared to the long-tailed distribution of the phase comparison seeker. The chances of the statistical comparisons being upset by extreme miss distances will be reduced. Target manoeuvre detection trials though should be conducted with the phase comparison seeker as the target glint appears more severe and should be easier to detect and monitor.

5. Target Fidelity Analysis

5.1. Introduction

By its very nature, the effect of glint on a missile is difficult to quantify deterministically. The engagement model provides a means of evaluating the effect of a target on a simulated missile. To minimise simulation times, we would like to use a model with the fewest scatterers possible. If the number of scatterers in a point scatterer model is reduced, only an approximation of the original radar cross section pattern can be re-created. The error between the original pattern and the pattern from the reduced model will increase as fewer scatterers are used. The error is a gauge of the fidelity of the reduced model to the original target.

This chapter is concerned with establishing the effects on the missile performance of reducing the model size and therefore the target fidelity. First, the approach to the experiments is described. Then the method of model reduction is detailed. The trial results are presented and finally conclusions are drawn about the effects of fidelity on missile performance.

5.2. Problem Definition and Existing Work

In order to minimise simulation times, models with the smallest number of scatterers must be used. We may often require the model to be an accurate representation of a specific target. The number of scatterers in the model must be traded against the accuracy of representation, or fidelity. If scatterers are removed from a high-fidelity model, the magnitude and phase characteristics of the remaining scatterers can be adjusted to approximate the original target. The adjustment is achieved using the constrained least squares fitting process detailed in appendix D.

Quantifying fidelity is difficult. The cost function defined in equation 3.1 is useful for indicating the difference between two radar cross section patterns. Unfortunately, different targets will have different characteristics and their cost functions are not related. Statistical measures, such as the Kolmogorov–Smirnov and Mann–Whitney statistical tests provide a convenient yardstick, if somewhat noisy, that are independent of radar cross section magnitude and distribution. Previous work on radar cross section fidelity has shown that it is often possible to reduce the number of scatterers while not altering significantly the radar cross section pattern [61].

The effects of a reduced fidelity model on missile miss distance are unknown. The problem is to establish how far models can be reduced before the trial miss distance distribution is significantly different to that seen with the original target.

5.3. Experimental Approach

For any statistical analysis to be reliable, the target must be analysed stochastically over 4π steradians. To achieve this coverage, synthetic missiles were fired towards a stationary target from random start points on a sphere. To establish the effects of fidelity on the missile, repeated trials of the missile have been performed against the reduced targets. The following hypothesis was tested for each of the reduced models:

Null Hypothesis H_0: The reduced model causes similar missile miss distances to the reference model.

Alternative Hypothesis H_1: The models cause different miss distances.

For each model, one thousand trials were performed, each from a randomly chosen start point. A range of five kilometres was chosen for all of the engagements. This range is sufficiently far enough away to minimise the effects of range on the glint errors, but maintains a short execution time for the simulation. One thousand trials were performed in order to obtain good statistical results but with an acceptable processing overhead. Every trial for every model was begun from a randomly chosen position. Two sets of reference data were generated, one for the fifty-point model and one for the one hundred point model. Each set of data contains the miss distance results of 1000 trials. Two statistical tests were performed on the experimental data for each of the model types and sizes. Each test compares the appropriate set of reference data to the set of trial data being investigated.

The Kolmogorov–Smirnov statistical test compares the distribution of a pair of data sets. The test is applied at $\alpha = 0.05$ to reject H_0. The Mann–Whitney test is a test of means and is also applied at the $\alpha = 0.05$ level. Both of the tests are distribution free and are described in appendix D.

5.4. Reduced Model Generation

The reduced fidelity models for the trials were generated from the fifty-point and one-hundred-point models described in appendix B. The iterative reduction method (section 3.2) was used to generate quickly a set of reduced models. This set was used to seed the genetic algorithm described in chapter 3.

Full 4π steradian coverage is required for the missile trials. The reference target echo pattern was generated from $10,000$ samples taken at random positions generated using equation 4.31. This equation gives uniform coverage of the target. The coefficients of the scatterers in the reduced models were fitted using the constrained least squares process. The $10,000$ azimuth and elevation data used to generate the reference echo pattern are also used in the fitting of each reduced model. The $10,000$ points used can only give an approximation to the true 4π steradian radar cross section. Using more points would improve the accuracy but at a rapidly increasing computational cost. The tree structure described in section E.3 may be used to reduce the genetic algorithm processing burden by storing previously generated chromosomes and objectives to prevent unnecessary repeat calculations.

The final Pareto set of results may not be the true optimum though for the set of $10,000$ trial points chosen. The genetic algorithm was stopped after 300 generations. Allowing the algorithm to run for longer may have improved the set of reduced models but at increasing computational cost. If a different set of points are chosen over which to match the target echo pattern, a different set of models may be generated.

Figure 5.1 shows the results of reducing the 50-point model. The upper plot shows the Kolmogorov-Smirnov significance levels generated by comparing the radar cross section of the reduced models to the original model. Here, error significance is plotted against model size with the 5% limit drawn on the graph. The sections of the curve that are below the line correspond to H_0 being rejected, ie. the radar cross sections are not the same. Applying the hypothesis test at $\alpha = 0.05$, the smallest model where the hypothesis H_0 is accepted has 36 scatterers out of 50. The lower plot shows the mean squared error between each reduced model's radar cross section pattern and the reference pattern (equation 3.1).

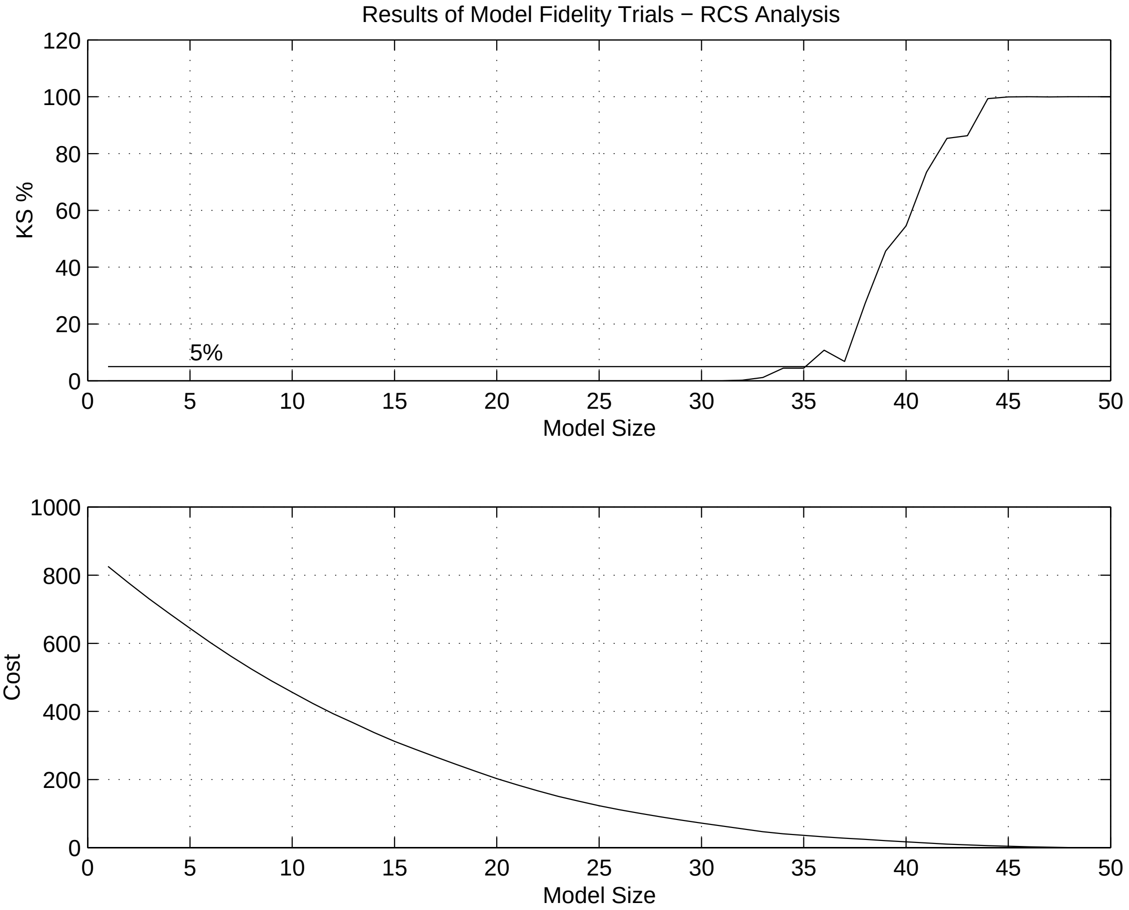


Figure 5.1: Results of reducing 50-point model for fidelity trials

Figure 5.2 shows the results of reducing the 100-point model. The model was produced with the scatterers positioned at random to give a similar physical extent to that of the 50-point model. It is interesting to note that the minimum model size

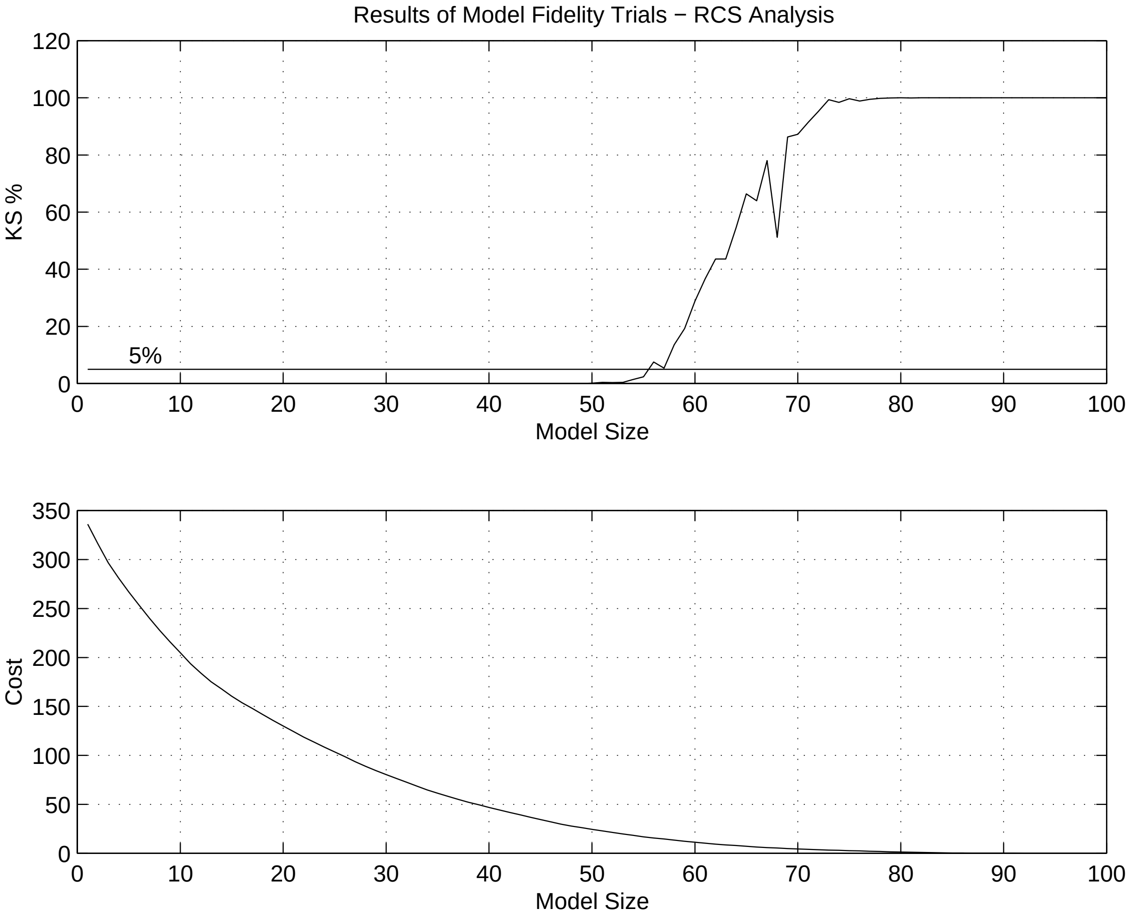


Figure 5.2: Results of reducing 100-point model for fidelity trials

where the hypothesis H_0 is accepted has 56 scatterers out of 100. The difference in the reduction ratio is probably due to the 50-point model's aircraft-like structure. It is clear from comparing the cost function plots of figures 5.1 and 5.2 that mean squared error is not a good gauge of fidelity as the curves' scales are related to the different radar cross section patterns.
The experiment is required to establish:

1. If the missile is influenced by target structure.

2. The reduction factor that can be applied before the target fidelity is compromised.

3. If the $\alpha = 0.9$ level used in the model fitting process will ensure a high fidelity model.

The launch position for the missile will be chosen at random. The missile will be fired from a 5km range towards a stationary target. Table 5.1 shows the missile configuration used in the trials. Two experiments will be performed:

Parameter	Value
Source power, $\mathcal{S}$	100W
Target detection range, R_N	10km
Seeker type	Amplitude Comparison
Frequency	10 Ghz
Beam-width, α	3°

Table 5.1: Missile configuration for fidelity trials

a. 1000 missile trials against each of the models derived from the 50-point target.

b. 1000 missile trials against each of the models, with an even number of scatterers, derived from the 100-point target. Only the even-sized models will be used to keep the processing overhead within reasonable limits.

A total of $100,000$ missile trials will be required to complete the experiments.

5.5. Experiment Results

Table 5.2 summarises the results of the experiments. Figure 5.3 shows the results of the trial miss-distance comparisons for the models derived from the 50-point model (Experiment *a*). The upper plot shows the Kolmogorov–Smirnov significance against model size with the 5% limit drawn on the graph. The sections of the curve that are below the line correspond to H_0 being rejected, ie. the miss distances are not the same. The lower trace shows the Mann-Whitney error figure against model size with the 5% limit drawn on the graph. The sections of the curve that are above the line correspond to H_0 being rejected, ie. the miss distances are not the same. Both tests agree that the smallest model where H_0 is accepted has 25 scatterers.

Parameter	Experiment a	Experiment b
Number of trials	50,000	50,000
Reference model	50-point	100-point
Minimum H_0 K–S accept (RCS)	36 scatterers	56 scatterers
Minimum H_0 K–S accept (Miss Dist)	25 scatterers	54 scatterers
Minimum H_0 M–W accept (Miss Dist)	25 scatterers	54 scatterers

Table 5.2: Results of target fidelity experiments

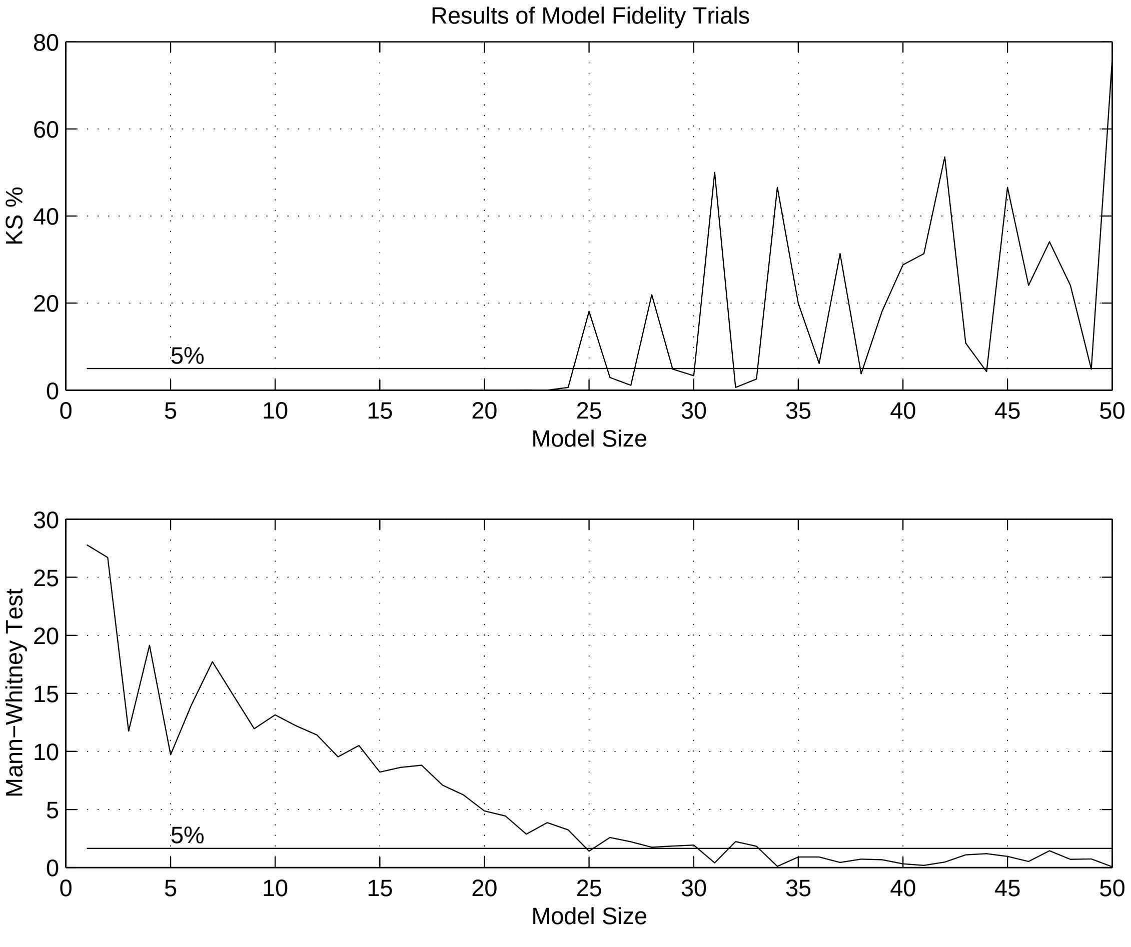


Figure 5.3: Results of experiment a of the fidelity trials (50-point model)

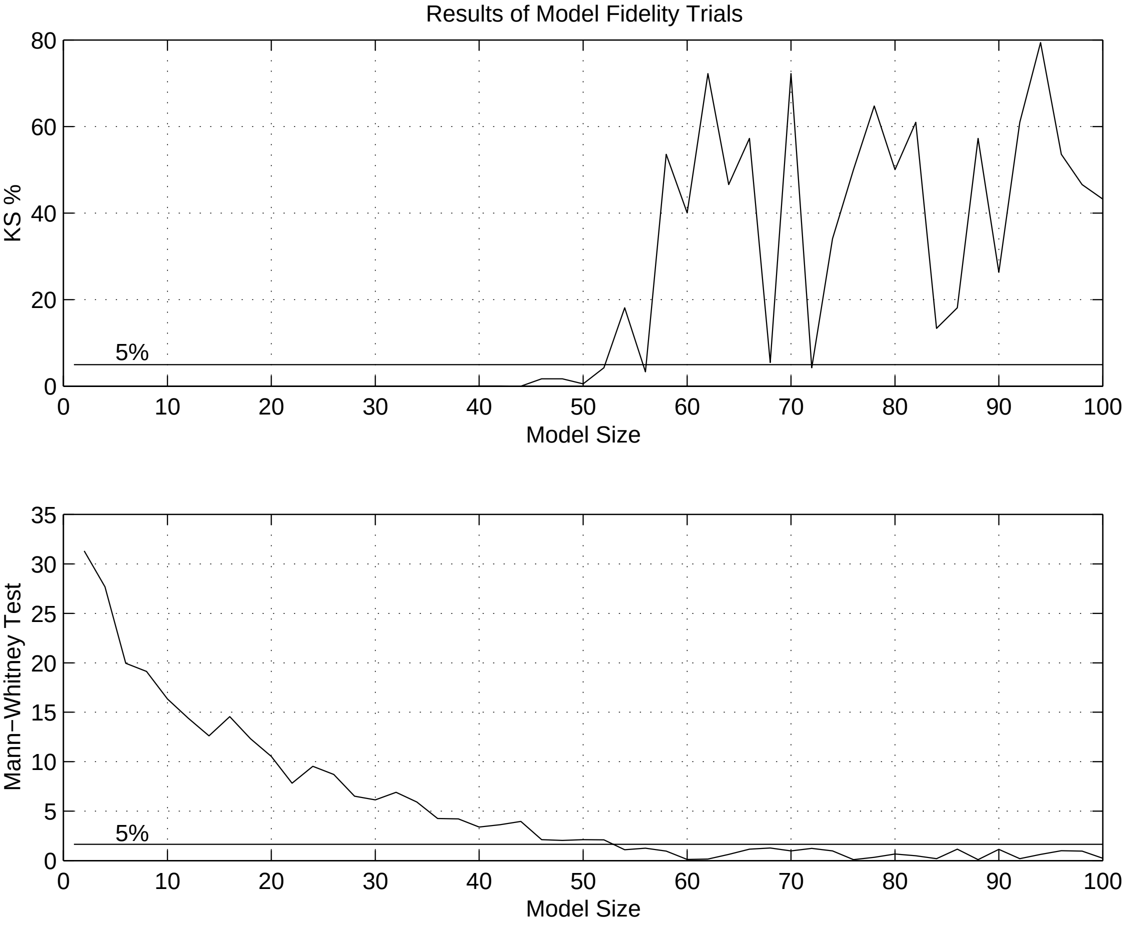


Figure 5.4: Results of experiment *b* of the fidelity trials (100-point model)

Figure 5.4 shows the results of the trial miss-distance comparisons for the models derived from the 100-point model (Experiment *b*). The upper plot shows the Kolmogorov–Smirnov significance against model size and the lower trace shows the Mann-Whitney error figure against model size. Both tests agree that the smallest model where H_0 is accepted has 54 scatterers.

5.6. Conclusions

The key results of the experiments are as follows:

1. The reduction factor to achieve $\alpha = 0.05$ for the Kolmogorov-Smirnov test of the model radar cross section may be influenced by the scatterer distribution.

2. The reduction factor to achieve $\alpha = 0.05$ for either the Kolmogorov-Smirnov test or the Mann-Whitney test of the trial miss distances appears to be independent of the model structure. The reduction factor is $\approx \frac{1}{2}$.

3. A model that is reduced to the $\alpha = 0.05$ limit of radar cross section will be within the $\alpha = 0.05$ limit of the trial miss distance distribution.

Result 3 is important as it confirms that the $\alpha = 0.9$ acceptance level used in the image-to-model conversion process should result in high-fidelity models. The experiments have shown that the K-S statistical measure can be used to gauge the level of target fidelity. To minimise the processing overhead of the target manoeuvre detection experiments in chapter 6, a high-fidelity reduced model has been created. The twenty seven scatterer model derived from the 50-point truth model has been chosen. The twenty seven scatterer model is higher-fidelity than the minimum 25-point model, but is not too large to compromise computational efficiency.

6. Target Manoeuvre Detection Using Glint

6.1. Introduction

Current target tracking algorithms are highly sophisticated and capable of tracking highly agile targets. Unfortunately, even agile targets spend most of their time in straight, level flight. Tracking algorithms that are designed to track manoeuvring targets are usually poor at following non-manoeuvring vehicles. The ability to switch rapidly between different tracking regimes is of paramount importance.

I have proposed that it may be possible, through observing glint noise, to detect when a target has changed its orientation in preparation for a rapid manoeuvre. As current aircraft performance is often pilot limited, in order to perform a high-g manoeuvre, the aircraft must bank steeply. The rapid target rotation associated with the bank is likely to produce a stream of glint spikes in the bore-sight error signal.

Any ability to detect potential manoeuvres can be used to greatly enhance the homing capability of missiles. The use of glint to augment target tracking is novel and could be used to create a significant tactical advantage in a radar guided missile for little cost.

This chapter first investigates the existing techniques used for target manoeuvre detection and glint processing. The noise characteristics of the bore-sight error signal are examined and the construction of a fuzzy-logic based manoeuvre detector is described. The experimental method is outlined and the trial results presented. Finally the potential of the technique is discussed and recommendations are made for further work.

6.2. Existing Manoeuvre Detection Techniques

6.2.1. Introduction

Two categories of manoeuvre detector have been identified:

1. **Prediction Methods** – These methods attempt to identify a manoeuvre by predicting the expected target position and comparing it to the measured track.

2. **Optical Methods** – The detector attempts to identify changes in the targets orientation by monitoring images of the target.

6.2.2. Prediction Methods

Prediction is the conventional approach to target manoeuvre detection. The current estimate of the target's position, velocity and acceleration are compared to the known target track. If the target deviates from its previous course, the target is manoeuvring. Unfortunately, the estimate of the target's position is subject to noise. This uncertainty prevents manoeuvres from being detected rapidly.

Most of the manoeuvre detection schemes use Kalman filter techniques to estimate the target parameters [67]. One of the most successful methods uses the interacting multiple models (IMM) algorithm [68]. The algorithm allows the kinematic equations in the Kalman filter to represent a set of different manoeuvring hypothesis. The ability to use different target models enables the Kalman filter to use a tracking regime that best suits the target manoeuvre.

Prediction methods are quite slow and typically require a few seconds to detect a manoeuvre. This speed of reaction is satisfactory for a ground based tracking radar, but not for a missile.

6.2.3. Optical Methods

Kendrick et al. [69] first investigated the use of target orientation information from optical sensors to augment tracking algorithms. A pattern recognition algorithm was used to predict the target orientation from the images. The image processing is quite computationally intensive. Sworder and Hutchins [70] also apply imaging techniques in order to estimate target radial acceleration.

Shetty and Alouani [71] have used three centroid position measurements of the target image to detect manoeuvres. This approach does not require any *a-priori* knowledge of the target structure and is relatively quick to process. Romine and Kamen [2] provide a detailed description of a Kalman filter based optical manoeuvre detector. Their results show that the optical technique can detect a manoeuvre in under half a second.

Laneuville and Mariton [72] have proposed using an interacting multiple model tracking algorithm with an image based manoeuvre detector. The number of pixels in the image was suggested as a feature to use in the manoeuvre detector. They observed a significant improvement in the tracking performance over the basic interacting multiple model algorithm approach.

6.2.4. Discussion

The predictive methods are inherently slow in detecting rapid manoeuvres. The performance is often satisfactory for ground-based target trackers, but unsuitable for use in missiles. A combination of these sophisticated tracking techniques and a fast manoeuvre detector would provide a better solution to the tracking problem.

Most of the optical methods are based on the assumption that the target must bank before performing a high-g turn. The bank manoeuvre is required to prevent the pilot experiencing excessive lateral g forces. Most of the methods operate by counting the number of pixels that form the image of the target. As the target manoeuvres, different views of the target are seen. For example, if the target is being viewed broad-side with its wings in the horizontal plane, a long, thin image will be seen. If the target performs a bank manoeuvre as part of a high-g turn away from the observer, the wings of the target will rotate into the vertical plane. The underside of the target will now be seen, presenting a much larger image area and therefore a greater number of pixels.

The optical methods all require large amounts of image processing. Expensive extra hardware would be required in a missile. Some of the techniques also require knowledge of the target structure and characteristics. This is undesirable. Optical techniques are more susceptible to climatic effects when compared to radar and are only practically suitable for medium to short range engagements.

Sworder and Hutchins [73] provide a detailed investigation into the effects of different frame rates on the probability of detecting manoeuvres. They conclude that even at very high frame rates, certain manoeuvres where there is only a small change in the image, are very difficult to detect reliably.

6.3. Glint Processing Methods

In order to detect target manoeuvres from glint, an investigation into the techniques used to remove glint errors was conducted. These techniques give an insight into the processing methods currently in use to identify glint spikes.

