Phase-Only Adaptive Nulling with a Genetic Algorithm

Randy L. Haupt, Senior Member, IEEE

Abstract—This paper describes a new approach to adaptive phase-only nulling with phased arrays. A genetic algorithm adjusts some of the least significant bits of the beam steering phase shifters to minimize the total output power. Using small adaptive phase values results in minor deviations in the beam steering direction and small perturbations in the sidelobe level in addition to constraining the search space of the genetic algorithm. Various results are presented to show the advantages and limitations of this approach. In general, the genetic algorithm proves to be better than previous phase-only adaptive algorithms.

Index Terms—Genetic algorithms, phased arrays.

I. INTRODUCTION

LOW sidelobes do not guarantee adequate reception of a desired signal in the presence of interfering sources. Adaptive nulling complements the low sidelobe antenna by placing nulls in a few low sidelobes to reject the strongest interfering sources. An ideal adaptive algorithm for a phased-array antenna has the following desirable characteristics:

• places multiple deep nulls in the directions of interference;
• rejects interference over the bandwidth of the antenna;
• places the nulls very quickly;
• complements existing phased array technology;
• minimizes pattern perturbations.

An adaptive algorithm possesses some of these characteristics, but no adaptive algorithm meets all the characteristics. Selection of the adaptive algorithm, hence, the desirable characteristics depend upon the antenna, the cost, the performance requirements, and the interference environment.

Most adaptive antenna algorithms multiply the quiescent weights by the inverse of the sampled covariance matrix to get the adapted weights. The resulting complex weights place nulls in the far-field pattern in the directions of interference. A sampled covariance matrix is formed from the complex signals received at each element in the array. Although mathematically elegant and fast, these methods impose two impractical hardware requirements on the antenna array. First, the array must have an expensive receiver or correlator at each element. Most arrays have a single receiver at the output of the summer, so the antenna must be designed especially for the algorithm. Not only are multiple receivers expensive, but the receivers require a sophisticated method for calibration [1]. Second, the array must have variable analog amplitude and phase weights at each element. Usually, a phased array has only digital beam steering phase shifters at the elements. The feed network is fixed and determines the amplitude weights. There are two problems from an algorithmic standpoint as well. First, digital phase shifters only approximate the continuous phase calculated by the adaptive algorithms. This weight quantization error limits null placement. Second, these algorithms get stuck in local minima [2]. As a result, they do not find the optimum weights to reject the interference at hand. Some common adaptive algorithms include least mean-square algorithm and Howells–Applebaum adaptive processor, and examples can be found in [2] and [3]. These methods are very fast, but the difficulties mentioned prohibit their wide-spread use, particularly for arrays with more than a handful of elements.

Another class of algorithms adjusts the phase shifter settings to reduce the total output power from the array [4]–[6]. These algorithms are cheap to implement because they use the existing array architecture without expensive additions such as adjustable amplitude weights or correlators. Their drawbacks include slow convergence and possibly high-pattern distortions.

There are four approaches to phase-only nulling, the last of which is the topic of this paper. The first approach is the random search algorithm [2]. Random search algorithms randomly sample a small fraction of all possible phase settings in search of the minimum output power. The search space for the current algorithm iteration can be narrowed around the regions of the best weights of the previous iteration. This approach is usually too slow for beam steering and radar applications. An array with $2^N$ elements and $B$ phase shifter bits has $2^{NB}$ possible phase settings (assuming the phase has an odd symmetry), some of which reduce the interference. It is less likely to get stuck in a local minimum and does not require an expensive receiver at each element. A second approach forms an approximate numerical gradient and uses a steepest descent algorithm to find the minimum output power [7]. This approach has been experimentally implemented but is slow. A finite-difference approximation to the partial derivatives of the output power with respect to a small phase change at each element is measured and placed in a vector. This vector is a finite-difference approximation of the gradient. Assuming the array is symmetric, a central differencing scheme requires $N$ power measurements to form the gradient during each iteration of the algorithm. After many iterations, the algorithm descends to a minimum in the vicinity of the starting point. This algorithm, and others like

Manuscript received January 19, 1996; revised October 2, 1996.

