Progress In Electromagnetics Research, PIER 49, 71–86, 2004

BAYESIAN OPTIMIZATION TECHNIQUES FOR ANTENNA DESIGN

M. J. Inman, J. M. Earwood, A. Z. Elsherbeni and C. E. Smith

Center for Applied Electromagnetics Systems Research
Department of Electrical Engineering
University of Mississippi
University, MS 38677, USA

Abstract—Optimization and parameter estimation techniques have been employed for many years as a method of improving and exploring designs in numerous areas. As the designs of antennas and antenna arrays become more complex in nature, optimization techniques such as Bayesian estimation or genetic algorithms have become more necessary in the design process. These techniques provide methods for not only the design process, but also for operation simulations such as element failure corrections as well. This paper will deal with Bayesian optimization techniques for antenna and antenna array design as an alternative to other techniques. Through the use of Bayesian inference techniques, probability and information theory can be applied to a design problem to improve the operation within a range of specifications. Examples provided show that how this method allows for the examination of an entire parameter space of a linear array so that the best fitting solutions can be quickly and efficiently examined and improvements can be implemented.

1. INTRODUCTION

In the course antenna and antenna array design, often many parameters and variables need to be taken into account to achieve proper operation of the design in order to meet the design criteria. In many antenna design methodologies simplification is achieved by holding certain sets of parameters constant, such as the element spacing in an array, so that other parameters, such as the element phasing and amplitudes, may be determined independently. In the past, analytic approximation techniques, such as Fourier or Woodward-Lawson synthesis, have been used to assist in the design. These methods, however, can be lacking in the flexibility that is often required in many cases. Conversely, trying to fully explore a design with many parameters becomes difficult due to computational limitations once the number of parameters becomes large. As more parameters are added to a design, simplification techniques used to find an answer can become a huge disadvantage. This leads to designs that can be larger, more costly, and less efficient.

One approach that many researchers have explored is the use of genetic algorithms to search for the best fitting parameters of a design. These methods attempt to use the principles of generic evolution to try to "evolve" the parameters to find the best possible design parameters. This method is not based on any mathematical theory and employs decisions based on anecdotal observations. While there has been much improvement on the technique recently, there is still a great deal left unknown about it. Bayesian parameter estimation attempts to accomplish many of the same goals as genetic algorithms, while trying to base all the decisions in the optimization process on probability and information theory. A parallel will be drawn between generic and Bayesian techniques to compare and contrast their implementation. The Bayesian estimation technique will be demonstrated to have the ability to computationally explore a much wider range of parameters, in a faster and more complete manner, than would be possible using traditional and other alternative algorithms. In addition to detailing the basic techniques used in Bayesian estimation, a few improvements that enable faster convergence, such as the use of a space-filling curve and simulated annealing will be explained.

2. PARAMETER ESTIMATION FUNDAMENTALS

The goal of any estimation or optimization technique is to formulate an efficient method to navigate through a large parameter space in order to find the best set of parameters. In most techniques this involves either an iterative approach of discretely examining the parameters or a "random walk" through the parameters as a way to explore the parameter space. Random walk methods have been employed in statistics and information theory for many years as a way to effectively sample or explore unknown distribution spaces. If done correctly the "sampled" space will be able to represent the entire parameter space with only its limited samples. In genetic algorithms this sampling is achieved by combining the "genes" of two selected parents and then performing a "mutation". The hope is that by picking two parent parameter sets who each have fairly good results, a combination of them will achieve even better results. A random "mutation" can be included that allows for a certain degree of randomness to be included. The actual implementation of this can vary many different ways. In Bayesian estimation a similar process will be done, but instead of using evolutionary metaphors to randomly sample, a sampling technique chosen from information theory will be used to move around our space. As specified by the tenets of information theory, to sample correctly there must be a sufficient number of independent samples from the parameter space. The different techniques to sample from the parameter space have varying degrees of speed to gather independent samples. As might be noticed, what is being performed is a form of Markov chain Monte Carlo (MCMC) simulation. The chains are being assembled through the use of the random walk samples. Older random walk techniques such as Metropolis or Gibbs [1] require many iterations of sampling before a new sample can be considered independent. So much care has to be given to choosing an efficient method of sampling. Genetic algorithms can also share this same problem even though it may not be so formally addressed.

