

Non-Uniformly Spaced Linear Array Antennas for Side Lobe Reduction using Genetic Algorithm

Asst. Lect. Sadiq K. Ahmed

*Electrical Engineering Dept., College of Engineering
Al-Mustansiriya University, Baghdad, Iraq*

Abstract

Genetic Algorithms are presented for calculating the distances between the elements of asymmetrical non-uniformly spaced linear array antenna for side lobe reduction, and describes the basic genetic algorithms. The amplitude of the excitation is assumed to be constant. The array design is optimum, and the simplicity of this method will allow arrays to be designed quickly.

الخلاصة

استخدمت الخوارزميات الجينية لحساب المسافات بين العناصر لمصفوفة هوائيات خطية متماثلة غير منتظمة بالمسافات وذلك لغرض تقليل الفصوص الجانبية. وأيضاً تم في هذا البحث وصف أساسيات الخوارزمية الجينية. أفترض مقدار التهيج (التيار) متساوي لعناصر الهوائي. تمتلك هذه الطريقة أفضل تصميم وكذلك بساطتها بالتالي تسمح هذه الطريقة بتصميم الهوائي بشكل سريع.

1. INTRODUCTION TO GENETIC ALGORITHMS (GA)

The use of non-uniformly spaced elements in antenna arrays is a relatively recent development, the merits of which are still being determined. A verity of investigations has been undertaken to see if the number of elements in an array of fixed size could be reduced significantly without causing serious deterioration of the pattern ⁽¹⁾.

It has been found that side lobes near the main beam may be reduced significantly using uniformly excited elements with variable spacing. This is sometimes accomplished at the expense of side lobes which further out, a price which is often not too costly ⁽¹⁾.

The success of genetic algorithms in the design of uniform distance array gives a stimulus for the application to the problem of non-uniformly distances array. The main concern in this paper is to find an appropriate element positions to obtain the least side lobe level. Though there have been many approaches to the problem of designing non-uniformly spaces arrays ⁽²⁾. Genetic algorithms may be used as a simple and flexible alternative to achieve the same objective, Which other methods can do, and more importantly, it has unique features to treat some complicated problem (arbitrary geometric layout, including mutual coupling), which can not be done by other methods ⁽³⁾.

Genetic algorithms are search and optimization algorithms which have very wide applications. Algorithm started with a set of solutions (represented by chromosomes) called population. Solutions from one population are taken to form a new population. This is motivated by a hope, that the new population will be better than the old one. Solutions which are selected according to their fitness the more suitable they are the more chances they have to reproduce. This is repeated until some condition is satisfied ^(3,4).

Conventional GA's with binary coding and binary genetic operation are non-convenient and inefficient for array pattern synthesis problem to optimize real or complex numbers ⁽³⁾. This approach avoids coding and directly deals with real weighting vector.

1-1 Construction of Chromosomes

Using genetic algorithms for non-uniform distance array radiation patterns corresponding to living beings and array-distances vectors correspond to chromosomes. Genetic algorithms were invented to manipulate a string of binary coding. Conventional genetic algorithms encode the parameters in binary chromosomes and perform binary genetic operation. In this approach, chromosomes are represented directly by real weighting vector ⁽³⁾.

$$\mathbf{d} = [d_1 d_2 d_3 \dots d_n \dots d_N] \dots\dots\dots (1)$$

where d_n (known as a genetic material in a GA) represents the excitation of the n^{th} radiator and d is set of real numbers. N is the length of the weighting vector. This simple representation explicitly shows the relation between chromosomes in genetic algorithms and array distances vectors ⁽³⁾.

1-2 Initial Population

For fast convergence of genetic algorithms iteration, the initial population can include approximate distances by other simple techniques (Harrington’s design and short design), and distances by guess based on experience and/or at random ⁽³⁾.