The author is with the Department of the Air Force, HQ USAFA/DFEE, USAF Academy, CO 80840–6236 USA.

Publisher Item Identifier S 0018–926X(97)04260-9.
it (Powell’s, downhill simplex, conjugate gradient, etc.) [8], are extremely nearsighted and only explore minima in the immediate vicinity of the starting point. These algorithms are based on the assumption that the cost surface is quadratic. Requiring small phase shifts to accurately approximate the derivative localizes the algorithm too. As a result, the best phase settings to achieve appropriate nulls are usually not found. In addition, the small phase shifts produce a small change in the output power that may be difficult to detect, especially in a large array. The third approach is a beamspace algorithm that assumes the location of the interference is known. This algorithm forms a cancellation beam in the direction of the interference. The height of the cancellation beam is adjusted to cancel the sidelobes and place a null in the interference direction. This approach is fast, but requires knowledge of the interference locations and a reasonably accurate estimate of the amplitude and phase weights at each element.

This paper describes a simple technique suitable for implementation on existing phased arrays. The approach combines a genetic algorithm with the hardware limitations of the array to place nulls in the directions of interference with small perturbations to the far-field pattern. Excellent nulling results are possible for most interference scenarios.

II. PROBLEM FORMULATION

The mathematical model for a linear array of point sources with \( \sin \phi \) element patterns lying along the \( x \) axis is given by

\[
AF(u) = \frac{\sin \phi}{2N} \sum_{n=1}^{2N} u_n e^{i(nN - \Delta_n)\psi}
\]  

where
- \( 2N \) number of elements;
- \( u_n = a_n e^{i\delta_n} \) complex array weight at element \( n \);
- \( \Psi = kdu + \tau \);
- \( k = 2\pi/\lambda \);
- \( \lambda \) wavelength;
- \( d \) spacing between elements;
- \( u \) \( \cos \phi \);
- \( \phi \) angle of incidence of electromagnetic plane wave;
- \( \tau \) beam-steering phase.

The amplitude weights are fixed. Lowering the sidelobe levels requires an even phase shift about the center of the array [9], while nulling requires an odd phase shift [10]. Since nulling is of importance here, (1) simplifies to

\[
AF(u) = 2\sin \phi \sum_{n=1}^{N} a_n \cos[(n - 1)\psi + \Delta_n + \delta_n].
\]  

Note that \( \Psi = kdu \) and no longer includes the beam-steering phase. Since the beam-steering phase is quantized, it differs from \( (n-1)\Psi \) and must be represented by \( \Delta_n \). The steering phase \( \Delta_n \) is calculated first and the beam steered to the proper angle before the nulling phase \( \delta_n \) is found. Fig. 1 is a diagram of a phase-only adaptive array with \( \Delta_n = 0 \).

The digital phase shifters have \( B \) bits. \( B \) needs to be as small as possible to reduce the cost of the phase shifter but should be large enough to maintain low sidelobes over the scan angles of the array. The quantized steering phase at element \( n \) is given by

\[
\Delta_n = \text{round}\left\{ \text{rem}\left\{ \frac{(n-1)\Delta}{2\pi} \right\} \frac{2B}{2\pi} \right\}
\]

where the nulling phase is represented by

\[
\delta_n = 2\pi \sum_{p=B-P+1}^{B} b_p 2^{-p}
\]

where
- \( B \) total number of bits in phase shifters;
- \( P \) number of phase bits used for nulling;
- \( [b_1 b_2 \cdots b_P] \) vector containing the nulling bits representing \( \delta_n \);
- \( \text{round}\{\ast\} \) round \( \ast \) to the nearest integer;
- \( \text{rem}\{\ast\} \) takes only the digits to the right of the decimal point of \( \ast \).

