Multi-objective Design of Linear Antenna Arrays with Particle Swarm Optimization

Nik Noordini Nik Abd Malik1, Mazlina Esa2, Syarifah Kamilah Syed Yusof, and Shipun Anuar Hamzah
Telematics Research Group, Faculty of Electrical Engineering, Universiti Teknologi Malaysia Johor Bahru, Johor, Malaysia, 1noordini@fke.utm.my, 2mazlina@fke.utm.my

1. Introduction

The presence of large sidelobe radiation beam levels of an antenna is undesirable as the antenna performance and efficiency will be greatly degraded. Techniques for designing antenna arrays can be realized by determining the physical layout, and a set of the amplitude and phase of the current excitations of the antenna array. The design problems strive for the desired radiation patterns which satisfy several conflicting objectives such as controllable beamwidth (HPBW), minimum sidelobe level (SLL), maximum directivity, and noise sensitivity. Thus, the design problem worsens if multi-objective criteria is considered.

Much effort has been introduced in antenna designs by using evolutional optimization algorithm such as genetic algorithm (GA) [1], [2], simulated annealing (SA), and particle swarm optimization (PSO) [3]. PSO has gained tremendous popularity as a new approach to optimization problems in electromagnetics. It has been used to obtain the excitation coefficients and optimise the amplitude, phase, spacing and position of an array antenna [4]-[7].

This paper applies an efficient PSO algorithm [8] to deal with the multi-objective problems (minimum sidelobe level (SLL) and controllable first-null beamwidth (FNBW)) at the same time. The PSO algorithm is used to locate the optimum element distance positions based on symmetric and even element linear antenna arrays (LAA) of isotropic radiators. The trade-off performance between the above two criteria is shown for different cases of 6- and 10-element LAA. The successfully generated radiation patterns demonstrate the capability of the developed PSO algorithm in achieving optimization. The results obtained showed that the PSO algorithm is capable of finding the optimal solution in almost all cases with superior performance over GA [1].

2. Array Factor and Geometry

Consider a one-dimensional symmetric LAA located along the x-axis depicted in Fig. 1. It has even number of elements up to \( N \). The array factor is written as:

\[
AF(\theta) = \frac{2}{\lambda} \sum_{n=1}^{N} I_n \cos(kd_n \cos(\theta)).
\]

where \( k \), \( I_n \), \( \theta \), and \( d_n \) are the wavenumber \( k = 2\pi/\lambda \), excitation amplitude (assume \( I_n = 1 \)), observation angle, and location of the \( n \)th element from the reference node at the origin, respectively. The distance \( d_n \) represents \( d_1 = x_1; d_2 = d_1 + x_2; d_3 = d_2 + x_3; \ldots; d_N = d_{N-1} + x_N \).

PSO will search for the optimum element distance positions \( x_N \), thus dealing with the multi-objective design problems of controllable FNBW and minimum SLL.

3. Particle Swarm Optimization

PSO is a computational intelligence optimizer which based on the behaviour and movement of a swarm of bees, a flock of birds or a school of fish during their food searching activities [8]. The bees are referred as particles which represents a potential solution to the optimization. With its own
directional velocity and position, the particle accelerates towards the best personal, pbest, and best overall location, gbest, while continuously verifying the value of its current location.

In this work, the PSO starts with generating randomly initial position \( X \) and velocity \( V \) for each particle as in Fig. 2. In an N-dimensional problem with \( i \) particles, \( X = [x_{i1}, x_{i2}, x_{i3}, \ldots, x_{iN}] \), \( X \) as the distance position of the elements, \( x_{ij} \) is limited between two boundaries, \( U_i \) and \( L_i \), i.e. \( (L_i \leq x_{ij} \leq U_i) \) and \( x_{iN,1} \) is limited between two other boundaries, \( U_N \) and \( L_N \), i.e. \( (L_N \leq x_{iN,1} \leq U_N) \). Each row of the position matrix, \( X \), represents a possible solution to the optimization problem.

The \( i \)-th particle in the solution space is determined by a fitness function value which depends on the distance position. The particle value will be improved by regards to the value of the fitness function i.e. \( F_{\text{min}} = F(g_{\text{b}}) \) is minimized. The best previous position, \( \text{pbest} \) of the \( i \)-th particle can be defined as position matrix for an individual particle’s best fitness function, \( \text{pbest} = [p_{i1}, p_{i2}, p_{i3}, \ldots, p_i] \). The global best position, \( \text{gbest} \), is the position in the search space at which the best fitness function was achieved among all particles. It is defined as, \( G = [g_{1}, g_{2}, g_{3}, \ldots, g_{N}] \). The positions, \( X \) and velocities, \( V = [v_{i1}, v_{i2}, v_{i3}, \ldots, v_{iN}] \) of particles are then updated in every iteration according to (2) and (3). Therefore, each particle should know its pbest and gbest:

\[
v_{iN}(t + 1) = w v_{iN}(t) + c_1 \text{rand}(t)[p_{i}(t + 1) - x_{iN}(t)] + c_2 \text{Rand}(t)[g_{N}(t + 1) - x_{iN}(t)]
\]

(2)

