A Numerical Pattern Synthesis Algorithm for Arbitrary Arrays

Phan Anh, E& Center, Faculty of Tech, Hanoi National University, Vietnam*
Hoang Dinh Thuyen, Dao Duc Kinh, Le Quy Don University, Vietnam*
phananh@fpt.vn ;* ddkinh.@yahoo.com

Abstract— **In this paper, a numerical method for antenna array pattern synthesis is presented. By this method, the designer can efficiently verify both mainlobe shaping, null steering and sidelobe levels. When a large number of interfering signals occurred at various angles throughout the sidelobe region, the sidelobes are controlled by an iterative method based on adaptive array theory. The values of the weighting function in the L_2 norm, interpreted as imaginary jammers, and are iterated to minimize exceedance of the desired sidelobe levels and minimize the absolute difference between desired and achieved mainlobe patterns. Simulation examples, including both nonuniform linear and planar arrays, are shown to illustrate the effective of this method.**

I. Introduction

Due to the increasing pollution of the electromagnetic environment, pattern synthesis technique with nulls steered to the interference directions become currently important.

The synthesis of equi-spaced linear array pattern with shaped beams has been considered by several authors in the specialized literature [1, 2, 3, 4, 6]. Particularly, the Schelkunoff Polynomial Method is very suitable for synthesizing linear arrays with a radiation pattern specified by several nulls. In this method, the array factor is viewed as a polynomial with roots located in the complex ω plane.

When the number of interfering sources is much less than a half of the total number of elements in the linear array, it is possible to optimize the pattern as well as to suppress interfering signals. However, when a large number of closely spaced interfering signals are assumed to be incident on the array from the sidelobe region, so the array cannot easy place a null on each interfering signal but a compromise pattern that minimizes the sidelobe levels is instead.

A classic paper by Dolph [1] showed how to obtain the weights for an uniform linear array (ULA) to achieve a Chebyshev pattern, with optimal in the sense that it yields a minimum uniform sidelobe level for a given mainlobe width.

Olen and Compton [7] presented a numerical synthesis algorithm that can be used for arbitrary arrays with arbitrary element. This algorithm is very effective and generally yields satisfactory array patterns. However, there is no pattern control mechanism in mainlobe region.

In this paper, we present a pattern synthesis algorithm for arbitrary arrays based on adaptive array theory. The imaginary jammer powers are varied depending on desired sidelobe levels, and adjusted by an iterative procedure.

We establish the problem as finding the optimal array weighting vector that minimizes the weighted L_2 norm of the difference between synthesized pattern and desired pattern, with null constraints, or using our algorithm for iterating the value of the weighting function in order to minimize the exceedance of the desired sidelobe levels and to minimize the absolute difference between desired and achieved patterns in the mainlobe region.

II. The Problem Formulation

The problem of array pattern synthesis can be stated as follows. With a given number of array elements and their positions, we have to find a set of complex weights w_i so that the array pattern $P_y(\theta)$ has a maximum at the desired direction θ_d with a certain beamwidth and also the sidelobe levels meet the specified values. First, we consider the sum of a weighted pattern errors E over the set of angles $\theta_1, \theta_2, \ldots, \theta_M$,

$$E = \sum_{i=1}^{M} f(\theta_i) \left| P_y(\theta_i) - P_r(\theta_i) \right|^2 \tag{1}$$

where

$$P_y(\theta) = V_s^H(\theta) W \tag{2}$$

and

$$V_s(\theta_i) = [g_1(\theta_i)e^{j\phi_1(\theta_i)} \; g_2(\theta_i)e^{j\phi_2(\theta_i)} \ldots g_N(\theta_i)e^{j\phi_N(\theta_i)}]^H$$

is the steering vector of the array, the superscript H denotes the complex conjugate transpose; $g_i(\theta)$ is the ith element pattern; $P_r(\theta_i)$ is the reference pattern; $f(\theta_i)$ is the weighting function; $\phi_i(\theta) = k.x_i$ is the phase due to propagation where k is wavenumber vector and x_i is ith element position; $W = [w_1, w_2, \ldots, w_N]^T$ is weighting vector. Then the error E can be rewritten as

$$E = \sum_{i=1}^{M} f(\theta_i) \left| V_s^H(\theta_i) W - P_r(\theta_i) \right|^2 \tag{3}$$