Once a new set of parameters has been chosen, the next step is to evaluate the "fitness" of the parameter set. In the case of designing antennas, the fitness is related to how closely the parameter set allows the antenna to perform as was specified in the design criteria. This, perhaps, could be how close it comes to matching a specific radiation pattern or perhaps shows certain sidelobe levels. In the case of Bayesian estimation, the evaluation of the parameter set will give a number related to the probability that the parameters are the correct ones. By using related probability, the user will have at the end some degree of certainty with the results. This is an important advantage

over genetic algorithms that will simply provide an answer with no measure of confidence in the assumption that the returned data is the best set possible.

In both techniques these two steps are done in parallel with many separate and independent sets performed simultaneously. This allows for both a more complete and a faster exploration of the parameter space and also for an easily parallelizable code. After the evaluation of the various sets has taken place, a final step is used to allow for better use of the number of sets available. In genetic algorithms this becomes a "survival" step in which many, if not all, of the "weaker" or less fit sets are thrown away in favor of repeating some of the stronger parameter sets. There are many and varied implementations of this in genetic algorithms, which can lead to the loss in the efficiency gained by this step, or even prevent the technique from converging if not done correctly. In the Bayesian techniques, this can be achieved by importance or weighted sampling of the various sets available. This type of resampling effectively allows the system to throw away some of the least likely sets using its own probability to determine if it stays. By allowing some of the sets with a low probability to remain, the system is allowed to explore a greater portion of the space. The sets that are replaced by ones with a higher probability then allow the system to focus more of its computations on areas that have a greater likelihood of producing high probability sets. The whole process is then repeated until some set final conditions are met.

It should be made clear that genetic algorithms often share many of the same concepts as Bayesian parameter estimation in that both are in essence Markov chain Monte Carlo methods as has been explained in [1]. However, genetic algorithms often make choices that are not based in any mathematical theory, but rather in broad mathematic metaphors to evolutionary processes. While Bayesian methods are based on information and sampling theory, as well as statistical methods.

3. PROGRAM ORGANIZATION

This section will detail one possible implementation of the Bayesian parameter estimation technique as well as some improvements to possible problems encountered in general optimization techniques. The technique will be similar to the order presented in the above section and is shown in Table 2.

The first step shown in the process is to initialize all the independent chains. During each iteration, the separate chains will move around the parameter space independent from one another. The

Table 1. Comparison of genetic and Bayesian techniques.

Step	Genetic Algorithms	Bayesian Estimation
Choosing of a parameter set to be evaluated	Combining the genes of selected parents then a possible random mutation	Use of an efficient sampling technique from Monte Carlo methods
Evaluation of Parameter Set	Various methods, up to programmer	Use of Bayesian techniques to establish a related probability
Resample Parameter Set	Various methods related some survivability of the fittest	Use of weighted or importance sampling
Exit Process	A parameter set passes below some arbitrarily set error level	The criteria for sampling theory has been met

Table 2. Program psuedo-code.

Step	
1.	Choose random starting points for n-number of independent chains
2.	Start simulated annealing loop with t=0
3.	Use slice sampling to move from old parameter set to new set
4.	Evaluate probabilities for all new parameter sets
5.	Resample parameter sets
6.	Increase annealing temperature parameter
7.	Repeat until annealing complete
8.	Use all collected samples to characterize parameter space and determine best answer

use of many separate chains allows the technique to simultaneously explore a greater area of the parameter space leading to faster convergence. Initially, each chain will be randomly assigned a position to start from.