Many methods exist that attempt to improve target tracking accuracy in the presence of glint. Most tracking radars utilise Kalman filters to estimate optimally the target motion from noisy radar data. Unfortunately, the Kalman filter is highly susceptible to non-Gaussian noise. It has been shown that glint noise is highly non-Gaussian and has a long-tailed distribution [74]. The levels of noise due to glint also increase with decreasing range. Wu [75, 76] has developed a tracking filter that can deal directly with the non-Gaussian glint noise by using multiple models internally in a Kalman filter.

A different approach has been taken by Hewer et al. [74] where the error signal is pre-processed in an attempt to make the glint noise approximate a Gaussian distribution. The pre-processing allows a conventional Kalman filter to be used to remove the noise without compromising the filter stability. Das and Yoganandam [77, 78] apply non-coherent processing to amplitude comparison monopulse signal envelope data in an attempt to reduce glint and the effects of receiver phase imbalances. This method is robust and achieves a consistent improvement in glint error.

It has been shown that glint spikes are highly correlated to deep echo amplitude fades and yet uncorrelated with glint observed at other frequencies or aspect angles [79]. These principles have been applied to glint reduction in a number of ways. Borden [80] derives methods that use either frequency or angular agility to integrate multiple bursts and therefore average out the glint spikes. The techniques work but may require sophisticated seeker heads in order to function properly. Guest [81] utilises the correlation between amplitude fades and glint spikes to censor the data being passed to the Kalman filter when fades occur. The censoring gives a general improvement but small glint spikes still pass through to the Kalman filter.

In general, target tracking becomes more and more difficult the closer the radar gets to the target. In the last few hundred metres, the glint noise can be extreme and highly non-Gaussian in nature. Of the glint reduction methods surveyed, all can give an improvement in the tracking capabilities of the radar but none can eliminate the effects of glint totally.

The approaches that use a combination of the bore-sight error signal and the radar cross section seem to be the most promising. Antenna noise effects are related to the target radar cross section and must be accounted for in the manoeuvre detector processing.

6.4. Bore-sight Error-Signal Noise Characteristics

6.4.1. Introduction

The noise on the bore-sight error-signal consists of two main components:

1. **Antenna Noise** – The antenna noise levels seen in the bore-sight error signal decay with decreasing range but are also related to the target radar cross section. Most of the noise is thermally generated and has a Gaussian distribution. Antenna noise is the dominant source at long and medium range.

2. **Target Glint** – The noise is highly non-Gaussian and is related to the relative rate of tangential rotation between the target and the missile. Glint is the noise to be used to detect a manoeuvre. Target glint noise also occurs due to changes in range. This range-glint occurs mainly at short ranges and is the dominant source of noise in the last few kilometres of an engagement.

These mechanisms and the approaches taken to account for the noise are detailed below. Noise induced by countermeasures is beyond the scope of this feasibility study.

6.4.2. Antenna Noise

Antenna noise is a combination of channel noise, receiver noise and the thermal noise in the signal processing stages. The noise is approximately Gaussian and the signal from each antenna may be defined as $N(\mu, k^2)$, where μ is the mean of the signal and k^2 is the variance of the noise. The noise variance, k^2, is a characteristic of the missile and essentially remains constant throughout the engagement. An approximation for k may be calculated from the operational characteristics of the missile. Equation 6.1 details this calculation, where the RMS noise level k is set to give a unity signal to noise ratio of the received echo, at the range R_N metres against a $1m^2$ target. The source power of the missile is denoted by $\mathcal{S}$, normalised to give an antenna with unity gain.

$$k = \frac{\sqrt{\mathcal{S}}}{4\pi R_N^2} \tag{6.1}$$

As part of the bore-sight error-signal processing, a sum and difference of the four receiving antennas in the seeker head is formed. If the seeker is aligned with the target, the difference channel will be a function of antenna noise only. Equations 6.2 and 6.3 show the noise functions of the difference and sum signals, where

z represents the magnitude of the echo signal from the sum channel.

$$\begin{aligned} d &= N(0, 4k^2) && (6.2) \\ s &= N(z, 4k^2) && (6.3) \end{aligned}$$

The monopulse ratio may be approximated from the difference and sum signals. Equation 6.4 shows the approximation used for dividing two Gaussian noise signals. The equation has been derived empirically and is detailed further in appendix F.

$$\frac{N(\mu_1, v_1)}{N(\mu_2, v_2)} \approx N\left(\frac{\mu_1}{\mu_2}, \frac{\mu_1^2 v_2 + \mu_2^2 v_1 + v_1 v_2}{\mu_2^2(\mu_2^2 + v_2)}\right) \tag{6.4}$$

Applying equation 6.4 gives the result shown in equation 6.5.

$$\frac{d}{s} \approx N\left(0, \frac{4k^2}{z^2}\right) \tag{6.5}$$

For phase comparison monopulse, the imaginary part of the monopulse ratio is extracted. By reference to the noise model in equation 4.18, the effects on the noise will result in equation 6.6.

$$\Im\left(\frac{d}{s}\right) \approx N\left(0, \frac{2k^2}{z^2}\right) \tag{6.6}$$

If a small number of bore-sight error and sum signal samples are taken (10 in this study), the mean radar cross section, z, can be approximated from the mean of the sum signal, $\bar{\mathbf{s}}$. Thus the antenna noise may be approximated as shown in equation 6.7, where the signals now represent short vectors of samples. Equation 6.7 may also be applied to amplitude comparison monopulse where the real part of the complex difference is used instead of the imaginary part as here.

$$\mathbf{N}_b = N(0, k^2) \approx \frac{\bar{\mathbf{s}}}{\sqrt{2}} \Im\left(\frac{\mathbf{d}}{\mathbf{s}}\right) \tag{6.7}$$

The set of samples, $\mathbf{N}_b$, may be normalised using equation 6.8 to give $\hat{\mathbf{B}}$, which is an approximation of the bore-sight error signal with an antenna noise component of zero mean and unity variance.

$$\hat{\mathbf{B}} \approx \frac{\mathbf{N}_b - \overline{\mathbf{N}}_b}{k} \tag{6.8}$$

The standard deviation of $\hat{\mathbf{B}}$ may now be monitored. The effects of antenna noise, now corrected for range and radar cross section, should lie within two standard deviations for 95% of the time. Any signals outside this range are likely to be glint.

The theoretical analysis is only approximate and will need to be addressed if a more refined manoeuvre detector is to be developed. The optimum number of samples to use is unknown. More samples may delay detection of the manoeuvre and less samples will increase the chances of false detection.

6.4.3. Target Glint

Target glint is an artifact of monopulse processing and represents an apparent positional error of the target. Monopulse processing effects are described earlier in section 4.3.2. Glint spikes are highly correlated to the target radar cross section pattern and most often occur when the radar cross section amplitude fades. The effect is that the signal in the sum channel of the seeker head becomes very small and the bore-sight error signal approaches a divide-by-zero condition. Radar cross section amplitude fades occur due to echos from scatterers being out of phase and cancelling at the receiver. Target glint is associated both with target rotation and range. In proportional navigation homing missiles, a zero rate of line-of-sight rotation is maintained. Thus glint spikes caused by rotation should only occur during a manoeuvre.

In figure 6.1, the seeker antenna is pointed directly at scatterer-A but is receiving echos from both scatterers. If the scatterers are the same magnitude, at very long ranges, distance r is approximately the same as distance q. As the range reduces, the distance q must be represented as $q = \sqrt{r^2 + x^2}$. When the difference between r and q is half a wavelength, the scatterers will cancel, causing a glint spike. This type of noise is range dependent and has been termed *range-glint*. As we are interested in active seekers, the distance that the radar pulse/echo travels is twice the range to the scatterer. Equation 6.9 details the condition for the furthest glint spike.

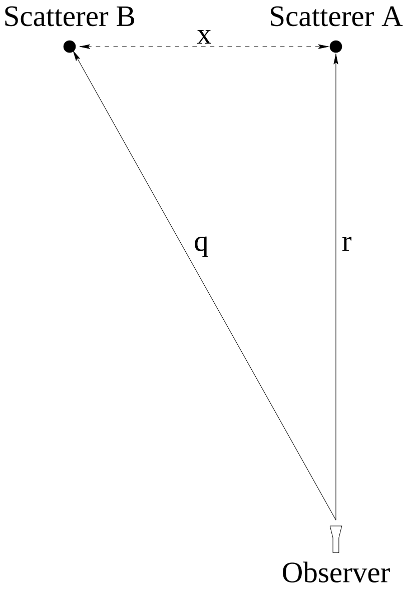


Figure 6.1: Scatterer configuration for glint calculations

$$\begin{aligned}
2q - 2r &= \frac{\lambda}{2} \\
\sqrt{r^2 + x^2} - r &= \frac{\lambda}{4} \\
x^2 &= \left(\frac{\lambda}{4}\right)^2 + \frac{\lambda r}{2}
\end{aligned}$$

$$r = 2\frac{\left(\left(\frac{\lambda}{4}\right)^2 + x^2\right)}{\lambda} \tag{6.9}$$

As $\left(\frac{\lambda}{4}\right)^2$ is very small in comparison to x^2 at the frequencies and distances of interest, the maximum range may be simplified as shown in equation 6.10.

$$r = 2\frac{x^2}{\lambda} \tag{6.10}$$

For an n-point scatterer model, there are $n(n-1)/2$ possible inter-scatterer distances. Figure 6.2 shows a histogram of the distribution of the 1225 inter-scatterer distances associated with the 50-point model detailed in appendix B. For comparison, figure 6.3 shows a histogram of the distribution of the 4950 inter-scatterer distances associated with the 100-point model detailed in appendix B. The scatterers in the 100-point model were distributed randomly. The distance histogram has only one peak. To determine if the multiple peaks in the histogram of the 50-point model are characteristic of the target shape, the 128-point model produced from the real 2D data in chapter 3 was investigated. Figure 6.4 shows a histogram of

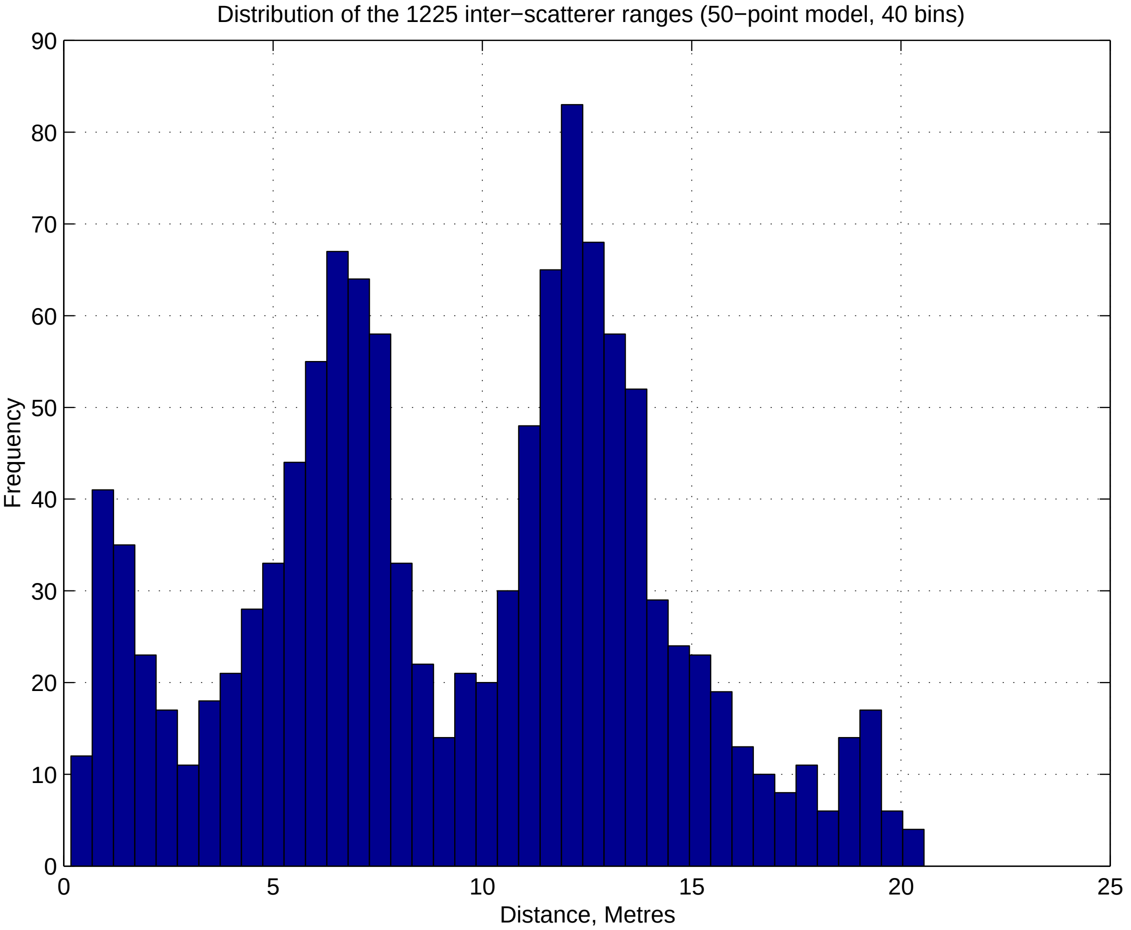


Figure 6.2: Histogram of inter-scatterer distances (50-point realistic model)

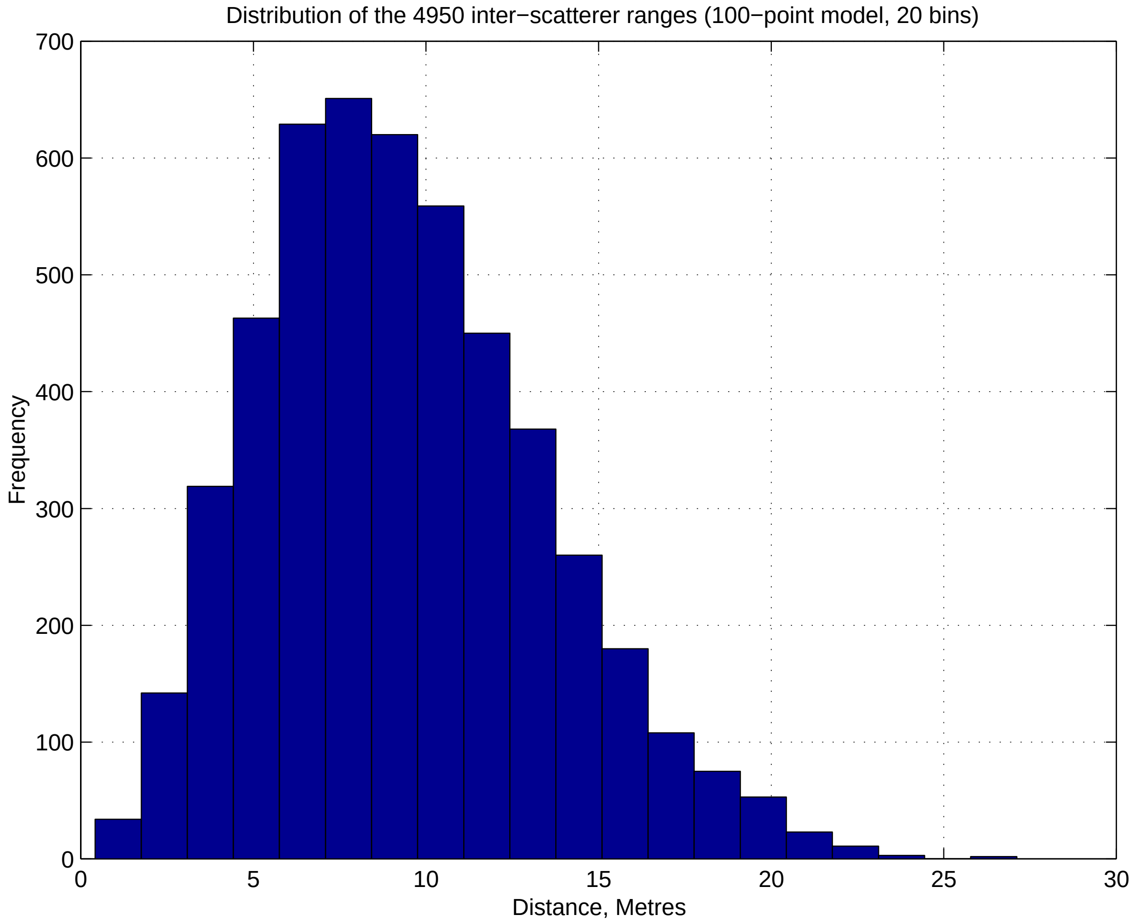


Figure 6.3: Histogram of inter-scatterer distances (100-point random model)

the distribution of the 8128 inter-scatterer distances associated with the 128-point model produced from the real 2D data. The multiple peaks are clearly visible and the locations correspond well to the peaks in figure 6.2. Clearly the target structure is important for generating realistic targets. The difference in peak sizes is probably due to the real data only being two-dimensional and fitted over an narrow angular range.

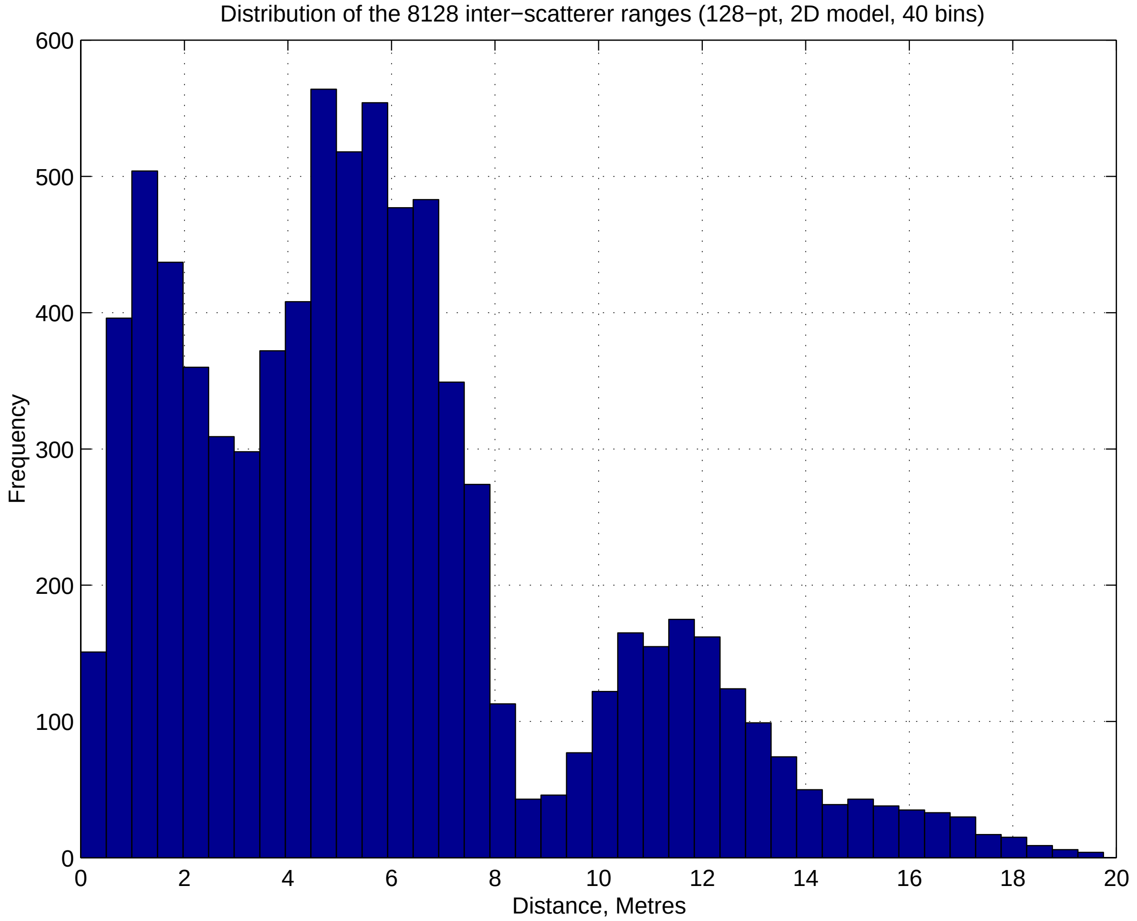


Figure 6.4: Histogram of inter-scatterer distances (128-point 2D model)

Figure 6.5 shows a histogram of the maximum distance for the range-glint spikes associated with the inter-scatterer distances shown in figure 6.2. The multiple peaks are again a dominant feature of the distribution. Consequently, there is an unexpected increase in the frequency of glint spikes due to changing range at around ten kilometres. The 50-point target has a twenty metre wing-span. Smaller targets would produce shorter range-glint distributions. In practice, the antenna never points exactly at a scatterer. The distances r and q in equation 6.9 become more similar as the antenna moves between the scatterers. The range at which each spike will occur will therefore be reduced slightly. Thus in practice, the distribution shown in figure 6.5 will be continuous.

The seeker head used in this investigation has a sample time of two milli-seconds. This sample rate is far too slow to see the rise and fall of each glint spike caused

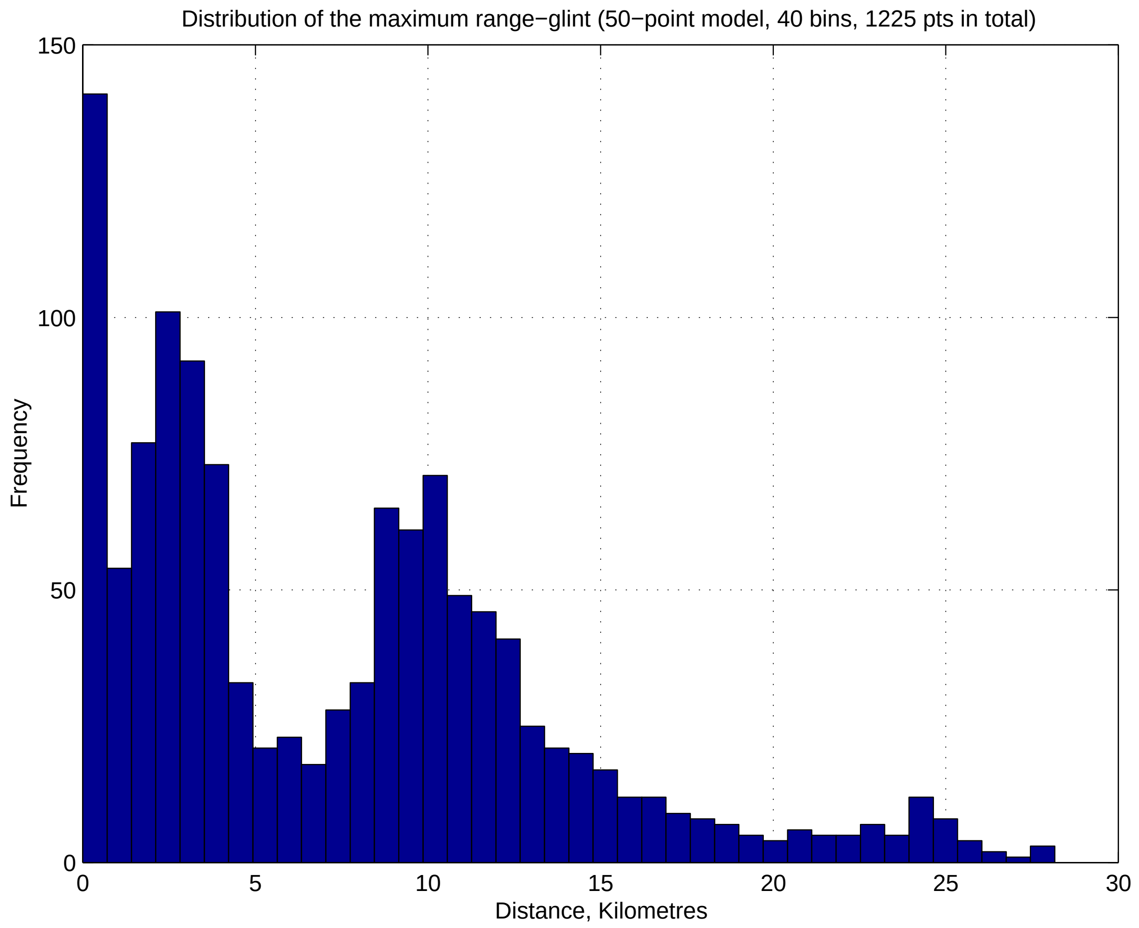


Figure 6.5: Histogram of maximum range-glint distances

by target rotation. The glint spikes from a manoeuvre appear as high-level wide-band noise on the bore-sight error signal. Range-glint spikes occur singly at medium range, with their frequency increasing with reducing range. It is possible for range-glint spikes to extend across many samples. The effect can influence the standard deviation of $\hat{\mathbf{B}}$ and cause false triggering of the manoeuvre detector. To counter these low frequency signals, the standard deviation of a high-pass filtered version of $\hat{\mathbf{B}}$ is also monitored. The high-pass filter is easily achieved by numerical differentiation of $\hat{\mathbf{B}}$, giving $\frac{d\hat{\mathbf{B}}}{dt}$. Observations indicate that the standard deviation of $\frac{d\hat{\mathbf{B}}}{dt}$ is about twice that of $\hat{\mathbf{B}}$ (assuming $dt = 1$). This empirical result is sufficient for use in the fuzzy-logic detector.

Figure 6.6 shows the bore-sight error signal for a typical engagement from a range of 10km. The engagement shown included a $2g$ coordinated manoeuvre lasting 1.3 seconds. Extreme glint spikes in the bore-sight error signal have been cropped to ±2 to aid legibility. The manoeuvre is indicated by the dashed line on the plot, the sloping sections indicating the periods when the target was banking. Figures 6.7 and 6.8

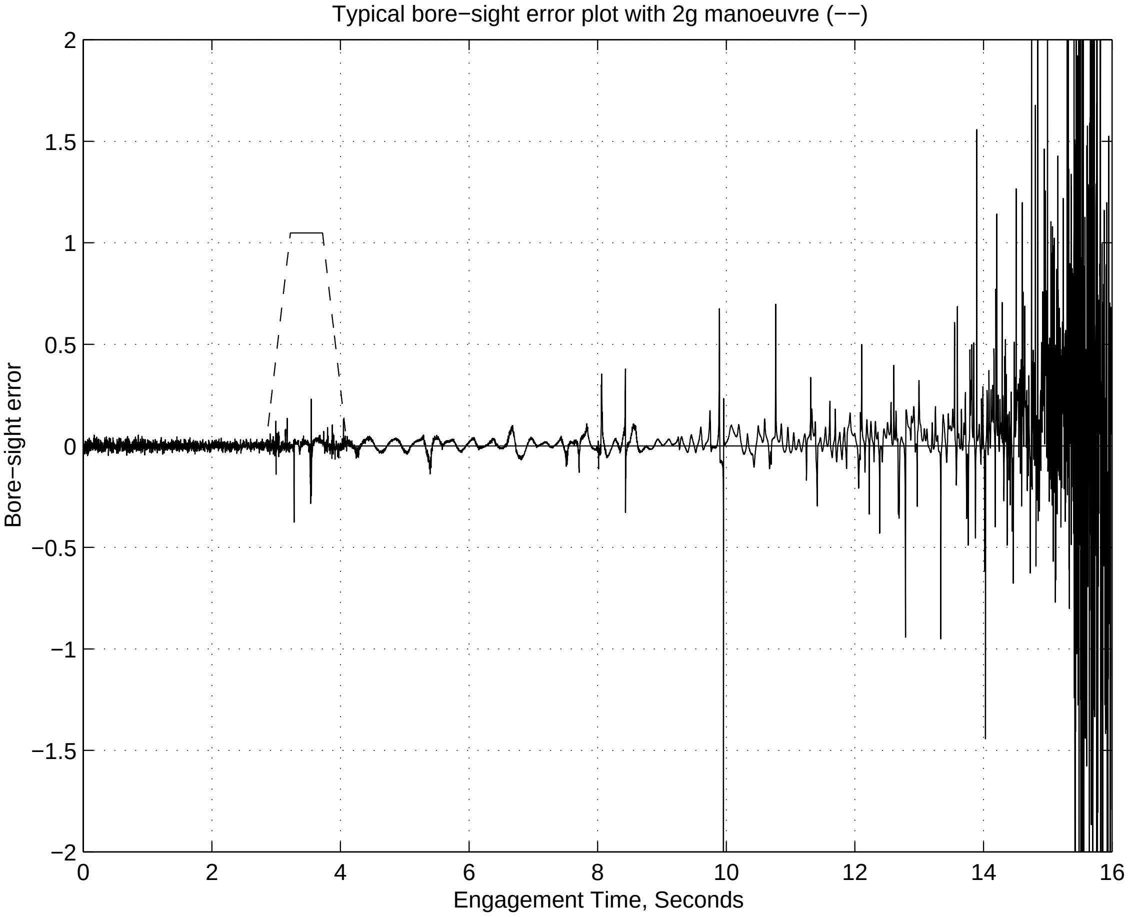


Figure 6.6: Typical bore-sight error plot (10km engagement)

show the plots of $\hat{\mathbf{B}}$ and $\frac{d\hat{\mathbf{B}}}{dt}$ respectively for the first eight seconds of the engagement. The manoeuvre is quite clear on these figures.