To minimally perturb the array pattern, the adaptive algorithm assumes \( P < B \). Nulling low sidelobes requires less significant bits than nulling high sidelobes.

III. THE ADAPTIVE ALGORITHM

A phase-only adaptive algorithm modifies the quantized phase weights based on the total output power of the array. If no interference is present, then the algorithm tries to minimize the desired signal. To prevent desired signal degradation, the algorithm should only be turned on at low array signal-to-interference-plus-noise ratios or when the nulling phase shifts should be small. By only using some of the least significant bits of the phase shifters for the adaptive nulling, the damage the algorithm can do to the main beam is very limited. The disadvantages of the phase-only algorithms listed earlier make them unlikely candidates for use with most antenna arrays.
This section presents a method that is as fast as the beam space algorithm, does not easily get stuck in local minima, and limits pattern distortion.

The adaptive algorithm is based on a genetic algorithm and uses a limited number of bits of the digital phase shifters. A genetic algorithm is a computer program that finds an optimum solution by simulating evolution in nature [11]. In this application the phase shifter settings evolve until the antenna pattern has nulls in the directions of jammers. A genetic algorithm was chosen for this application, because it is an efficient method to perform a search of a very large, discrete space of phase settings for the minimum output power of the array. Traditional approaches only explore solutions near an arbitrary starting point and have difficulty working with quantized parameters. An adaptive phase-only array has $2^{NB}$ possible phase settings, many corresponding to local minima in the total power output. Such a large number of phase settings and local minima make random-search and gradient-based algorithms impractical to use.

Fig. 2 shows a flowchart of the adaptive genetic algorithm. It begins with an initial population consisting of a matrix filled with random ones and zeros. Each row of the matrix (chromosome) consists of the nulling bits for each element placed side-by-side. There are $NP$ columns and $M$ rows. The output power corresponding to each chromosome in the matrix is measured and placed in a vector (Fig. 3). $M$ must be large enough to adequately search the solution space and help the genetic algorithm arrive at an excellent solution. On the other hand, $M$ needs to be small so the algorithm is fast. The speed of the algorithm is also a function of $N$ and $P$. As $N$ and $P$ increase in size, $M$ needs to be larger to keep the algorithm out of local minima, and the number of iterations required for convergence increases. The output power vector and associated chromosomes are ranked with the lowest output power on top and the highest output power on the bottom. The next step discards the bottom $X\%$ of the chromosomes because they have the greatest output power. New nulling chromosomes to replace those discarded are created from the chromosomes that were kept (Fig. 4). Two chromosomes are selected at random. Chromosomes with lower output power receive a correspondingly higher probability of selection. Next, a random point is selected and bits to the right of the random point are swapped to form two new chromosomes. These new chromosomes are placed in the matrix to replace two settings that were discarded and their output powers are measured. When enough new chromosomes are created to replace those discarded, the chromosomes are ranked and the process repeated. A small number (less than 1%) of the nulling bits in the matrix can be randomly switched from a one to zero or visa versa. These randomly induced errors allow the algorithm to try new areas of the search space while it converges on a solution. Usually, the best phase setting is not randomly altered. More general descriptions of genetic algorithms can be found in [11] and [12]. The next section shows results for various interference scenarios.
IV. RESULTS

Results presented here are for an array with 40 elements and a 30-dB Chebychev amplitude taper and for a 100-element uniform array. Elements are spaced 0.5λ₀ apart at the center frequency $f_0$. Phase shifters must have six bit accuracy to keep the quantization lobe slightly below the general sidelobe level over the scan angles of the array ($\pm 30^\circ$ from broadside). The goal of the genetic algorithm is to reduce the total noncoherent output power of the array. The cost function is given by

$$
\text{cost} = 20 \log_{10} \left\{ \sum_{i=1}^{N_I+1} s_i \sin \phi_i \sum_{n=1}^{N} a_n \cos \left[(n-1)k_d \cos \phi_i + \Delta_n + \delta_n\right] \right\}
$$

where

- $s_i$ signal strength of source $i$;
- $\phi_i$ angular location of source $i$;
- $\delta$ 1 is the desired signal;
- $N_I$ number of interference sources.