The second step is the main loop from which the process will iterate across. In this particular implementation, a simulated annealing process is introduced. This process, detailed in Section 4, allows the

underlying probability distribution for the parameter space to be slowly introduced. This allows for a more complete sampling of the entire space without having to increase the number of samples.

The movement about the parameter space for each chain will be determined by the use of slice sampling [2]. In order to adequately characterize the entire parameter space, many independent samples are required. Slice sampling was developed by Neal as a highly efficient method to generate independent samples from a parameter space. It allows for independent samples to be generated with an order of magnitude fewer iterations than random-walk methods such as Metropolis or Gibbs. This efficiency actually increases as compared to Metropolis or Gibbs as the dimensionality is increased. If applied correctly in the ideal situation, every sample from the slice method would be independent. Once an independent sample has been generated from the parameter space, its characteristics will be compared with the ideal solution.

Slice sampling works by allowing the sampling routine to adaptively move around the area underneath the distribution (the area of interest), this is achieved by alternately slicing along the probability and the parameters. In other words, the sampler first chooses a target probability from the uniform space between 0 and the probability at the current parameter set as is represented by the vertical dotted line in Figure 1. The sampler then attempts to find a new point in the parameter space above that target probability. This is done by expanding outward from the original parameter set until a slice is created that all points within are of at least the new target probability shown by the horizontal dashed line in Figure 1. The sampler then chooses a point randomly from this slice, which then becomes the new parameter set.

After slice sampling has been completed, new sample points have been generated. These new points must be evaluated for their related

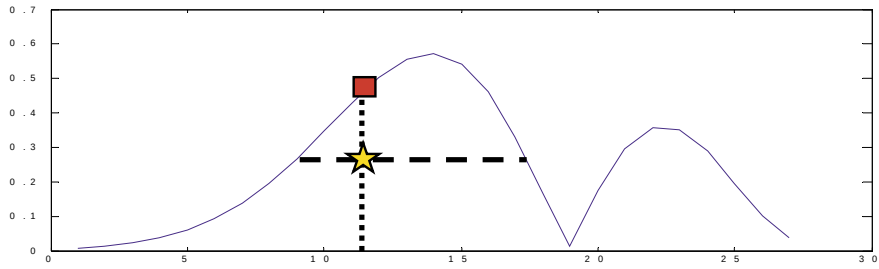


Figure 1. Graphical representation of slice sampling in one dimension.

probabilities. This is accomplished in this implementation via an energy function. This energy is related to the probability that its current position in the parameter space correctly allows the model to fit the data or specifications exactly. The range for this energy function is from $-\infty \rightarrow +\infty$ where parameter set that perfectly matches the data or desired results will return a $-\infty$. This energy is defined as,

$$E = \log\left[\sum D^2 + \sum M^2 - 2DM\right] \tag{1}$$

where D is the set of data points (ideal pattern, etc.) and M is the set of points that the model yields for the Markov chains current position in the parameter space. It should be noted that there is no requirement for the data points or the model points to be a real number, in many cases they can be complex. The final energy, however, will be a real number so it becomes important to understand what is happening with complex data.

The next step is to resample the parameter set in order to reduce the randomness and increase the effectiveness of the samples. In this case there is a collection of parameter sets each with an associated probability. Importance of sampling allows some of the chains with sets at a low probability to have their location replaced with those sets of higher probability. The replacement points are picked from the original collection of parameter sets in such a way that a point with a high probability of being the correct solution has a high probability of being picked as a replacement point. This process of importance sampling reduces the randomness of the sampled points and speeds convergence to the most probable solution.

Once the chains have been resampled, the "temperature" for the annealing process is increased slightly. This is an adaptive procedure that increases the annealing temperature based upon the current set of samples. This adaptive system allows more iterations while the chains are wandering around the parameter space and speeds up when they begin to converge. At this point the whole process starts over until full temperature has been reached.