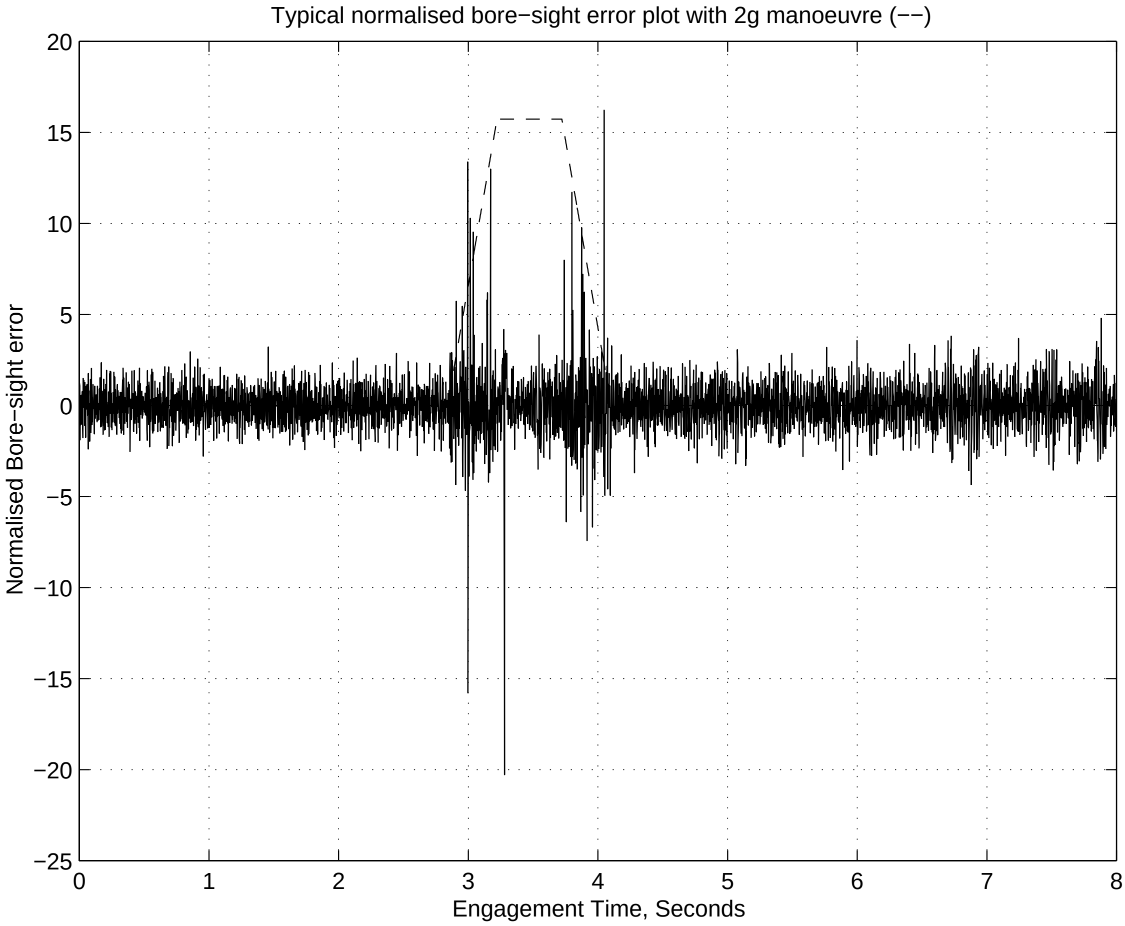


Figure 6.7: Normalised bore-sight error plot of figure 6.6 ($\hat{\mathbf{B}}$)

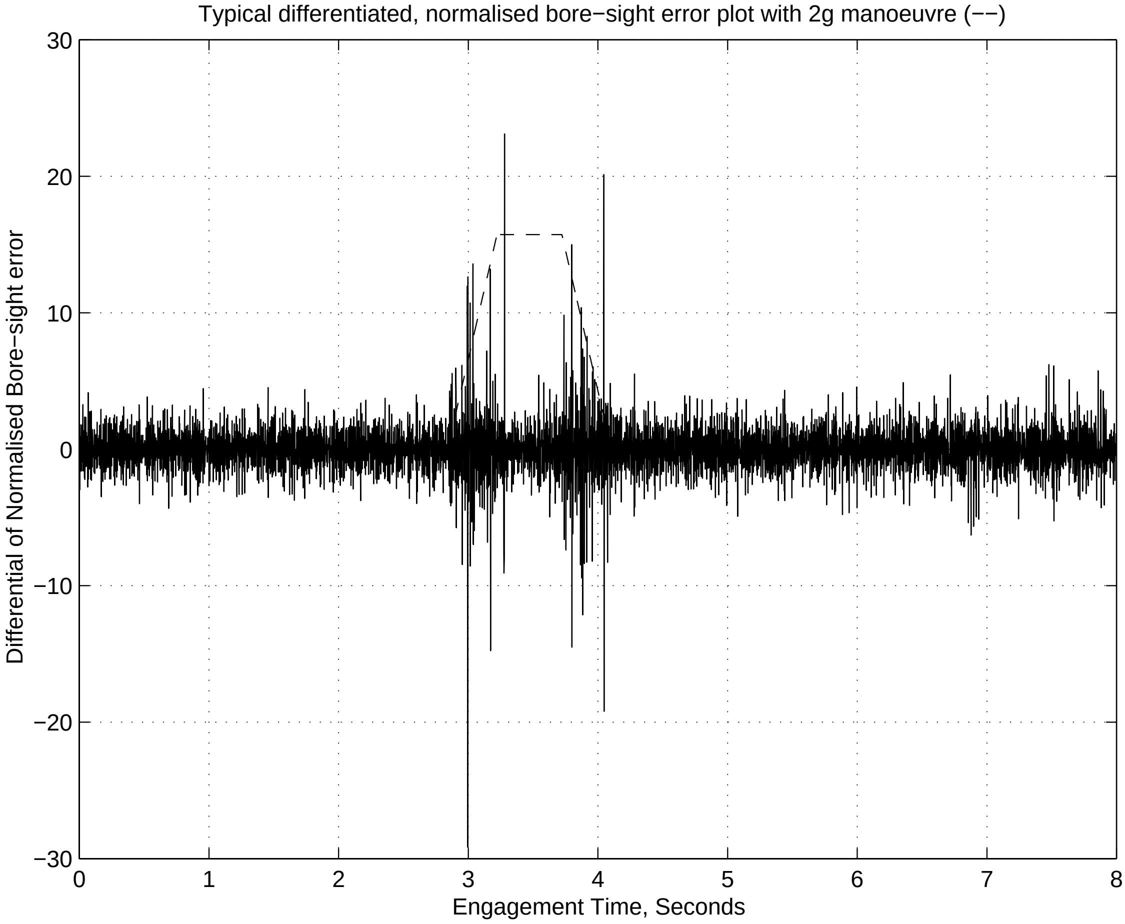


Figure 6.8: Differential of figure 6.7 ($\frac{d\hat{\mathbf{B}}}{dt}$)

It is proposed that to trigger the manoeuvre detector, the standard deviation of both $\hat{\mathbf{B}}$ and $\frac{d\hat{\mathbf{B}}}{dt}$ should be large (≈ 2 or greater for $\hat{\mathbf{B}}$, ≈ 4 or greater for $\frac{d\hat{\mathbf{B}}}{dt}$ for 95% inclusion). The seeker head generates signals for both horizontal and vertical errors. A significant signal in either channel should trip the manoeuvre detector. At medium and short ranges, there should be a slight delay between the first glint spike being detected and activation of the manoeuvre detection signal. Indication of a manoeuvre should only be generated if the glint spikes are still present at the end of the delay period. The delay helps prevent the solitary range-glint spikes from causing false alarms, at the expense of an increased turn-on delay. At long ranges the glint spikes from a manoeuvre may be small in size and widely spaced. Therefore, only a very short delay should be used.

Once a manoeuvre has been detected, the manoeuvre detector must be able to coast over small gaps between the spikes. The wider the gap that can be tolerated, the less likely are the chances of terminating falsely the manoeuvre indication signal. The longer the hold time, the longer it will take for the detector to signal the end of a manoeuvre. While the detector is activated, a large standard deviation of either $\hat{\mathbf{B}}$ or $\frac{d\hat{\mathbf{B}}}{dt}$ would be sufficient to maintain the detection signal.

In the last few kilometres of the engagement, the normalised bore-sight error signal level, $\hat{\mathbf{B}}$, becomes very susceptible to range-glint, causing false triggering of the manoeuvre detector. Luckily, the levels of glint due to a manoeuvre are also very large in this region. It is suggested that the standard deviation of the raw bore-sight error signal is used directly, with a suitable gain to bring the manoeuvre glint spikes into the range of ≈ 3 standard deviations. This gain needs to reduce with reducing range to account for the increasing glint levels. Typical signal levels may be seen in the last three seconds of the engagement shown in figure 6.6. In the engagement used to generate figure 6.6, the missile was unable to recover from the $2g$ manoeuvre and missed the target by 117 metres.

6.5. Manoeuvre Detector Construction

6.5.1. Introduction

As only a feasibility study is being performed into target manoeuvre detection using glint, a fuzzy-logic approach was adopted for speed of development and simplicity. The fuzzy-logic detector is only intended as a proof-of-principle model and therefore only a minimum of tuning was performed.

Fuzzy-logic is a convenient method for mapping input data into a new output space. The form of the mapping is controlled essentially by the membership functions used to fuzzify the input data, a set of rules that determine which output functions are active, and the output function shapes. An excellent description of fuzzy-logic is given in the *MATLAB* Fuzzy Logic Toolbox Manual [82].

6.5.2. Concept

The manoeuvre detector has been constructed from four different fuzzy inference system module types. Figure 6.9 shows how the modules are inter-connected. The functions of the four modules are as follows:

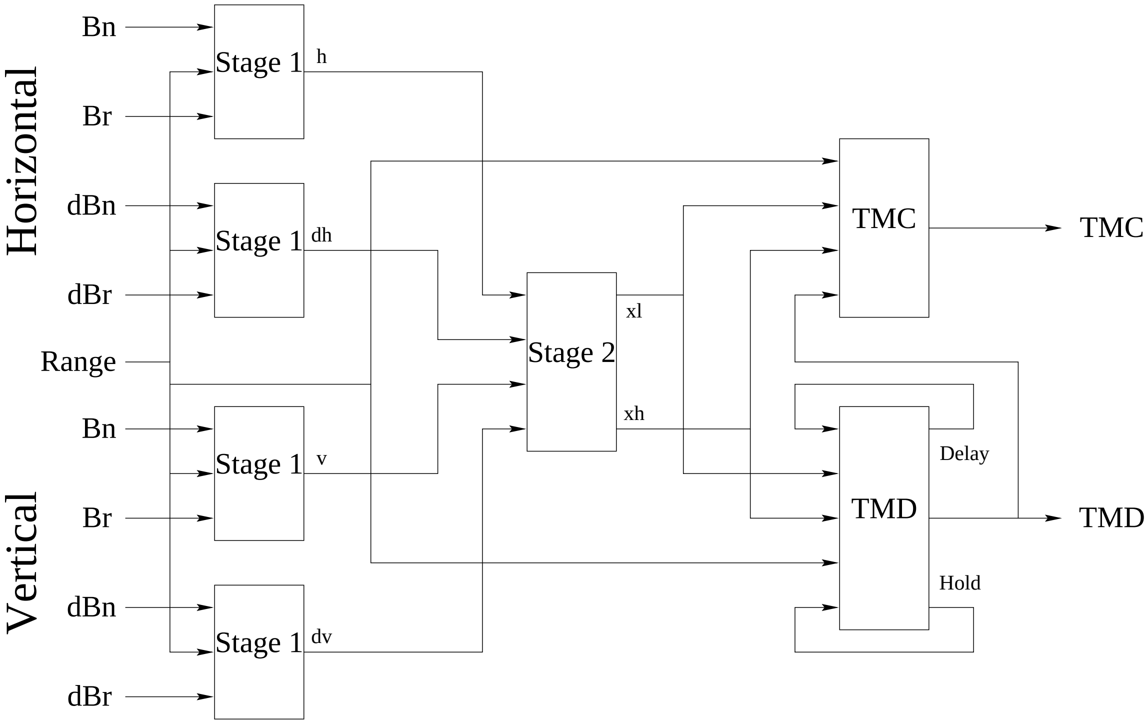


Figure 6.9: Block diagram of fuzzy-logic manoeuvre detector

1. **Stage 1** – This module removes the range dependent characteristics of the bore-sight error signals. The signals B_n and B_r represent the standard deviations of the normalised and raw bore-sight error signals respectively. The signals dB_n and dB_r represent the standard deviations of the differentiated versions of the normalised and raw bore-sight error signals. At long ranges, the normalised signals are passed straight through. At short ranges, the raw error signal is output through a range-dependent gain.

2. **Stage 2** – The module combines the four processed standard deviation signals for the normal horizontal, differentiated horizontal, normal vertical, and differentiated vertical components. Two signals are generated. The first, xl, is active when all the signals are low. The second, xh, is active when either both horizontal signals or both vertical signals are large.

3. **TMD** – The manoeuvre detector module uses the activity signals from stage-2 to provide the manoeuvre detection signal. The trigger delay and detection hold functions are implemented by using a state feedback loop for each function. The delay and hold times are range dependent.

4. **TMC** – This module provides an indication of the manoeuvre detection signal accuracy. When the delay or hold functions are operating, the signals from the stage-2 module may contradict the manoeuvre detector output. The confidence signal should therefore be low accordingly. The confidence level is also reduced at both long and short ranges to reflect the difficulty of detection.

Details of the software for each of the modules is given in appendix G.

Figure 6.10 shows the output of the manoeuvre detector for an example $6g$ coordinated turn. The detector turned on in 80 milli-seconds and turned off in 104 milli-seconds.

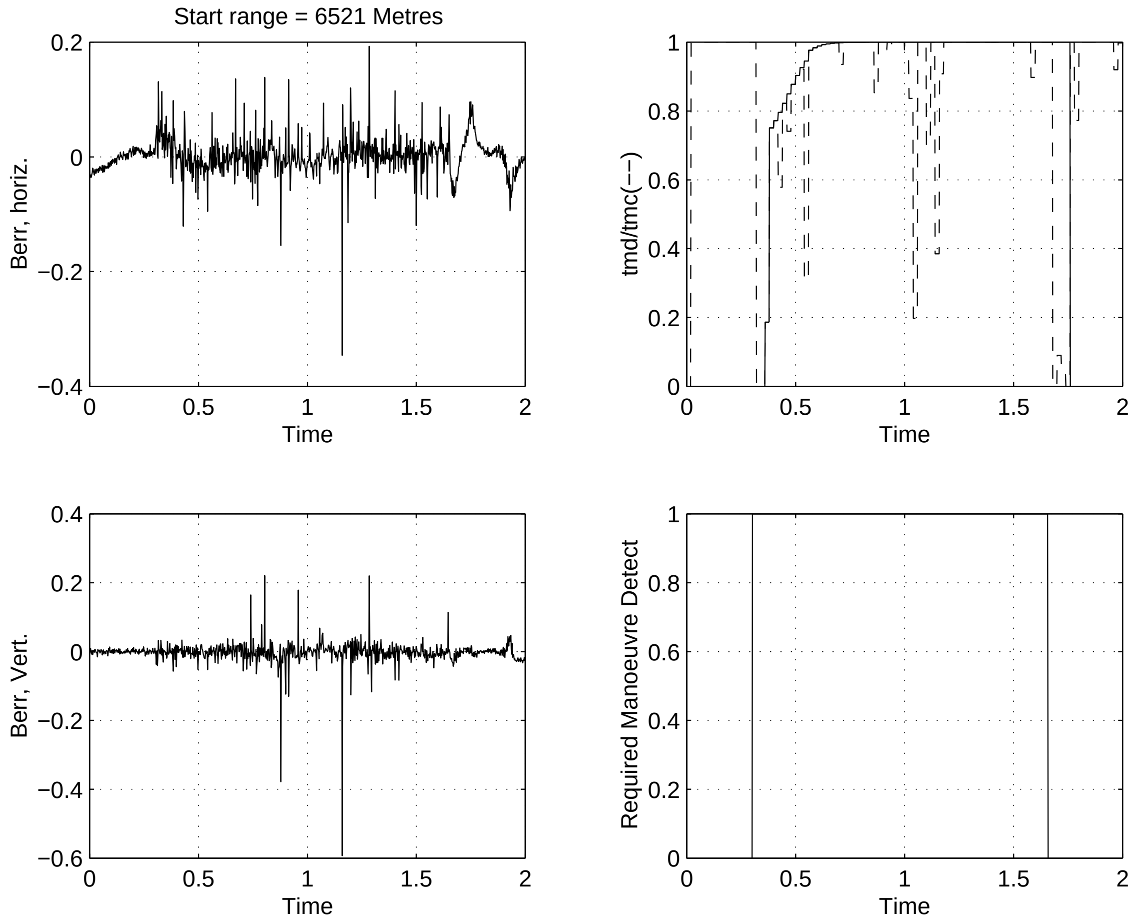


Figure 6.10: Example manoeuvre detector response

6.6. Experimental Method

In order to determine the general performance of the manoeuvre detector, an experiment was devised that could characterise the following performance criteria.

1. Turn on time
2. Turn off time
3. Reliability of detecting manoeuvre
4. Reliability of detecting manoeuvre start
5. Reliability of detecting manoeuvre progress

6. Reliability of detecting manoeuvre end

As the noise characteristics are range dependent, the performance criteria must be evaluated for engagements at different ranges. The performance of the confidence output, *tmc*, is not required to be evaluated.

The effects of different launch positions need to be minimised in this experiment, therefore the launch positions will be chosen at random using equation 4.31 to give uniform 4π steradian coverage. The launch range will be chosen at random from between 1km and 20km. A random delay lasting between 0.2 and 1 second will be given before the manoeuvre to allow the missile to stabilise. The manoeuvre will be a $6g$ coordinated turn. This manoeuvre involves the target banking to 80.4° to maintain a maximum lateral force of $1g$ on the pilot. The bank will take 0.51 seconds to complete. The turn will be maintained for a period of between 0.25 and 0.75 seconds before the target reverse banks and returns to straight, level flight. The reverse bank manoeuvre will last for 0.51 seconds. The engagement will be terminated 0.5 seconds after the manoeuvre has been completed. A $6g$ coordinated turn is likely to seriously disturb the missile guidance and so there will be no benefit in extending the simulation after the manoeuvre. The target model used in the simulation will be the 27-point model that resulted from the fidelity trials in chapter 5. The model is detailed in appendix B. Five thousand trials will be run to give one trial approximately every four metres in range.

The missile uses a proportional navigation homing guidance system with a phase comparison monopulse seeker. The seeker configuration is shown in table 6.1.

Parameter	Value
Source power, $\mathcal{S}$	100W
Target detection range, R_N	10km
Seeker type	Phase Comparison
Frequency	10 Ghz
Beam-width, α	10°
Head Offset, x	0.06m
Model	27-Point

Table 6.1: Missile configuration for manoeuvre detector trials

Figure 6.11 shows the timing details of the manoeuvre detection signal. Pre-detection errors will occur in the time period t_s. The rise time of the manoeuvre detector is t_r. Drop-out errors, where the detector falsely switches off, will occur during time t_p. The fall time of the manoeuvre detector is t_f. Post-detection errors will occur in the time period t_e.

The following details of the trials will be logged:

1. Starting range of manoeuvre.

2. Turn-on time.

3. Turn-off time.

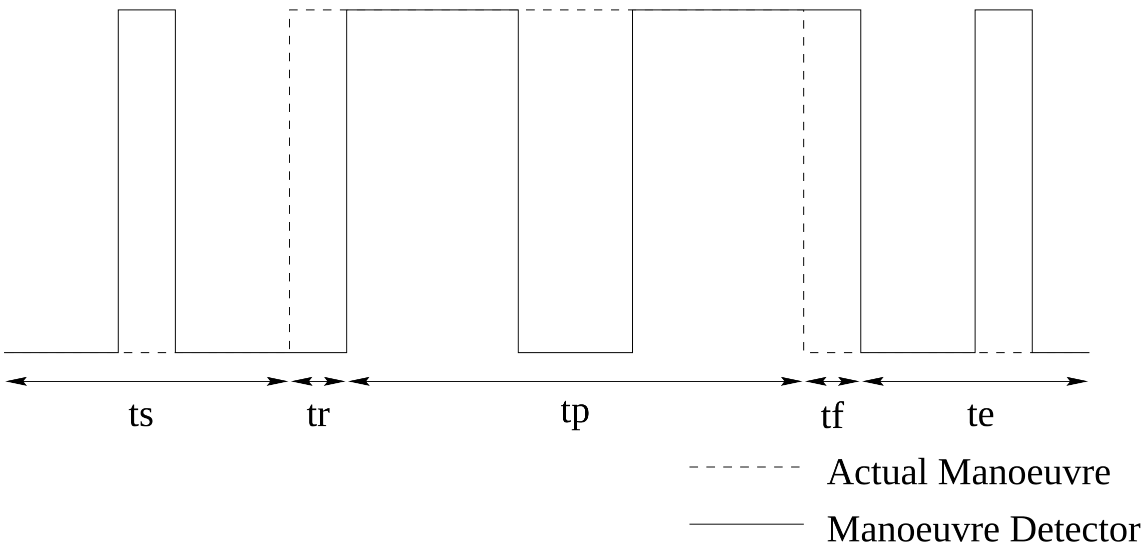


Figure 6.11: Timing details of the manoeuvre detection signal

4. The number of total detection failures, where the detector failed to respond to the manoeuvre.
5. The number of start-detection errors, where the detector was already tripped when the manoeuvre occurred.
6. The number of end-detection errors, where the detector failed to turn off within 0.5 seconds after the manoeuvre ended.
7. The proportion of the pre-detection time (t_s) that the detector was falsely tripped.
8. The proportion of detection time (t_p) that the detector had dropped out.
9. The proportion of the post-detection time (t_e) that the detector was falsely tripped.

6.7. Trial Results

The results of the trials were sorted by manoeuvre start range before plotting. Table 6.2 summarises the main results of the experiments.

Figure 6.12 shows the distribution of turn-on delay with respect to range. The peak around 9km is coincident with the peak in the range-glint distribution of figure 6.5. The peak also seems to be related to the change in the delay time of the fuzzy detector. An experiment with a different delay time still maintained the bulk of the peak, suggesting it is a target characteristic. The mean turn-on delay time is 96 milli-seconds. This delay is comparable to the reaction speeds of optical manoeuvre detectors.

Parameter	Result
Number of trials	5000
Mean turn-on delay	96 milli-seconds
Mean turn-off delay	148 milli-seconds
Number of total failures	2
Number of detection failures	14
Number of turn-off failures	403
Minimum range of manoeuvre start	850 metres
Maximum range of manoeuvre start	19891 metres

Table 6.2: Results of target manoeuvre detection experiment

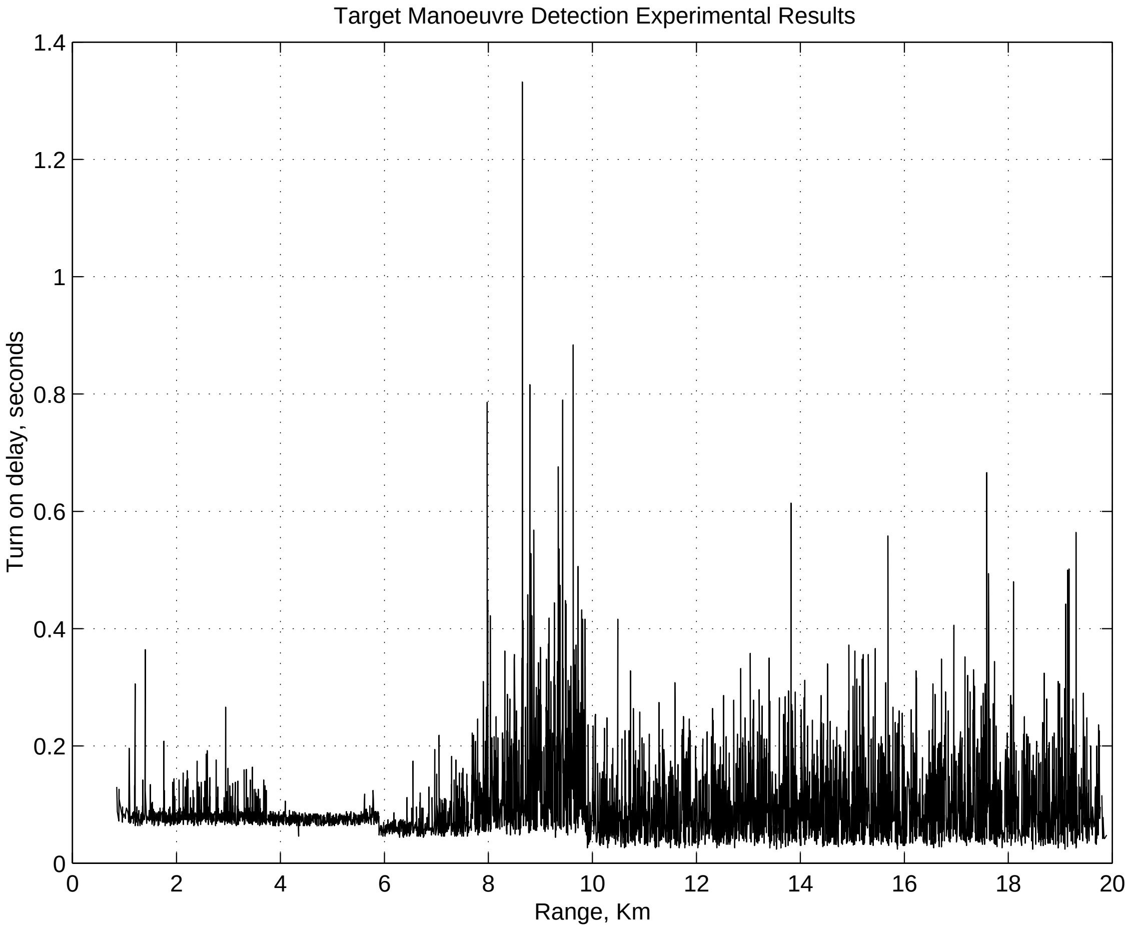


Figure 6.12: Turn-on delay with respect to range

Figure 6.13 shows the distribution of the turn-off delay times with respect to range. The influence of the fuzzy membership function for range in the TMD module is clearly visible between ten and twenty kilometres. The mean turn-off delay time is 148 milli-seconds. Again, the turn-off time is comparable to the reaction speeds of optical manoeuvre detectors.

