Phase-Only Beam Broadening of Contiguous Uniform Subarrayed Arrays

Barry K. Daniel, *Senior Member, IEEE*, Adam L. Anderson, *Senior Member, IEEE*

***Abstract*—In modern antenna systems, beam broadening of subarrayed arrays provides continuous coverage of a wide angular extent in a cost-effective manner. While many methods have been published that address beam broadening of traditional (non-subarrayed) arrays, there is a knowledge gap in the published literature with respect to efficient beam broadening of contiguous uniform subarrayed arrays. This paper presents efficient methods for beam broadening of contiguous uniform subarrayed arrays where the excitation of each element is not individually controlled but the elements of the array are grouped together as subarrays to have the same element excitations. Particularly, this paper focuses on phase-only optimization to preserve maximum power output. Three modified iterative Fourier transform (IFT) methods and one genetic algorithm (GA) are presented to efficiently search the vast solution space of possible phase settings for a solution that satisfies the desired broadened pattern. These methods are evaluated on idealized 1x40 and 1x80 linear arrays with five element subarrays and 40x40 and 80x80 element rectangular arrays with 5x5 element subarrays. The proposed modified IFT methods are found to be faster than the GA approach while the GA approach only offer a few percentage points of better effectiveness.**

***Index Terms*—Antenna pattern synthesis, beam broadening, iterative Fourier transform, subarrayed arrays.**

I. Introduction

MANY of the instrumentation radars being delivered to military test centers around the world are designed with phase-only adjustable contiguous uniform subarrayed arrays to reduce cost while giving acceptable performance. These instrumentation radars utilize beam broadening for continuous measurement of objects over wide areas to increase the probability that unexpected phenomenology produced from prototype projectiles can be captured and characterized. Therefore, the generation of efficient and effective broadened beam patterns for contiguous uniform subarrayed arrays is essential and is the focus of this paper.

The architecture of a contiguous uniform subarrayed array reduces cost by limiting the number of control elements but introduces problems with the generation of broadened beam patterns because the elements are grouped together in a manner that only allows the amplitude and phase of each subarray, not each element, to be adjusted independently. Only allowing excitation adjustments at the subarray level makes it more difficult to form a low-ripple broadened main beam and to reduce the sidelobes. Subarrayed architectures also produce grating lobes which limit the effective electronic scanning of these types of arrays. However, currently delivered subarrayed instrumentation radars utilize mechanical movement to steer the widened beams instead of electronic steering.

These delivered instrumentation radars utilize solid-state transmit amplifiers which operate most efficiently in saturation (unity gain), it is often desired to utilize phase-only approaches to beam broadening as presented in [1]–[3] to optimize power efficiency. However, these published methods of beam broadening assume that the phase of each element in the array can be independently adjusted which is not true of contiguous uniform subarrayed arrays.

There is currently a gap in the available literature regarding efficient methods to broaden the main beam of subarrayed arrays when the synthesis is constrained to be phase-only, utilize contiguous uniform subarrays, and produce low sidelobes. Papers such as [4], [5] address synthesis with subarrayed architectures but they concentrate on optimization of combined sum and difference patterns, not broadening the main beam. The author in [6] proposes a synthesis method for subarrayed arrays but this method is amplitude only, utilizes subarrays of unequal sizes, and does not address beam broadening. The papers [7], [8] approach our desired domain space but utilize hybrid subarrays where the amplitude is dependent on the subarray amplitude but the phase of each element in the subarray can still be adjusted independently. To the best of our knowledge, the closest published material found is [9] where a beam broadening approach for subarrayed architectures is constrained to be phase-only and utilize contiguous subarrays. Unfortunately, this method constrains the widened full array beam widths to only those beam widths that are multiples of the subarray beam width often producing unacceptable sidelobe levels.