Genetic algorithms have a certain “black magic” associated with them. There are no set rules for picking the various population sizes, the mutation rate, etc. In general, these specifications come from experience with genetic algorithms and the problem at hand. After running the algorithm for many different scenarios, the following values were attained:

- initial population: 20 chromosomes;
- iteration population: ten chromosomes;
- size of mating pool: four chromosomes;
- discarded each iteration: six chromosomes.

A genetic algorithm is called a “global” optimization technique, but it usually cannot find the true global minimum. It is much more capable of exploring a large region of the cost surface than conventional methods, hence, it has increased odds of finding a better minimum.

The algorithm begins with 20 chromosomes. Only the four best are kept for mating. The parents produce six new chromosomes, so only ten chromosomes are in the population each iteration. These numbers are a compromise between algorithm speed and performance for arrays with 40 to 100 elements. The population sizes that produce the best results are a function of the array size, and the antenna engineer must experiment to find the population sizes that are most desirable for a given application.

The first scenario is a simple example of two interference sources at $u = 0.62$ and $u = 0.72$, each 60-dB stronger than the desired signal power. Since this is a low sidelobe array, only two least significant bits are needed to perform the nulling ([11.25°, 5.625°]). The resulting phase settings and far-field pattern appear in Figs. 5 and 6, respectively. The main lobe is only slightly perturbed by the adaptive weights. Figs. 7–9 are graphs of the algorithm’s performance for this case. Iteration zero is the value for the quiescent antenna pattern. In Figs. 7 and 8, the solid line represents the best chromosome of the population, while the dashed line is the average of the population. The SNR plot provides a quality rating of the algorithm performance. Unfortunately, the signal-to-noise ratio (SNR) cannot be used by the genetic algorithm to rank chromosomes, because there is no way to calculate the SNR for an actual antenna that corresponds to the model used here. Instead, the total output power is used as the cost associated with each chromosome. In Fig. 9, the solid line represents one null and the dashed line represents the other null. After eight iterations, the algorithm reaches a set of adaptive weights which have effectively suppressed the two interferers. At this point, the algorithm required 20 + 8 × 6 = 68 power measurements. In comparison, a gradient search method using a central differencing scheme requires 40 power measurements per iteration for this array. Such an approach requires many iterations to converge, thus, it is quite slower than the genetic algorithm. Other problems with the gradient approach include:

1) the digital phase shifters only form approximate derivatives for the gradient;
2) the gradient search only investigates a small section of the cost surface; and
3) a small change in the phase setting of one phase shifter in a large array produces corresponding small change in the array output power that is difficult to detect and use in forming the gradient. An exhaustive search of this array requires checking $4^{20}$ different
Fig. 7. SNR of the array as a function of the number of iterations of the genetic algorithm. The solid line is the SNR of the best chromosome and the dashed line is the SNR of the average chromosome of the population.

Fig. 8. Output power as a function of iteration of the genetic algorithm. This is the cost that the genetic algorithm is minimizing. The solid line is the output power of the best chromosome, and the dashed line is the output power of the average chromosome of the population.

Fig. 9. These are the depths of the two nulls as a function of iteration number.

Fig. 10. Adapted (solid line) and quiescent (dotted line) array patterns for a 100-element uniform array with interference sources at $u = 0.03$ (54 dB) and $u = 0.73$ (46 dB).

phase settings for the minimum output power. In a random selection of 5000 phase settings the best produced a SNR = 4.4 dB and an output power of 0.44 dB. In $5000/68 \approx 74$ times the number of power measurements, the random search could not outperform the genetic algorithm. The superiority of the genetic algorithm becomes even more striking as the number of elements increases.