In this work short design method can be used. The position of the $(n-1)^{\text{th}}$ element in terms of the n^{th} element is given by ⁽²⁾.

$$d_{n-1} = d_n \frac{(2d_n - 1)}{(2d_n + 1)} \dots\dots\dots (2)$$

1-3 Fitness Function

The initial populations are produced, and their fitness's correspond to side lobe levels for the selection of suitable chromosome to compete for next generation. The radiation pattern generated by an $2N$ -elements antenna array is given by ^(2,3).

$$AF(\theta) = \frac{\left[\sum_{n=1}^N \cos(\beta x_n \sin \theta) \right]}{N} \dots\dots\dots (3)$$

where AF array factor.

$$\beta = 2\pi/\lambda = \text{wave number.}$$

λ the wavelength.

and $2N$ is the total number of elements. For convenience a change of variables

will be made where $d_n = x_n/\lambda$ and $v = 2\pi \sin \theta$. This gives ⁽²⁾

$$AF(v) = \left[\sum_{n=1}^N \cos(d_n v) \right] / N \dots\dots\dots (4)$$

And relative side lobe level in dB is computed from the radiation pattern using ⁽⁵⁾.

$$SLL_{dB} = 20 \log \frac{\max |E_{rad}(\theta)|}{E_{max}} \dots\dots\dots (5)$$

where SLL is the side lobe level in dB.

$\max |E_{rad}(\theta)|$ is the pattern value of the maximum of the highest side lobe magnitude.

E_{max} is the peak of main beam.

From the equation (5), the objective function is obtained as:

$$\text{error} = |SLL_{desired} - SLL_{obtained}| \dots\dots\dots (6)$$

then

$$\text{The fitness function} = 1/\text{error} \dots\dots\dots (7)$$

where the objective function is the calculation of its associated fitness.

The fitness function is measure of the quality of a chromosome ⁽⁶⁾.

1-4 Selection

There are many mating techniques available to pick two parent chromosomes to produce child chromosome ⁽³⁾.

1-4-1 Roulette Wheel Selection

Parents are selected according to their fitness. The better chromosomes are, the more chances to be selected they have. Imagine a roulette wheel where are placed all chromosomes in the population, every has its place big accordingly to its fitness function ⁽⁴⁾.

1-4-2 Rank Selection

Rank selection first ranks the population and then every chromosome receives fitness from this ranking. The worst will have fitness 1, the second worst 2 etc, and the best will have fitness P (where P is the number of chromosomes in population). After this all the chromosomes have a chance to be selected ⁽³⁾. In this work Roulette Wheel selection is used.

1-5 Crossovers

Crossover is another process that involves exchange of genetic materials between two parent chromosomes to make child chromosome. The simplest way how to do this is to choose randomly some crossover point and then everything before this point copies from the first parent and then everything after a crossover point copies from the second parent. There are many types of crossover ^(3,4).

1-5-1 Single Point Crossover

One crossover point is selected, string from the beginning of chromosome to the crossover point is copied from one parent and the rest is copied from the second parent ⁽⁴⁾.

1-5-2 Two Points Crossover

Two points crossover are selected, string from the beginning of chromosome to the first crossover point is copied from one parent, the part from the first to the second crossover point is copied from the second parent and the rest is copied from the first parent ⁽⁴⁾.

1-5-3 Uniform Crossover

Genes are randomly copied from the first or from the second ⁽⁴⁾.

1-5-4 Arithmetic Crossover (Linear Crossover)

Some arithmetic operations are performed to make a new off spring (average, and extrapolation crossover from midpoint)^(3,4). In this work single point crossover is used.

1-6 Mutation

After a crossover is performed, mutation takes place. This is to parent falling all solutions in population into a local optimum of solved problem. Mutation changes randomly the new offspring (children). There are many types of accomplishing mutation (binary mutation, and real mutation). Real mutation is used according to the following equation, as ^(4,7):

$$d_i' = \begin{cases} r(L_o, U_p) & \text{if } z' \leq P_m \\ d_i & \text{otherwise} \end{cases} \dots\dots\dots (8)$$

where

- z' is random number.
- $r(L_o, U_p)$ is random number with limited range (L_o, U_p) .
- d_i is the value of gene before mutation.
- d_i' is the value of gene after mutation.
- P_m is probability of mutation equal to (0.5%-1%).