It is well known that the Fourier transform (FT) can be used to compute the array factor (AF) of a phased array antenna with uniformly spaced antenna elements [10] where the AF is defined as the radiation pattern of the array minus the effects of the individual element patterns. A single application of a simple transform method such as FT cannot be used effectively for phase-only synthesis applications because simple transform methods assume that the amplitude and phase of the elements or subarrays can be adjusted. Therefore, iterative processes using transform methods such as FT were developed. Early works such as [11], [12] show the benefits of utilizing iterative processes based on the FT to synthesize beam broadened array patterns. In [13], iterative use of the FT and the inverse FT was shown to enable generation of AFs with low sidelobes and directional nulls. In [14], [15], this iterative process was improved by detailing the usage of the more computationally

B. K. Daniel is with the Oak Ridge National Laboratory, Oak Ridge, TN, 37831 USA (e-mail: danielbk@ornl.gov).

A. L. Anderson is with MRSL Real-Time Systems Lab, Monterey, CA, 93933 USA e-mail: adam.anderson@mrsl.com.

efficient fast Fourier Transform (FFT) and the inverse fast Fourier transform (IFFT). However, these published IFT approaches assume that the phase of each element can be independently adjusted which is not true for subarrayed arrays.

This paper contributes to the currently published literature by presenting new efficient beam broadening methods for contiguous uniform subarrayed arrays by proposing three enhanced IFT methods and adapting the GA method for subarrayed arrays. The proposed IFT methods extend previously published IFT methods by accounting for contiguous uniform subarrayed architectures through the use of mean pooling techniques to aggregate the elements into subarrays and to ensure symmetry in the array pattern. Likewise, the presented GA method has been adapted for contiguous uniform subarrayed arrays. The GA method was selected for adaptation because it has been commonly utilized in previous non-subarrayed array synthesis literature such as [16], this is the first time to the authors' knowledge that it has been adapted for contiguous uniform subarrayed arrays. The proposed methods are applied to linear subarrayed arrays and are then expanded for use in planar subarrayed arrays. Simulation results for the beam broadening and shaping of the main beam for linear arrays containing eight and sixteen subarrays, each containing five elements, are given to demonstrate the validity and efficiency of the proposed method. Simulation results for planar arrays containing 8x8 and 16x16 subarrays, each containing 5x5 elements, are also presented.

The mathematical foundation for synthesis is presented in Section II and is followed by Section III which presents the three enhanced IFT methods and the GA method that are modified to account for subarrayed architectures. Section IV shows the simulated results of the proposed methods for linear and planar subarrayed arrays and compares the efficiency and effectiveness of each method. Finally, the paper is concluded in Section V.

II. Mathematical Foundation

It is well known that the far-field array pattern of a linear array of N elements each spaced a distance d apart can be written as the product of the element factor, EF, and the array factor AF

$$F(u) = EF(u)AF(u) \tag{1}$$

where $u = \sin\theta$, and θ is the angle relative to the array normal. For isotropic elements, $EF(u) = 1$; therefore, the array pattern, $F(u)$, depends solely on the array factor which can be represented as

$$AF(u) = \sum_{n=0}^{N-1} W_n e^{jkndu}, \tag{2}$$

where N is the total number of elements in the linear array, W_n is the complex excitation of the nth element, k is the wavenumber $(2\pi/\lambda)$, and λ is the wavelength. The complex excitation, W_n, for a normal phased array where the amplitude and phase of each element can be independently adjusted can be written as

$$W_n = A_n e^{j\phi_n}, \tag{3}$$

where A_n is the amplitude and ϕ_n is the phase of the element excitation.