We note that E may be interpreted as average output power of a "sidelobe canceler " with a main channel response $P_r(\theta)$ to a collection of jammers (Fig. 1), where the ith jammer has the location θ_i and the power $f(\theta_i)$. The key to this algorithm is that the jammer powers are adjusted to emphasize selected parts of the achieved pattern, particularly the mainlobe and sidelobe peaks.

Suppose $\theta_{01}, \theta_{02}, \ldots, \theta_{0k}$ are the localized nulls, which have to synthesize. We have

$$V_s^H(\theta_{0i}) W = 0 \qquad i = 0, 1, 2, \ldots, k \tag{4}$$

(4) can be written by a matrix equation

$$CW = h \tag{5}$$

where

$$C = \left[V_s^H(\theta_{01}), V_s^H(\theta_{02}), \ldots, V_s^H(\theta_{0k}) \right]^T$$
$$h = [0, 0, \ldots, 0]^T$$

The pattern synthesis problem may be outlined as follows. We should find the weighting vector W to minimize the error E, subject to the constraint (5).

$$\min_{w} \sum_{i=1}^{M} f(\theta_i) \left| V_s^H(\theta_i) W - P_r(\theta_i) \right|^2,$$

subject to $CW = h$

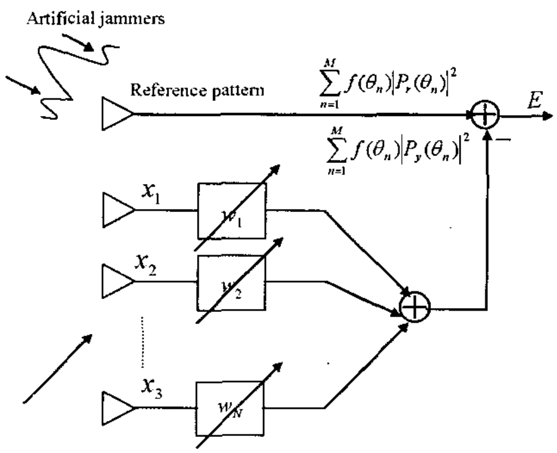


Fig. 1. An sidelobe canceler interpretation

This constrained minimization can be accomplished by forming the Lagrangian [6].

$$J = \sum_{i=1}^{M} f(\theta_i)\left|V_s^H(\theta_i)W - P_r(\theta_i)\right|^2 + \lambda[h - CW] + [h - CW]^H \lambda^H \tag{6}$$

where λ is a Lagrange multipliers vector.

Equation (6) can be written by

$$J = (V^H W - P)^H [F](V^H W - P) + \lambda[h - CW] + [h - CW]^H \lambda^H \tag{7}$$

where $[F]$ is a weighting matrix of the weighting function, it is a diagonal matrix

$$[F] = [diag\, f(\theta_i)]_{M \times M} \tag{8}$$

$$V^H = \left[V_s^H(\theta_1), V_s^H(\theta_2), \ldots, V_s^H(\theta_M)\right]^T \tag{9}$$

and

$$P = [P_r(\theta_1), P_r(\theta_2), \ldots, P_r(\theta_M)]^T \tag{10}$$

The minimization of (7) is obtained by setting the partial derivatives of J with respect to both the real and imaginary parts of W equal to zero, or equivalently with respect to W^H [6]. The solution for optimal weight vector is

$$W_{opt} = R_s^{-1}R_d + R_s^{-1}C^H (CR_s^{-1}C^H)^{-1}(h - CR_s^{-1}R_d) \tag{11}$$

where R_s is the covariance matrix and R_d is the cross-correlation vector defined as

$$R_s = VFV^H \tag{12}$$

$$R_d = [VF]P \tag{13}$$

The error E is found to be [6]

$$E = E_{\min} + (W - W_{opt})^H R_s (W - W_{opt})$$

where $E_{\min}$ is the minimized error E

$$E_{\min} = \sum_{i=1}^{M} f(\theta_i)|P_r(\theta_i)|^2 - W_{opt}^H R_s W_{opt}$$

