Adaptive Wide Nulling for Arbitrary DBF Array Using Particle Swarm Optimization

Xiaoli Xu and #Yilong Lu

School of Electrical & Electronic Engineering, Nanyang Technological University
Nanyang Avenue Singapore 639798
eylu@ntu.edu.sg

1. Introduction

Digital beamforming (DBF) is a rapidly developing technology which is the most advanced approach to array antenna pattern control. When implemented at the array element level, DBF enables full utilization of the maximum number of degrees of freedom in the array, leading to significant improvements in beamforming of simultaneous multiple independent beams, adaptive pattern nulling, space-time adaptive processing (STAP), and direction finding (DF), compared to traditional analog array control techniques. In order to suppress the interference from the environment, nulls are placed in the antenna patterns in the direction of interfering source and signal reflections. The nulls can be narrow nulls or wide nulls. Because of the increasing electromagnetic pollution of the environment, the technique of placing wide nulls to suppress the interference from wideband jammer becomes more important nowadays [1]-[3].

Steering nulls and adjusting SLL can be achieved by controlling the amplitude and phase of signal at each antenna element. In a DBF array, controlling the amplitude and phase of a signal is realized by multiplying a complex number, often called a complex weight. The complex weights are altered adaptively to maximize the communication channel. In this study, a powerful and efficient Particle Swarm Optimization (PSO) technique [4] is studied and applied to adaptive wide nulling for arbitrary DBF Array. Compared to another very popular optimization tool, Genetic Algorithm (GA) [5], PSO is much simpler and easier to implement because the particles inside the swarm update only based on the internal velocity, there are no crossover and mutation operations involved. Moreover, the study indicates that PSO can converge faster when solving the beamforming problem [6], [7]. Detailed information about this approach will be provided in the following context.

2. Problem Formulation

It is well known that for beamforming, the pattern function of an arbitrary N-element array with identical elements can be expressed by (1):

$$\mathbf{F}(\theta,\phi) = \mathbf{w}^T \cdot \mathbf{S}(\theta,\phi) \tag{1}$$

where $\mathbf{w} = \{w_n\}$ is a complex weighting column vector, $\mathbf{S}(\theta,\phi) = \{\exp(jk\mathbf{r}_n \bullet \mathbf{a}_r)\}$ is the steering column vector, and $\mathbf{r}_n$ the element location vectors, $\mathbf{a}_r$ the unit vector of distance ray,

$$\mathbf{r}_n = \mathbf{a}_x x_n + \mathbf{a}_y y_n + \mathbf{a}_z z_n$$

$$\mathbf{a}_r = \mathbf{a}_x \sin\theta\cos\phi + \mathbf{a}_y \sin\theta\sin\phi + \mathbf{a}_z \cos\theta$$

θ and ϕ are elevation and azimuth angles of the spherical coordinate, respectively.

The major task of adaptive digital beamforming is to find the proper complex weight for each antenna element, so that desired pattern shape, including sidelobe level (SLL) suppression and null formation, can be achieved.

3. Particle Swarm Optimization and Solution Procedures

Particle Swarm Optimization (PSO) is a stochastic optimization technique which mimics the social behaviour of bird flocks and fish swarms [4]. The initial swarm is generated randomly within the bounded space. The particle inside the swarm updates its position based on its own experience and the experience of other particles, until any of termination criteria is satisfied. The interactions among particles inside the swarm are described as "Swarm Intelligence".

We use the standard PSO algorithm to optimize (1) and the detailed procedures are described as follows.

3.1 Variable Encoding and Initialization

The complex weights are represented by the particles inside the swarm. Each particle is a vector of float numbers and the length of the vector equals to double of the antenna array's length. The first half of the vector represents the real part of the complex weight, and the second half represents the imaginary part. The real part and imaginary part are combined together when evaluating the fitness, but are handled separately during updating operations.

The optimizer initially generates a group of, say M, random weighting vectors as potential solutions. Each weighting vector is called a particle with its fitness value evaluated from the specified objective function. In this study, we choose M=100 and also introduce a pattern in the initial swarm from the classic Chebyshev solution of the specified SLL in order to accelerate the optimization process.

3.2 Fitness Definition and Evaluation

The input beam pattern is formed based on the input parameter vector which is denoted by [null pointing direction, null width, null depth, sidelobe level, main beam pointing direction, main beam width]. The dotted line in Figure 1 illustrates a sample ideal beam pattern with the parameter vector equals to [36, 14, 30, -20, 90, 30].