2. RESULTS AND DISCUSSION

Many configurations of non-uniformly spaced arrays have been designed by various investigators. Each design has its own peculiarities regarding side lobe level and the complexity of calculations of element positions.

To illustrate the characteristics of an array based on the present design a comparison is made among 12-element arrays designed by short design, Harrington and by Genetic Algorithms. Table (1) shows element positions, side lobe level (sll) and half power beam width (HPBW) for Harrington design, short design, and Genetic Algorithms method. The pattern calculated for array (using short design) has a half-power beam width of 6.4° and maximum side lobe level of -7.33dB near main beam while outer (sll) which rises to -12dB . Harrington’s array has beam width of 6.8° and low side lobe level -20dB near the main beam

while outer side lobe level which rises to -6.37dB. And the pattern calculated for array (using Genetic Algorithms) has a half-power beam width of 7.34° and maximum side lobe level of -14 dB near the main beam while outer side level which decrease to -18.41 dB. The patterns calculated for array (using three methods) are shown in Figure (1).

As shown in the first column of Table (2), represents best array positions achieved from genetic Algorithms, the second and third columns represent the side lobe levels near main beam and outer side lobes respectively, the last column refers to half-power beam width. The pattern of arrays for (d_1, d_2, d_3, d_4) shown in Figures (2,3,4,5). The GA's are carried out in ten runs of 25 iterations using MATLAB software.

3. CONCLUSION

Simple and flexible GA's is proposed as a general purposed tool for non-uniformly spaced linear array. Side lobe level in GA's near main beam is lower than Sll in short design, while its higher than Sll in Harrington's design. In other hand outer side lobe level of GA's is the lowest from Sll in short and Harrington's design. HPBW is slightly greater than Harrington's and short design.

4. REFERENCES

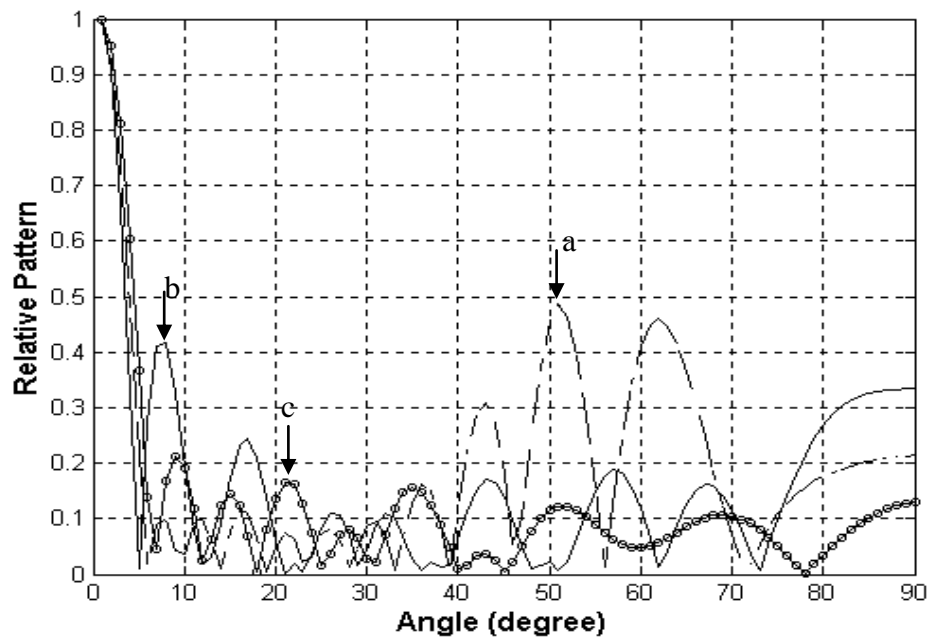
1. Hansen R. C., "*Microwave Scanning Antennas*", United States of America, 1985.
2. John P. Basart, "*A Short Design Method for Non-Uniformly Spaced Antenna Arrays*", IEEE Antenna and Propagation, November, 1974.
3. Keen-Keng Yan, and Yilong, "*Side Lobe Reduction in Array Pattern Synthesis Using Genetic Algorithms*", IEEE, Antenna and Propagation, Vol. 45, July, 1997.
4. Obitko M., "*Introduction to Genetic Algorithms*", Hochschule for Technik and Wirtschaft Dresden, September, 1998.
5. Stutzman W. L., "*Antenna Theory and Design*", McGraw-Hill, Third Edition, 1981.
6. Jain L. C., and Jain R. K., "*Hybrid Intelligent Engineering Systems*", World Scientific Publishing, 1997.
7. Goldberg D. E., "*Genetic Algorithms in Search, Optimization, and Machine Learning*", Addison-Wesley, 1989.