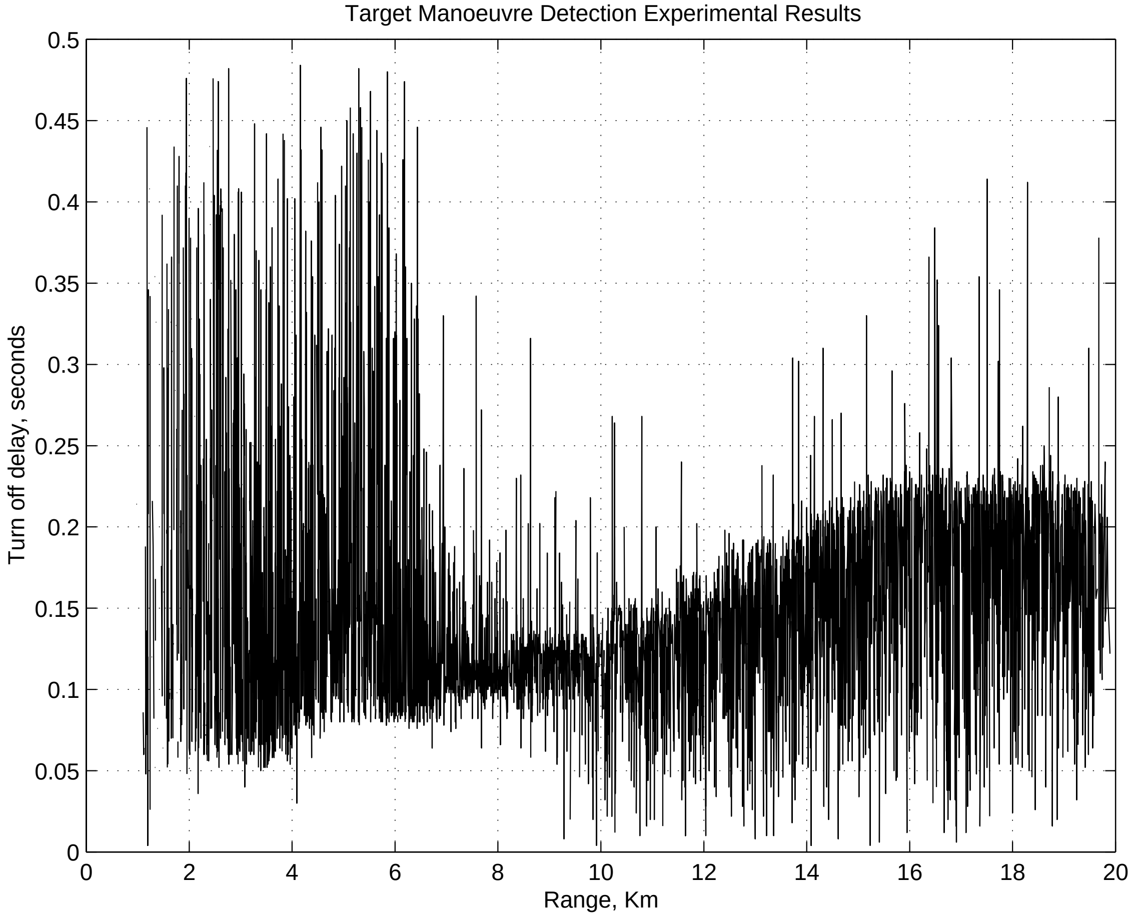


Figure 6.13: Turn-off delay with respect to range

Figure 6.14 shows the proportions of pre-detection, during-detection, and post-detection errors. The pre-detection errors are relatively few and occur mostly at long range. At long ranges, the turn on delay is short and so the detector responds to any isolated spikes. The during-detection errors reflect the shape of the range-glint distribution shown in figure 6.5. The errors are caused because the glint spikes occur mostly when the radar cross section fades. The increased frequency of glint spikes around 9km causes the mean of the sum channel signal to be artificially reduced. The mean, $\bar{s}$, is no longer a good estimate of the noise level, z. The use of $\bar{s}$ may be too crude for a more sophisticated detector.

The post detection errors are worst at low ranges where the glint effects of the manoeuvre seriously affect the tracking ability of the missile. At the end of the manoeuvre, the missile is unable to track the target and deviates away from the required collision course. This deviation appears as a manoeuvre and is consequently

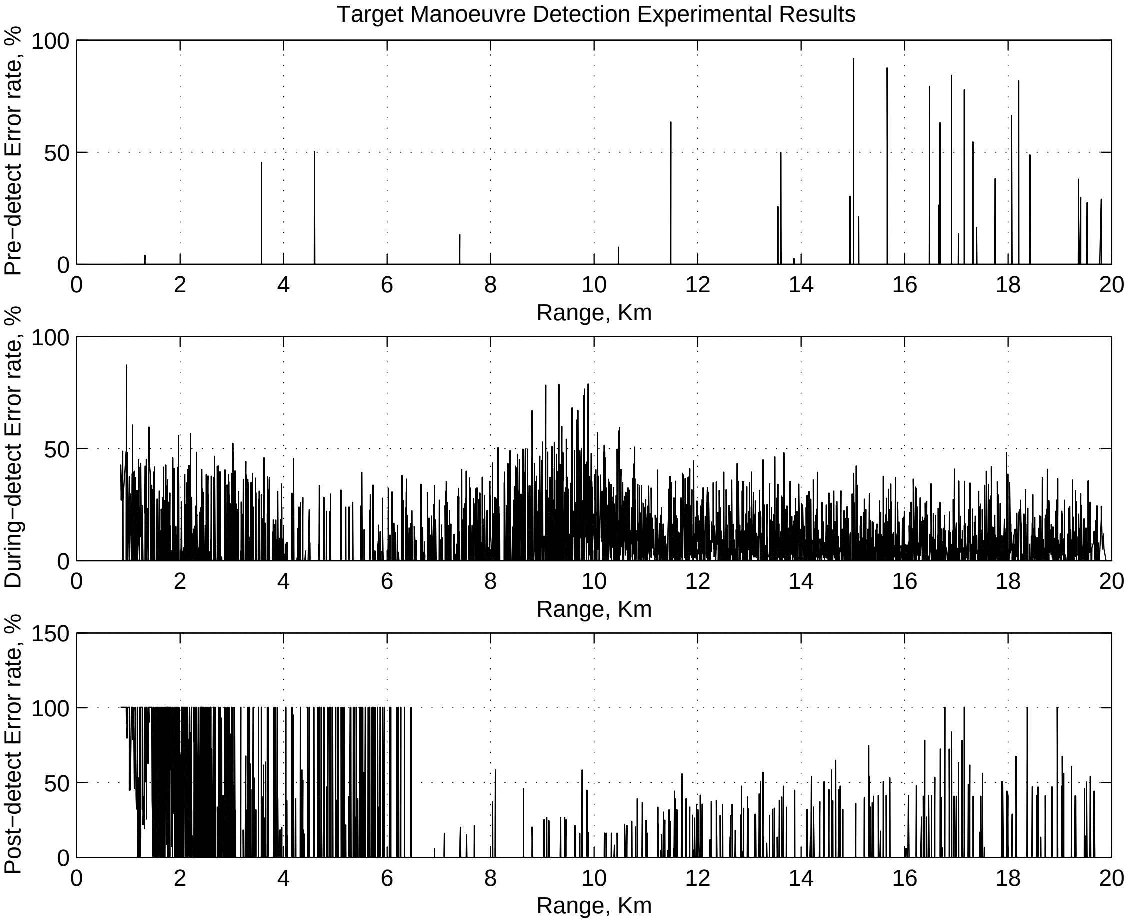


Figure 6.14: Detection errors with respect to range

detected. Results at very short ranges may be biased towards tail chase engagements as only complete manoeuvres are used. At short ranges, head-on engagements do not last long enough for the manoeuvre to be completed before impact.

There were only two engagements out of 5,000 where the manoeuvre detector failed to respond to the target manoeuvre. Figure 6.15 shows a histogram of the pre-detection and post-detection failures with respect to range. Each bar of the histogram represents the total number of failures per kilometre. The pre-detection failures occur most frequently at long range, although less than 0.3% of the trials were affected. The post-detection failures are mainly due to the missile loosing track of the target after the manoeuvre. The high concentration of failures at very short ranges and the peak around 10km are probably due to range-glint falsely triggering detection. From three to six kilometres, the missile is badly affected by the manoeuvre. Between six and nine kilometres, the missile performance is at its optimum and is often able to track the target after the manoeuvre. The good performance of the missile leads to a reduced failure rate in this region. Above eleven kilometres, the failures are mainly due to the missile loosing track, rather than to range-glint.

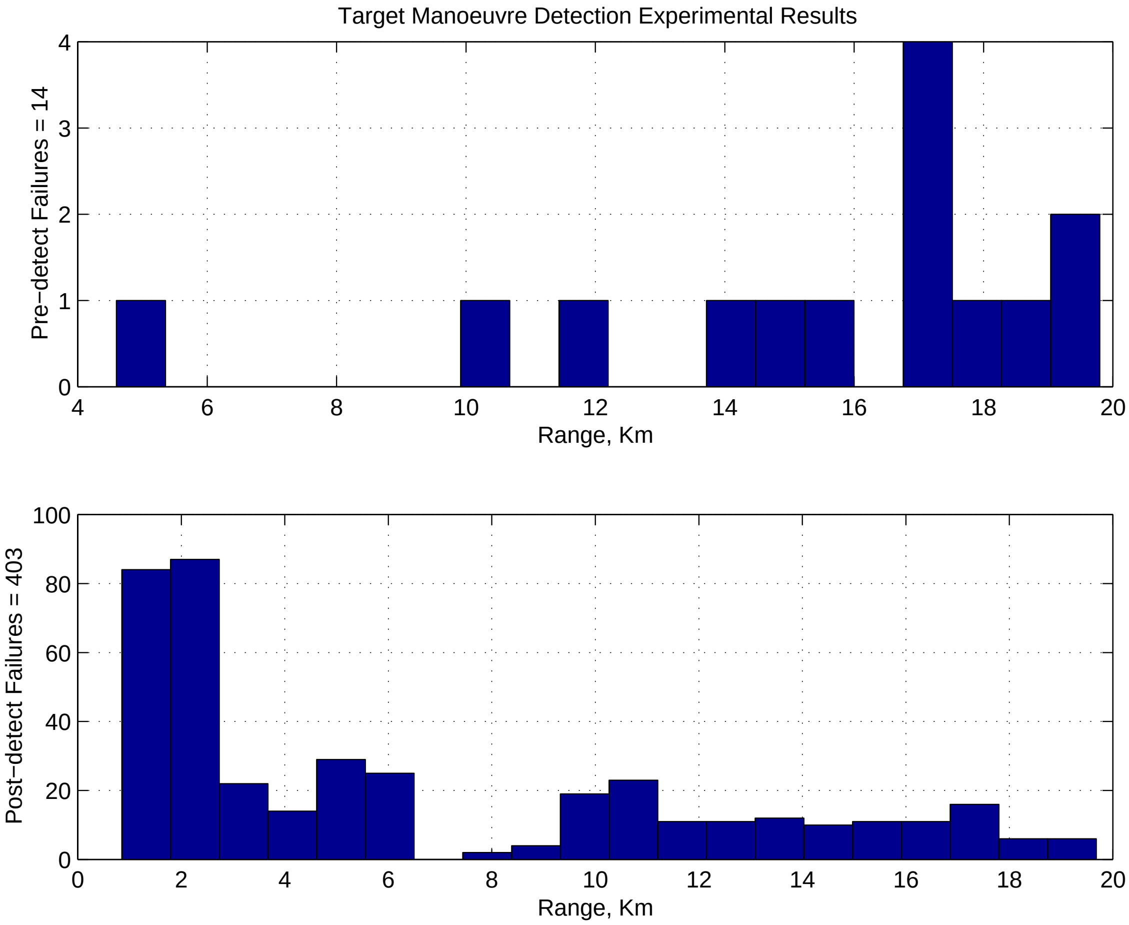


Figure 6.15: Detection failures with respect to range

6.8. Conclusions

The performance of the missile has had a significant effect on the trial results. Details of the manoeuvre termination performance are mainly related to the missile and not the manoeuvre detector, although the turn-off times reflect the shape of the fuzzy membership functions. If the missile were able to track the target well at all times, only the banking manoeuvres would probably be seen. The detector would therefore drop-out during the turn-only section of the manoeuvre. This manner of operation is still of great benefit in augmenting tracking algorithms. The short range results were biased towards tail chase engagements (first near head-on engagement occurred at 1.3km). The effects of range-glint in this region have far outweighed any influences of the biasing though.

The detector operated without error in over 91% of the trials. Despite the fuzzy detector only being very roughly tuned, the detector only failed totally for 0.04% of the trials. The average detection time of 96 milli-seconds and the high reliability make target manoeuvre detection using glint very attractive. The main strengths of the method are as follows:

- No *a-priori* target knowledge required.
- Fast response.
- Effective for long, medium and short range engagements.
- Very reliable.
- Low processing overhead.
- Uses existing sensors.
- Low cost.

6.9. Recommendations for Future Work

The fuzzy-logic proof-of-principle target manoeuvre detector showed that the technique is effective and very reliable. It is suggested that an extended Kalman filter or fuzzy-logic/artificial neural network solution is investigated. The noise characteristics developed for this study are only approximate. More rigorous analysis of the system noise characteristics should be performed. If a fuzzy-logic approach is used, tuning of the membership functions and the hold and delay times would be necessary. The tuning may be best accomplished with an evolutionary optimisation algorithm approach.

7. Conclusions

7.1. Introduction

This thesis has looked at the problems associated with generating realistic target models for simulating engagements with homing guidance missiles. A point-scatterer model approach has been taken to solve the modelling problems. The novel use of a binary space partition tree structure allows complex model structures to be created easily. The tree structure provides a framework that allows models of different sizes and complexities to form a single entity. The models are:

- **Fast to process** – Radar cross section data are simple to calculate from point scatterer models. The tree structure allows the correct scatterer model for the current viewing aspect to be retrieved quickly.
- **High fidelity** – Models can be constructed that produce an accurate representation of the source target's radar cross section pattern.
- **High resolution** – The point scatterer model will create realistic approximations of the radar cross section for any interpolated view aspects that were not in the original target data.
- **Correlated** – The radar cross section is properly correlated to range and the target motion.
- **Low storage requirements** – The point scatterer models are very compact and have low storage requirements.

A suite of genetic algorithms have been used to create realistic point scatterer models from ISAR data. The effects of reducing the fidelity of the target have been established. Limits on model reduction have been determined, allowing realistic target models to be generated that minimise simulation times.

An efficient, high fidelity model has been applied to trials of a novel target manoeuvre detection technique. A fuzzy-logic proof-of-principle model has been developed to test the theory that glint can be used to detect target manoeuvres. The results of the manoeuvre detector trials have been outstanding, showing that the use of target glint to detect manoeuvres is:

- Fast
- Operates over wide range envelope
- Very reliable
- Has a low processing overhead

- Uses existing sensors
- Cheap

The requirements for the thesis defined in section 1.4 (page 9) are discussed in the conclusions below.

7.2. Item 1 – Automating model conversion

The process of generating point scatterer models from ISAR images has been automated and the processing overhead reduced when compared to existing techniques. The conversion process may be broken into three elements. These elements have been addressed as follows:

- A novel multi-species genetic algorithm for locating multiple scatterers in each pass of the data. This algorithm dramatically reduces the processing overhead involved in generating ISAR images of the model during the conversion process.
- Genetic algorithm based fine-tune method to allow low, medium or high resolution images to be processed easily.
- Model reduction using a genetic algorithm to reduce the processing overhead involved in optimising the model structure. It allows the designer to trade between the model size and the fidelity of reproduction.

The process has been applied successfully to a real two-dimensional image and a simulated three-dimensional image.

7.3. Item 2 – Integration into engagement scenario

Two seeker models have been created for use with a synthetic missile in simulated engagements. These models have been successfully integrated into the object-oriented engagement model software (see [65] for code). Trials on the two seekers have established that the trial miss distance distribution for the amplitude comparison seeker is compact, while the the phase comparison seeker has a long-tailed distribution. The miss distance distributions appear to be influenced mostly by the effects of glint. The contribution of thermal noise to the miss distance is small.

7.4. Item 3 – Establishing model fidelity

The reduction factors that can be applied to models and still retain a high-fidelity representation have been established. It has been determined that reducing the model complexity to give a K–S significance of $\alpha = 0.9$ for the radar cross section pattern, will not significantly affect the performance of the missile against the target. The use of a significance level of $\alpha = 0.9$ for the image conversion process has been justified. A high-fidelity reduced model has been produced to allow the target manoeuvre detection trials to be performed with minimal computational overhead.

7.5. Item 4 – Target manoeuvre detection using glint

The characteristics of the bore-sight error signal have been established. A method for extracting target manoeuvre information from the patterns of glint spikes has been proposed. A manoeuvre detector based on the theory has been implemented. A fuzzy-logic approach was taken for its ease of application and simplicity. Trials of the fuzzy-logic manoeuvre detector have shown that even the crude, proof-of-principle model is capable of outstanding reliability and performance. The technique forms a very attractive solution to the problem of target manoeuvre detection in radar guided homing missiles.

7.6. Future Work

Important areas of work where further investigation is required have been identified.

- Research into methods for tuning the evolutionary algorithms used for model extraction. Some advances have been made towards optimally tuned PBIL algorithms but tuning genetic algorithms can be difficult.

- Further research into the effects of population size on the ability of evolutionary algorithms to find global optima. It has been established that for PBIL algorithms with high learning rates, the probability of finding the global optimum follows a binomial distribution. Genetic algorithms need to be investigated and the general case for PBIL needs to be established.

- Investigations into glint reduction and filtering techniques. Frequency and angular diversity methods could be combined with fuzzy-logic and artificial neural network techniques to provide an adaptive solution.

- Further research into target manoeuvre detection using target glint. The use of extended Kalman filters with fuzzy-logic/artificial neural networks for extracting manoeuvre information from the glint signal may prove fruitful. It may be possible to use target manoeuvre information to augment the glint filtering process.

- Research into data fusion from multiple sensors for target manoeuvre detection. Manoeuvre detection using glint is practical for medium and long range, but is more difficult at short ranges. Conversely, optical detectors perform best at short and medium ranges. A hybrid system would perform better in a wider range of engagement scenarios and may be able to improve performance in the presence of countermeasures such as flares or chaff.

7.7. Acknowledgements

I would like to thank Dr J. Chadwick of the DERA, Malvern, UK (agency of UK MoD) for supplying the measured radar cross section data. I would also like to thank my wife, Jo, for her support during the work and for proof reading this thesis.

References

[1] James R. Cloutier, Johnny H. Evers, and Joseph J. Feeley. Assessment of air-to-air missile guidance and control technology. *IEEE Control Systems Magazine*, pages 27–34, October 1989.

[2] J. B. Romine and E. W. Kamen. Target manoeuvre detection using image features. *Proceedings of the SPIE - The International Society for Optical Engineering*, 2561:420–434, 1995.

[3] Y. T. Chan and F. Couture. Manoeuvre detection and track correction by input estimation. *IEE Proceedings–F*, 140(1):21–28, February 1993.

[4] I. Yaesh and J. Z. Ben Asher. Optimum guidance with a single uncertain time lag. *Journal of Guidance, Control and Dynamics*, 18(5):981–988, September–October 1995.

[5] Ronald G. Cottrell. Minimizing interceptor size using neural networks for terminal guidance law synthesis. *Journal of Guidance, Control and Dynamics*, 19(3):557–562, May–June 1996.

[6] S. K. Mishra, I. G. Sarma, and K. N. Swamy. Performance evaluation of two fuzzy-logic-based homing guidance schemes. *Journal of Guidance, Control and Dynamics*, 17(6):1389–91, November–December 1994.

[7] Fumiaki Imado. Some aspects of a realistic three-dimensional pursuit-evasion game. *Journal of Guidance Control and Dynamics*, 16(2):289–293, March–April 1993.

[8] Hongren Zhou and K. S. P. Kumar. A "current" statistical model and adaptive algorithm for estimating maneuvering targets. *Journal of Guidance, Control and Dynamics*, 7(5):596–602, September–October 1984.

[9] Eugene F. Knott, John F. Shaeffer, and Michael T. Tuley. *Radar Cross Section.* Artech House, 1985.

[10] T. T. Moon and P. J. Bawden. High resolution RCS measurements of boats. *IEE Proceedings-F*, 138(3):218–222, June 1991.

[11] Atul Jain and Indu Patel. Dynamic imaging and RCS measurements of aircraft. *IEEE Transactions on Aerospace and Electronic Systems*, 31(1):211–226, January 1995.

[12] Bernard D. Steinberg, Donald L. Carlson, and Woosung Lee. Experimental localized radar cross sections of aircraft. *Proceedings of the IEEE*, 77(5):663–669, May 1989.

[13] Robert R. Dybdal. Radar cross section measurements. *Proceedings of the IEEE*, 75(4):498–515, April 1987.

[14] Jihad S. Daba and Mark R. Bell. Statistics of the scattering cross-section of a small number of random scatterers. *IEEE Transactions on Antennas and Propagation*, 43(8):773–783, August 1995.

[15] Donald R. Wehner. *HIGH RESOLUTION RADAR*. Artech House, 1987.

[16] Elizabeth C. Botha, Etienne Barnard, and Charl J. Barnard. Feature-based classification of aerospace radar targets using neural networks. *Neural Networks*, 9(1):129–142, January 1996.

[17] Nabil H. Farhat. Microwave diversity imaging and automated target identification based on models of neural networks. *Proceedings of the IEEE*, 77(5):670–681, May 1989.

[18] G. S. Sandhu and A. V. Saylor. A real-time statistical radar target model. *IEEE Transactions on Aerospace and Electronic Systems*, 21(4):490–507, July 1985.

[19] Bert-Eric Tullsson. Monopulse tracking of Raleigh targets: A simple approach. *IEEE Transactions on Aerospace and Electronic Systems*, 27(3):520–531, May 1991.

[20] William B. Gordon. Statistical moments of radar cross section. *IEEE Transactions on Antennas and Propagation*, 41(4):506–508, April 1993.

[21] Brett H. Borden and Michael L. Mumford. A statistical glint/radar cross section target model. *IEEE Transactions on Aerospace and Electronic Systems*, 19(5):781–785, September 1983.

[22] Allen E. Fuhs. *The no-See-Um Book, RADAR CROSS SECTION LECTURES*. American Institute of Aeronautics and Astronautics, 1985.

[23] B. Haywood, W. C. Anderson, J. T. Morris, and R. Kyprianou. Generation of point scatterer models for simulating ISAR images of ships. In *RADAR '97 Conference*, pages 700–704, Edinburgh, UK, 14–16 October 1997. IEE Pub. No. 449.

[24] John S. Asvestas. The physical-optics integral and computer graphics. *IEEE Transactions on Antennas and Propagation*, 43(12):1459–1460, December 1995.

[25] Juan M. Rius, Miguel Ferrando, and Luis Jofre. High-frequency RCS of complex radar targets in real-time. *IEEE Transactions on Antennas and Propagation*, 41(9):1308–1319, September 1993.

[26] Nazih N. Youssef. Radar cross section of complex targets. *Proceedings of the IEEE*, 77(5):722–734, May 1989.

[27] P. A. Lees and M. R. Davies. Computer prediction of RCS for military targets. *IEE Proceedings, Part F*, 137(4):229–236, August 1990.

[28] S. D. Turner. RESPECT: Rapid electromagnetic scattering predictor for extremely complex targets. *IEE Proceedings, Part F*, 137(4):214–220, August 1990.

[29] K. Emir and E. Topuz. Simulation of ISAR images of ships for localization of dominant scatterers. In *RADAR '97 Conference*, pages 273–275, Edinburgh, UK, 14–16 October 1997. IEE Pub. No. 449.

[30] M. Domingo, F. Rivas, J. Pérez, R. P. Torres, and M. F. Cátedra. Computation of the RCS of complex bodies modeled using NURBS surfaces. *IEEE Antennas and Propagation Magazine*, 37(6):36–47, December 1995.

[31] D. M. Elking, J. M. Roedder, D. D. Car, and S. D. Alspach. A review of high-frequency radar cross section analysis capabilities at McDonnell Douglas Aerospace. *IEEE Antennas and Propagation Magazine*, 37(5):33–47, October 1995.

[32] D. J. Andersh, M. Hazlett, S. W. Lee, D. D. Reeves, D. P. Sullivan, and Y. Chu. XPATCH: A high-frequency electromagnetic-scattering prediction code and environment for complex three-dimensional objects. *IEEE Antennas and Propagation Magazine*, 36(1):65–69, February 1994.

[33] Edward H. Newman and Ronald J. Marhefka. Overview of MM and UTD methods at the Ohio state university. *Proceedings of the IEEE*, 77(5):700–708, May 1985.

[34] Hyeongdong Kim and Hao Ling. Electromagnetic scattering from an inhomogeneous object by ray tracing. *IEEE Transactions on Antennas and Propagation*, 40(5):517–525, May 1992.

[35] Rajan Bhalla and Hao Ling. A fast algorithm for signature prediction and image formation using the shooting and bouncing ray technique. *IEEE Transactions on Antennas and Propagation*, 43(7):727–731, July 1995.

[36] Vijaya Shankar, William F. Hall, and Alireza H. Mohammadian. A time-domain differential solver for electromagnetic scattering problems. *Proceedings of the IEEE*, 77(5):709–721, May 1989.

[37] Allen Taflove and Korada R. Umashankar. Review of FD–TD numerical modeling of electromagnetic wave scattering and radar cross section. *Proceedings of the IEEE*, 77(5):682–699, May 1989.

[38] Richard Holland. Two-pass finite-difference time-domain (FDTD) calculations on a fighter aircraft. *IEEE Transactions on Antennas and Propagation*, 44(5):659–664, May 1996.

[39] M. S. Towers, A. McCowen, and J. A. R. McNab. Electromagnetic scattering from an arbitrary, inhomogeneous 2-D object — a finite and infinite element solution. *IEEE Transactions on Antennas and Propagation*, 41(6):770–777, June 1993.

[40] J. M. Williams. Radar cross-section of large structures with complex microgeometry. *IEE Proceedings, Part F*, 137(4):221–228, August 1990.

[41] J. Li, Z.Bi, Z.-S. Liu, and K. Knaell. Use of curvilinear SAR for three-dimensional target feature extraction. *IEE Proceedings on Radar, Sonar and Navigation*, 144(5):275–283, October 1997.