When the process has completed the technique should return all of the sample points collected during the annealing process. If implemented correctly these samples should be sufficient in characterizing the entire probability space for the parameter space. If this is possible to be done, then simple statistical methods can be used such as taking mean or median in each parameter to find the set with the highest probability.

4. SIMULATED ANNEALING

In order to enhance the techniques ability to fully explore the parameter space the process of simulated annealing is utilized. This also has the added benefit of speeding up convergence for the technique as it allows for more effective sampling. This process is analogous to annealing metal, which is the slow cooling of a forged metal piece in order to make it stronger. The goal is to allow the Markov chains to freely explore the parameter space, without regard to local extremes, during the beginning of the simulation, yet have the majority of the chains located at the most probable solution(s) position(s) by the end of the simulation. In order to ensure full exploration, simulated annealing modifies the probability distribution, P, by scaling the returned energy function, E

$$P = E^{\lambda} \tag{2}$$

The power factor, λ, is a real number with a range [0,1] that is given by:

$$\lambda = \lambda_{old} + \left(\frac{R_c}{\sigma_p} \right) \tag{3}$$

where R_c is a user-defined constant with a magnitude of much less than one, and σ_p is the standard deviation of the energies of all the Markov chains.

At the beginning of the processes (akin to a hot metal), λ is equal to zero and the probability distribution is unity throughout. As the simulation progresses, σ_p becomes smaller and λ becomes larger at a rate defined by R_c, slowly allowing the underlying probability distribution to emerge. Once λ has reached or exceeded unity, the simulation ends. This helps prevent chains from getting stuck in local extremes. A graphical representation of this process is shown in Figure 2.

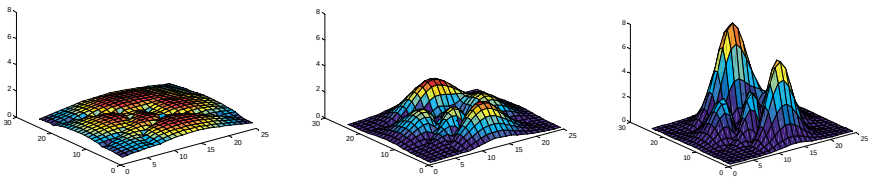
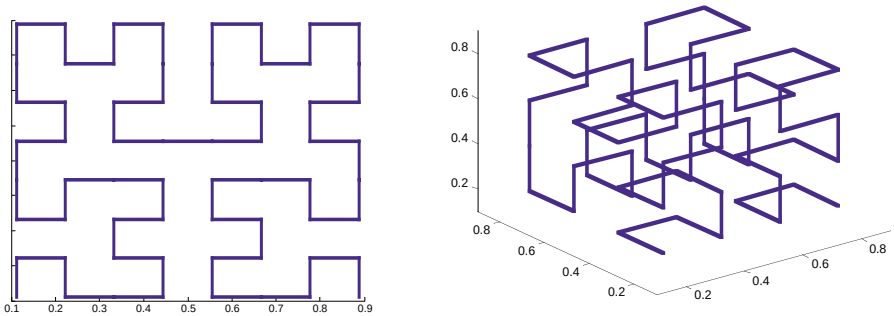


Figure 2. Three-dimensional plot of the annealing process.

5. SPACE-FILLING CURVE

A major problem encountered in implementing slice sampling in situations with high-dimensionality parameter sets, is that the slice-sampling routine will break down as it attempts to slice along all the dimensions. When slicing is performed on high dimensional problems, certain assumptions and the need for detailed balance, which are required for the slice sampling to be effective, will begin to fail. To solve this problem, a Hilbert curve (Fig. 3) will be used to map the coordinates from each dimension into a single dimension. This process of mapping from multiple dimensions to a single dimension is non-trivial, as there is no algebraic form relating the two. Butz [3], however, introduced an algorithm in which the coordinates are converted into binary numbers, and a series of binary operations are performed to convert from a single, one-dimensional coordinate into multiple coordinates. Lawder [4] expanded on this algorithm by noting that a conversion from binary to grey-code is used in one step. Lawder also presented the algorithm to convert back from multiple dimensions into one dimension. The implementation presented in [3, 4] are used in her to perform the conversion between the 1-D and N-D spaces.