Table (1) Element Positions, Side Lobe Levels and Half-Power Beam Width for Three Methods

Methods of Design	Element Positions	Sll Near Main Lobe (dB)	Sll Outer Main Lobe (dB)	HPBW (Degree)
Harrington's Design	0.449 λ 1.32 λ 2.198 λ 3.234 λ 4.511 λ 5.866 λ	-20	-6.37	6.8
Short Design	1.463 λ 2.284 λ 3.146 λ 4.036 λ 4.944 λ 5.866 λ	-7.33	-12	6.4
GA Method	0.4245 λ 1.1149 λ 1.9072 λ 2.7535 λ 3.6325 λ 4.2937 λ	-14	-18.41	7.34

Table (2) Element Positions, Side Lobe Levels and Half-Power Beam Width for Best Solutions of GA's

GA Arrays	Element Positions	Sll Near Main Lobe (dB)	sll Outer Main Lobe (dB)	HPBW (Degree)
d_1	0.4245 λ 1.1149 λ 1.9072 λ 2.7535 λ 3.6325 λ 4.2937 λ	-14	-18.41	7.34
d_2	0.4822 λ 1.189 λ 1.9932 λ 2.9499 λ 3.7387 λ 5.3668 λ	-18.4	-10.8	6.74
d_3	0.343 λ 1.0268 λ 1.8346 λ 2.7073 λ 3.4808 λ 4.4145 λ	-16	-14	7.4
d_4	0.4932 λ 1.2162 λ 2.0436 λ 2.9275 λ 3.8461 λ 4.5879 λ	-13.5	-17.7	7



**Fig.(1) 12-Element Array Patterns for Element Positions in Table (1) using
a) Harrington's Design b) Short Design c) Genetic Algorithms**