The velocity of each particle depends on the distance of the current position to the position with the best fitness function. In eq. (2), \( t+1 \) and \( t \) refer to the time index of the current and previous iterations, \( \text{rand}(t) \) and \( \text{Rand}(t) \) are functions that generate random numbers between 0.0 and 1.0. The parameters \( c_1 \) and \( c_2 \) are the relative weight of the pbest and gbest with value of 2.0 [4]. The inertial weight, \( w \) is linearly damped with iterations starting at 0.9 and decreasing linearly to 0.4 at the last iteration. After a time step, the new position of the particle is then given by:

\[
X(t + 1) = X(t) + V(t + 1)
\]

(3)

The PSO is employed to optimize the optimum distance position of the element in order to deal with multi-objective problem of the LAA. The objective of the algorithm is to find the gbest coordinates, \( G \), that correspond to the minimum value of the fitness function, \( F_{\text{min}} \) or \( F(g_1, g_2, g_3, \ldots, g_N) \). The fitness function is computed using:

\[
F = \min \sum_{i=1}^{s} |A F(\theta)_{ij}|^2
\]

(4)

where \( s \) is the regions where the SLL is minimized, \( i \) and \( j \) is the angles where the FNBW starts and stops. As the fitness function decreases, the radiation pattern improves with related particle’s position \( x_{iN} \). Therefore, the PSO algorithm will terminates successfully when the fitness function discovers its optimized minimum value. Fig. 2 shows the PSO flow-chart.

4. Simulation Results

Several cases are simulated and analysed with different number of antenna elements (6- and 10-elements of LAA). The experiments are performed for a controllable FNBW, which corresponds to the LAA design by using genetic algorithm (GA) as in [1]. In this work, the LAA is assumed to be symmetric about the x-axis. Initially, the particle’s position, \( X \) is generated randomly in the range of 0 to \( \lambda_o \), to produce more diverse possible solutions. The swarm size used is 30; the number of iterations is 500-2000. Some boundary conditions are also defined to \( d_{ij} \) which is allowed to vary from 0.25\( \lambda_o \) to 1.0\( \lambda_o \) except for \( d_i \) which is allowed to vary from 0.15\( \lambda_o \) to 1.0\( \lambda_o \). The results in Table 1 are compared to that obtained using GA result in [1].

Case 1. A 6-element LAA is simulated for controlled FNBW of 66° as in Fig. 3. Results are compared to the corresponding results obtained using GA in [1]. It can be clearly illustrated that the SLL suppression is improved. All the minor lobes have been minimised with the first SLL less than -20dB i.e. -20.04dB. The size of aperture of LAA obtained by using PSO (maximum=3.4546\( \lambda_o \) for FNBW of 66°) is slightly larger than that obtained using GA [1] (maximum=3.44\( \lambda_o \) for FNBW of 66°). Fig 4 shows the narrower controlled FNBW of 42°. The trade-off performance between SLL and controllable FNBW can be clearly seen in this behaviour. The SLL increases with the narrower
FNBW. However, the maximum SLL of -14.75 dB has been achieved by using PSO. From Table 1, it is also noted how much the aperture has been enlarged with narrower FNBW.

Case 2. The next array considered is a 10-element LAA as shown in Figs. 5 and 6. In both figures, the PSO results are better that GA results in [1]. A first observation from these plots is that a good performance of radiation pattern obtained from PSO even with a narrower FNBW. For a controlled FNBW of 26°, the SLL achieved -17.31 dB as in Fig. 5. Also, it is noted that 10-element LAA with FNBW of 28° maintains a low SLL throughout the angles. It is observed that the PSO algorithm generates a set of distance position of antenna element that provides a radiation pattern with the SLL is minimized until -20.31 dB.

5. Conclusions

The PSO algorithm was implemented to optimize the distance spacing between the neighbour element of 6- and 10- element LAA. The PSO can efficiently compute the trade-off performance between SLL and controllable FNBW. It can improve the radiation pattern of the LAA according to the desired multi-objective design for simultaneous minimum SLL with desired controllable FNBW. The simulation of the radiation pattern for the LAA is presented, and the design requirements are satisfied. It shows that the PSO can converge very fast. The developed PSO algorithm has successfully optimized the distance position of the array elements and demonstrate better performance than those presented in [1] by using GA.

![Fig.1 2N linear array geometry.](image)

![Flow-chart of the developed PSO.](image)

<table>
<thead>
<tr>
<th>Number of Elements (2N)</th>
<th>SLL(dB)</th>
<th>FNBW (deg)</th>
<th>[x1, x2, x3, ..., x/N] in λ₀’s</th>
<th>Aperture 2x(∑xᵢ) in λ₀’s</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>-20.0441</td>
<td>66</td>
<td>[0.4000, 0.3800, 0.9473]</td>
<td>∑=3.4546</td>
</tr>
<tr>
<td></td>
<td>-14.7522</td>
<td>42</td>
<td>[0.4900, 0.8100, 1.0956]</td>
<td>∑=4.7912</td>
</tr>
<tr>
<td>10</td>
<td>-17.3140</td>
<td>26</td>
<td>[0.4500, 0.7500, 0.9761, 0.9753, 1.1989]</td>
<td>∑=8.7006</td>
</tr>
<tr>
<td></td>
<td>-20.3097</td>
<td>28</td>
<td>[0.4000, 0.7500, 0.9263, 1.0789, 1.2646]</td>
<td>∑=8.9836</td>
</tr>
</tbody>
</table>
References