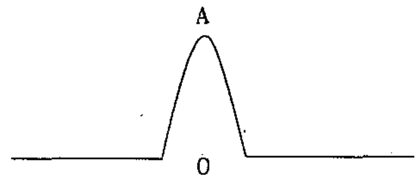


Fig.2. The reference pattern $P_r(\theta)$

When constraints are not needed, the optimal weight vector is

$$W_{opt} = R_s^{-1} R_d \tag{14}$$

The array response at each angular location depends on the weighting function $f(\theta_i)$. Different values of $f(\theta_i)$ put different emphasis on array responses at pertinent directions and, therefore, result a relevant array pattern. By making $f(\theta_i)$ large enough, it is possible to ensure sidelobe peaks do not exceed a certain value.

III. The Pattern-Synthesis Algorithm

The response of an array pattern to interfering signals depends on the number of interfering signals in relation to the number of freedom order in the array. An N-element array has N–1 order of freedom in its pattern. One order of freedom is needed to form a pattern maximum on the desired signal. The remaining N–2 order are available to null interference signals. If N–2 or less of interference signals are incident on the array, the array usually forms a null on each interference signal. However, if more than N–2 interference signals are incident, the array can not normally make null the individual interference signals but instead, it will be found a compromise pattern that minimizes the total interference power at the array output.

The most common objective for pattern synthesis is to obtain a pattern with sidelobe level lower than a specified value over a certain regions while maintaining a certain gain at looking angle θ_d. Here, we choose the reference pattern $P_r(\theta)$ as shown in Fig. 2, in which all the responses in sidelobe regions are zeros and the mainlobe peak response is of a value A. The mainlobe shape is specified by the designer, it may be a parabola, for example. While it is impractical to have all zero sidelobe levels, we can make lower and lower sidelobes by increasing the weighting function $f(\theta)$ in selected areas. We use a realistic desired pattern $P_d(\theta)$ to iteratively adjust $f(\theta)$ until the sidelobe requirements in $P_d(\theta)$ are met.

The weighting function is up dated through an iteration procedure similar to that of Olen and Compton [7], which leads to a satisfactory array pattern. The iteration process is as in (15) and (16), shown on the bottom of the page, where n =1,2,,.. M index the points in angle over which we are interested in controlling the pattern.

$$f_{k+1}(\theta_n) = \begin{cases} h_k(\theta_n), & \theta_n \ \textit{in mainlobe region} \\ \max\{f_k(\theta_n) + K_p[P_{yk}(\theta_n) - P_d(\theta_n)], 0\} & \theta_n \ \textit{in sidelobe region} \end{cases} \tag{15}$$

$$h_k(\theta_n) = \begin{cases} f_k(\theta_n), & \textit{if}\ \left|P_{yk}(\theta_n) - P_d(\theta_n)\right| \le \varepsilon \\ f_k(\theta_n) + K_m\left|P_{yk}(\theta_n) - P_d(\theta_n)\right|, & \textit{otherwise} \end{cases}$$

(16)

The $f_k(\theta_n)$ and $P_{yk}(\theta_n)$ are weighting function and synthesized pattern, respectively, at the kth iteration and ε is a very small quantity for an error tolerance between the synthesized pattern and the desired pattern in mainlobe region. The $P_d(\theta_n)$ is desired pattern; K_m and K_p are iteration gains. Usually, K_p is specified to be much smaller than K_m for exa-mple, $K_p = 3$ and $K_m = 1000$. We note that for θ_n in the main-lobe region, $f_k(\theta_n)$ has never decreased from its initial value. The desired pattern $P_d(\theta)$ is set up to facilitate the iteration process whereas the reference pattern $P_r(\theta)$ is used to define the pattern errors that are to be minimized. In general, $P_d(\theta)$ and $P_r(\theta)$ are the same in mainlobe regions but different in sidelobe regions. The sidelobe part of, $P_d(\theta_n)$ should be chosen according to a realistic specification or a reasonable estimation, for example, the uniform sidelobe level.