Let us consider a subarrayed linear array where these N elements are separated into contiguous groups of N_s elements to form S subarrays. Each subarray has a complex excitation consisting of an amplitude and phase value that drives all elements in the subarray as visualized in Fig. 1. For subarrayed

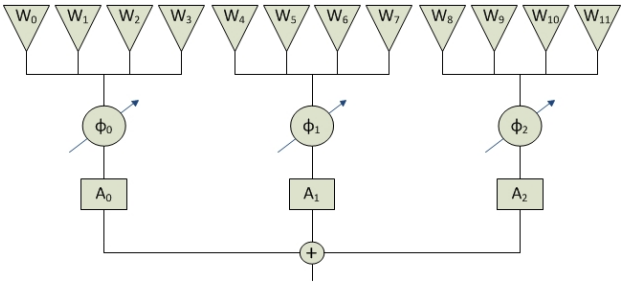


Fig. 1. Subarrayed architecture geometry. This figure shows an example of a contiguous uniform subarrayed linear array antenna with N = 12 antenna elements, N_s = 4 elements per subarray, and S = 3 subarrays.

architectures (2) can be rewritten as

$$AF(u) = \sum_{s=0}^{S-1} \sum_{n_s=0}^{N_s-1} W_{(N_s s+n_s)} e^{jkd(N_s s+n_s)u}, \tag{4}$$

where the complex excitation variable $W_{N_s s+n_s}$ is the complex excitation applied to each subarray. For instance, in Fig. 1 the first subarray consists of the first four elements of the array. Since this first subarray is given a complex excitation of $A_0 e^{j\phi_0}$, the excitation of each of the first four elements is identical, $W_0 = W_1 = W_2 = W_3 = A_0 e^{j\phi_0}$. The response of each subarray is given by

$$AF_s(u) = \sum_{n_s=0}^{N_s-1} W_{(N_s s+n_s)} e^{jkdn_s u}. \tag{5}$$

$$AF(u) = \sum_{s=0}^{S-1} AF_s(u) e^{jkdN_s su}. \tag{6}$$

As mentioned above, the excitation of each element in the subarray is identical; so the excitation variable $W_{N_s s}$ is a constant within each subarray and can be pulled outside of the summation as a variable only dependent on the subarray number as

$$AF_s(u) = W_{(N_s s)} \sum_{n_s=0}^{N_s-1} e^{jkdn_s u}. \tag{7}$$

Now if we recognize that the summation in (7) is the common pattern for the structure of each subarray in the uniform contiguous subarrayed array, we can reduce (7) to

$$AF_s(u) = W_{(N_s s)} B(u), \tag{8}$$

where $B(u)$ represents the common pattern for the structure of each subarray.

Substituting (8) into (4), the resulting array factor for the subarrayed architecture is

$$AF(u) = B(u) \sum_{s=0}^{S-1} W_{(N_s s)} e^{jkdN_s su}. \tag{9}$$

Notice that the summation in (9) is using subarray spaced (dN_s) elements to represent the substructure of the positioning of the subarrays in the full array and that $B(u)$ is a constant outside of the summation. Using this information, (9) can be reduced further to

$$AF_{\text{ss}}(u) = AF(u)/B(u) = \sum_{s=0}^{S-1} W_{(N_s s)} e^{jkdN_s su} \ . \tag{10}$$

Expanding the scope of the mathematical model for use in planar subarrayed arrays, (2) can be extended to apply to 2-D arrays with $N \times M$ elements arranged in a rectangular grid with spacing d_x and d_y as

$$AF(u,v) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} W_{n,m} e^{jk[nd_x u + md_y v]}, \tag{11}$$

where $W_{n,m}$ is the complex excitation of the element in the nth column and mth row, k is the wavenumber $(2\pi/\lambda)$, λ is the wavelength, d_x is the separation between the columns of elements, d_y is the separation between the rows of elements, $u = \sin\theta\cos\phi$, and $v = \sin\theta\sin\phi$.

Similarly, (4) can be extended to represent planer subarrayed architectures as

$$AF(u,v) = \sum_{s=0}^{S-1} \sum_{t=0}^{T-1} \sum_{n_s=0}^{N_s-1} \sum_{m_s=0}^{M_s-1} W_{x,y} e^{jk[xd_x u + yd_y v]}, \tag{12}$$

where $x = N_s s + n_s$ and $y = M_s t + m_s$.

