Comparative Study of Linear Array Pattern Synthesis Using Iterative Fourier Transform & Genetic Algorithm

Rakesh Kumar[1], R Avinash[1], Purender Reddy B[2] and Dr. A Vengadarajan[1]
[1]LRDE (DRDO), CV Raman Nagar, Bangalore-560093,
[2]PESIT Bangalore-560085
rakesh.kumar@lrde.drdo.in

Abstract:

This paper presents Iterative Fourier Technique (IFT) and Genetic Algorithm (GA) for the synthesis of low Side lobe patterns for linear array with uniform element spacing. A comparative study is done with Array Element failure is introduced at various location of linear array in distributed and concentrated form. Quantitative results are presented to show effectiveness of the algorithms for failure correction in terms of pattern recovery. The MATLAB simulation results are presented for 50 and 80 elements arranged in regular grid for antenna sum pattern.

Key Words: Linear Array, Iterative Fourier Technique, Genetic Algorithm.

I INTRODUCTION

An antenna array with traditional analog beamforming, if one or more elements are damaged by an unforeseen reason, the array may have to be pulled out from operation due to unacceptable pattern distortion, for example, a significant increase of sidelobe level (SLL) [1]. With digital beamforming, the defective elements of an antenna array need not to be replaced. Instead, the beamforming weights of the remaining elements can be recalculated to form a new pattern that is close to the original. The possibility of failure correction for digital beamforming arrays provides a cost-effective alternative to hardware replacement which might be too late or too time-consuming, especially for arrays performing critical operations, such as, for instance, in the battlefield. From the open literature, no analytic technique has been devised to yield a set of new beamforming weights that effectively corrects the deformed pattern. Since a failed array can be considered as a nonuniformly spaced array, analytic approaches are generally unable to tackle this kind of problem. In recent years, numerical algorithms have been proposed to correct the deformed patterns. However, due to the arbitrariness of the geometrical layout of the remaining functional array elements and of the desired beam shape, array failure correction even for numerical approaches is a very challenging problem. From literature review, only a few research results have been reported. In this paper, IFT and GA has been presented and compared in terms of simulation time, pattern recovery in terms of side lobe level and complexity of implementation.

The IFT[2] approach uses the property that for an array having a uniform spacing of the elements, an inverse Fourier transform relationship exists between the array factor (AF) and the element excitations. Because of this relationship, a direct Fourier transform performed on AF will yield the element excitations. The underlying approach relies on the repeatedly use of both types of Fourier transforms. At each iteration, the newly calculated AF is adapted to the side lobe requirements, which then is used to derive a new set of excitation coefficients. Only those excitation coefficients belonging to the array are used to calculate a new AF. A key characteristic of this iterative synthesis method is that the algorithm itself is very simple, highly robust and very easy to implement in software requires only a few lines of code when programmed in MATLAB. The computation speed is very high because the core calculations are based on direct and indirect fast Fourier transforms.

Genetic algorithms [3] are “global” numerical-optimization methods, patterned after the natural processes of genetic recombination and evolution. The algorithms encode each parameter into binary sequences, called a gene, and a set of genes is a chromosome. These chromosomes undergo natural selection, mating and mutation, to arrive at the final optimal solution.

II ITERATIVE FOURIER TRANSFORM

The far-field F(u) of a linear array with M elements arranged along a periodic grid at distance d apart, can be written as the product of the embedded element pattern EF and the array factor AF

$$\mathrm{F}(u) = EF\,(u)\,AF(u) \qquad (1)$$

$$\mathrm{F}(u) = \sum_{m=0}^{M-1} A_m\, e^{jkmdu} \qquad (2)$$

Where A_m is the complex excitation of the m^{th} element, k is wave number ($2\pi/\lambda$), λ is the wavelength, u = sinθ and θ angular coordinate measured between far-field direction and the array normal. Equation (2) forms a finite Fourier series that relates the element excitation coefficients A_m of the array to its AF through a discrete inverse Fourier transform. AF is periodic in u-dimension over the interval d/λ. Since AF is related to the element excitations through a discrete inverse Fourier transform, a discrete direct Fourier transform applied on AF over the period λ/d will yield the element excitations A_m. These Fourier transform relationships are used in an iterative way to synthesize low sidelobe pattern for arrays with a periodic element arrangement.