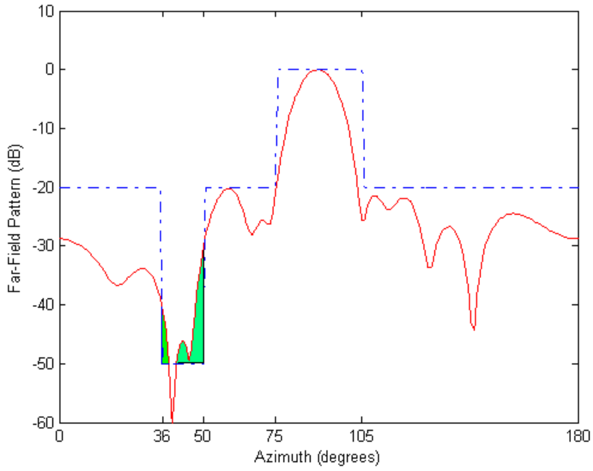


Figure 1. Desired pattern template and fitness evaluation.

The error of a solution is measured by summation of the area of a test pattern exceeding the ideal beam pattern template, as the shading area shown in Figure 1. The optimization objective is to minimize the error or find the set of complex weights for a pattern “particle” with zero error or the smallest error compared to other pattern “particles”.

3.3 Swarm Update and Termination Criteria

Each particle will keep track of its positions in the problem space and store the position with the best fitness value it has achieved so far as the population best, say, $\mathbf{P}_{best}$. The optimizer will also keep track of the particle position that has the best fitness value obtained so far by any particle in the population. This position is taken as the global best, say, $\mathbf{G}_{best}$. The population best and the global best are initialized as the following:

$$\mathbf{P}_{best}^{j} \leftarrow \mathbf{\Phi}^{j} \text{ and } \mathbf{G}_{best} \leftarrow \arg\min Obj_{Total}(\mathbf{\Phi}^{j}),\ j = 1,\ldots,M \qquad (2)$$

After initialization, iteration progress will begin. In each iteration, particles are employed to evaluate their fitness values based the objective function, and the fitness values are taken to update the $\mathbf{G}_{best}$ and $\mathbf{P}_{best}$. Afterwards each particle would be updated according to (3) and (4):

$$\mathbf{V}(i+1)=\alpha\cdot\mathbf{V}(i)+\beta\cdot\mathbf{U}(0,1)\odot\left(\mathbf{P}_{best}(i)-\mathbf{X}(i)\right)+\gamma\cdot\mathbf{U}(0,1)\odot\left(\mathbf{G}_{best}(i)-\mathbf{X}(i)\right) \tag{3}$$

$$\mathbf{X}(i+1)=\mathbf{X}(i)+\mathbf{V}(i) \tag{4}$$

where $\mathbf{X}(i)$ is the particle position of the i^{th} iteration and $\mathbf{X}(i+1)$ is the particle position of the following iteration, $\mathbf{V}(i)$ is the velocity of the i^{th} iteration, $\mathbf{V}(i+1)$ the next iteration, α is inertia weight, β and γ are learning factors, $\mathbf{U}(0,1)$ is a random vector with elements uniformly distributed in the region of [0,1], operator $\odot$ means the Hadamard matrix operator. The selection of parameters α, β, and γ can be referred to [8], that is, $\beta = \gamma = 2$ and α has an initial value around 1 and gradually declines towards 0. In this study, we set α to change gradually from 0.95 to 0.3. PSO will iterate until the desired fitness value is achieved or a given maximum number of iterations is reached.

The maximum number of iterations and the desired error goal are defined before the loop starts. PSO exit the loop if either one of the above criteria is satisfied. The figure which plots the pattern function together with the ideal beam pattern is updated at every 5 iterations. A log file of the PSO process in terms of the increasing fitness, and the matrix representing the swarm are stored in the hard disk for future reviewing.

4. Simulation Results

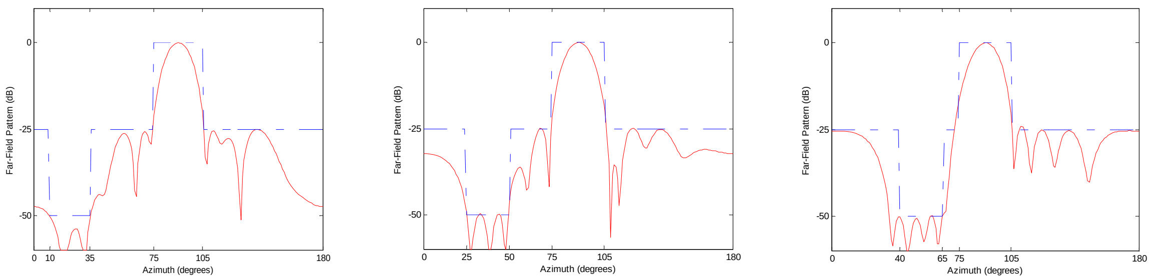
To show the effectiveness of this approach, the proposed PSO approach is tested by solving beamforming for adaptive wide nulling of a linear array.