The second scenario tests the algorithm for a 100-element array with two jammers. One jammer at $u = 0.03$ is 54-dB more powerful than the desired signal and the second at $u = 0.73$ is 46-dB stronger. Since this array has much higher sidelobe levels, four least significant bits [45°, 22.5°, 11.25°, 5.625°] must be used to produce a null in one of the higher sidelobes. Only minor improvements to the antenna performance are possible with three or fewer bits. Six bit phase shifters may be overkill for a uniform array, but keeping the number of bits at six allows us to compare with the results from the previous scenario. Fig. 10 shows the nulled far-field pattern superimposed on the quiescent pattern. These nulls were quickly placed in the pattern as shown by the graph of the SNR versus iteration in Fig. 11. The depth of the null at $u = 0.73$ does not have to be as deep as the null at $u = 0.03$. Fig. 12 shows the null depths versus iteration for the two nulls. The main beam has suffered some loss in gain and has been steered away from broadside. In addition, the sidelobe levels have increased. For this example, an exhaustive search would require $16^{20}$ power measurements to find the best nulls. Randomly tripping across a good set of adaptive phase weights becomes more and more unlikely as the number of elements in the array increases. A gradient search method using a central difference scheme requires 100 power measurements every iteration of the algorithm. The genetic algorithm, on the other hand, found an excellent set of adaptive weights in only 16 iterations or $20 + 16 \times 4 = 84$ power measurements. This is less time than the gradient search requires for a single iteration.

Many other scenarios were tested. The algorithm had difficulties in two situations. First, it cannot place a null in a quantization lobe. This result is not surprising though, since the quantization lobe arises from the approximation of the phase. Second, interference sources at symmetric or nearly symmetric locations about the main beam cannot both be nulled. This
nulling. Using one to three bits for nulling has very little impact on the main beam. Thus, the adaptive algorithm does not have to be turned off when no interference is present. Using four bits can do some damage to the main beam. Using five or six bits for nulling does significant damage to the main beam. The genetic algorithm should not need to use more than the four least significant bits for the examples shown here.

V. CONCLUSION

Genetic algorithms are well known for their ability to find excellent optimized solutions where traditional hill-climbing algorithms fail but are typically slow. This paper presented a real-time application of the genetic algorithm for phase-only adaptive nulling. Previous methods of phase-only nulling were quite slow or needed special extra knowledge like locations of the interference sources to operate fast. Surprisingly, the genetic algorithm proved to be much faster than random search methods or gradient type methods.

The algorithm performed well for two jammers that were separated by an angular width of at least one sidelobe and were not symmetric in angular distance from the main beam. Both the 40-element low sidelobe array and the 100-element uniform array showed fast convergence, deep nulling capability, and small pattern distortions. Using only a few (two and four in the examples) of least significant bits and small phase perturbation values for the MSB are a key to the algorithm performance. This algorithm has the important advantage of being simple to implement on existing phased arrays. Disadvantages include little success at nulling interference entering a quantization sideloop and interference at symmetric angles about the main beam.

REFERENCES


**Randy L. Haupt** (M’82–S’84–SM’90) received the B.S. degree in electrical engineering from the USAF Academy, CO, the M.S. degree in engineering management from Western New England College, Springfield, MA, the M.S. degree in electrical engineering from Northeastern University, Boston, MA, and the Ph.D. degree in electrical engineering from University of Michigan, Ann Arbor, MI, in 1978, 1981, 1983, and 1987, respectively.

He is currently a Lieutenant Colonel in the U.S. Air Force and a Professor of electrical engineering at the USAF Academy, CO. He worked as a Project Engineer on the Over-the-Horizon Radar and as a Research Engineer for Rome Air Development Center, Hanscom AFB, MA.

Dr. Haupt was Federal Engineer of the Year in 1993. He is a member of Tau Beta Pi, URSI Commission B, and the Electromagnetics Academy.