Implementation of the IFT algorithm for the synthesis of low sidelobe patterns for linear arrays using amplitude-only element weighting proceeds as follows.

1. Start the synthesis with a uniform excitation for M elements in case of the sum pattern.
2. Compute AF from {Am} using a K-point inverse FFT with K>M.
3. Adapt AF to the prescribed sidelobe constraints.
4. Compute {A_m} for the adapted AF using a K-point direct FFT.
5. Truncate {A_m} from K samples to M samples by making zero all samples outside the array.
6. Make the phase of the M samples of {Am} equal to the phase of initial excitation at Step 1.
7. Set the magnitude of the excitations violating the amplitude dynamic range constraint to the lowest permissible value.
8. Enforce the optional defective element constraint. Take element failures into account by setting their excitation values to zero.
9. Repeat Steps 2-9 until the prescribed sidelobe requirements for AF are satisfied or the allowed number of iterations is reached.

The above algorithm refers to the amplitude only synthesis.

III GENETIC ALGORITHM

Genes are the basic building blocks of genetic algorithms. A gene is a binary encoding of a parameter. A chromosome is an array of genes in an algorithm. Each chromosome has an associated cost function, assigning a relative merit to that chromosome. The algorithm begins with a large list of random chromosomes. Cost functions are evaluated for each chromosome. The chromosomes are ranked from the most-fit to the least-fit, according to their respective cost functions. Unacceptable chromosomes are discarded, leaving a superior species-subset of the original list. Genes that survive become parents, by swapping some of their genetic material to produce two new offspring. The parents reproduce enough to offset the discarded chromosomes. Thus, the total numbers of chromosomes remains constant after each iteration. Mutations cause small random changes in a chromosome. Cost functions are evaluated for the offspring and the mutated chromosome, and the process is repeated. The algorithm stops after a set number of iterations, or when an acceptable solution is obtained [4-5].

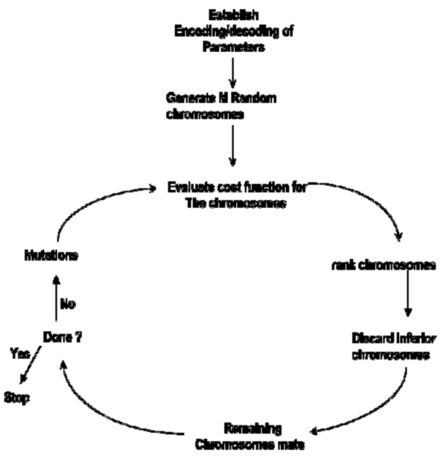

Establish Encoding/decoding of Parameters
Generate N Random chromosomes
Evaluate cost function for The chromosomes
rank chromosomes
Discard Inferior chromosomes
Remaining Chromosomes mate
Done ?
No
Mutations
Yes
Stop

Fig.1 A flowchart of a genetic algorithm.

IV SIMULATION RESULTS

A quantitative comparison is made between IFT and GA for linear array of 50 and 80 elements arranged in regular grid. 10 % concentrated and distributed element failure is introduced in the array and pattern difference was observed. Initially Chebyshev window is used to reduce the SLL to -30 dB which will be termed as reference pattern in this paper. All comparison is done with reference pattern. Fig.2 and Fig.3 are for concentrated element failure for 50 and 80 elements respectively. Fig.4 and Fig.5 are for distributed element failure for 50 and 80 elements respectively. Concentrated failure causes more pattern distortion than distributed failure. In distributed failure, level of pattern distortion depends on location of element failure. Failure at centre causes more pattern distortion than failure at edge. The performance of IFT and GA on linear array pattern is tabulated below in Table 1 for concentrated element failure and in Table 2 for distributed element failure.