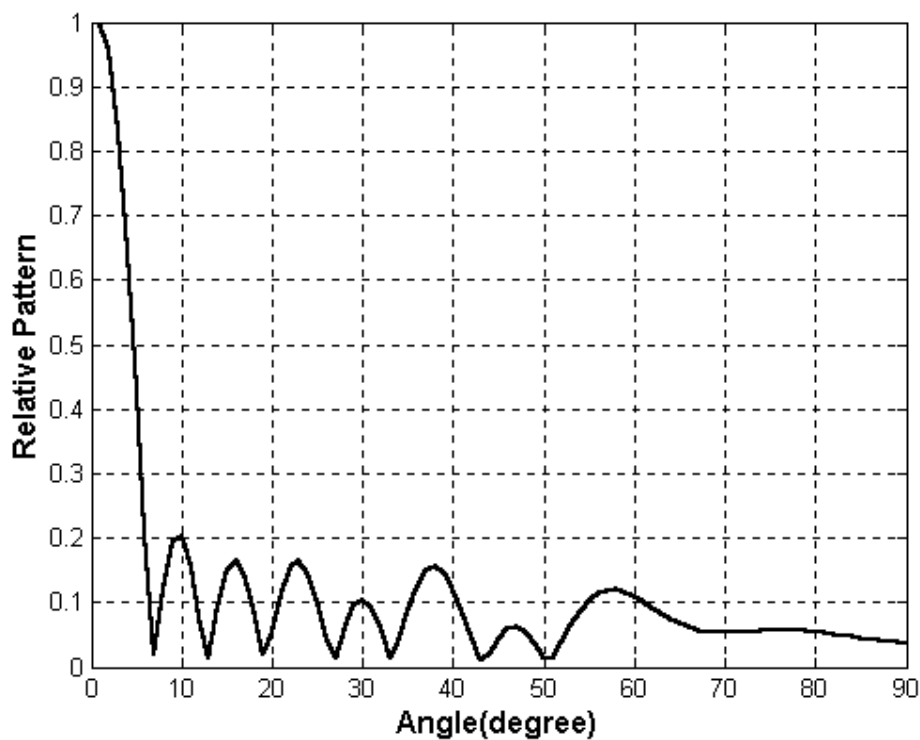


Fig.(2) 12-Element Array Pattern using GA for Element Positions of Array d_1 in Table (2)

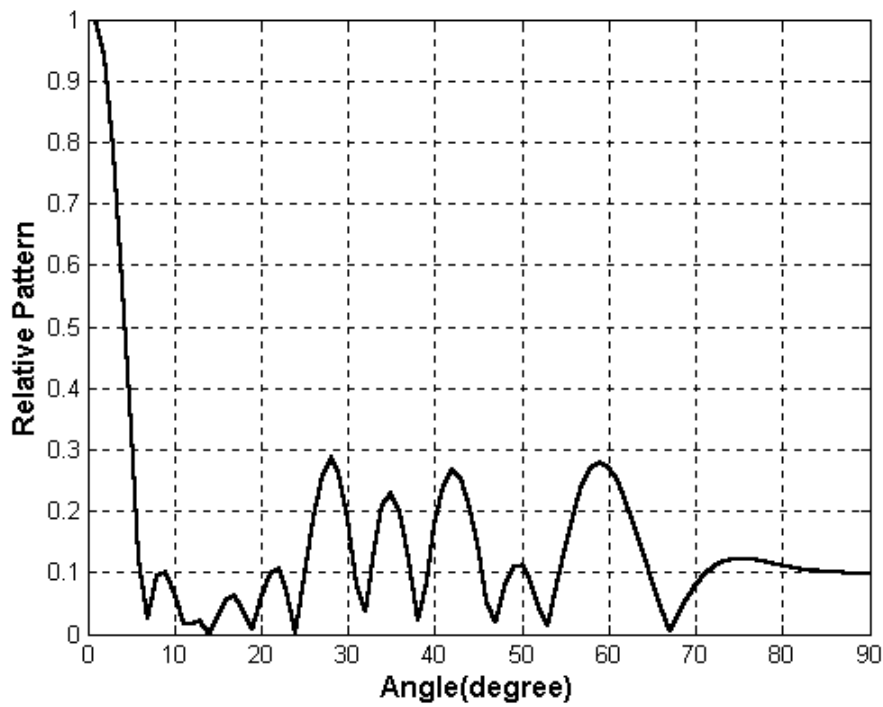


Fig.(3) 12-Element Array Pattern using GA for Element Positions of Array d_2 in Table (2)