Assuming a linear antenna array with N isotropic sources that are equally spaced, we have the steering vector expressed by (5):

$$\mathbf{S}(\phi)=\left\{\exp\left(jkd\,(n-\frac{N+1}{2})(\cos\phi-\cos\phi_m)\right)\right\} \tag{5}$$

where ϕ_m is the main bam pointing direction provided the elevation is fixed at 90°. If the main beam pointing direction in above equation is changed, the same set of complex weights for the main beam at broadside can still be used.

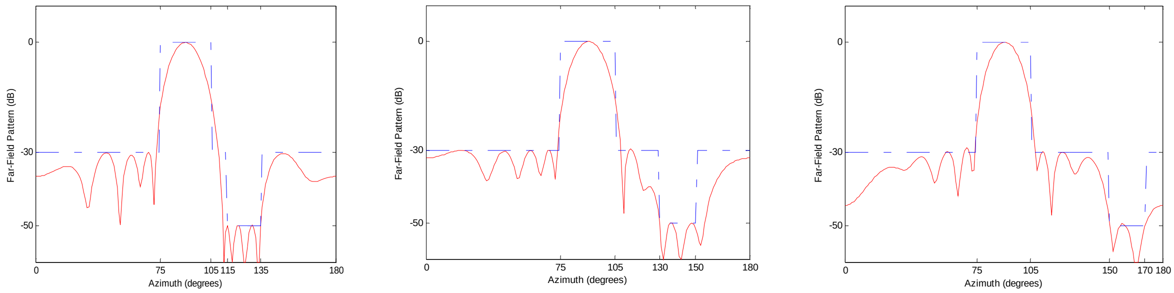
PSO progresses not as fast as GA at the first few iterations. However, when iterations go on, the PSO solution gets improved much fast than that of GA at the same number of generations. To get a satisfactory solution, the number of iterations for PSO is averaged at about 300. For the examples shown in Figures 2 and 3, the dot-dash lines depict the template casted for that case while the solid line represent the resultant patterns computed by the PSO with the optimum weights.



(a) (b) (c)

Figure 2. Wide null of 25 degree width and 25 degree depth with respect to SLL.

Figure 2 shows null steering examples with same SLL of –25dB and a wide null with the depth of –25dB and width of 25 degrees azimuth. In Figure 2(a), the null is steered to stretch from 10 to 35 degrees azimuth, and it is later steered to extend from 20 to 50 degrees and 40 to 65 degrees in Figure 2(b) and Figure 2(c) respectively.



(a) (b) (c)

Figure 3. Wide null of 25 degree width in the right side and 35 degree depth with respect to SLL.

Figure 3 gives more wide nulling and steering examples with same SLL of –30 dB and the null locating on the right side of the main beam.

Conclusion

In this study, Particle Swarm Optimization (PSO) is applied for adaptive wide nulling or null steering in digital beamforming arrays. The presented PSO approach has been proven to be much simpler and efficient than many genetic algorithms (GA) based approaches for beamforming including sidelobe suppression, wide nulling, and null steering. The presented approach can be easily applied to any other beamforming for arbitrary arrays.

References

[1] I. El-Azhary, M.S. Afifi, P.S. Excell,, "Fast cancellation of sidelobes in the pattern of a uniformly excited array using external elements," *IEEE Trans. Antennas Propagat.*, vol. 38, no. 12, Dec 1990.

[2] T. Gao, Y. Guo and J. Li. "Wide null and low sidelobe pattern synthesis for phased array antennas," *Proc. 1993 Asia-Pacific Microwave Conf.*, vol. 1, pp.42-45, Hsinchu, Taiwan , 18-20 Oct 1993.

[3] Y.L Lu, B.K. Yeo, "Adaptive wide null steering for digital beamforming array with the complex coded genetic algorithm", *Proc. 2000 IEEE Int. Conf. on Phased Array Systems and Technology*, pp.557-560, Dana Point CA, USA, 21-25 May 2000.

[4] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Proc. IEEE Conf. on Neural Networks*, vol. 4, pp.1942-1948, Perth, Australia, 27 Nov-1 Dec 1995.

[5] L. Chambers, *Practical Handbook of Genetic Algorithms: Applications*, vol. 1. Boca Raton, CRC, 1995.

[6] K.K. Yan and Y.L. Lu, "Sidelobe reduction in array pattern synthesis using genetic algorithm", *IEEE Trans. Antennas Propagat*, vol. 45, no. 7, pp.1117-1122, 1997.

[7] B.K. Yeo and Y.L. Lu, "Array failure correction with a genetic algorithm," *IEEE Trans. Antennas Propagat.*, vol. 47, no. 5, pp.823-828, May 1999.

[8] I.C. Trelea, "The particle swarm optimization algorithm: convergence analysis and parameter selection," *Information Processing Lett.*, vol. 85, pp.317-325, 2003.