RADAR CROSS SECTION MODEL OPTIMISATION USING GENETIC ALGORITHMS

E J Hughes and M Leyland

Royal Military College of Science, Cranfield University, UK

ABSTRACT

Many missile–target simulation systems use random numbers to mimic the effects of a fluctuating target Radar Cross Section (RCS) in an attempt to minimise simulation times. In this paper a Genetic Algorithm is used to optimise the complexity of a point-scatterer model with a realistic radar cross section, ultimately allowing real measured data to be used in simulations. The performance of the genetic algorithm is compared against an iterative optimisation method and a known optimum solution. The radar cross section models described in this paper are designed to be used with a synthetic homing guidance missile in 3-dimensional virtual engagement scenarios. The models allow measured RCS data to be combined with synthetic RCS details, creating a realistic target radar cross section with 4π steradian coverage.

INTRODUCTION

It is seldom practical to use real radar cross section data with sufficient resolution due to the massive data storage requirements involved. A series of point scatterer models can be fitted to high resolution RCS data (1, 2) and then combined using a Binary Space Partition Tree structure (3, Pages 675–680) to allow the correct point scatterer model to be rapidly retrieved for any aspect angle. Any gaps in the coverage of the measured data can be filled with synthetically generated RCS information by including further point scatterer models in the tree structure. Each point scatterer model is typically valid for only a few degrees of coverage, therefore many individual models are required to cover 4π steradians.

The electric field from a point scatterer model at wavelength λ may be described as shown in equation 1.

$$\xi = \sum_{k=1}^{n} a_k \, e^{j\left(\frac{4\pi d_k}{\lambda} + \phi_k\right)} \quad (1)$$

$$\sigma = |\xi|^2 \quad (2)$$

The field returned from the target is a complex quantity with magnitude and phase. It is defined as the coherent sum of the echos from the n scatterers, each scatterer with its own magnitude, a_k, and phase, ϕ_k, and at a distance d_k from the observation point. The radar cross section, σ, is a scalar quantity and represents the effective area of the target in square metres and is defined by equation 2.

The model generation process involves first identifying the scatterer locations (1, 2) and then weighting each scatterers amplitude and phase using Least-Squares Fitting Methods (4, Pages 509–520) to reproduce the required field. The process can yield models with large numbers of scatterers. This large amount of model data can create handling and storage problems, with correspondingly long simulation times.

If we accept that a measured or calculated field will never be a perfect representation of the real target (5, 6), small degradations in data fidelity are acceptable. Therefore, if we remove some of the scatterers in an n-point model, we should be able to re-adjust the model to give an approximation to the desired field pattern. A simple iterative reduction method may be used that removes each scatter in turn from the n point model and fits weights to the remaining $(n-1)$ point model. The scatter whose removal caused the least error in the new field pattern is discarded. The process is repeated until the field and therefore the radar cross section becomes unacceptable.

As RCS is governed by scatterer interactions, the iterative method detailed may not always select the best combination of scatterers. A small scatterer may have little effect on its own but may be dominant when paired with another similar scatterer. For a small model with 20 scatterers, there are $2^{20} = 1048576$ possible combinations to search to find the optimum solution.

An exhaustive search of all possible model combinations in an attempt to find an optimum solution is often impractical. A *Genetic Algorithm* (GA) (7, 8) has been developed in an attempt to identify efficiently, good combinations of scatterers. Similar large combinatorial problems exist when attempting to thin linear arrays. Genetic algorithms have been applied to this problem area with great success (9, 10).

OPTIMISATION USING ITERATIVE REDUCTION

To achieve an optimised model, the required number of least influential scatterers has to be removed and the remaining points adjusted in an attempt to compensate for the loss. The iterative optimisation method developed is very fast and produces reasonable reduced models. The Least-Squares Method is used to fit scatterer amplitude and phase data whilst keeping the locations fixed. The scatterer that has the least influence is removed and a new set of coefficients fitted to compensate for the loss. The process is repeated, systematically reducing the model. The mean squared error of the optimised model field pattern compared with the original required pattern is calculated at each stage by applying equation 3, allowing the effects of the reduction to be monitored for the current region of optimisation. In equation 3, r_i is the complex field of the optimised model at point i; g_i is the required complex field; N is the number of data samples and E is a measure of the error.

$$E = \frac{1}{N} \sum_{i=1}^{N} |r_i - g_i|^2 \tag{3}$$

OPTIMISATION USING GENETIC ALGORITHMS

Genetic Algorithms 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*. This is the genetic description of the solution and may be broken into n sections called *genes*. Each gene representing a single parameter in the problem, 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 individuals 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 a 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 numerical optimisation problems, genetic algorithms that have a large population 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.