We next use $f_{k+1}(\theta_n)$ to compute new weights. Let θ_L and θ_R be the boundary points for mainlobe region, i.e. $\theta_L \le \theta_n \le \theta_R$ defines the mainlobe. Since the reference pattern is zero outside of this region, the cross-correlation vector and the covariance matrix become

$$R_d(k+1) = VF(k+1)P$$

$$= \sum_{n=\theta_L}^{\theta_R} f_{k+1}(\theta_n) P_r(\theta_n) V^H(\theta_n) \tag{17}$$

$$R_s(k+1) = \sigma^2 I + VF(k+1)V^H \tag{18}$$

where $F(k+1)$ is the weighting matrix (8) at the kth iteration, and a small quantity σ^2 is added to each diagonal element of the covariance matrix to prevent it from being ill conditioned [8], for example, 0.0001. Then the next weight vector is

$$W_{opt}(k+1) = R_s^{-1}(k+1) R_d(k+1) \tag{19}$$

The iteration stops when the errors between $P_{yk}(\theta)$ and $P_d(\theta)$ are small enough in the mainlobe region and the sidelobe levels of $P_{yk}(\theta)$ are equal to or lower than $P_d(\theta)$.

TABLE I

A nonuniform linear array of 21-element

Element Nos.	Position	Element Nos.	Position
1,21	$\pm 5.0\lambda$	6,16	$\pm 2.3\lambda$
2,20	$\pm 4.6\lambda$	7,15	$\pm 1.9\lambda$
3,19	$\pm 3.9\lambda$	8,14	$\pm 1.5\lambda$
4,18	$\pm 3.3\lambda$	9,13	$\pm 0.7\lambda$
5,17	$\pm 2.9\lambda$	10,12	$\pm 0.3\lambda$
		11	0λ

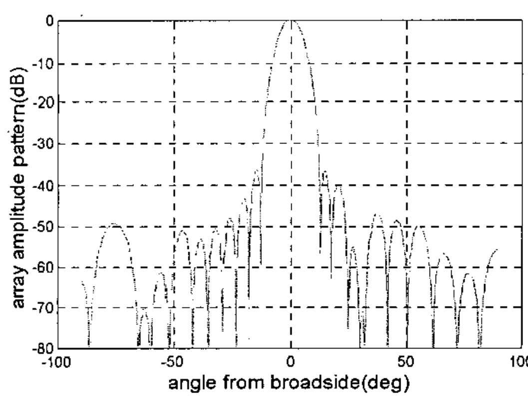


Fig. 3. The synthesized pattern the localized nulls at -65°, -60° , 25° 30°

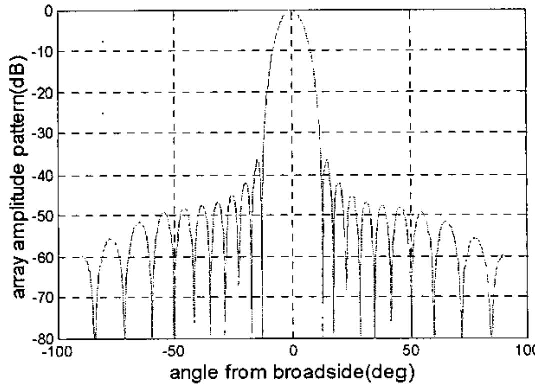


Fig. 4. Initial pattern.