(a) (b)

Figure 3. The space filling curve in (a) a two-dimensional parameter space and (b) a three-dimensional parameter space.

By performing the slice sampling in one dimension along the space-filling curve, the need for detailed balance and other essential assumptions required, will remain intact. This allows for the use of slice sampling and will permit the program to become more effective.

6. OBJECT-ORIENTED IMPLEMENTATION

A severe limitation to using a binary implementation of the Hilbert curve is the number of bits that can be stored in a variable. On most personal computers, the maximum number of bits that can be stored in a single variable is 32. However, given a system with 10 parameters and a 10th order Hilbert curve, 100 bits would need to be stored to represent each position on the one-dimensional curve. One could use an array of data to store this information, but indexing each bit position would be unwieldy and would require a complicated bit-masking algorithm.

Use of an object-oriented programming language, such as C++, easily solves this problem by defining a data type that is an array of single bits. This "*BitArray*" class can insert or retrieve a single bit at any index in the array. It has built-in functions that can expand or contract its size as needed, add another "*BitArray*" object to itself, or perform a two's-complement conversion on itself. One can also instance an array of BitArray objects, so that all the information from every dimension in the parameter space can be stored in a single variable array.

Another advantage to using an object-oriented method is that each Markov chain can be instanced separately, without dependence on other Markov chains. Each Markov chain object contains its current position in both the one-dimensional, as well as the multi-dimensional parameter space. Each Markov chain object also carries with it a copy of the target data so that each can independently determine its own probability. Each chain also has the ability to slice-sample itself, in that it needs no separate code to direct it to the next point. Using this object-oriented method minimizes the looping and computations needed by the overall search engine.

7. PARALLEL IMPLEMENTATION

Since each Markov chain is independent from every other chain, parallelization becomes simply a matter of assigning a number of chains to each processor. Using the popular message passing interface, or MPI, method of parallel processing [5], all of the processors will send their value of the standard deviation of the energy for their chains, and processor 0 will receive them, determine the largest standard deviation, and compute the next value of λ. Processor 0 will then broadcast the value of λ to be used in the slice-sampling routine, to all of the other processors. Once λ has achieved a value of unity, all processors will send their chain's position to processor 0 for output to file.

8. RESULTS

In order to investigate the effectiveness of the technique in common situations, a test was proposed by Haupt and Haupt in [6] to run the simulation for a highly undulating function. Since most real life applications of Bayesian and GA techniques require evaluation of often very complex "cost" functions, this test was meant as a basis for the evaluation. The function was defined as:

$$f(x, y) = x \sin(4x) + 1.1y \sin(y) \text{ for } 0 \leq x,\ y \leq 10 \tag{4}$$

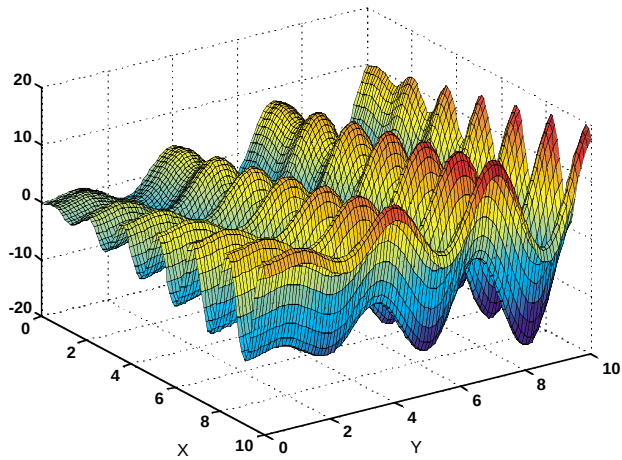


Figure 4. Plot of the function in (4).