ALGORITHM CONSTRUCTION

For a 20-point point scatterer model, a chromosome with 20 genes has been used to define the model structure. Each gene can take the values 0 or 1 and corresponds to a specific scatterer in the model. If a gene is '1', the corresponding scatterer is present in the model, if it is '0', the scatterer is omitted. Scatterer locations are kept fixed, magnitudes and phases are fitted to the selected points using the Least Squares method each time a new individual is created.

The GA 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. This 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. Create a random chromosome for any offspring with null chromosomes (all genes zero).
8. Offspring become new population, assess the performance of each individual.
9. Record best individual.
10. Repeat from step 3 for required number of generations.

The performance of each individual is calculated by first fitting weightings and phases using the Least Squares Method to the selected scatterers defined by its chromosome and then generating the N field pattern data samples for the region of optimisation. The generated and required field patterns are then compared using the cost function previously shown in equation 3 to give a measure of the error, E.

The genetic algorithm is designed to search for the best solutions for models with a specific number of points. This presents us with two objectives, to minimise the error, E, and to find a model with a specific number of scatterers. The algorithm is intended to be run repeatedly in an attempt to identify the best solutions for each size of model. These results will form a *Pareto Optimal Set* (11, Pages 197–201) where no single solution is better than any of the others when both objectives are taken into account. For example, models with large numbers of scatterers are undesirable but have a low error, while small models tend to have higher error but are more desirable for processing purposes.

An optimisation method called *Goal Attainment* may be used to combine the two requirements into a single objective for the algorithm to process. If we are trying to minimise the two functions, the method works by comparing them to some pre-determined goals and normalising the results. The maximum of the two values is then selected as the single objective to be minimised. This has the effect of pushing the functions down towards the set goals. The dominant function is therefore always the one that is proportionally furthest from its goal. The goals are set as the error, E, and number of scatterers achieved with the iterative method.

The standard goal attainment function has been modified in this algorithm to allow the number of scatterers in the model to be specified more precisely. An objective value, $O(x)$ is calculated using a modified goal attainment multi-objective optimisation approach as shown in equation 4. This will reduce both the error and size of the model until a model of the correct size or smaller is found. The objective will then be based purely on field pattern error if the correct number of scatterers are present in the model. The two functions to be minimised are as described in equation 5. The function C_1 is related to the difference in the modelled and required field patterns and C_2 is related to the number of scatterers in the model.

$$O(x) = \begin{cases} \max(C_1(x), C_2(x)) & , C_2(x) > 0 \\ C_1(x) & , C_2(x) \leq 0 \end{cases} \quad (4)$$

$$\begin{aligned} C_1(x) &= \frac{E - G_1}{G_1} \\ C_2(x) &= \frac{\sum_{j=1}^{20} \mathcal{X}(x)_j - G_2}{G_2} \end{aligned} \quad (5)$$

In equation 5, E is the error between the modelled and required field pattern, $\mathcal{X}(x)_j$ is the j^{th} gene of the 20-gene chromosome of individual x and G_1, G_2 are the cost goals that the optimisation is to aim for.

The fitness value $F(x)$ is assigned according to rank position p_x of individual x. The individual with the greatest $O(x)$ (least fit) being assigned a rank position of 1 and the best individual being assigned rank position M. Equation 6 details the calculation of $F(x)$.

$$F(x) = \frac{2(p_x - 1)}{M - 1} \quad (6)$$

A technique called Stochastic Universal Sampling (7, Page 57) is used to select M individuals at random from the population for breeding. Each individual has a probability of selection defined in equation 7.

$$\text{Prob}(x \text{ selected}) = \frac{F(x)}{\sum_{i=1}^{M} F(i)} \quad (7)$$

The individuals selected are paired up for breeding. Which individual pairs with which other is chosen at random to ensure a good genetic mix. A method of generating new chromosomes is used called Single Point Crossover (7, Page 38). Two new offspring are generated from each pair of parents by swapping sections of their chromosomes. A position in one parents chromosome is randomly chosen, dividing it into two sections. The second parents chromosome is then also divided at the same position. The chromosomes of the two parents may now be represented as the strings of genes $[a_1\ a_2]$ and $[b_1\ b_2]$. The chromosomes of the two offspring may now be defined as $[a_1\ b_2]$ and $[b_1\ a_2]$.