[42] C. L. Breuille and E. Caille. Modeles multi-points brillants de signatures radar de cibles. In *Radar International Conference*, pages 325–329, Paris, France, 3–6 May 1994. Soc. Electr. & Electron.

[43] Qing Li, Edward J. Rothwell, Kun-Mu Chen, and Dennis P. Nyquist. Scattering center analysis of radar targets using fitting scheme and genetic algorithm. *IEEE Transactions on Antennas and Propagation*, 44(2):198–207, February 1996.

[44] Michael P. Hurst and Raj Mittra. Scattering center analysis via Prony's method. *IEEE Transactions on Antennas and Propagation*, 35(8):986–988, August 1987.

[45] Rob Carrière and Randolph L. Moses. High resolution radar target modeling using a modified Prony estimator. *IEEE Transactions on Antennas and Propagation*, 40(1):13–18, January 1992.

[46] Rajan Bhalla and Hao Ling. Three-dimensional scattering centre extraction using the shooting and bouncing ray technique. *IEEE Transactions on Antennas and Propagation*, 44(11):1445–1453, November 1996.

[47] S. Y. Wang and S. K. Jeng. Generation of point scatterer models using PTD/SBR technique. In *IEEE Antennas and Propagation Symposium*, pages 1914–1917, Newport Beach, CA., June 1995. IEEE.

[48] A. M. S. Zalzala and P. J. Flemming, editors. *Genetic algorithms in engineering systems.* The Institution of Electrical Engineers, 1997.

[49] J. Michael Johnson and Yahya Rahmat-Samii. Genetic algorithms in engineering electromagnetics. *IEEE Antennas and Propagation Magazine*, 39(4):7–25, August 1997.

[50] M. J. Byrd, B. D. Jersak, B. D. Krenek, and A. J. Blanchard. Demonstration of 3D microwave holography. In *IGARSS '94 International Geoscience and Remote Sensing Symposium*, pages 918–920, Pasadena, CA, USA, 8–12 August 1994. IEEE.

[51] David E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley Publishing Company, Inc., 1989.

[52] Zbigniew Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs.* Springer–Verlag, 2nd edition, 1992.

[53] D. E. Goldberg and J. Richardson. Genetic algorithms with sharing for multimodal function optimization. In *Second International Conference on Genetic Algorithms: Genetic Algorithms and Their Applications*, pages 41–49, 1987.

[54] S. W. Mahfoud. A comparison of parallel and sequential niching methods. In *Sixth International Conference on Genetic Algorithms*, pages 136–143, San Mateo, CA, USA, 1995. Morgan Kaufmann.

[55] David Beasley, David R. Bull, and Ralph R. Martin. A sequential niche technique for multimodal function optimization. *Evolutionary Computation*, 1(2):101–125, 1993.

[56] E. J. Hughes, M. Leyland, and B. A. White. A multi-species genetic algorithm applied to radar scattering centre identification in three-dimensions. In *GALESIA '97 Conference*, pages 472–477, Glasgow, UK, 1–4 September 1997. IEE Pub. No. 446.

[57] August W. Rihaczek. Radar resolution of ideal point scatterers. *IEEE Transactions on Aerospace and Electronic Systems*, 32(2):842–845, April 1996.

[58] N. Srinivas and Kalyanmoy Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3):221–248, 1995.

[59] Rajeev Kumar and Peter Rockett. Assessing the convergence of rank-based multiobjective genetic algorithms. In *Genetic Algorithms in Engineering Systems: Innovations and Applications. (GALESIA '97)*, pages 19–23, Glasgow, 2–4 September 1997. IEE Conference Publication No. 446.

[60] Evan James Hughes. Transfer report: Radar cross section modelling. Royal Military College of Science, Cranfield University, February 1997.

[61] E. J. Hughes and M. Leyland. Radar cross section model optimisation using genetic algorithms. In *RADAR '97 Conference*, pages 458–462, Edinburgh, UK, 14–16 October 1997. IEE Pub. No. 449.

[62] James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. *Computer Graphics: Principles and Practice.* Addison-Wesley, 2nd edition, 1990.

[63] John C. Holmes. Binary space partitioning (BSP) trees. http://www.cis.ufl.edu /∼jch/is-final.ps, 1995. Last accessed 24 April 1998.

[64] Samuel M. Sherman. *Monopulse Principles and Techniques.* Artech House, 1984.

[65] Evan James Hughes and Paul Creaser. Engagement model software defining specification. Technical Report DAPS/EJH/17/97, Royal Military College of Science, Cranfield University, 1997.

[66] Paul Creaser. Transfer report. Royal Military College of Science, Cranfield University, March 1997.

[67] Y. T. Chan and F. Couture. Maneuvering detection with input estimation. In *EUSIPCO 90: 5th European Signal Processing Conference*, volume 1, pages 309–312, Barcelona, Spain, 18–21 September 1990. Elsevier.

[68] Francois Dufour and Michel Mariton. Tracking a 3D maneuvering target with passive sensors. *IEEE Transactions on Aerospace and Electronic Systems*, 27(4):725–739, July 1991.

[69] J. D. Kendrick, P. S. Maybeck, and J. G. Reid. Estimation of aircraft target motion using orientation measurements. *IEEE Transactions on Aerospace and Electronic Systems*, 17(2):254–260, March 1981.

[70] D. D. Sworder and R. G. Hutchins. Maneuver estimation using measurements of orientation. *IEEE Transactions on Aerospace and Electronic Systems*, 26(4):625–638, July 1990.

[71] Shreenath Shetty and Ali T. Alouani. Image-sensor-based target maneuver detection. *Optical Engineering*, 32(11):2735–2740, November 1993.

[72] D. Laneuville and M. Mariton. Image based target maneuver detection. In *Proceedings of the 30th IEEE conference on Decision and Control*, volume 2, pages 2066–2067, Brighton, UK, 11-13 December 1991. IEEE.

[73] D. D. Sworder and R. G. Hutchins. Image-enhanced tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 25(5):701–710, September 1989.

[74] G. A. Hewer, R. D. Martin, and J. Zeh. Robust preprocessing for Kalman filtering of glint noise. *IEEE Transactions on Aerospace and Electronic Systems*, 23(1):120–128, January 1987.

[75] Wen-Rong Wu and Peen-Pau Cheng. A nonlinear IMM algorithm for maneuvering target tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 30(3):875–885, July 1994.

[76] Wen-Rong Wu. Target tracking with glint noise. *IEEE Transactions on Aerospace and Electronic Systems*, 29(1):174–185, January 1993.

[77] R. Jitendra Das and Y. Yoganandam. Performance analysis of a noncoherent monopulse processor in the presence of target glint and receiver imbalances. *IEE Proceedings, Part F*, 138(5):445–452, October 1991.

[78] R. Jitendra Das and Y. Yoganandam. Effect of system noise on the performance of a noncoherent monopulse processor. *IEE Proceedings, Part F*, 139(1):61–66, February 1992.

[79] Brett H. Borden. Diversity methods in phase monopulse tracking — a new approach. *IEEE Transactions on Aerospace and Electronic Systems*, 27(6):877–880, November 1991.

[80] Brett H. Borden. Requirements for optimal glint reduction by diversity methods. *IEEE Transactions on Aerospace and Electronic Systems*, 30(4):1108–1114, October 1994.

[81] Ian W. Guest and Christoff K. Pauw. Rank detector preprocessor for glint reduction in a tracking radar. *IEEE Transactions on Aerospace and Electronic Systems*, 29(2):527–531, April 1993.

[82] Ned Gulley and J.-S. Roger Jang. *Fuzzy Logic Toolbox Users Guide.* The MathWorks, Inc., 24 Prime Park Way, Natick, Mass., USA., January 1995.

[83] Edward Pervin and Jon A. Webb. Quaternions in computer vision and robotics. ftp://ftp.netcom.com/pub /hb/hbaker/quaternion/cmu-cs-82-150.ps.gz, 1982. Last accessed 24 April 1998.

[84] P. M. Morse and H. Feshbach. *Methods of Theoretical Physics: Part 1*, chapter 1, pages 73–76. McGraw-Hill, 1953.

[85] S. B. Bell and D. C. Mason. Tesseral quaternions for the octree. *The Computer Journal*, 33(5):386–397, 1990.

[86] Joseph R. Mautz and Roger F. Harrington. Computational methods for antenna pattern synthesis. *IEEE Transactions on Antennas and Propagation*, 23(7):507–512, July 1975.

[87] S. Barnett and T. M. Cronin. *Mathematical Formulae.* Longman Scientific & Technical, Harlow, England, fourth edition, 1988.

[88] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *NUMERICAL RECIPES The Art Of Scientific Computing (FORTRAN Version).* Cambridge University Press, 1989.

[89] S. Baluja. Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning. Technical Report CMU-CS-95-163, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA, 1994.

[90] Ignatious Thithi. Control system parameter identification using the population based incremental learning (PBIL). In *IEE Conference Publication No. 427, UKACC International Conference on Control (CONTROL '96)*, pages 1309–1314, London, 2–5 September 1996. IEE.

[91] J. R. Greene. A role for simple, robust 'black-box' optimisers in the evolution of engineering systems and artifacts. In *Genetic Algorithms in Engineering Systems: Innovations and Applications. (GALESIA '97)*, pages 427–432, Glasgow, 2–4 September 1997. IEE Conference Publication No. 446.

[92] Markus Höhfeld and Günter Rudolph. Towards a theory of population-based incremental learning. In *1997 IEEE International Conference on Evolutionary Computation*, Ch. 127, pages 1–5, Indianapolis, 13–16 April 1997. IEEE.

[93] Erwin Kreyzig. *Advanced Engineering Mathematics.* John Wiley & Sons, Inc., sixth edition, 1988.

A. Quaternions

A.1. Introduction

Quaternions were discovered by Sir William Rowan Hamilton in October 1843. The quaternion is a four dimensional complex number, where the complex part is defined by

$$\mathbf{i^2 = j^2 = k^2 = ijk = -1}$$

Dot and cross products of vectors were discovered as part of the quaternion product.

The quaternion is an extension of the usual complex form $a+\mathbf{j}b$, where $j = \sqrt{-1}$,

$$\mathcal{Q} = a + \mathbf{i}b + \mathbf{j}c + \mathbf{k}d$$

where $\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = -1$ and $\mathbf{ij} = -\mathbf{ji} = \mathbf{k}$, $\mathbf{jk} = -\mathbf{kj} = \mathbf{i}$ and $\mathbf{ki} = -\mathbf{ik} = \mathbf{j}$ and a, b, c and d are real. This form of representation was referred to by Hamilton as Standard Quadrinormal Form [83].

The quaternion may also be represented in a form analogous to Euler's Equation for the imaginary exponential [84]

$$me^{\phi(\mathbf{i}\alpha+\mathbf{j}\beta+\mathbf{k}\gamma)}$$

where m is the magnitude of the quaternion, ϕ is a rotation angle and $[\ \alpha \quad \beta \quad \gamma\ ]$ is a unit vector defining an axis of rotation, ie. $\alpha^2 + \beta^2 + \gamma^2 = 1$. The Euler form of the equation may also be represented by

$$m(\cos(\phi) + (\mathbf{i}\alpha + \mathbf{j}\beta + \mathbf{k}\gamma)\sin(\phi))$$

Quaternions are normally represented as four component vectors,

$$[\ a \quad [\ b \quad c \quad d\ ]\ ]$$

where a represents the real part of the quaternion and $[\ b \quad c \quad d\ ]$ the imaginary parts. The components are often expressed in a vector form,

$$\tilde{q} = [a, \overline{b}]$$

where $\tilde{q}$ is the notation for a quaternion[1] and $\overline{b}$ represents the imaginary part of the quaternion, expressed as a vector.

[1] $\overline{q}$ and $\dot{q}$ are sometimes used in other papers.

A.2. Arithmetic

The following rules apply to quaternion arithmetic for the set of all quaternions, $\mathcal{Q}$ [83]:

1. Addition:

   1.1. Closure: if $P, Q \in \mathcal{Q}$ then $P + Q \in \mathcal{Q}$

   1.2. Commutativity: $P + Q = Q + P$ for all $P, Q \in \mathcal{Q}$

   1.3. Associativity: $(P + Q) + R = P + (Q + R)$ for all $P, Q, R \in \mathcal{Q}$

   1.4. Identity: There is a $0 \in \mathcal{Q}$ such that $0 + P = P + 0 = P$

   1.5. Inverse: For any $P \in \mathcal{Q}$ there exists a $(-P) \in \mathcal{Q}$ such that $P + (-P) = (-P) + P = 0$

2. Multiplication:

   2.1. Closure: If $P, Q \in \mathcal{Q}$ then $PQ \in \mathcal{Q}$

   2.2. Associativity: $(PQ)R = P(QR)$ for all $P, Q, R \in \mathcal{Q}$

   2.3. Identity: There is a $1 \in \mathcal{Q}$ such that $1P = P1 = P$

   2.4. Inverse: If $P \neq 0$, then there is a P^{-1} such that $PP^{-1} = P^{-1}P = 1$

3. Distributivity:

$$P(Q + R) = PQ + PR \text{ and } (Q + R)P = QP + RP \text{ for each } P, Q, R \in \mathcal{Q}$$

4. No zero divisors: If $PQ = 0$, then either $P = 0$ or $Q = 0$.

A.3. Magnitude, Conjugate and Inverse

If we have the quaternion

$$\tilde{q} = [a, \overline{b}]$$

then its magnitude is given by

$$m = |\tilde{q}| = \sqrt{a^2 + \overline{b} \cdot \overline{b}} = \sqrt{\tilde{q} \cdot \tilde{q}}$$

The conjugate is the original quaternion but with its vector part negated,

$$\tilde{q}^* = [a, -\overline{b}]$$

and the inverse is the conjugate divided by the magnitude squared

$$\tilde{q}^{-1} = \frac{\tilde{q}^*}{m^2} = \frac{\tilde{q}^*}{\tilde{q} \cdot \tilde{q}}$$

A.4. Operators

The quaternion dot product may be defined for two quaternions $\tilde{q} = [w, (x, y, z)]$ and $\tilde{p} = [a, (b, c, d)]$ as

$$\tilde{q} \cdot \tilde{p} = wa + xb + yc + zd$$

The quaternion product of two quaternions $\tilde{p} = [p_o, \overline{p}]$ and $\tilde{q} = [q_o, \overline{q}]$ may be defined by

$$\tilde{p}\tilde{q} = [p_o q_o - \overline{p} \cdot \overline{q}\ ,\ \overline{p_o \overline{q} + \overline{p} q_o + \overline{p} \times \overline{q}}]$$

where '·' and '×' imply the vector dot and cross product operations respectively.

A.5. Rotations

Quaternion rotation of vectors is described by the transformation

$$[0, (x', y', z')] = \tilde{q}[0, (x, y, z)]\tilde{q}^*$$

where (x, y, z) is the vector to be rotated, (x', y', z') is the rotated vector and

$$\tilde{q} = [\cos(\theta/2), [\alpha, \beta, \gamma] \sin(\theta/2)]$$

with $|\tilde{q}| = 1$ and θ defining the required rotation angle and $[\alpha, \beta, \gamma]\,|_{\alpha^2+\beta^2+\gamma^2=1}$ defining the axis to rotate around. The vector (x, y, z) is effectively turned into a quaternion with a zero scalar part to allow the rotation to be evaluated. The resulting quaternion will also have a zero scalar part.

Alternatively, a unit quaternion $[w, (x, y, z)]$ may be converted into matrix form

$$M = \begin{pmatrix} 1 - 2y^2 - 2z^2 & 2xy + 2wz & 2xz - 2wy \\ 2xy - 2wz & 1 - 2x^2 - 2z^2 & 2yz + 2wx \\ 2xz + 2wy & 2yz - 2wx & 1 - 2x^2 - 2y^2 \end{pmatrix}$$

If $\tilde{q} = me^{\theta(\mathbf{i}\alpha + \mathbf{j}\beta + \mathbf{k}\gamma)}$, then the quaternion product $\sqrt{\tilde{q}}[0, (x, y, z)]\{\sqrt{\tilde{q}}\}^*$ will rotate by an angle θ about unit axis (α, β, γ) and scale by a factor m.[85]

B. Truth Models

This appendix describes the single-point, 50-point and 100-point truth models. The high-fidelity 27-point model that was derived from the 50-point model is also detailed. The models are designed to be used within a suite of MATLAB programmes for radar cross section simulation. The models are held as $N \times 6$ arrays, where each row defines one of N scatterers. The format for each row is as follows

$$\left[\begin{array}{cccccc} x & y & z & \text{Magnitude} & \text{Phase} & \rho \end{array}\right]$$

where x, y and z define the scatterer position and ρ is defined by

$$\rho = \begin{cases} 1 & \text{, scatterer active} \\ 2 & \text{, scatterer passive} \end{cases}$$

The returned echo voltage, ξ, from an n–point scattering centre model may be found using equation B.1.

$$\xi = \sum_{p=1}^{n} \frac{a_p\sqrt{\mathcal{S}}}{(\sqrt{4\pi}d_p)^{\rho_p}}\, e^{j\left(\frac{2\pi\rho_p d_p}{\lambda}+\phi_p\right)} \tag{B.1}$$

> Where a_p is the magnitude of scatterer p, $\mathcal{S}$ is the radar source power, d_p is the distance from the view point to scatter p, λ is the wavelength and ϕ_p is the phase of scatterer p relative to the radar.

Table B.1 defines the one point truth model. Distance is in metres, magnitude is $\sqrt{\sigma}$ in metres and phase in radians. Tables B.2 and B.3 defines the 50-point truth model, and tables B.4, B.5 and B.6 define the 100-point truth model. Table B.7 defines the twenty seven point model that was used for the target manoeuvre detection trials in chapter 6. The model was created by the fidelity reduction trials on the 50-point model detailed in chapter 5.

x	y	z	Magnitude	Phase	Ac./Pa.
0	0	0	29.77	0	2

Table B.1: 1-point truth model

Scatterer	x	y	z	Magnitude	Phase	Ac./Pa.
1	10.00	0.00	0.00	1.75	0	2
2	4.80	3.15	0.20	1.94	0	2
3	2.50	9.50	0.60	3.74	0	2
4	-1.50	1.00	-0.11	4.72	0	2
5	-1.50	-1.00	-0.10	4.28	0	2
6	2.50	-9.55	0.60	4.09	0	2
7	4.80	-3.20	0.20	1.93	0	2
8	-6.40	0.00	3.60	8.54	0	2
9	-9.00	0.10	0.00	6.77	0	2
10	-6.00	-0.54	-0.10	1.88	0	2
11	10.80	-0.27	-0.57	3.08	0	2
12	4.65	2.66	0.56	6.34	0	2
13	1.78	8.77	1.42	4.22	0	2
14	-0.61	1.57	-0.61	2.93	0	2
15	-1.68	-1.09	0.62	5.50	0	2
16	1.76	-9.85	0.54	4.59	0	2
17	5.57	-3.30	0.21	4.68	0	2
18	-7.22	0.62	3.80	6.58	0	2
19	-9.68	0.96	0.64	6.36	0	2
20	-6.86	-0.24	0.41	5.68	0	2
21	9.58	0.43	-0.02	6.60	0	2
22	4.87	3.75	0.54	1.44	0	2
23	2.53	9.91	0.96	2.97	0	2
24	-2.29	1.48	-0.71	2.10	0	2
25	-1.67	-1.96	0.73	3.43	0	2
26	2.65	-8.78	1.33	5.88	0	2
27	5.55	-3.15	0.98	2.93	0	2
28	-6.52	-0.07	3.69	3.12	0	2
29	-8.54	-0.77	-0.72	5.38	0	2
30	-5.26	-0.11	-0.20	2.43	0	2
31	9.42	-0.16	0.26	1.89	0	2
32	5.80	3.51	-0.37	5.18	0	2
33	1.81	9.86	0.03	5.23	0	2
34	-1.24	0.41	-0.95	2.82	0	2
35	-1.27	-0.33	-0.32	1.88	0	2
36	1.50	-9.13	1.50	1.29	0	2
37	3.80	-2.54	1.10	0.08	0	2
38	-5.85	-0.81	3.38	1.28	0	2
39	-8.55	-0.74	-0.46	6.55	0	2
40	-6.36	-0.01	0.28	1.63	0	2

Table B.2: 50-point truth model

Scatterer	x	y	z	Magnitude	Phase	Ac./Pa.
41	10.64	-0.80	0.20	4.69	0	2
42	4.07	2.59	-0.35	0.95	0	2
43	2.30	9.77	0.24	1.08	0	2
44	-1.30	1.39	0.29	3.24	0	2
45	-2.15	-0.41	-0.87	5.73	0	2
46	3.16	-9.16	1.13	5.42	0	2
47	4.12	-2.69	0.25	3.71	0	2
48	-5.42	0.34	3.71	4.93	0	2
49	-9.49	0.37	0.18	2.11	0	2
50	-6.53	-1.43	-0.44	0.90	0	2

Table B.3: 50-point truth model

Scatterer	x	y	z	Magnitude	Phase	Ac./Pa.
1	4.05	-0.57	3.21	3.10	0	2
2	1.63	-0.60	1.75	2.53	0	2
3	4.57	1.27	-5.01	1.93	0	2
4	-3.41	-4.48	3.00	0.08	0	2
5	-4.53	1.93	10.50	0.60	0	2
6	1.37	2.04	3.34	0.95	0	2
7	0.20	-3.42	3.43	1.76	0	2
8	-3.81	0.14	-1.10	0.33	0	2
9	-0.27	-1.43	7.93	0.57	0	2
10	-6.74	3.40	-3.18	0.13	0	2
11	2.71	0.54	7.10	0.27	0	2
12	1.02	3.04	1.43	0.44	0	2
13	2.66	-5.91	4.36	1.94	0	2
14	4.70	0.88	-5.01	0.15	0	2
15	-2.65	-4.82	0.57	1.06	0	2
16	-1.81	0.83	-3.03	2.26	0	2
17	-0.39	-2.72	0.21	1.77	0	2
18	2.75	2.55	-1.34	1.22	0	2
19	4.26	-3.23	5.90	5.07	0	2
20	1.57	0.32	2.47	0.12	0	2
21	-0.22	0.59	10.54	0.79	0	2
22	4.60	-0.92	4.99	0.54	0	2
23	-1.61	6.72	1.32	1.09	0	2
24	5.71	4.95	0.40	1.55	0	2
25	-4.73	-2.65	-8.40	0.75	0	2
26	-0.94	-7.97	4.58	3.66	0	2
27	-1.61	-8.95	-0.58	0.34	0	2
28	-9.24	-3.44	-0.59	1.27	0	2
29	-3.02	-8.51	1.13	2.36	0	2
30	-0.56	-0.98	0.04	1.00	0	2
31	-1.85	2.60	2.62	2.35	0	2
32	3.07	5.93	-11.06	2.49	0	2
33	-2.11	2.28	-3.68	1.26	0	2
34	2.56	-2.69	-1.60	3.50	0	2
35	-1.67	-5.51	4.96	0.09	0	2
36	-4.40	4.88	1.97	1.54	0	2
37	4.49	-0.69	8.23	4.47	0	2
38	1.58	11.24	-6.12	0.19	0	2
39	-4.06	-0.26	1.59	2.28	0	2

Table B.4: 100-point truth model

Scatterer	x	y	z	Magnitude	Phase	Ac./Pa.
40	1.03	-2.29	-1.36	1.30	0	2
41	-0.14	-2.15	-9.15	1.31	0	2
42	-1.60	5.76	1.23	1.16	0	2
43	-2.44	-2.76	6.23	0.56	0	2
44	2.04	1.27	-2.64	3.89	0	2
45	4.10	9.34	0.50	2.66	0	2
46	-1.40	2.40	-1.73	1.53	0	2
47	5.66	0.97	3.03	2.19	0	2
48	1.29	4.79	8.13	0.42	0	2
49	4.08	3.64	-6.02	0.54	0	2
50	-2.11	-0.65	2.71	1.03	0	2
51	2.18	1.78	-2.61	2.08	0	2
52	-7.22	-0.90	5.27	1.38	0	2
53	8.78	-0.32	8.10	2.25	0	2
54	-0.83	-3.27	-1.13	2.78	0	2
55	2.86	-0.57	-9.44	0.17	0	2
56	5.01	-0.84	-2.73	1.07	0	2
57	3.09	3.24	7.39	2.03	0	2
58	-4.48	-0.76	-1.31	0.50	0	2
59	-1.09	-0.19	0.45	1.07	0	2
60	-1.16	5.53	-0.84	2.03	0	2
61	1.54	3.88	1.16	0.46	0	2
62	0.62	-4.25	-4.20	1.20	0	2
63	0.32	9.61	2.04	3.73	0	2
64	7.83	6.59	3.10	3.84	0	2
65	7.84	-4.16	-4.38	2.43	0	2
66	8.97	0.79	0.01	2.58	0	2
67	4.55	1.79	-0.96	2.36	0	2
68	-3.56	0.35	-2.37	0.14	0	2
69	-7.00	1.59	9.26	0.47	0	2
70	-3.04	-2.70	-0.90	1.69	0	2
71	2.83	1.97	-2.81	3.22	0	2
72	-0.26	-0.16	3.05	0.54	0	2
73	-0.32	2.89	-2.96	1.59	0	2
74	-0.66	3.80	-4.34	0.62	0	2
75	5.09	1.51	3.87	0.31	0	2
76	4.10	0.67	5.05	0.85	0	2
77	-0.47	6.17	1.47	1.54	0	2
78	2.09	1.20	-0.90	0.15	0	2
79	6.22	0.60	-1.07	1.58	0	2

Table B.5: 100-point truth model

Scatterer	x	y	z	Magnitude	Phase	Ac./Pa.
80	-0.88	-0.34	-2.76	2.31	0	2
81	-11.46	5.96	1.79	3.34	0	2
82	4.27	-2.08	2.97	0.85	0	2
83	-12.18	1.26	5.06	0.89	0	2
84	0.74	5.14	-8.52	0.80	0	2
85	-0.33	-5.90	-8.18	0.01	0	2
86	-1.25	-3.99	-2.47	0.70	0	2
87	-2.69	2.08	0.76	0.10	0	2
88	2.41	2.54	-4.50	1.62	0	2
89	5.11	0.41	-1.98	0.15	0	2
90	2.24	2.21	3.30	1.68	0	2
91	1.74	-6.17	1.38	1.50	0	2
92	-0.60	-1.70	-2.53	2.36	0	2
93	3.97	-3.98	-1.89	0.71	0	2
94	2.31	-2.85	1.46	3.49	0	2
95	-8.56	-6.70	9.92	0.50	0	2
96	1.41	1.05	4.33	2.95	0	2
97	1.94	-1.19	3.52	4.47	0	2
98	-1.95	2.57	1.29	0.52	0	2
99	-0.24	-6.91	-3.73	2.38	0	2
100	-2.66	-3.58	-4.36	1.76	0	2

Table B.6: 100-point truth model

Scatterer	x	y	z	Magnitude	Phase (Radians)	Ac./Pa.
1	2.50	9.50	0.60	3.79	-0.01125	2
2	-1.50	1.00	-0.11	4.69	0.00093	2
3	-1.50	-1.00	-0.10	4.22	-0.01354	2
4	2.50	-9.55	0.60	4.05	0.00108	2
5	-6.40	0.00	3.60	8.56	0.00047	2
6	-9.00	0.10	0.00	6.78	0.00981	2
7	4.65	2.66	0.56	6.29	-0.02221	2
8	1.78	8.77	1.42	4.21	0.02974	2
9	-1.68	-1.09	0.62	5.48	0.00782	2
10	1.76	-9.85	0.54	4.58	-0.00976	2
11	5.57	-3.30	0.21	4.65	0.01484	2
12	-7.22	0.62	3.80	6.58	0.00921	2
13	-9.68	0.96	0.64	6.32	0.00587	2
14	-6.86	-0.24	0.41	5.74	0.01305	2
15	9.58	0.43	-0.02	6.80	0.00001	2
16	-1.67	-1.96	0.73	3.42	-0.03191	2
17	2.65	-8.78	1.33	5.95	-0.00636	2
18	-8.54	-0.77	-0.72	5.35	0.00530	2
19	5.80	3.51	-0.37	5.19	0.01741	2
20	1.81	9.86	0.03	5.26	0.02003	2
21	-8.55	-0.74	-0.46	6.62	0.02005	2
22	10.64	-0.80	0.20	4.71	0.00321	2
23	-1.30	1.39	0.29	3.26	-0.00696	2
24	-2.15	-0.41	-0.87	5.82	0.00420	2
25	3.16	-9.16	1.13	5.48	0.02065	2
26	4.12	-2.69	0.25	3.68	0.05130	2
27	-5.42	0.34	3.71	4.88	-0.03022	2

Table B.7: 27-point fitted model

C. Constrained Least Squares Fitting

The complex weightings of the individual scatterers in a model must be chosen so the correct field pattern can be generated. If the field pattern describing the N target radar cross section datum is denoted by a vector g_o and the M model amplitudes and phases are described by a vector f, equation C.1 describes a linear operator T that relates the two [86].

$$[T]_{N\times M}[f]_{M\times 1} \approx [g_o]_{N\times 1} \tag{C.1}$$

Equation C.2 defines the standard unconstrained Least Squares solution, where T^* indicates the complex conjugate and T^T indicates matrix transpose.

$$f = [[T^*]^T T]^{-1}[T^*]^T g_o \tag{C.2}$$

Unfortunately, the simple least squares method can give wild over-estimates for scatterer magnitude values in the model. A better method is to use a constrained least squares approach [86].