Due to the complex nature of this function, as can be seen in Fig. 4, determining a global minimum for the function can be difficult due to the high number of local minima, a characteristic common in many real world situations. In [6], the authors choose to let their GA simulation run using various population sizes (analogous to the number of chains here) and stop when a point was found to have a value of less that -17, where the global minimum for the specified range is -18.5547 at $x = 9.0390,\ y = 8.6682$.

When the Bayesian technique is run for this case over a variety of chain sizes, the data in Fig. 5 shows how the convergence rate is tied to the number of chains. As could be expected after a certain point, increasing number of chains does little to improve the convergence rate. For this case a chain size of 80 has an optimal balance of convergence rate to simulation time. As could be inferred, the smaller number of

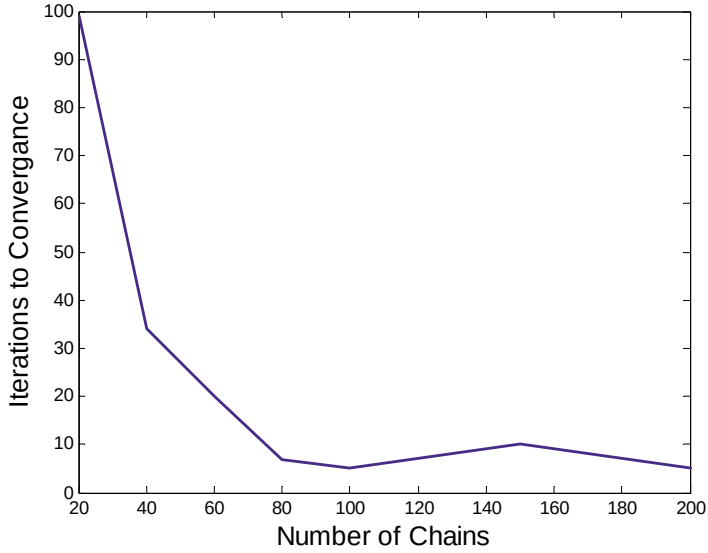


Figure 5. Number of iterations needed vs number of chains.

iterations required for a simulation the fewer calls to the cost/energy function, while the larger the chain size the more calls would have to be made. So by determining the "Sweet spot" of simulation parameters, the total time to simulate can be reduced. Examinations between the number of cost/energy function calls between Bayesian and GA techniques for optimal simulation settings shows a comparable number of cost/energy function calls in the order of 200 to 400 for both techniques.

As a second test to check the operation of the technique, the simulation was run with a large number of chains over a larger number of iterations. The sampling data was then gathered to make sure the system was indeed fully exploring the parameter space. Figure 6 shows histograms for samples along the X and Y axes for the undulating function. The histograms show that the simulation was able to effectively locate and sample from each of the minima areas and successfully locate the deeper ones by taking more samples.

Preliminary results show that this method is quite effective for determining the required element amplitudes needed to reproduce a variety of radiation patterns in linear arrays. One must take care, however, in the definition of the model as multi-modal results can occur. For the following antenna array examples, the following model

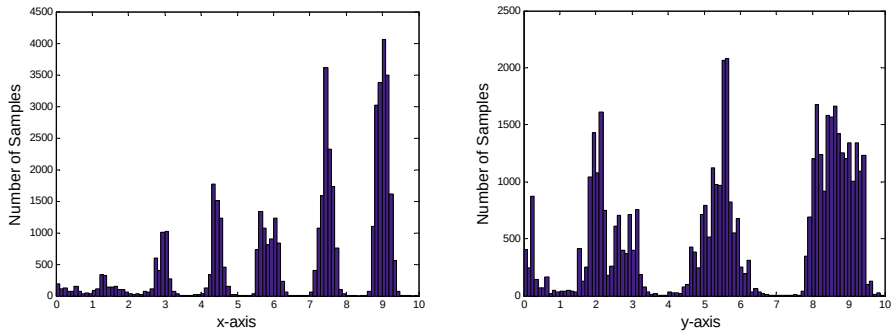


Figure 6. Histrogram of sample locations.