The GA is terminated after 100 generations and the best overall individual is recorded as the solution. A population of $M = 25$ individuals was used for the experiments, therefore giving 2500 objective calculations for each run of the GA. As there are 20 different model sizes, 50,000 objective calculations are required to generate a set of models, compared to a total of 209 for the iterative method and $2^{20} = 1048576$ for an exhaustive search. The exhaustive search method is computationally more expensive for problems with 16 or more scatterers.

RESULTS

Figure 1 shows the results of the fitting processes on the RCS as the model is thinned with a $\pm 1^\circ$ region of optimisation. The comparisons are made using the *Kolmogorov-Smirnov (K–S)* statistical test (4, Pages 472–475) with 800 sample points in each data set. This test compares cumulative distribution

curves and allows comparisons between the fitting process over different angular intervals to be compared easily. A $\pm 1^\circ$ angle of interest is chosen for demonstration purposes as this allows a relatively small 20-point model to be used. The 5% and 20% levels of significance (*fair* and *high*) have been drawn on the graph for comparison.

For the iterative method, the K–S values of the eleven and twelve point models straddle the two significance levels. The better model (with twelve points) is chosen as the optimised model. Figure 2 shows the field pattern plots from the twelve point reduced model to indicate resulting fidelity. The solid line denotes the pattern from the twelve point model while the broken line is the original field.

The Genetic Algorithm performs significantly better. A *fair* model may be represented by six scatterers and a *good* model by eight, rather than the twelve point model chosen by the iterative method. The results are compared against the ideal results generated by performing an exhaustive search for the 20-point model. Table 1 summarises the optimisation results.

TABLE 1 - Best Model Sizes For Different Reduction Methods

K–S Fit	No. of Scatterers		
	Ideal	GA	Iterative
Fair (K–S < 1.3580)	6	7	12
High (K–S < 1.0727)	7	8	12

CONCLUSIONS

Point scatterer models allow complex RCS patterns to be recreated quickly and efficiently but fitting real RCS data to point scatterer models may result in models with very large numbers of scatterers.

A model with a large number of scatterers may be optimised by removing the least significant points and adjusting the weights on the others to compensate for the loss. An iterative optimisation method where the scatterer that has least effect is removed at each cycle can provide a quick but crude model reduction. A Genetic Algorithm has been proposed that provides superior solutions but at the expense of increased processing overhead. The computation time is $\frac{1}{20}$ of that required for an exhaustive search of a 20-point model. Slightly larger models should not require larger populations or increased numbers of generations to achieve good results, therefore, for a 30-point model, the computational effort would be $\frac{1}{21475}$ of the exhaustive search, a significant saving. The optimisation tests suggest that although the theoretical minimum number of scatterers needed to represent a 3-dimensional target is four, a *realistic* reproduction of the RCS can be generated with six or more.

REFERENCES

1. Bhalla, R. and Ling, H., 1996. "Three-dimensional scattering centre extraction using the shooting and bouncing ray technique". IEEE Transactions on Antennas and Propagation, 44(11), 1445–1453.

2. Breuille, C. L. and Caille, E., 1994. "Modeles multi-points brillants de signatures radar de cibles". In Radar International Conference, pages 325–329, Paris, France. Soc. Electr. & Electron.

3. Foley, J. D., van Dam, A., Feiner, S. K., and Hughes, J. F., 1990. "Computer Graphics: Principles and Practice". Addison-Wesley, 2nd edition.

4. Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T., 1989. "NUMERICAL RECIPES The Art Of Scientific Computing (FORTRAN Version)". Cambridge University Press.

5. Turner, S. D., 1990. "RESPECT: rapid electromagnetic scattering predictor for extremely complex targets.". IEE Proceedings, Part F, 137(4), 214–220.

6. Williams, J. M., 1990. "Radar cross-section of large structures with complex microgeometry.". IEE Proceedings, Part F, 137(4), 221–228.

7. Michalewicz, Z., 1992. "Genetic Algorithms + Data Structures = Evolution Programs". Springer–Verlag, 2nd edition.

8. Heitkötter, J. and Beasley, D., 1996. "The hitch-hiker's guide to evolutionary computation: A list of frequently asked questions (FAQ)". USENET:comp.ai.genetic, available from anonymous ftp at rtfm.mit.edu:/pub/usenet/news .answers/ai-faq/genetic/.

9. Haupt, R. L., 1994. "Thinned arrays using genetic algorithms". IEEE Transactions on Antennas and Propagation, 42(7), 993–999.

10. Haupt, R. L., 1995. "An introduction to genetic algorithms for electromagnetics". IEEE Antennas and Propagation Magazine, 37(2), 7–15.