The table 1 & table 2 explain quantitatively how the pattern changes in terms of side lobe level due to element failure. Table 1 also shows time taken to accomplish the simulation. Simulation time is computer specific. The computer configuration used here is Intel Core 2 Duo processor running at 2.33 GHz and 4 GB of RAM memory. The coding of the algorithm was done in MATLAB R2009a.

Table 1: IFT & GA simulation result for Concentrated Element Failure

Technique	No Of Elements	Reference Pattern SLL(db)	SLL After Failure (dB)	SLL After Correction (dB)	SLL SUPPRESSION (dB)	Time Taken (min)
IFT	50	-30	-21.9	-30	8.1	0.08
GA	50	-30	-21.9	-25.58	3.68	6
IFT	80	-30	-20.89	-24.93	4.04	0.07
GA	80	-30	-20.89	-24.52	3.63	3.50

Table 2: IFT & GA simulation result for Distributed Element Failure

Technique	No Of Elements	Reference Pattern SLL(db)	SLL After Failure (dB)	SLL After Correction (dB)	SLL SUPPRESSION (dB)	Time Taken (min)
IFT	50	-30	-20.9	-30	9.1	0.05
GA	50	-30	-20.9	-28.03	7.13	5
IFT	80	-30	-22.51	-28.6	6.09	0.05
GA	80	-30	-22.51	-29	6.49	1.50

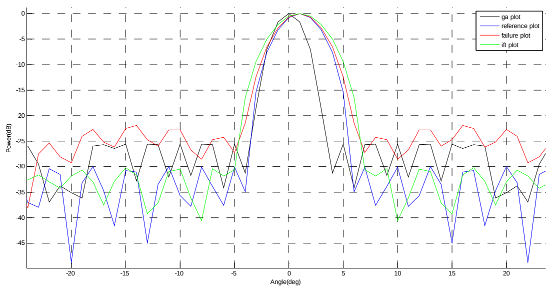


Fig.2 performance of IFT & GA for concentrated failures for 50 element linear array. Failure locations are [5, 6, 7, 8, 9]. For this scenario GA population has been taken 20000 and mutation rate is 0.1. The result shows Array pattern with element failure has recovered to reference pattern in terms of SLL but main lobe with has slightly changed.

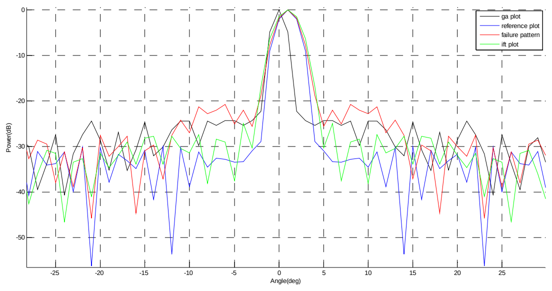


Fig.3 performance of IFT & GA for concentrated failures for 80 element linear array. The failure position are [10, 11, 12, 13, 14, 15, 16, 17].For this scenario GA population has been taken 40000 and mutation rate is 0.1.

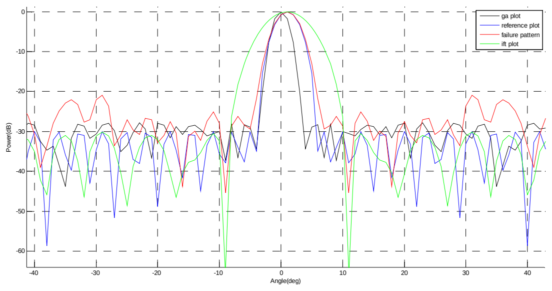


Fig.4 performance of IFT & GA for distributed failures of 50 elements linear array. The failure positions are [5, 10, 15, 40, 45]. For this scenario GA population has been taken 40000 and mutation rate is 0.1.