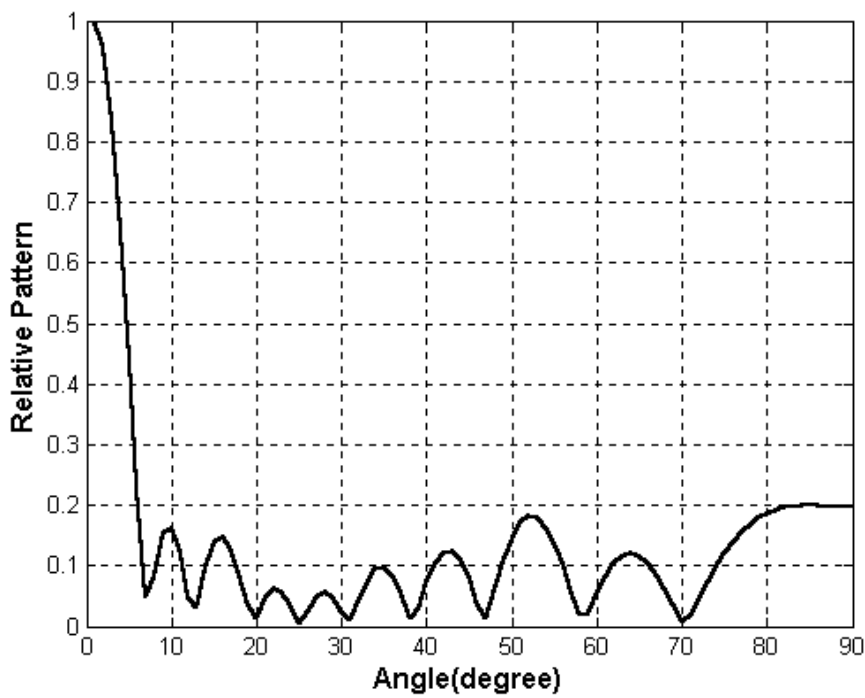


Fig.(4) 12-Element Array Pattern using GA for Element Positions of Array d_3 in Table (2)

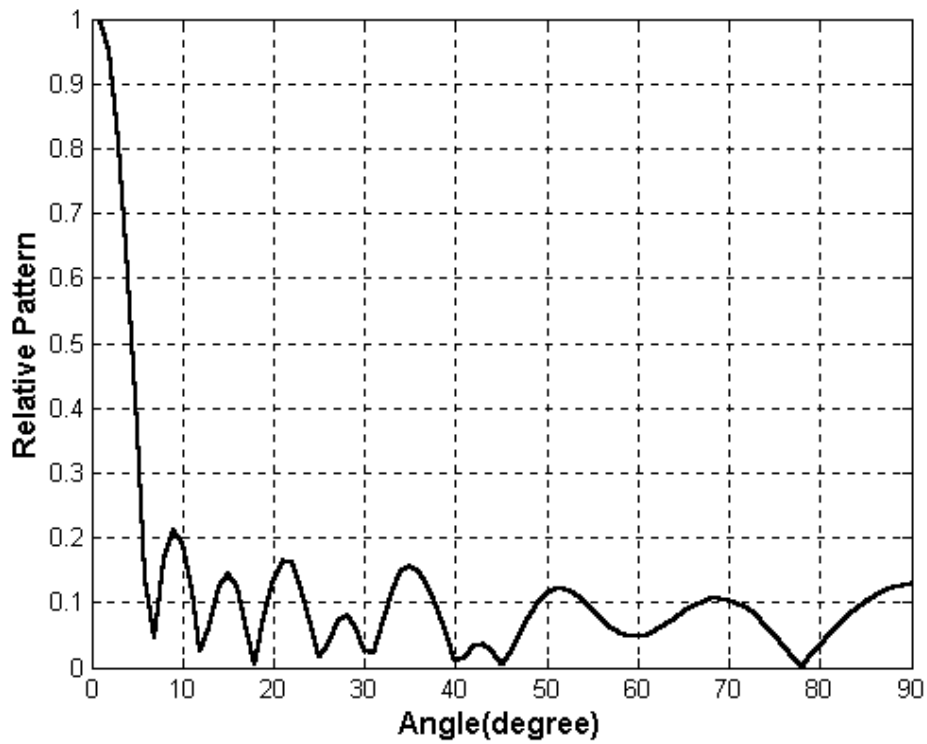


Fig.(5) 12-Element Array Pattern using GA for Element Positions of Array d_4 in Table (2)