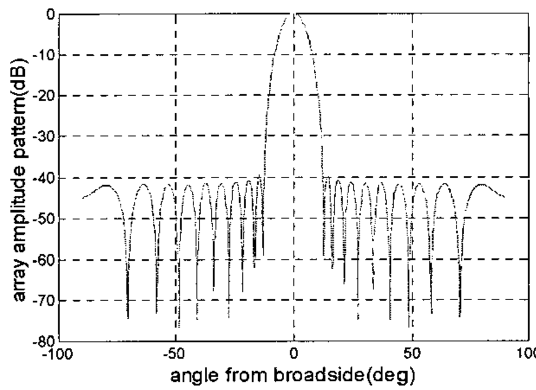


Fig. 5. Intermediate pattern.

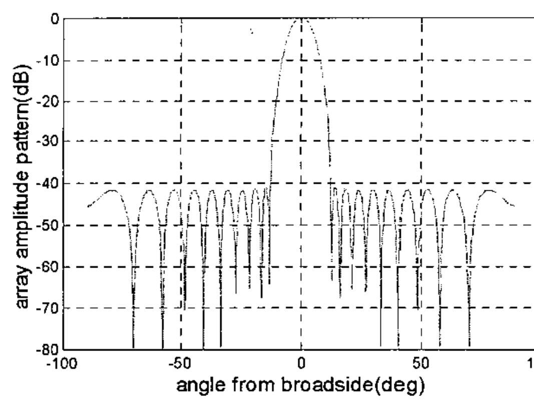


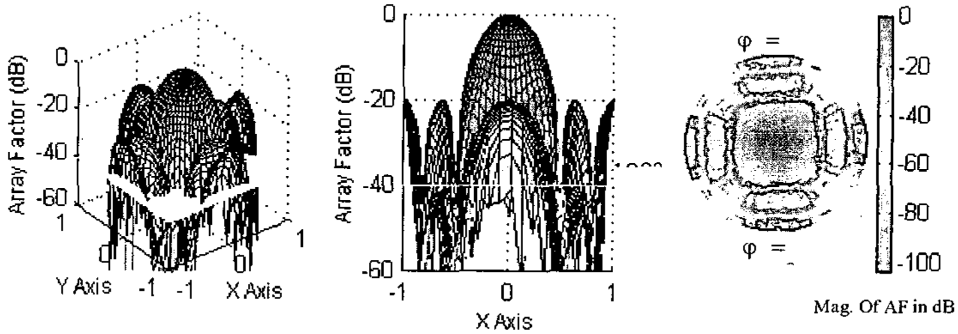
Fig. 6. The synthesized pattern.

IV. Simulation Examples

In this section, we will show a few pattern synthesis examples using our algorithm. The array elements used in the examples are assumed to be isotropic although such an assumption is not necessary in our algorithm.

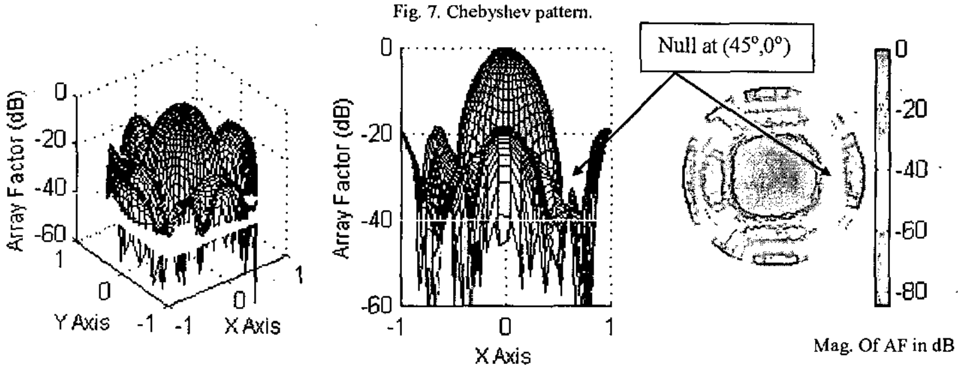
Example 1: We consider a pattern synthesis for a non-uniform linear array of 21 elements with element positions shown in the Table I. The selected reference pattern is $P_r(\theta)=\cos^2(7\theta)$ in the main lobe. We used 1° spacing from -90° to 90° for placing values of the weighting function.