11. Goldberg, D. E., 1989. "Genetic Algorithms in Search, Optimization, and Machine Learning". Addison-Wesley Publishing Company, Inc.

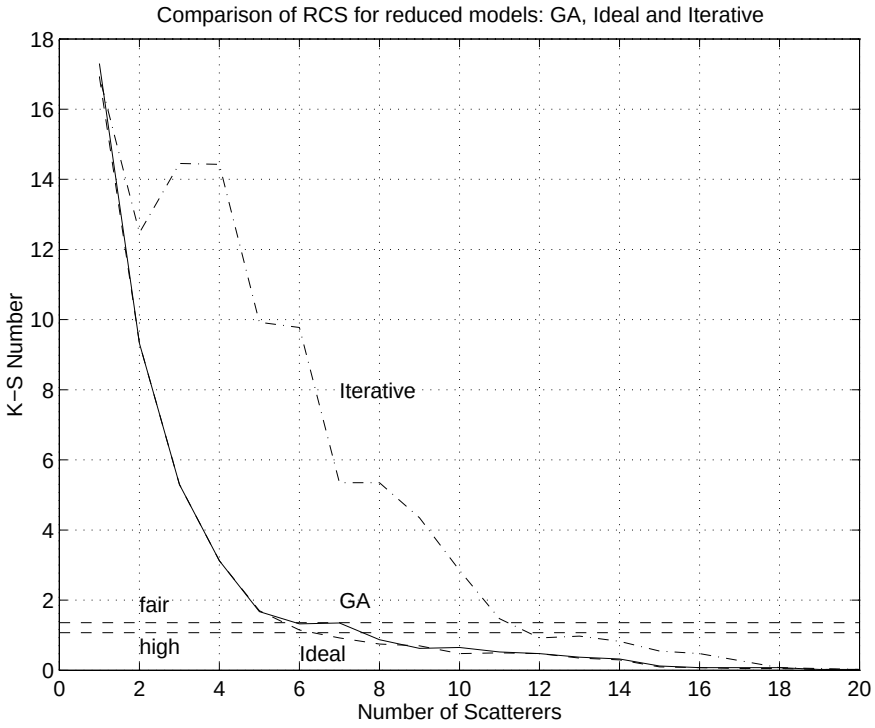

Figure 1: Comparison of Reduction Methods and Ideal Curve

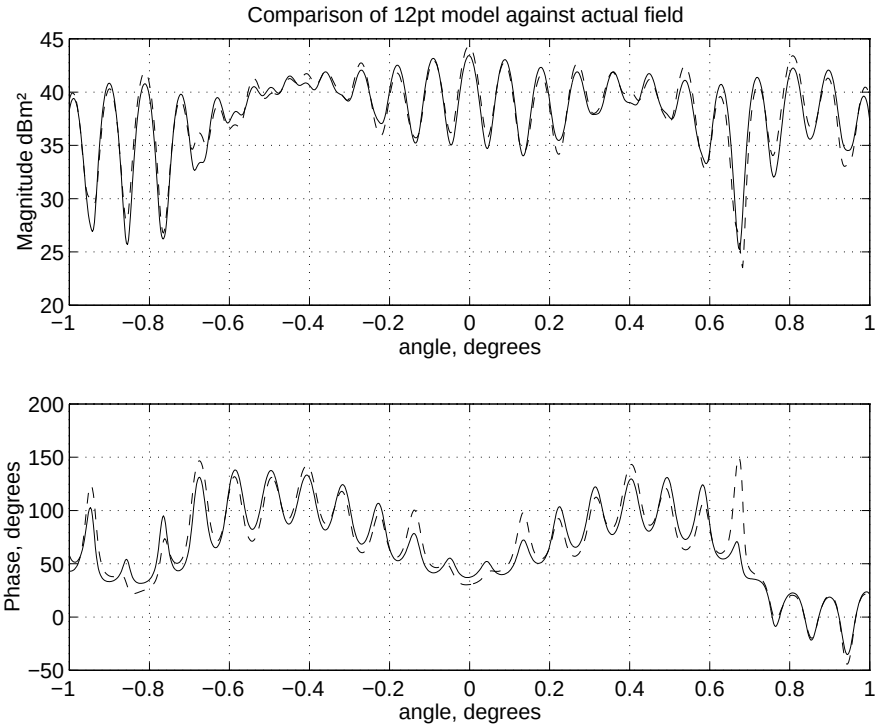

Figure 2: Comparison of 12 point model and required field pattern