If Φ_i and λ_i denote the i^{th} Eigen vector and Eigen value of the matrix $[T^*]^T T$, ie.

$$[[T^*]^T T]\Phi_i = \lambda_i \Phi_i$$

The constrained least squares approximation of f is defined by equation C.3.

$$f = \sum_{i=1}^{M} \frac{c_i}{(\lambda_i + \alpha)} \Phi_i \tag{C.3}$$

where

$$c_i = [\Phi^*]^T [T^*]^T g_o$$

The value α may be found using the Newton–Raphson iterative method [87, Page 55]. Equation C.4 details the calculation, where C is the applied constraint. A starting value of $\alpha_0 = 1$ is suggested. Care must be taken if $\alpha_0 \leq 0$ as the function is discontinuous below zero and the results of the iterative method are unpredictable. A possible solution is to replace negative values of α_0 with a positive random number, typically in the range $[1, 100]$. The algorithm will then re-start at a random position.

$$\alpha_1 = \alpha_0 - \left(\frac{C - \sum_{i=1}^{M} \frac{|c_i|^2}{(\lambda_i + \alpha_0)^2}}{2\sum_{i=1}^{M} \frac{|c_i|^2}{(\lambda_i + \alpha_0)^3}} \right) \tag{C.4}$$

The square of the norm of the vector f, as defined in equation C.5, is limited by the value of the constraint C.

$$||f||^2 = \sum_{i=1}^{M} |f_i|^2 \tag{C.5}$$

If the model is being used to re-construct an ISAR image, the required constraining value for $||f||^2$, ie. C, is found by calculating the squared norm of the image. Equation C.5 is applied to the image with f_i representing each picture element. As the image intensity is determined by scatterer amplitudes, the squared norm should be approximately the same as the squared norm of the model.

D. Statistical Tests

D.1. Introduction

The purpose of both statistical tests is to determine whether two independent groups of data have been drawn from the same distribution.

D.2. Kolmogorov-Smirnov Statistical Test

The Kolmogorov-Smirnov (K–S) statistical test [88, Pages 472–475] gives a figure of merit for the similarity between the cumulative distribution functions of two sets of data. The K–S number, λ_{KS}, may be related to a probability that the sets of data are drawn from the same distribution.

Taking two Cumulative Distribution Functions S_1 and S_2 of size N_1 and N_2 respectively, the Kolmogorov–Smirnov statistic is as shown in equation D.1.

$$\lambda_{KS} = \sqrt{\frac{N_1 N_2}{N_1 + N_2}} \left(\max_{-\infty<x<\infty} |S_1(x) - S_2(x)| \right) \tag{D.1}$$

In the case of the null hypothesis 'sets of data drawn from the same distribution', the distribution of the K–S statistic can be calculated giving the significance of any observed non-zero value of λ_{KS}.

The significance may be calculated using equation D.2, which is monotonic with limiting values $Q_{KS}(0) = 1$ and $Q_{KS}(\infty) = 0$.

$$Q_{KS}(\lambda) = 2\sum_{j=1}^{\infty}(-1)^{j-1}e^{-2j^2\lambda^2} \tag{D.2}$$

In terms of this function, the significance level of an observed value of λ_{KS} (as disproof of the null hypothesis) is given approximately by

$$\text{Prob}(\lambda_{KS} > \text{observed}) = \alpha = Q_{KS}(\lambda_{KS})$$

The approximation becomes asymptotically accurate as N becomes large. Typically $N > 20$ is acceptable. Table D.1 summarises some commonly used significance levels (α) and their K–S number equivalents. The significance levels are the probability of a type one error, ie. the null hypothesis is rejected erroneously.

D.3. Mann-Whitney Statistical Test

The Mann-Whitney test (sometimes called Wilcoxon Test) uses ranking methods to produce an indication of how the sets of data overlap. The test statistic is normally

% Significance, α	K–S number, λ_{KS}
99%	0.4410
95%	0.5196
90%	0.5712
10%	2.2239
5%	1.3580
1%	1.6720

Table D.1: Typical Kolmogorov–Smirnov test significance levels

distributed for sufficient numbers of observations. The test is performed on the two sets of data as follows.

1. Label the two sets of data X and Y, having n and m members respectively. Set X should be the smallest set, ie. $n \leq m$.

2. Put the two sets of data together to form a single $X + Y$ group with $(n + m)$ members.

3. Rank the $(n + m)$ data with rankings 1 to $(n + m)$, with the smallest value getting rank 1 and the largest rank $(n + m)$.

4. If $R(X_i)$ denotes the rank of X_i, equation D.3 gives the test statistic.

$$U = \sum_{i=1}^{n} R(X_i) - \frac{n(n+1)}{2} \tag{D.3}$$

Thus, if the population X lies totally below population Y, $U = 0$. While if X lies totally above Y, $U = nm$. If $n, m > 20$, the test statistic asymptotically approximates the normal distribution with the mean and standard deviation shown in equation D.4.

$$\begin{aligned} U_z &= \frac{U - \mu}{\sigma} \\ \mu &= \frac{nm}{2} \\ \sigma &= \sqrt{\frac{nm(n+m+1)}{12}} \end{aligned} \tag{D.4}$$

Table D.2 summarises some commonly used significance levels (α) and their Mann-Whitney U_z equivalents. The significance levels are the probability of a type one error, ie. the null hypothesis is rejected erroneously.

% Significance, α	Mann-Whitney Statistic, U_z
10%	1.283
5%	1.645
1%	2.327

Table D.2: Typical Mann-Whitney test significance levels

E. Population Based Incremental Learning

E.1. Introduction

This appendix covers advances in tuning the search parameters of Population Based Incremental Learning (PBIL) algorithms. The basic PBIL algorithm has been enhanced to include automatic run termination and optimum setting of some control parameters. In many cases, only one parameter now needs to be tuned for efficient operation of the algorithm. A simple binary-tree structure is also described for storing past chromosomes, giving around a 30% processing reduction for complex objective functions. The enhanced PBIL algorithm may be more easily applied to a wide range of engineering applications. Example *MATLAB* routines are included to demonstrate the simplicity of the algorithms.

Population based incremental learning algorithms are considered to be among the simplest evolutionary optimisation techniques currently available. They are able to find optimum solutions to problems which are multi-modal or lack gradient information. They have been shown to outperform conventional deterministic and stochastic optimisation techniques on a wide range of problems and yet are simple to code [89, 90].

The algorithm was first described in 1994 [89] and has been improved recently [91]. This algorithm has three control parameters; *population size* (p), *learning rate* (l) and *forgetting factor* (f). The algorithm presented here has been enhanced further by automatically terminating the run when the process has converged on a single solution. One new operating parameter has been introduced into the algorithm to control the algorithm termination. This parameter has been called the *termination factor* and is denoted by τ. The *forgetting factor* is now calculated from a more intuitive parameter that allows the analysis of the algorithm operation to be simplified. This parameter has been called the *search rate* and is denoted by s.

Empirical values have been derived for two of the four parameters and empirical conditions for optimality have been established for a third. The remaining parameter of population size is used to control the search. Small populations yield rapid but crude results, large populations will give more accurate results but at a processing cost. The processing overheads can be reduced by storing previous chromosome structures and their objective values. A simple technique based on binary space partition trees is described.

E.2. Algorithm operation

The PBIL algorithm is a stochastic guided search process that obtains its directional information from the previous best solutions. The problem parameters are represented as a binary chromosome of total length b bits. Each variable is coded in a binary form and then concatenated to any previous parameters to form a single chromosome.

A *prototype vector* (**P**) is used to bias the generation of bits in a population of chromosomes. The prototype vector has b elements, one for each bit location. At each location, the prototype vector holds the probability that the corresponding bit is a '1'. Each location is initially set to 0.5 which corresponds to unbiased bit generation. A population of candidate solutions is generated using the prototype vector to bias the generation of bits. For each chromosome in the population, the bits are selected by generating a uniformly distributed random number in the range [0,1] for each bit. The chromosome bit is set to one if the random number is less than the corresponding prototype vector element, zero otherwise. All the chromosomes are then evaluated by the objective function and the best identified.

Equation E.1 is then applied to the prototype vector to incorporate the directional information of the best chromosome. This equation is a variant of the process described in [91].

$$\mathbf{P}_{n+1} = ((1-l)\mathbf{P}_n + l \cdot \mathbf{C}_B)(1-f) + \frac{f}{2} \tag{E.1}$$

Where $\mathbf{C}_B$ is the best chromosome and consists of a pattern of ones and zeros.

Figure E.1 shows the changes in the prototype vector during a typical optimisation run. The learning mechanism in equation E.1 leads to a change in each prototype vector element level, that follows an exponential profile. In order to investigate the algorithm further, the nature of the exponential function must be known. To establish the function characteristics, if we take the specific case of $C_B = 0$, the rate of fall of each prototype vector element, P, is given by equation E.2.

$$P_n = \frac{1}{2}\left(\frac{f(1-a^n)}{1-a} + a^n\right) \tag{E.2}$$

where

$$a = (1-l)(1-f)$$

and n is the generation number

The level to which each element of the prototype vector converges is found by taking the limit of equation E.2 at $n = \infty$ to give equation E.3, which defines the search rate, s. As the pattern of convergence is symmetrical about $P = 0.5$, the convergence properties for $C_B = 1$ may be determined by observation as $1 - P_n$. The search rate, s, is shown graphically on figure E.1. The search rate may also be considered as the probability of selecting a one instead of a zero element after an infinite number of generations. Equation E.4 is used to calculate the forgetting factor, f for any given value of s.

$$s = P_\infty = \frac{f}{2(1-a)} \tag{E.3}$$

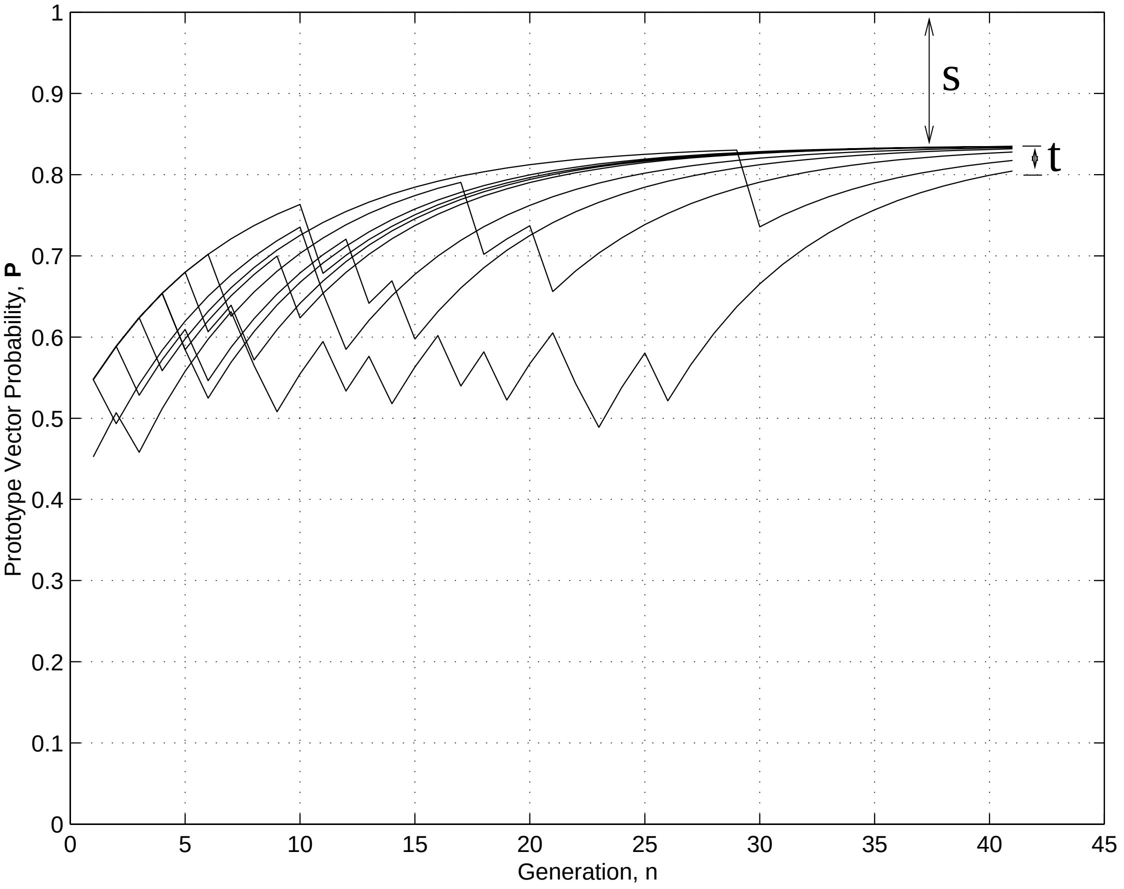


Figure E.1: Typical prototype vector plot (9 bits in **P**)

$$f = \frac{2sl}{1 - 2s(1 - l)} \tag{E.4}$$

If the search rate is set to zero, the elements of the prototype vector can converge to either 0 or 1. Once this terminal value has been reached, if the element has converged in the wrong direction, there is no way for it to be corrected. Increasing the search rate prevents the prototype vector elements converging exactly to 0 or 1. The search rate is analogous to mutation rate in genetic algorithms. The higher the value of s, the less likely the algorithm is to get stuck in local optima.

The algorithm is allowed to run until all the elements are within a bound t of the final convergence level of s or (1–s) as appropriate. The bound t is shown graphically in figure E.1. If we define t as shown in equation E.5, t varies dynamically with s. We can now calculate the minimum number of generations required for the algorithm to terminate, $n_{\min}$, shown in equation E.6.

$$t = (0.5 - s)\tau \tag{E.5}$$

$$\begin{aligned} P_n &= s + t \\ \frac{f(1-a^n)}{2(1-a)} + \frac{a^n}{2} &= \frac{f}{2(1-a)} + \left(0.5 - \frac{f}{2(1-a)}\right)\tau \\ f + a^n((1-a) - f) &= f + ((1-a) - f)\tau \\ a^n &= \tau \\ n_{\min} &= \frac{\log(\tau)}{\log(a)} \end{aligned} \tag{E.6}$$

Equation E.7 gives the minimum number of function evaluations, $F_{\min}$.

$$F_{\min} = pn_{\min} \tag{E.7}$$

The termination condition may be summarised as shown in equation E.8.

$$\max(0.5 - |\mathbf{P} - 0.5|) < s + (0.5 - s)\tau \tag{E.8}$$

A convenient empirical setting for τ is $\tau = 0.1$. Reducing τ will extend the length of the run and increasing τ will increase the risk of false termination. If τ is increased, at most $n_{\min}$ generations can be cut from the runtime.

Trials have shown that the final convergence phase occurs most often in the region where there should be, on average, at least one copy of the optimum chromosome in the population. This conditions are met when equation E.9 is satisfied.

$$p \prod \mathbf{P} > 1 \tag{E.9}$$

It must be noted that for the PBIL algorithm to converge on a final solution, each gene should influence the objective function. If inverting a gene value has no effect, the associated prototype vector element will drift randomly around the 0.5 average value. It has been demonstrated that given the condition of every gene having influence, the algorithm will eventually converge on a solution [92]. As the final convergence level of the prototype vector is limited by s, the condition in equation E.10 therefore has to be satisfied for algorithm convergence.

$$p > \frac{1}{(1 - s)^b} \tag{E.10}$$

It has been found through trials and observations that the minimum number of function evaluations required to reach convergence is governed by the probability of a chromosome being better than average, the search rate, and the number of bits. This probability, $P(\text{Chrom.} > \text{Aver.})$, appears to be approximately 0.5 for most real-world problems.

The ‘sum of bits’ problem (demonstrated in section E) with an even number of bits has a probability $P(\text{Chrom.} > \text{Aver.}) < 0.5$. The lower probability only has significant influence for low numbers of bits. The probability may be related to population size as described in equation E.11.

$$p_{opt} = \frac{1}{P(\text{Chrom.} > \text{Aver.})(1-s)^b} \tag{E.11}$$

This effect is demonstrated graphically in figure E.2. The minimum number of function evaluations for any given value of s occurs at P_{opt}. If s is increased, P_{opt} will increase, following the linear portion of the curve. Thus an optimum value of s may be found for any population size and thus the corresponding forgetting factor calculated. Using this philosophy, as population size is increased, s increases and so the chances of converging on a local optima reduces. The lines in figure E.2 are an average of 100 trials at each population size under the values of s shown and using the source code in section E.

Thus we may operate the algorithm by choosing a population size and then calculating the maximum value of s that minimises the number of function evaluations. Equation E.12 details this.

$$s_{opt} = 1 - (2/p)^{1/b} \tag{E.12}$$

A population size of 5 with an optimum value for s $(0 < s < 0.5)$ is often a good starting point. It is wise to start with low population sizes to assess how many function evaluations are required and then increase p to achieve sufficiently accurate results. By combining equations E.4, E.6, E.7 and E.12, equation E.13 gives the minimum number of function evaluations required for operation with $s = s_{opt}$.

$$F_{\min} = p \cdot \left\lceil \frac{\log(\tau)}{\log\left(1 - \frac{l}{1-2(1-l)\left(1-(2/p)^{1/b}\right)}\right)} \right\rceil \Bigg|_{s=s_{opt}} \tag{E.13}$$

where $\lceil x \rceil$ denotes the smallest integer $\geq x$

Figure E.3 shows the results of proving trials for a 7 bit sum of chromosome problem, where the theoretical minimum is denoted by ‘ .. ’, actual minimum by ‘ - - ’, mean by ‘ – ’ and maximum by ‘ .- ’. The results were generated from a PBIL algorithm with $\tau = 0.1$, $l = 0.1$ and $s = s_{opt}$ and 1000 trials run at each population size. The data for the minimum number of function evaluations match the theory well but it is unknown how the other curves are related. In practice, the algorithm will be operating in the region $p \ll 2^b$ where the curve is almost linear. The details of this near linear portion of the curve are specific to the objective function used.

The learning rate (l) determines the final accuracy of the solution. The lower the learning rate, the less likely it is that the algorithm will converge on a local optimum. Baluja [89, Page 17] observed that:

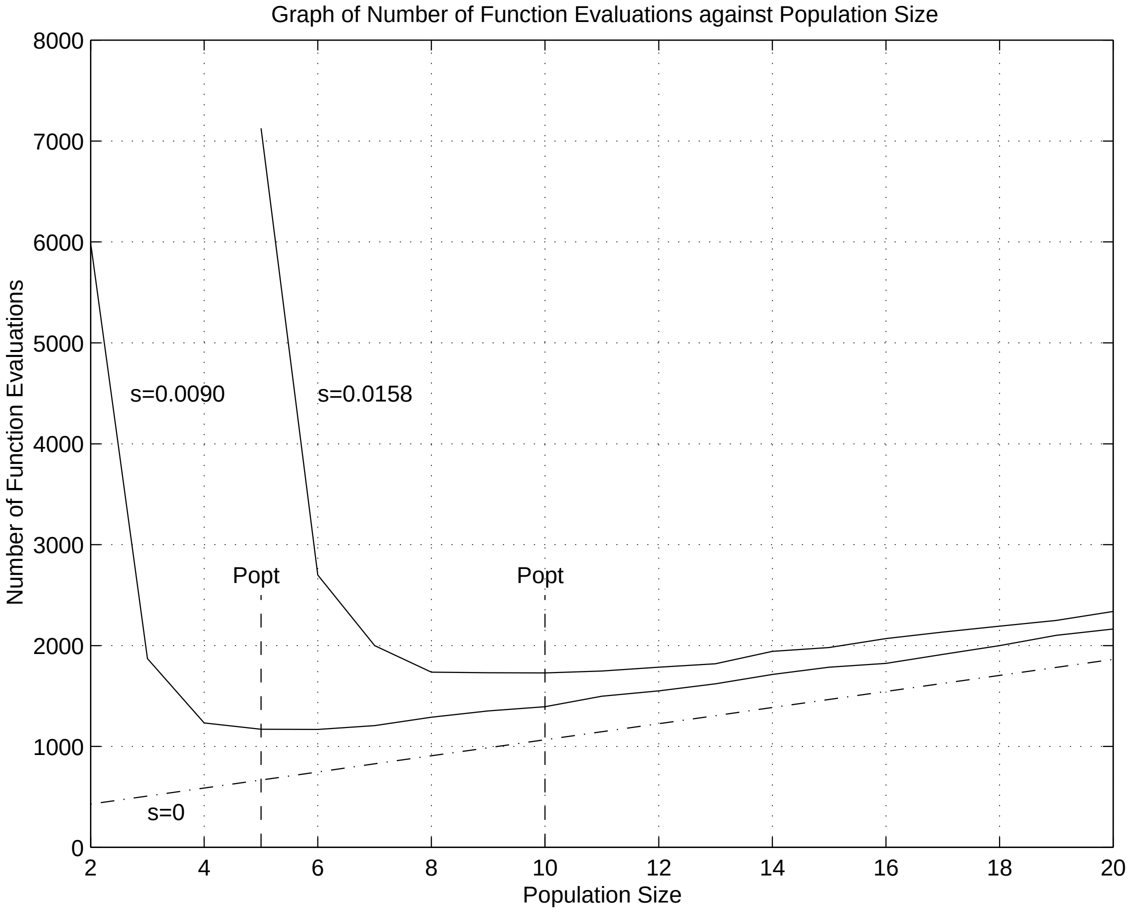


Figure E.2: Optimum operation

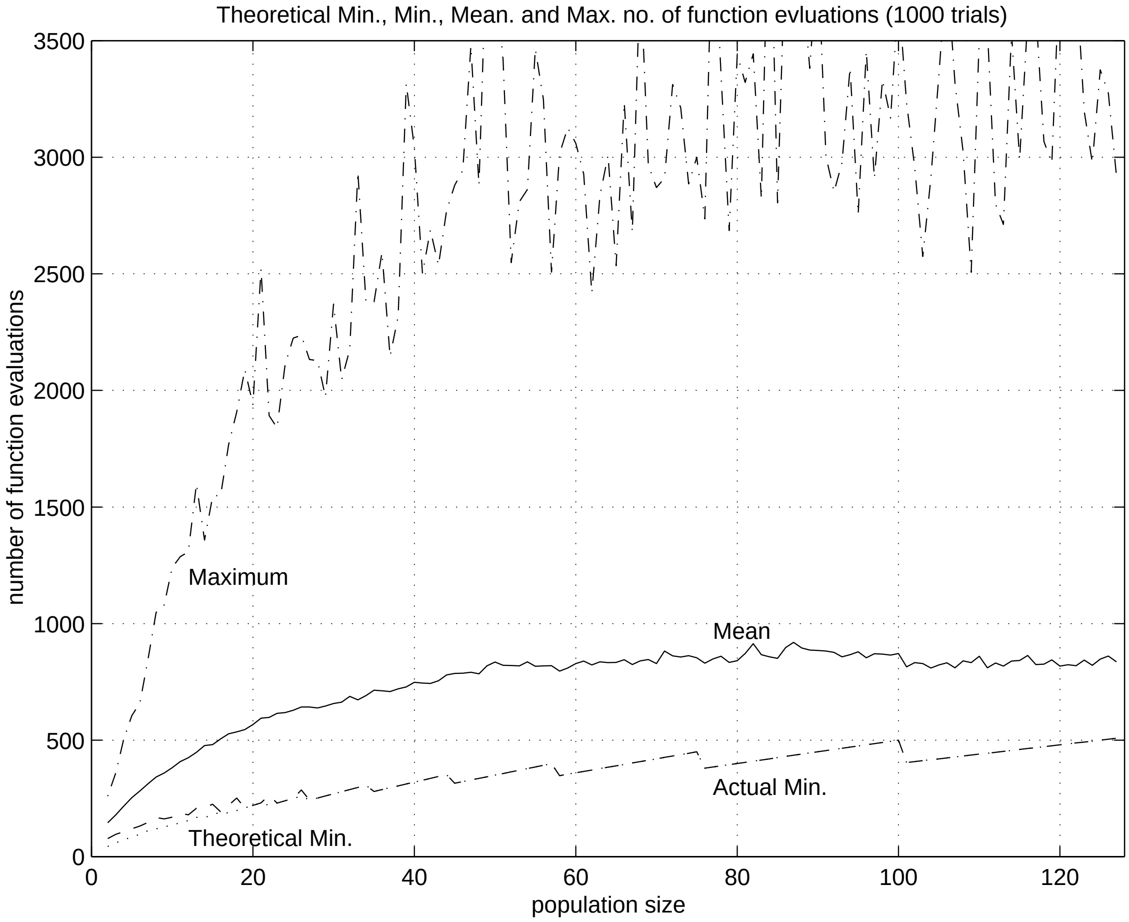


Figure E.3: Results of proving trials

> If the learning rate is high, the initial populations generated will largely determine the focus of the search, without enabling the algorithm to explore the function space. If the function space does not contain local optima, a high learning rate may work well. However, if local minima could be a problem, lower learning rates allow greater exploration.

Changing the learning rate directly affects the gradient of the near linear portion of the curves. Doubling l will approximately halve the gradient etc. (following equation E.6 as the gradient). Therefore as l is increased, the number of function evaluations reduces but the probability of premature algorithm convergence increases. An empirical range of $0.1 \leq l \leq 0.4$ has been found satisfactory for most problems.