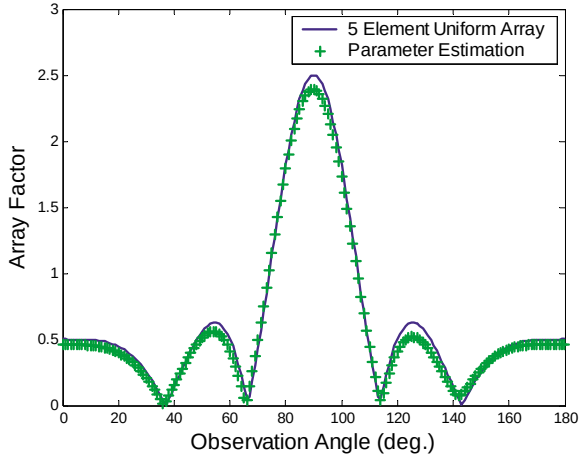


Figure 7. Target pattern and parameter estimated pattern.

is used.

$$F(\theta) = \sum_{n=1}^{N} a_n e^{j(n-1)\psi} \tag{5}$$

with

$$\psi = kd\cos(\theta) \tag{6}$$

where a_n is the complex element excitation, k is the wavenumber, and d is the inter-element spacing.

Figure 7 shows the radiation pattern computed for the computed element excitations when given the radiation pattern for a 5-element array with uniform excitation of $a_n = 0.5$. Very good agreement is observed between the estimated pattern and the target pattern. 50

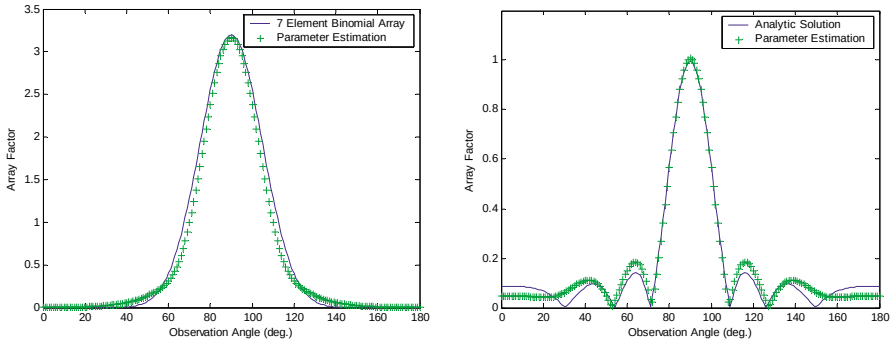


Figure 8. Target pattern and parameter estimated pattern for a 7-element binomial and Fourier synthesized array.

chains were used and the simulation was allowed to automatically anneal.

Figure 8 shows the radiation pattern for the computed element excitations when given the radiation pattern for a 7-element binomial array. Again, it is observed that the estimated values closely match the target data. While Figure 8 also shows the radiation pattern for the computed element excitations when the target data was a 7 element Fourier synthesized array with a beam-width of 20 degrees. Once again a very close match is observed across the main lobe. However, some degradation can be seen in the major sidelobe regions. Again 50 chains were used and the simulation was allowed to automatically anneal.

9. CONCLUSIONS

Through the use of Bayesian techniques, the entire parameter space of even dimensional problems can be characterized and analyzed. Histograms of the samples taken in the simulation show that all the peaks of the energy function can be located and utilized. The Bayesian technique allows for the elegant and efficient estimation optimization of many design problems. Improvements were detailed that allowed the technique to become more efficient and avoid possible problems. Results were also offered that illustrated the ease and versatility of the technique and how it compares to genetic algorithms.