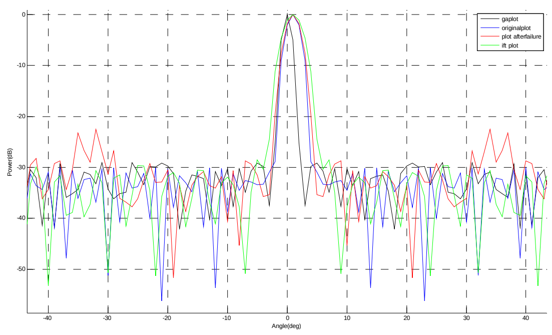


Fig.5 performance of IFT & GA for distributed failures for 80 element linear array. The failure positions are [5, 10,15,20,60, 65, 70, 75].For this scenario GA population has been taken 40000 and mutation rate is 0.1. The result shows Array pattern with element failure has recovered to reference pattern in terms of SLL but main lobe with has slightly changed.

CONCLUSION

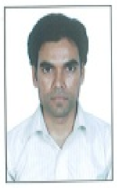
In this paper, comparative study has been done between IFT and GA. The study is performed for concentrated and distributed element failure. Concentrated failure causes more pattern distortion. The parameters chosen to compare is SLL, simulation time and complexity of implementation. Both IFT & GA can be used to recover array pattern to some extent in terms of SLL. The above figure clearly shows that element failure causes distortion in array pattern so it needs to be corrected to some extent. IFT and GA can be used for synthesis of low side lobe level i.e. pattern recovery in terms of SLL.Table 1 & 2 shows that IFT performs better in terms of SLL recovery. IFT is easy to implement since it requires direct Fourier transform, inverse Fourier transform as standard algorithm and required SLL (as constraint) and IFT is faster than GA. The results presented here are for antenna sum pattern. The algorithm can be used for difference pattern also.

REFERENCES

[1]. Beng-Kiong Yeo and Yilong Lu,” Array Failure Correction with a Genetic Algorithm” IEEE transactions on antennas and propagation, vol. 47, no. 5, may 1999.
[2]. Will P.M.N. Keizer,” Low Sidelobe Pattern Synthesis Using Iterative Fourier Techniques Coded in MATLAB”

[3]. R L. Haupt, “An Introduction to Genetic Algorithms for Electromagnetics,” IEEE Antennas and Propagation Magazine, Vol. 37, No. 2, April 1995.
[4]. Randy L. Haupt, “Genetic Algorithm Applications for Phased Arrays” ACES journal, vol. 21, no. 3, november 2006.
[5]. Keen-Keong Yan and Yilong Lu, “Sidelobe Reduction in Array-Pattern Synthesis Using Genetic Algorithm” IEEE Transactions on Antennas And Propagation, Vol. 45, No. 7, July 1997.

BIO DATA OF AUTHOR(S)



[1]Rakesh Kumar received the B.Tech. degree in Electronics & Telecommunication Engineering from National Institute of Technology Silchar, Assam, India, in 2007.Currently, he is working in Electronics & Radar Development Establishment (DRDO) as a Scientist. His main research interests are in the area of radar signal processing and antenna pattern synthesis.

.

[2]Avinash R received the B.Tech. degree in Computer Science & Engineering from Madras Institute of Technology, Chennai, Tamilnadu India, in 2007. Currently, he is working in Electronics & Radar Development Establishment (DRDO) as a Scientist. His main research interests are in the area of radar signal processing and antenna pattern synthesis.

[3]B Purender Reddy received the B.Tech. Degree in ECE from Sree Datta College Of Engineering, Hyderabad and M.Tech. degree from PESIT, Bangalore. His main research interest is antenna pattern synthesis. He can be contacted at purenderbireddy@gmail.com

[4]Dr. A Vengadarajan is working as a Scientist in DRDO since 1989. Received B.E degree in Electronics & Communication Engineering and M.Tech & Ph.D in Microwave Engineering. His areas of interest are Radar Signal Processing, Array Processing, STAP and System Engineering.