E.3. Binary Space Partition Tree for Chromosome Storage

By their very design, evolutionary algorithms can be inefficient with objective function calculations. In the first few generations of the algorithm, all of the chromosomes evaluated are likely to be different. As the population of chromosomes converge toward a solution, a small set of chromosomes may be evaluated repeatedly. If the objective function has heavy processing requirements, much processor time can be wasted. A variant of the binary space partition tree described in Foley et. al. [62, Pages 675–680] may be used to reduce the number of wasted calculations. The following algorithm may be applied to most evolutionary algorithm techniques.

The tree is used to store chromosome patterns and their corresponding objective values. In this variant of the standard tree, the chromosomes themselves are used as the partitioning structures. The tree is generated by inserting each new chromosome as its objective value is required by the evolutionary algorithm. The very first chromosome is treated as a special case and simply inserted into the first location in the tree. Its objective value is then calculated and stored in the tree too. Two extra data values are held along with each chromosome and objective. Initially, the data items are both zero, but are intended for storing pointers to the left and right branches of the tree.

Subsequent chromosomes are added to the tree only if they are not already present. The new chromosome is compared to the first chromosome in the tree. Because the number of bits in the chromosome may be much larger than the machine precision, the chromosomes are compared gene by gene. Starting from the left, each gene in the new chromosome is compared to the corresponding gene of the chromosome in the tree. If these genes are the same, the next pair of genes are tested. The first test that finds a difference in the chromosomes is used as the decision variable. If the gene in the new chromosome is less than the gene in the tree, the new chromosome is classed as being less than the tree chromosome. This algorithm is demonstrated in figure E.4. If the new chromosome is classed as smaller, the left branch of the tree is investigated. If it is greater, the right branch is chosen.

The tree is descended in a recursive fashion until either a matching chromosome is found or a chromosome with no sub-tree to follow is encountered. If a match is found, the previously recorded objective value is returned to the evolutionary algorithm.

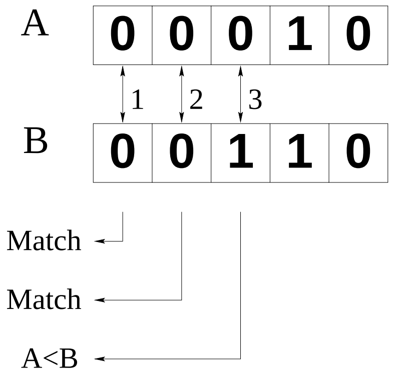


Figure E.4: Example chromosome comparison

If no match is found before the tree ends, an objective value is calculated for the chromosome. The objective is inserted into the tree along with the chromosome, and the index to the storage location recorded in the appropriate pointer of the last chromosome found in the tree. This process is demonstrated in figure E.5. If the same chromosome is generated by the evolutionary algorithm again, it can be retrieved quickly from the tree.

The tree structure is suited to both binary and integer chromosomes. Real valued chromosomes are difficult to store effectively as a tiny deviation in one gene is enough to prevent the chromosome being matched. Generating the tree by using the chromosomes as the partitioning structures may not lead to a very efficient tree structure though. A balanced tree has a uniform distribution of branches following each node. The trees generated using this method will not be balanced and therefore some look-up operations may be much more rapid than others. However, as the evolutionary algorithm is stochastic in nature, the spread of the tree unlikely to become excessively unbalanced.

If the objective is quick to calculate, it may be better not to use the tree. If the objective is computationally expensive or requires heavy disk usage, in a typical evolutionary algorithm, one third of the objective values may be returned from the tree. Example *MATLAB* code for implementing the tree structure in the PBIL algorithm is presented in section E.5

E.4. Conclusions

The enhancements to the algorithm reduce the tuning burden normally associated with stochastic optimisation techniques. In most cases, only the population size needs to be adjusted to trade repeatability against number of function evaluations. The simplicity of the algorithm allows it to be applied to new problems rapidly and can give excellent results with little or no tuning. The *MATLAB* code in section E.5 is a complete PBIL example where the function being optimised is the sum of the bits in the chromosome. This function is multi-modal in nature. The whole programme consists of 20 lines without the comments and should take approximately 180 generations to complete. Trials have shown that with a population of 10 in the

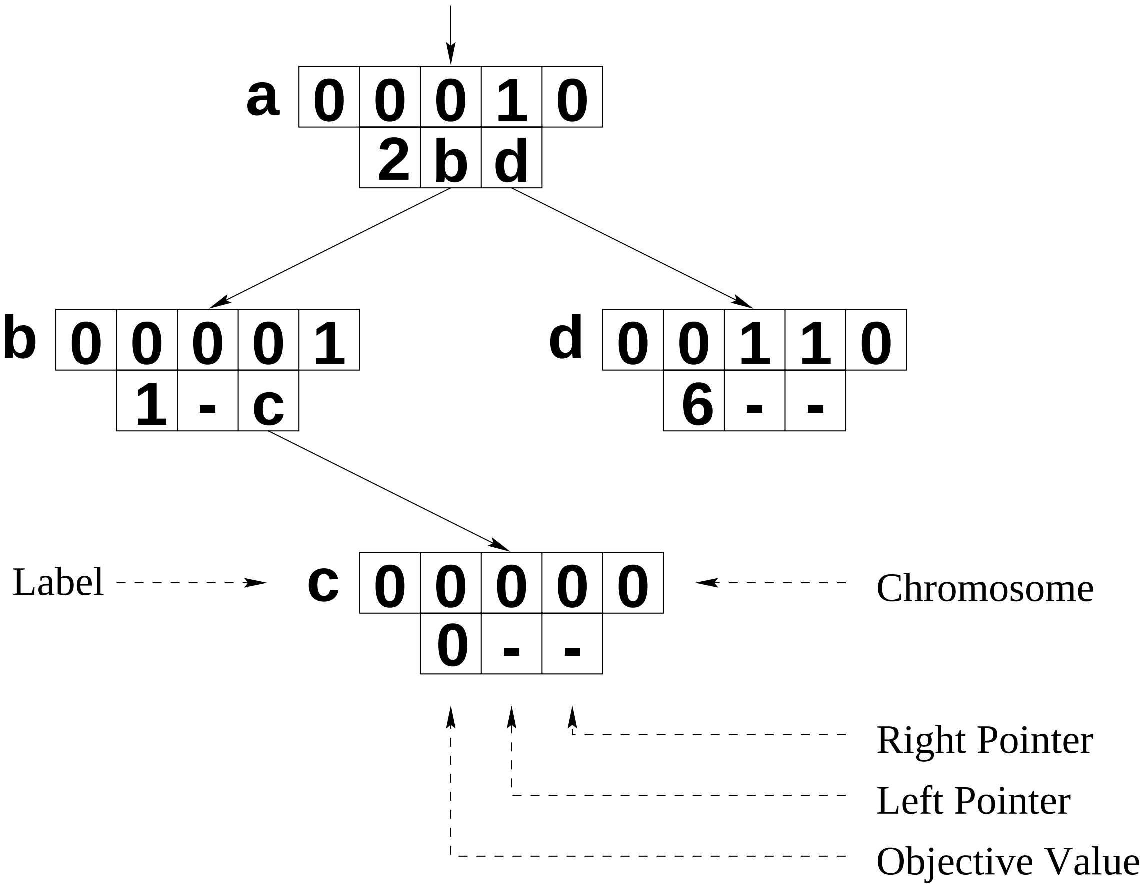


Figure E.5: Example tree structure

example, the correct answer can be obtained about 95% of the time.

E.5. *MATLAB* example

```
% Enhanced Population Based Incremental Learning
% E.J.Hughes  14/11/97
%% user control parameters
  maxgen=3000;      % maximum no. of generations
  b=101;            % no. of bits in chromosome
  l=0.1;            % learning rate
  p=10;             % population size
%% set other control parameters
  tau=0.1;               % termination factor
  s=1-(2/p)^(1/b);       % search rate
  f=2*s*l/(1-2*s*(1-l)); % forgetting factor
%% initialise
  pv=0.5*ones(1,b);
  pvx=zeros(maxgen,b);
%% main loop
  for n=1:maxgen
%% generate population
    chrom=rand(p,b)<(ones(p,1)*pv);
%% put objective here
    obj=sum(chrom')';  % calculate sum of bits
    [a,i]=max(obj);    % i=index of best chromosome
%% update prototype vector and stop if converged
    pv=((1-l)*pv+l*chrom(i,:))*(1-f)+f/2;
    pvx(n,:)=pv;
    if max(0.5-abs(pv-0.5))<(s+(0.5-s)*tau)
       break; end
  end
%% output results
  [(pv>0.5) a]
  plot(1:n,pvx(1:n,:));   % plot prototype vector
```

```
% Population Based Incremental Learning
% With Binary Space Partition Tree Enhancement
% for Objective Calculations
% E.J.Hughes  10/2/98
%% user control parameters
  maxgen=3000;      % maximum no. of generations
  b=101;            % no. of bits in chromosome
  l=0.1;            % learning rate
  p=10;             % population size
%% set other control parameters
  tau=0.1;               % termination factor
  s=1-(2/p)^(1/b);       % search rate
  f=2*s*l/(1-2*s*(1-l)); % forgetting factor
```

```
%% initialise
  pv=0.5*ones(1,b);
  pvx=zeros(maxgen,b);
%% Initialise BSP tree
 bsp=zeros(maxgen*p,3+b);
 curr_pos=1;
 hit_cnt=0;
%% main loop
  for n=1:maxgen
%% generate population
    chrom=rand(p,b)<(ones(p,1)*pv);
%% To calculate objective:
    obj=zeros(p,1);
    for k=1:p
%% Check BSP tree first
       ck=chrom(k,:);
       pos=1;next=0;
       objt=[];
       while(isempty(objt))
          kk=sign(ck-bsp(pos,4:3+b));
          tst=kk(min(find(kk~=0)));
          if isempty(tst)
             objt=bsp(pos,3);  %Chromosome exists
             hit_cnt=hit_cnt+1;
             tst=0;
          end
          next=bsp(pos,1+(tst==1)); % get next node
          if(~next)&isempty(objt)   % not found
%% put objective here
             objt=sum(chrom(k,:));  % calculate sum of bits
             bsp(curr_pos,3:3+b)=[objt ck];
             if(curr_pos~=1)
                bsp(pos,1+(tst==1))=curr_pos; % add to tree
             end
             curr_pos=curr_pos+1;
          end
          pos=next;
       end
       obj(k,1)=objt;
    end
    [a,i]=max(obj);    % i=index of best chromosome
%% update prototype vector and stop if converged
    pv=((1-l)*pv+l*chrom(i,:))*(1-f)+f/2;
    pvx(n,:)=pv;
    if max(0.5-abs(pv-0.5))<(s+(0.5-s)*tau)
       break; end
  end
%% output results
  [(pv>0.5) a]
  plot(1:n,pvx(1:n,:));   % plot prototype vector
```

F. Noise Approximations

F.1. Introduction

This appendix covers the derivation of the approximation used to divide two Gaussian noise signals. A survey of existing literature and mathematical texts failed to provide a solution to the problem and so an empirical approximation was found.

F.2. Derivation

An attempt was made to derive an analytical proof of the division of the two Gaussian noise sources detailed in equation F.1. Where $N(\mu, v)$ describes a Gaussian source with mean μ and variance v.

$$\frac{N(\mu_1, v_1)}{N(\mu_2, v_2)} \tag{F.1}$$

The following properties of noise distributions may be found from standard texts (for example [93]); where $D_1(\mu, v)$ and $D_2(\mu, v)$ etc. describe noise sources with an arbitrary distribution and mean μ and variance v (the distribution D' is not necessarily the same as D).

1. $D_1(\mu_1, v_1) \pm D_2(\mu_2, v_2) = D_3(\mu_1 \pm \mu_2, v_1 + v_2)$
2. $\mu + D_1(0, v) = D_1(\mu, v)$
3. $aD_1(\mu, v) = D_1(a\mu, a^2 v)$

The property

$$D_1(\mu_1, v_1)D_2(\mu_2, v_2) = D_4(\mu_1\mu_2, \mu_1^2 v_2 + \mu_2^2 v_1 + v_1 v_2)$$

may be derived from the above results.

The assumption that there exists an inverse that satisfies equation F.2 was made. This assumption is obviously flawed as the noise cannot be cancelled out by another random distribution but it does provide a useful mathematical construction.

$$D(\mu, v)D(\mu, v)^{-1} = D(1, 0) \tag{F.2}$$

The theoretical inverse shown in equation F.2 was calculated and is shown in equation F.3.

$$D(\mu, v)^{-1} = D\left(\frac{1}{\mu}, \frac{-v}{\mu^2(\mu^2 + v)}\right) \tag{F.3}$$

The negative variance is nonsense as it would represent imaginary noise. Empirical trials indicate that if the negative sign is removed, the equation does closely approximate the reciprocal of a noise source, as long as all of the noise samples lie to one side of zero, ie. $|\mu|$ is large with respect to the variance v. Observations of the noise distribution show that the noise is approximately Gaussian for large μ.

Using equation F.3 with positive variance will yield equation F.4.

$$\frac{N(\mu_1, v_1)}{N(\mu_2, v_2)} \approx N\left(\frac{\mu_1}{\mu_2}, \frac{\mu_1^2 v_2 + \mu_2^2 v_1 + 2v_1 v_2}{\mu_2^2(\mu_2^2 + v_2)}\right) \tag{F.4}$$

Empirical trials show that equation F.4 gives a good approximation to the quotient of the two noise sources. Observations and trials have shown that equation F.5 gives an even better approximation to the noise. Figure F.1 shows a histogram for the example equation $N(0,1)/N(10,1)$ demonstrating that the result is approximately Gaussian. The overlaid curve is for a normal distribution $N(0, 0.01)$, as provided by equation F.5. As μ_2 decreases, the distribution becomes very long-tailed with large outliers. The region where μ_2 is low is also the region where glint spikes occur. A proper solution to the quotient problem may provide an insight into the probability distribution of the glint noise.

$$\frac{N(\mu_1, v_1)}{N(\mu_2, v_2)} \approx N\left(\frac{\mu_1}{\mu_2}, \frac{\mu_1^2 v_2 + \mu_2^2 v_1 + v_1 v_2}{\mu_2^2(\mu_2^2 + v_2)}\right) \tag{F.5}$$

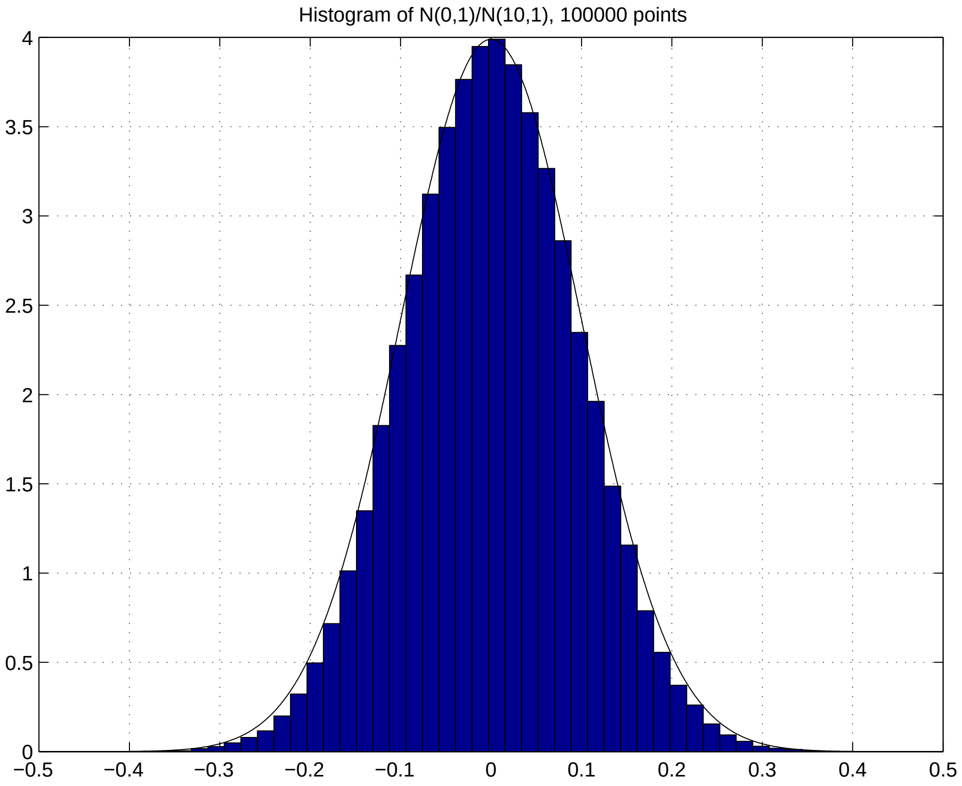


Figure F.1: Example noise histogram (50 bins)

G. Target Manoeuvre Detection Software

G.1. Introduction

This appendix details the *MATLAB* fuzzy-logic inference system modules and includes the software used for target manoeuvre detection. The source code is designed using the *MATLAB* Fuzzy Logic Toolbox [82]. The following source code is detailed in this appendix.

- Initialisation Code
- Detector Main Function
- Fuzzy Pre-Processor Module
- Fuzzy Combiner Module
- Fuzzy Detector Module
- Fuzzy Confidence Module

G.2. Detail of the Stage-1 Pre-Processor module

Figure G.1 shows the top-level structure of the stage-1 module. The module is a Sugeno type fuzzy inference system, where the outputs are functions of the input variables. Figure G.2 shows the membership function for the range input. There are no membership functions associated with the s_n and s_r inputs. There are four rules in the system. They are as follows:

1. If (range is long) then (out $= s_n$)
2. If (range is short) then (out $= 100s_r$)
3. If (range is close) then (out $= 30s_r$)
4. If (range is end) then (out $= 10s_r$)

The membership function for the range and the levels of the gains in rules 2,3 & 4 have been chosen empirically by observation.

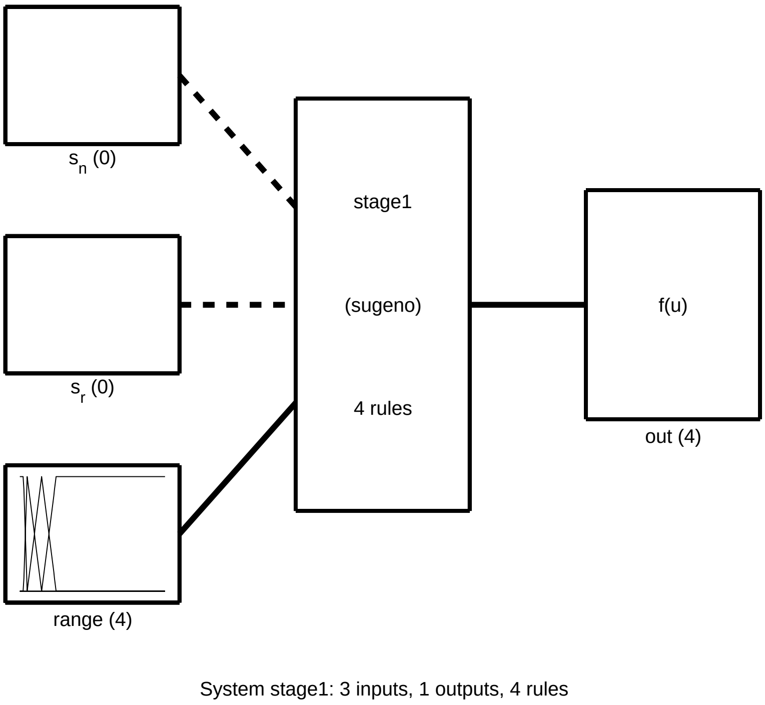


Figure G.1: Block diagram of Stage-1 module

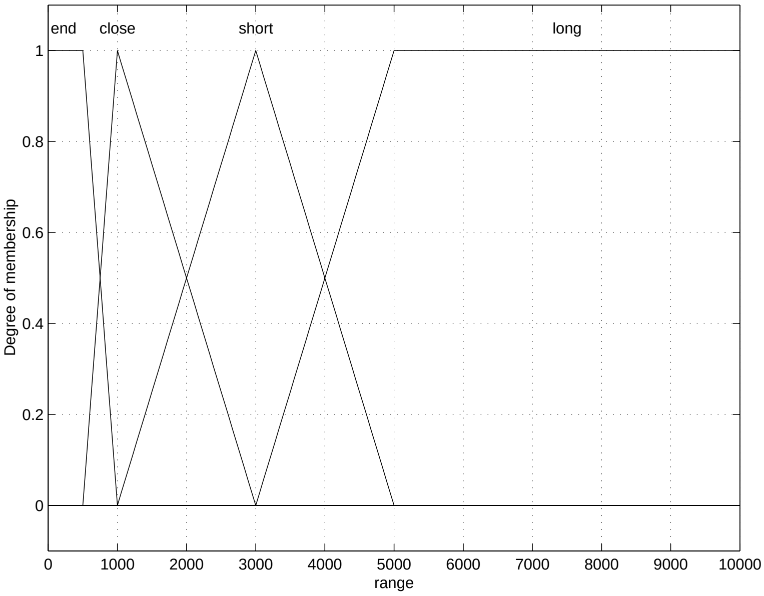


Figure G.2: Membership function for the Stage-1 module range input

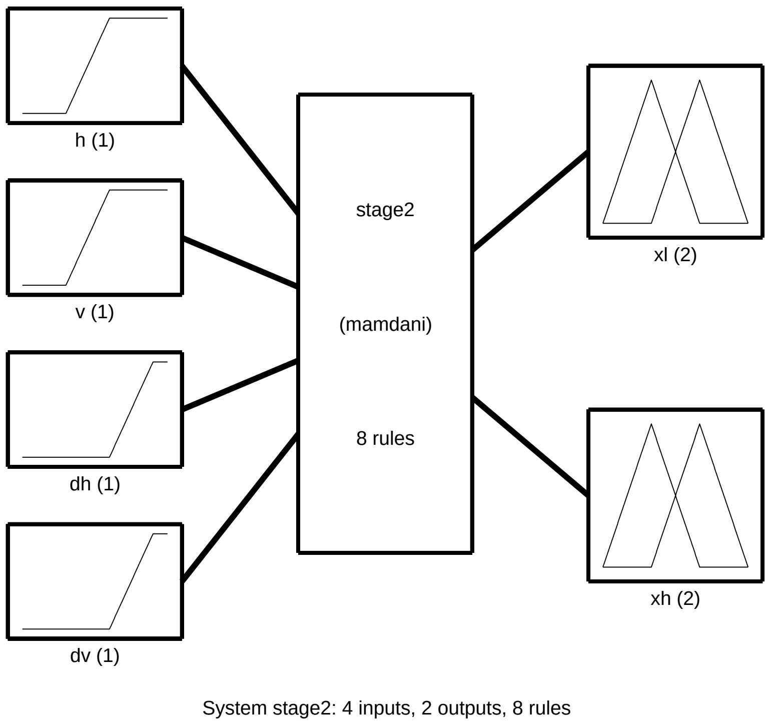


Figure G.3: Block diagram of Stage-2 module

G.3. Detail of the Stage-2 Combiner module

Figure G.3 shows the top-level structure of the stage-2 module. The module is a Mamdani type fuzzy inference system, where the outputs are membership functions. Figure G.4 shows the membership function for the normal signal inputs and figure G.5 shows the membership function for the differentiated signals. Figure G.6 shows the membership function for the outputs.

There are eight rules in the system. They are as follows:

1. If (h is hi) or (v is hi) or (dh is hi) or (dv is hi) then (xl is no)
2. If (h is not hi) and (v is not hi) and (dh is not hi) and (dv is not hi) then (xl is yes)
3. If (dh is not hi) and (dv is not hi) then (xh is no)
4. If (h is not hi) and (v is not hi) then (xh is no)
5. If (v is hi) and (dv is hi) then (xh is yes)
6. If (h is hi) and (dh is hi) then (xh is yes)
7. If (h is hi) and (dv is hi) then (xh is yes)
8. If (v is hi) and (dh is hi) then (xh is yes)

The membership functions have been chosen empirically by observation.