REFERENCES

1. McKay, D. J. C., *Information Theory, Inference, and Learning Algorithms*, Cambridge University Press, 357–396, 2003.

2. Neal, R. M., "Slice sampling" (with discussion), *Annals of Statistics*, Vol. 31, 705–767, 2003.
3. Butz, A. R., "Alternative algorithm for Hilbert's space-filling curve," *IEEE Trans. Comput.*, C-20 (Apr.), 424–426, 1971.
4. Lawder, J. K., "The application of space-filling curves to the storage and retrieval of multi-dimensional data," Ph.D. thesis, School of Computer Science and Information Systems, Birkbeck College, University of London, 2000.
5. Pacheco, P. S., *A User's Guide to MPI*, University of San Francisco Department of Mathematics, 1995.
6. Haupt, R. L. and S. E. Haupt, "Optimum population size and mutation rate for a simple real genetic algorithm that optimizes array factors," *ACES Journal*, Vol. 15, No. 2, 94–102, July 2000.
7. Barbisch, B. J., D. H. Werner, and P. L. Werner, "A genetic algorithm optimization procedure for the design of uniformly excited and nonuniformly spaced broadband low sidelobe arrays," *ACES Journal*, Vol. 15, No. 2, 34–42, July 2000.

Matthew Joseph Inman was born in Dayton, Ohio on Feb. 7th, 1978. He received his B.S. in Electrical Engineering in 2000 and his Masters in Electromagnetics in 2003 from the University of Mississippi. He currently is currently pursuing Ph.D. studies in electromagnetics there. He is currently employed at the University as a research assistant and graduate instructor. His interests involve electromagnetic theories, numerical techniques, antenna design and visualization, as well as teaching a number of undergraduate courses.

John M. Earwood received his B.S. in Electrical Engineering in 1999 from the University of Mississippi, where he graduated Magna Cum Laude. He is currently enrolled in the graduate program at the University of Mississippi, where he is pursuing a M.S. in Electrical Engineering. Upon entering graduate school, he was appointed to the Radar Power Technology program in which he is conducting research into phased array antenna optimization techniques. After completing his undergraduate degree, Mr. Earwood spent three years as a Radar Systems and Antenna engineer for Raytheon Co., where he acted as technical lead on the construction of two near-field antenna test facilities, as well as the upgrade of a third. He was also involved in the development of the new U.S. Army TPQ-47 Fire-Finder radar system.

Atef Z. Elsherbeni received an honor B.Sc. degree in Electronics and Communications, an honor B.Sc. degree in Applied Physics, and a

M.Eng. degree in Electrical Engineering, all from Cairo University, Cairo, Egypt, in 1976, 1979, and 1982, respectively, and a Ph.D. degree in Electrical Engineering from Manitoba University, Winnipeg, Manitoba, Canada, in 1987. He joined the faculty at the University of Mississippi in August 1987 as an Assistant Professor and advanced to the rank of Associate Professor on July 1991, and to the rank of Professor on July 1997. Dr. Elsherbeni has published 73 technical journal articles and 12 book chapters on applied electromagnetics, antenna design, and microwave subjects, and contributed to 210 professional presentations. He is the coauthor of the book entitled *MATLAB Simulations for Radar Systems Design*, CRC Press, 2003. Dr. Elsherbeni is a senior member of the Institute of Electrical and Electronics Engineers (IEEE).

Charles E. Smith was born in Clayton, AL, on June 8, 1934. He received the B.E.E., M.S., and Ph.D. degrees from Auburn University, Auburn, AL, in 1959, 1963, and 1968, respectively. In late 1968, he accepted the position of Assistant Professor of Electrical Engineering with The University of Mississippi, University, MS, and he advanced to the rank of Associate Professor in 1969. He was appointed Chairman of the Department of Electrical Engineering in 1975, and he is currently Professor and Chair of this department.