The firs ons of
obtain o



(a) (b) (c)

Fig. 7. Chebyshev pattern.



(a) (b) (c)

Fig. 8. Synthesized pattern

Fig. 3 shows the resulting synthesized pattern with the localized nulls at -65°, -60°, 25°, and 30°.

The second case is using an iterative method for array pattern synthesis with the uniform sidelobe levels. Fig. 4 shows the initial pattern, Fig. 5 shows an intermediate pattern and Fig. 6 is the final synthesized pattern which has a sidelobe level lower than -40dB. Here we selected K_p =5 and K_m =500.

Example 2: The problem is to synthesize a 2-D Chebyshev pattern for a 5×5 rectangular uniform planar array of 25 elements with haft-wavelength spacing. In this case θ_n is replaced with (θ_n, φ_m).

The initial 2-D pattern is plotted in Fig. 7(a) as a function of $x=\sin\theta\cos\varphi$ and $y=\sin\theta\sin\varphi$. A side view of the initial pattern is plotted in Fig. 7(b) and a top-down view is plotted in Fig. 7(c). Figs. 8(a), (b), and (c) show three views of the final synthesized pattern with a single null located at angle $(45^{\circ},0^{\circ})$. The null is indicated by the white regions at the specified direction.

V. Conclusion

In this paper a numerical pattern synthesis algorithm for arbitrary arrays has been presented. The optimal weighting vector is obtained by minimizing the sum of weighted squared errors between synthesized and desired patterns, with or without null constraints. The weighting functions are adjusted iteratively in both mainlobe and sidelobe regions to insure a desired mainlobe shape as well as desired sidelobe levels.

References

[1] R. E. Collin and F. J. Zucker, *Antenna Theory-Part 1.* New York : McGraw-Hill, 1969.

[2] W. L. Stuzman, "Synthesis of shaped-beam radiation pattern using the iterative sampling method," *IEEE Trans. Antennas Propagat.*, vol. 19, pp. 36-41, Jan. 1971.

[3] J. Perini, "Note on antenna pattern synthesis using numerical iterative methods," *IEEE Trans. Antennas Propagat.*, vol. 19, pp. 284-286, Mar. 1971.

[4] P. K. Murthy and A. Kumar, "Synthesis of linear antenna arrays," *IEEE Trans. Antennas Propagat.*, vol. 24, pp. 865-870, Nov. 1976.

[9] P. Y. Zhou, M. A. Ingram and P. D. Anderson, "Synthesis of minimax sidelobes for arbitrary arrays," *IEEE Trans. Antennas Propagat.*, vol. 46, pp. 1759-1760, Nov. 1998.

[6] R.A. Monzingo and T.W. Miller, *Introduction to Adaptive Arrays.* New York : Wilay, 1980.

[7] C. A. Olen and R. T. Compton, "A numerical pattern synthesis algorithm for arrays," *IEEE Trans. Antennas Propagat.*, vol. 41, pp. 1666-1676, Oct. 1990.

[8] C.-Y. Tseng and L. J. Griffiths, "A simple algorithm to achieve desired patterns for arbitrary arrays," *IEEE Trans. Signal Processing.*, vol. 40, pp. 2737-2746, Nov. 1992.

[9] P. Y. Zhou, and M. A. Ingram, "Pattern synthesis for arbitrary arrays using an adaptive array method," *IEEE Trans. Antennas Propagat.*, vol. 47, pp. 862-869, May. 1999.

[10] E. C. Dufort, "Pattern synthesis based on adaptive array theory," *IEEE Trans. Antennas Propagat.*, vol. 37, pp. 1011-1018, Aug. 1989.