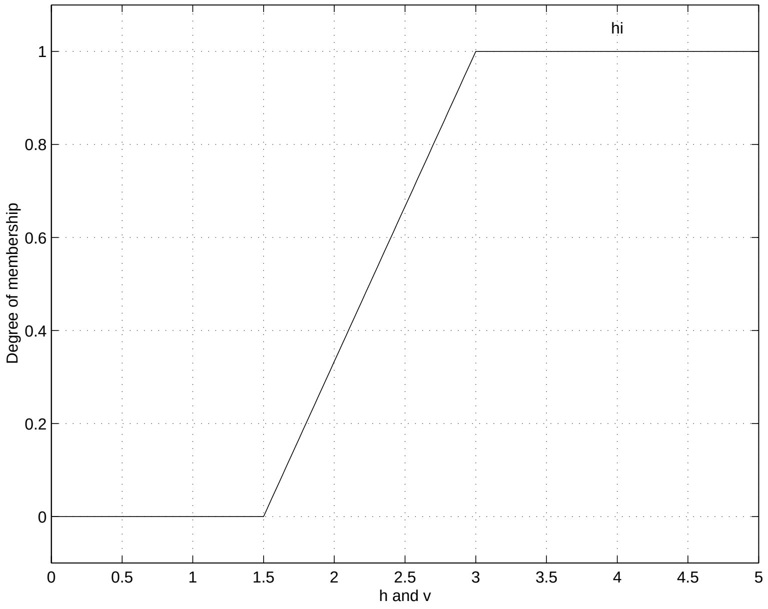


Figure G.4: Membership function for the Stage-2 module normal inputs

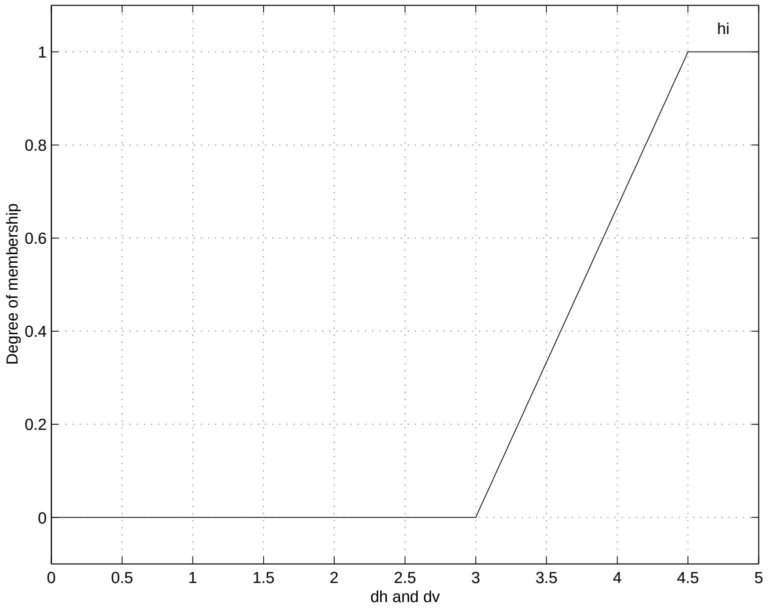


Figure G.5: Membership function for the Stage-2 module differentiated inputs

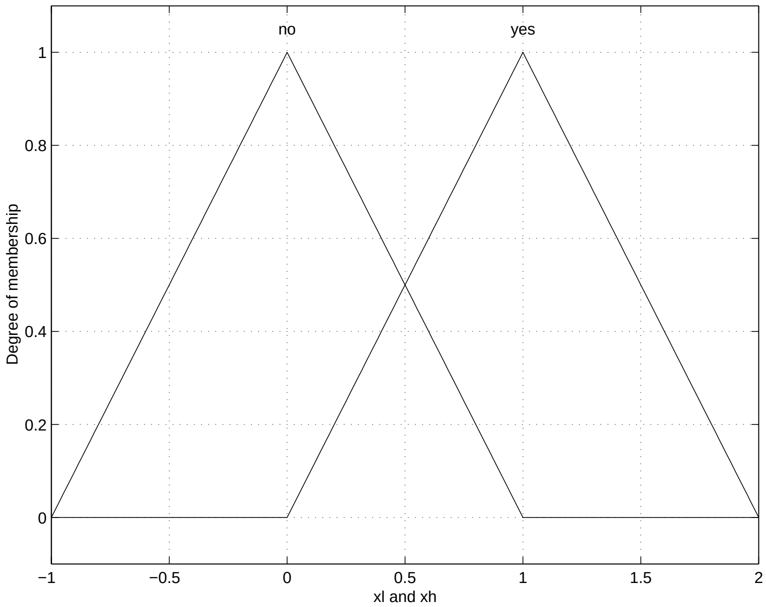


Figure G.6: Membership function for the Stage-2 module outputs

G.4. Detail of the TMD Detector module

Figure G.7 shows the top-level structure of the TMD, target manoeuvre detector module. The module is a Sugeno type fuzzy inference system. Figures G.8 and G.9 show the membership functions for the signal inputs. Figures G.10 and G.11 show the membership functions for the feedback signals to control the hold and delay times. Figure G.12 shows the membership function for the range input. There are sixteen rules in the system. They are as follows:

1. If (xh is trip) and (hold_i is off) and (range is long) then (delay $= 0.5$)
2. If (xh is trip) and (range is long) then (hold $= 10$)
3. If (xh is trip) and (hold_i is off) and (range is med) then (delay $= 1$)
4. If (xh is trip) and (range is med) then (hold $= 6$)
5. If (xh is trip) and (hold_i is off) and (range is short) then (delay $= 1.75$)
6. If (xh is trip) and (range is short) then (hold $= 4$)
7. If (xh is trip) and (hold_i is off) and (range is close) then (delay $= 2.5$)
8. If (xh is trip) and (range is close) then (hold $= 2$)
9. If (hold_i is off) then (tmd $= 0$)(delay $= 0$) (hold $= 0$)
10. If (delay_i is not off) and (hold_i is not off) then (tmd $= 0$)
11. If (delay_i is off) and (hold_i is not off) then (tmd $= 1$)(delay $= 0$)

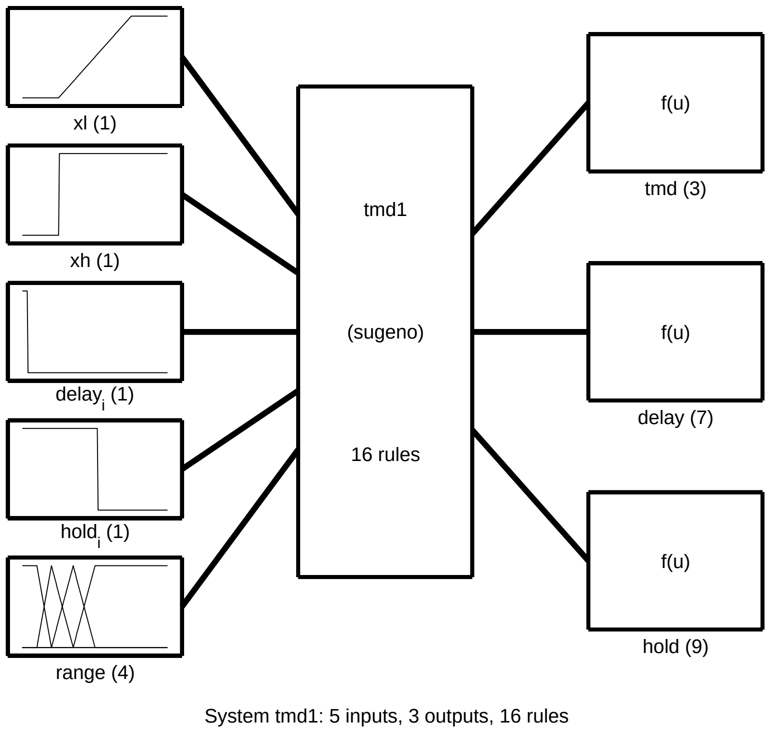


Figure G.7: Block diagram of TMD module

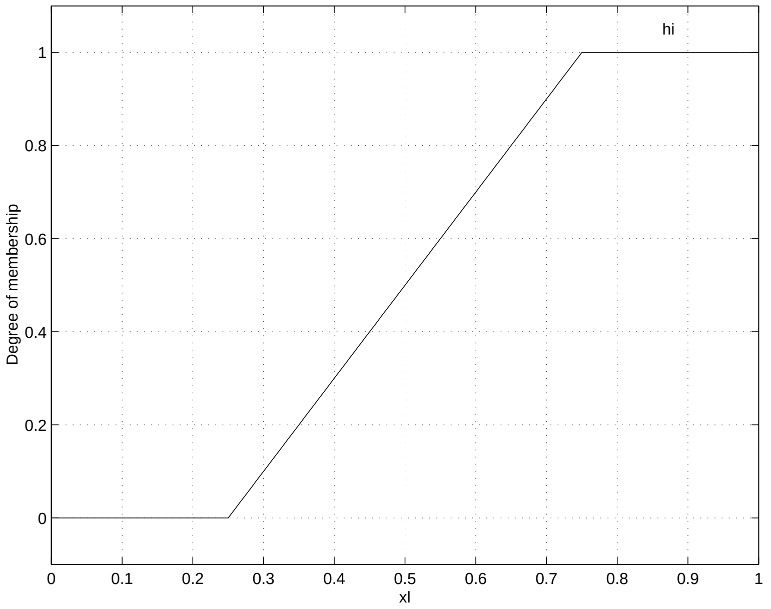


Figure G.8: Membership function for the TMD module xl input

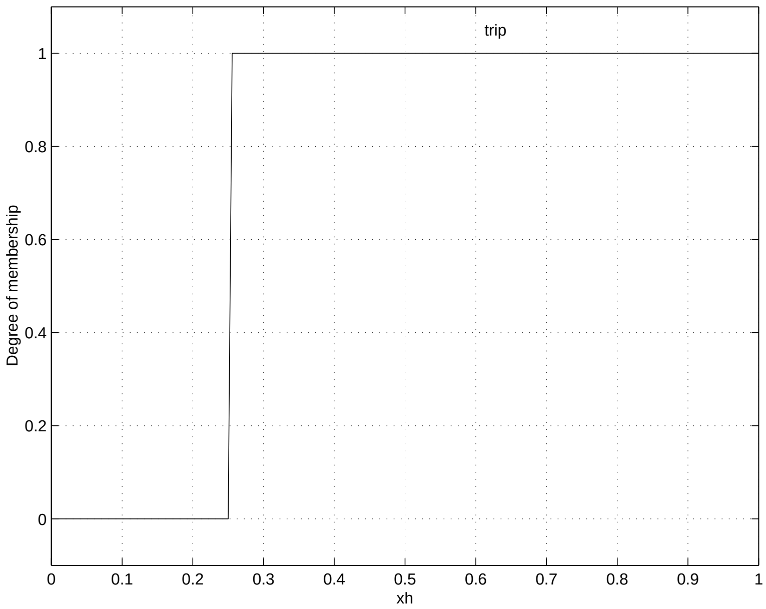


Figure G.9: Membership function for the TMD module xh input

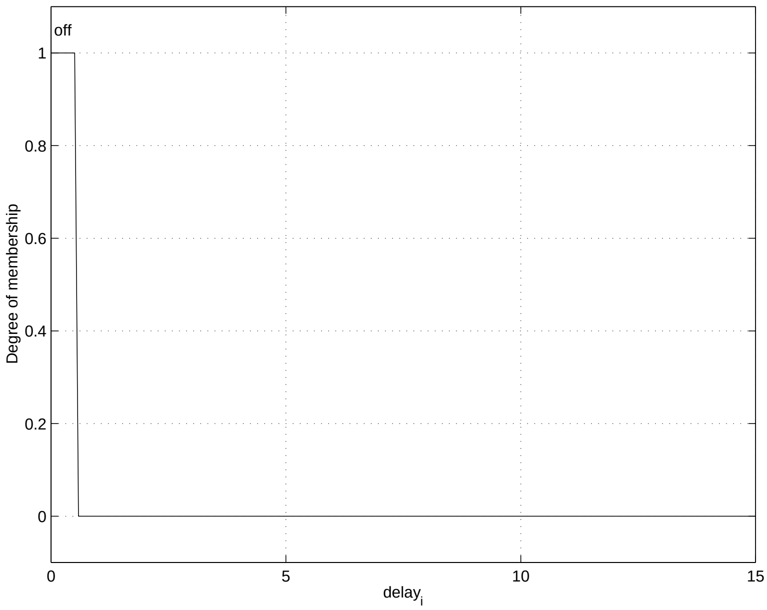


Figure G.10: Membership function for the TMD module delay$_i$ input

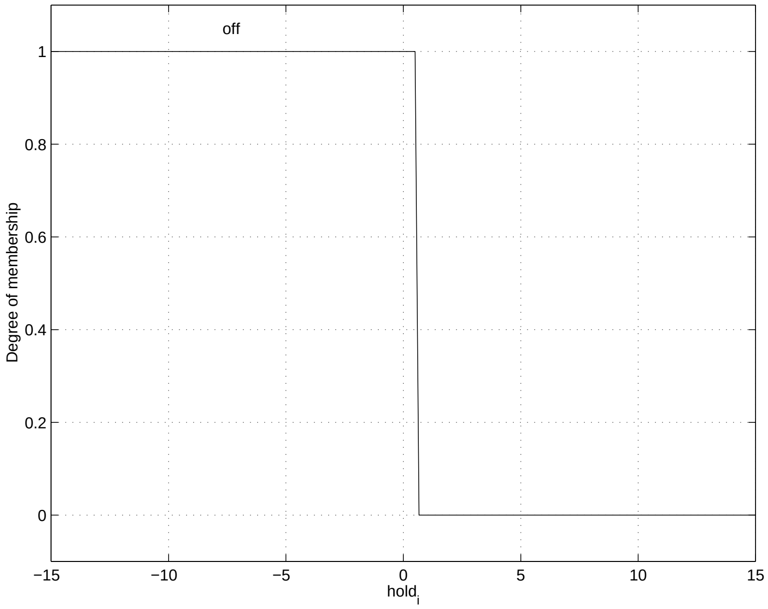


Figure G.11: Membership function for the TMD module hold$_i$ input

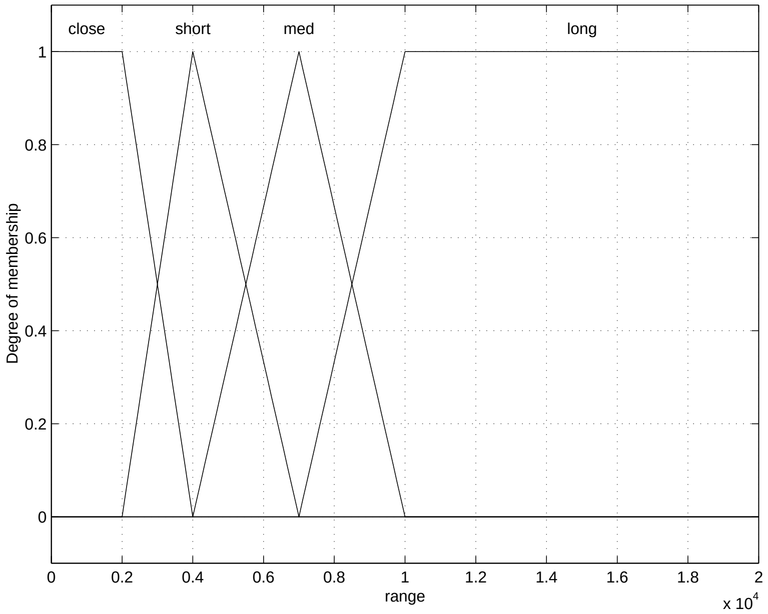


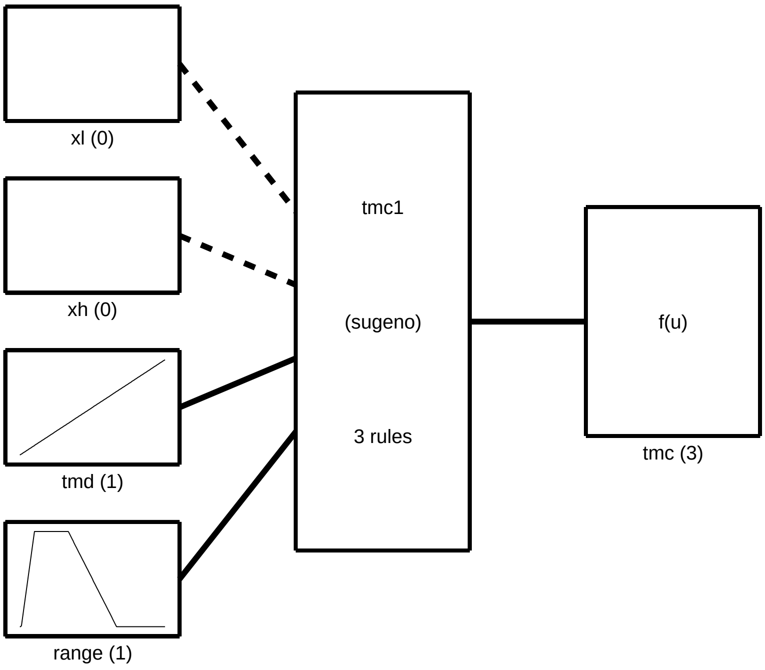
Figure G.12: Membership function for the TMD module range input

12. If (xl is not hi) and (delay$_i$ is not off) then (delay $=$ delay$_i - 1$)
13. If (xl is hi) and (hold$_i$ is not off) then (hold $=$ hold$_i - 1$)
14. If (xl is not hi) and (xh is not trip) and (delay$_i$ is off) then (hold $=$ hold$_i$)
15. If (xl is hi) and (hold$_i$ is off) then (delay $= 0$)(hold $= 0$)
16. If (xl is hi) and (delay$_i$ is not off) then (delay $= 0$) (hold $= -$hold$_i$)

The membership functions and the levels for the turn-on delay and hold have been chosen empirically by observation.

G.5. Detail of the TMC Confidence module

Figure G.13 shows the top-level structure of the TMC, target manoeuvre confidence module. The module is a Sugeno type fuzzy inference system. Figures G.14 and G.15 shows the membership functions for the *tmd* and range inputs. There are no membership functions associated with the xl and xh inputs. There are three rules in the system. They are as follows:



System tmc1: 4 inputs, 1 outputs, 3 rules

Figure G.13: Block diagram of TMC module

1. If (range is not good) then (tmc $= 0$)
2. If (tmd is not hi) and (range is good) then (tmc $= 0.5xl - 0.5xh + 0.5$)

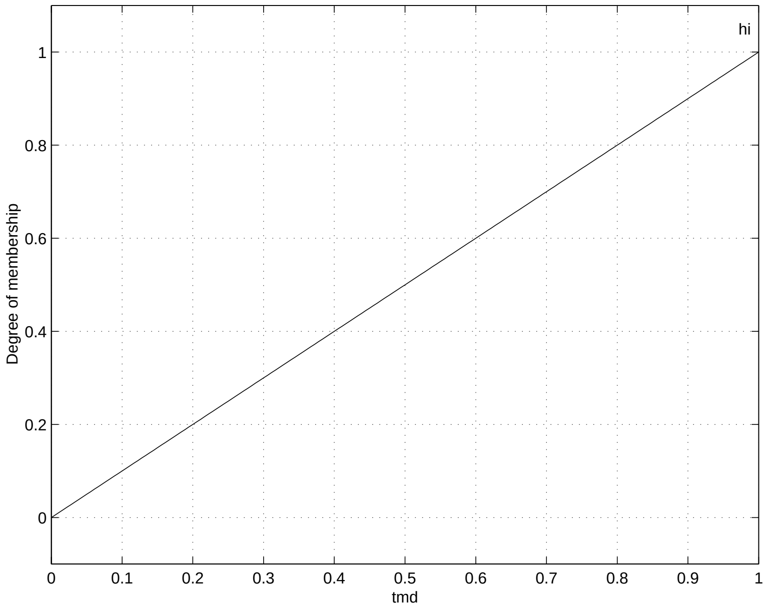


Figure G.14: Membership function for the TMC module manoeuvre detect input

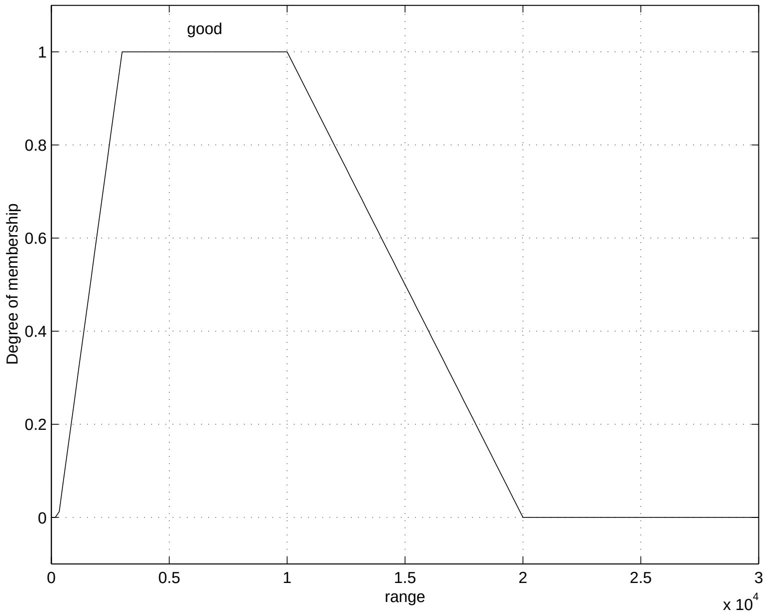


Figure G.15: Membership function for the TMC module range input

3. If (tmd is hi) and (range is good) then (tmc $= -0.5xl + 0.5xh + 0.5$)

The membership function for the range and the output functions have been chosen empirically by observation.

G.6. Initialisation Code

```
% Target Manoeuvre Detect Software
%
% E.J.Hughes 20/3/98
%
%% Initialisation

%% Load Fuzzy inference system modules
    global stage1 stage2 tmdfuzz tmcfuzz k tmx
    stage1=readfis('stage1');
    stage2=readfis('stage2');
    tmdfuzz=readfis('tmd1');
    tmcfuzz=readfis('tmc1');
%% correction for source pwr and noise range
    k=sqrt(100)/4/pi/(10000^2);
    tmx=[0 0 0];
```

G.7. Detector Main Function – tmdetect.m

```
% Target Manoeuvre Detect Software
%
% E.J.Hughes 20/3/98
%
%% Detector Function
%
% inputs berr_h and berr_v are a frame of
% bore-sight error samples (typ. 10).
%
% s_rcs is the corresponding frame of the
% radar cross section sum signal (complex).
%
% range is the mean range of the frame.
%
% Output tmd is the detection signal,
% zero no manoeuvre, one is manoeuvre.
%
% Output tmc is a confidence measure and
% lies in the range [0,1].

function [tmd,tmc]=tmdetect(berr_h,berr_v,s_rcs,range)

global stage1 stage2 tmdfuzz tmcfuzz k tmx

fsize=length(berr_h); % frame size

frame=[berr_h' berr_v' s_rcs'];
curr=frame(:,1:2);  %berr signals
s=abs(frame(:,3));  %sum signal
s=mean(s);
```

```
currx=curr*s;                                   %correct for RCS
currx=(currx-ones(fsize,1)*mean(currx))/sqrt(2)/k;%normalise
df=curr(2:fsize,:)-curr(1:(fsize-1),:);
dfx=currx(2:fsize,:)-currx(1:(fsize-1),:);  % differentials
ss=std(curr);
sp=std(df);
ssx=std(currx);
spx=std(dfx);   % std deviations
dx=[ssx' ss';spx' sp'];
dx=dx.*(dx<4.9)+4.9*(dx>=4.9);    %crop
dat1=evalfis([dx ones(4,1)*range],stage1);  % fuzzy pre-process
dx=dat1.*(dat1<4.9)+4.9*(dat1>=4.9);        %crop
dat=evalfis(dx',stage2);                    % fuzzy combine
tmx=evalfis([dat tmx(2:3) range],tmdfuzz);  % detect (tmd)
tmd=tmx(1);
tmc=evalfis([dat tmx(1) range],tmcfuzz);    % confidence (tmc)
```

G.8. Pre-Processor Module – stage1.fis

```
[System]
Name='stage1'
Type='sugeno'
NumInputs=3
NumOutputs=1
NumRules=4
AndMethod='prod'
OrMethod='probor'
ImpMethod='min'
AggMethod='max'
DefuzzMethod='wtaver'

[Input1]
Name='s_n'
Range=[0 5]
NumMFs=0

[Input2]
Name='s_r'
Range=[0 5]
NumMFs=0

[Input3]
Name='range'
Range=[0 20000]
NumMFs=4
MF1='end':'trapmf',[-500 0 500 1000]
MF2='close':'trimf',[500 1000 3000]
MF3='short':'trimf',[1000 3000 5000]
```

```
MF4='long':'trapmf',[3000 5000 20000 30000]

[Output1]
Name='out'
Range=[0 1]
NumMFs=4
MF1='a_long':'linear',[1 0 0 0]
MF2='b_short':'linear',[0 100 0 0]
MF3='b_close':'linear',[0 30 0 0]
MF4='b_end':'linear',[0 10 0 0]

[Rules]
0 0 4, 1 (1) : 1
0 0 3, 2 (1) : 1
0 0 2, 3 (1) : 1
0 0 1, 4 (1) : 1
```

G.9. Combiner Module – stage2.fis

```
[System]
Name='stage2'
Type='mamdani'
NumInputs=4
NumOutputs=2
NumRules=8
AndMethod='prod'
OrMethod='probor'
ImpMethod='prod'
AggMethod='sum'
DefuzzMethod='centroid'

[Input1]
Name='h'
Range=[0 5]
NumMFs=1
MF1='hi':'trapmf',[1.5 3 5 6]

[Input2]
Name='v'
Range=[0 5]
NumMFs=1
MF1='hi':'trapmf',[1.5 3 5 6]

[Input3]
Name='dh'
Range=[0 5]
NumMFs=1
MF1='hi':'trapmf',[3 4.5 5 6]
```

```
[Input4]
Name='dv'
Range=[0 5]
NumMFs=1
MF1='hi':'trapmf',[3 4.5 5 6]


[Output1]
Name='xl'
Range=[-1 2]
NumMFs=2
MF1='no':'trimf',[-1 0 1]
MF2='yes':'trimf',[0 1 2]

[Output2]
Name='xh'
Range=[-1 2]
NumMFs=2
MF1='no':'trimf',[-1 0 1]
MF2='yes':'trimf',[0 1 2]

[Rules]
 1  1  1  1,1 0 (1) 2
-1 -1 -1 -1,2 0 (1) 1
 0  0 -1 -1,0 1 (1) 1
-1 -1  0  0,0 1 (1) 1
 0  1  0  1,0 2 (1) 1
 1  0  1  0,0 2 (1) 1
 1  0  0  1,0 2 (1) 1
 0  1  1  0,0 2 (1) 1
```

G.10. Detector Module – tmd1.fis

```
[System]
Name='tmd1'
Type='sugeno'
NumInputs=5
NumOutputs=3
NumRules=16
AndMethod='prod'
OrMethod='probor'
ImpMethod='min'
AggMethod='max'
DefuzzMethod='wtsum'

[Input1]
Name='xl'
Range=[0 1]
```

```
NumMFs=1
MF1='hi':'trapmf',[0.25 0.75 1 2]

[Input2]
Name='xh'
Range=[0 1]
NumMFs=1
MF1='trip':'trapmf',[0.25 0.251 1 2]

[Input3]
Name='delay_i'
Range=[0 15]
NumMFs=1
MF1='off':'trapmf',[-1.5 0 0.5 0.55]

[Input4]
Name='hold_i'
Range=[-15 15]
NumMFs=1
MF1='off':'trapmf',[-20 -15 0.5 0.55]

[Input5]
Name='range'
Range=[0 20000]
NumMFs=4
MF1='close':'trapmf',[-1000 0 1000 5000]
MF2='short':'trimf',[1000 5000 10000]
MF3='med':'trimf',[5000 10000 15000]
MF4='long':'trapmf',[10000 15000 20000 30000]

[Output1]
Name='tmd'
Range=[0 1]
NumMFs=3
MF1='off':'constant',0
MF2='on':'constant',1
MF3='def':'constant',0

[Output2]
Name='delay'
Range=[0 5]
NumMFs=7
MF1='off':'constant',0
MF2='short':'constant',0.5
MF3='med':'constant',1.75
MF4='long':'constant',2.5
MF5='dec':'linear',[0 0 1 0 0 -1]
MF6='def':'constant',0
MF7='shortish':'constant',1.0
```

```
[Output3]
Name='hold'
Range=[-15 15]
NumMFs=9
MF1='off':'constant',0
MF2='short':'constant',4
MF3='med':'constant',6
MF4='long':'constant',10
MF5='dec':'linear',[0 0 0 1 0 -1]
MF6='coast':'linear',[0 0 0 1 0 0]
MF7='v_short':'constant',2
MF8='def':'constant',0
MF9='force':'linear',[0 0 0 -1 0 0]

[Rules]
0 1 0 1 4, 3 2 8 (1) : 1
0 1 0 0 4, 3 6 4 (1) : 1
0 1 0 1 3, 3 7 8 (1) : 1
0 1 0 0 3, 3 6 3 (1) : 1
0 1 0 1 2, 3 3 8 (1) : 1
0 1 0 0 2, 3 6 2 (1) : 1
0 1 0 1 1, 3 4 8 (1) : 1
0 1 0 0 1, 3 6 7 (1) : 1
0 0 0 1 0, 1 1 1 (1) : 1
0 0 -1 -1 0, 1 6 8 (1) : 1
0 0 1 -1 0, 2 1 8 (1) : 1
-1 0 -1 0 0, 3 5 8 (1) : 1
1 0 0 -1 0, 3 6 5 (1) : 1
-1 -1 1 0 0, 3 6 6 (1) : 1
1 0 0 1 0, 3 1 1 (1) : 1
1 0 -1 0 0, 3 1 9 (1) : 1
```

G.11. Confidence Module – tmc1.fis

```
[System]
Name='tmc1'
Type='sugeno'
NumInputs=4
NumOutputs=1
NumRules=3
AndMethod='prod'
OrMethod='probor'
ImpMethod='min'
AggMethod='max'
DefuzzMethod='wtsum'

[Input1]
```

```
Name='xl'
Range=[0 1]
NumMFs=0

[Input2]
Name='xh'
Range=[0 1]
NumMFs=0

[Input3]
Name='tmd'
Range=[0 1]
NumMFs=1
MF1='hi':'trimf',[0 1 2]

[Input4]
Name='range'
Range=[0 30000]
NumMFs=1
MF1='good':'trapmf',[300 3000 10000 20000]

[Output1]
Name='tmc'
Range=[0 1]
NumMFs=3
MF1='off':'constant',0
MF2='wait':'linear',[0.5 -0.5 0 0 0.5]
MF3='lock':'linear',[-0.5 0.5 0 0 0.5]

[Rules]
0 0 0 -1, 1 (1) : 1
0 0 -1 1, 2 (1) : 1
0 0 1 1, 3 (1) : 1
```