Likewise, (10) can be extended to represent planar subarray substructures as

$$AF_{\text{ss}}(u,v) = AF(u,v)/B(u,v) = \sum_{s=0}^{S-1} \sum_{t=0}^{T-1} W_{g,h} e^{jk[gd_x u + hd_y v]} \tag{13}$$

where $g = N_s s$ and $h = M_s t$.

A. General IFT Method

Close inspection of (2) reveals that this equation can be represented as a finite Fourier series that relates the element-level excitations, W_n, of the array to the array factor of the array through a discrete inverse Fourier transform. Since AF is related to the element excitations through a discrete inverse Fourier transform, a discrete direct Fourier transform applied to AF will yield the element excitations. The general IFT method utilizes these Fourier transform relationships between AF and W_n to iteratively find W_n that will generate a desired AF, AF_{des}. The general process is initiated with all W_n set to unity amplitude and zero phase. Then the process iteratively generates the AF associated with the current W_n, compares it to the AF_{des}, adjusts the AF to be more like the AF_{des}, and then generates new W_n from the adjusted AF. This iterative process is repeated for a maximum number of iterations, $i_{\max}$.

For computational efficiency the IFT process makes use of the IFFT and FFT operations. When W_n is used in the IFFT operation to generate the AF, it is zero padded to become W_{zp} where the size of W_{zp} is the nearest power of two greater than $10 \times N$. After the FFT process is used on the AF, W_{zp} is truncated to find W_n. A detailed step-by-step process of the general IFT approach can be found in [15].

B. Cost Function

In this paper a cost function determines how close the generated array factor, AF_{gen}, is to the desired array factor, AF_{des}. The desired array factor is defined as a normalized flat-top function with a main beam and sidelobes. The main beam consists of a desired angular width with an amplitude of 0 dB. The sidelobes are the points in AF_{des} outside this main beam with a desired sidelobe level defined in negative dB. AF_{gen} is computed by taking the IFFT of the element excitations (2) and is normalized before comparing it to AF_{des}. The cost function, C, is defined as:

$$C = \log(E_{\text{mb}} + E_{\text{sl}}) \tag{14}$$

where E_{mb} is the error calculated for the points that are in the main beam and below 0 dB, θ_{mb}, and is defined as:

$$E_{\text{mb}} = \sum_{\theta_{\text{mb}}} (|AF_{\text{des}}(\theta_{\text{mb}})| - |AF_{\text{gen}}(\theta_{\text{mb}})|)^2 \tag{15}$$

and E_{sl} is the error calculated for the points that are outside the main beam and above the desired sidelobe level, θ_{sl}, and is defined as:

$$E_{\text{sl}} = \sum_{\theta_{\text{sl}}} (|AF_{\text{gen}}(\theta_{\text{sl}})| - |AF_{\text{des}}(\theta_{\text{sl}})|)^2 \tag{16}$$

.

The logarithm is used in (14) to reduce the values returned to a more condensed range so that the convergence rate of the cost function can be more easily visualized. For a more tangible performance measure, (14) can be converted to a percentage pattern effectiveness metric as

$$P_{\text{eff}} = 10^{-\sqrt{\exp(C)/(\beta \times 100)}} \times 100, \tag{17}$$

where β represents the number of angular points used to evaluate AF_{gen} vs. AF_{des}.

III. Proposed Methods

This paper presents new efficient beam broadening methods for contiguous uniform subarrayed arrays by proposing three speed enhanced IFT methods that modify the general IFT approach [15] to account for contiguous uniform subarrayed architectures and to broaden the main beam. This paper also presents a GA method that has been altered for use with contiguous uniform subarrayed architectures. The previously presented (2) assumes that the amplitude and phase of each individual array element can be independently adjusted. This assumption does not hold true for subarrayed array architectures where the amplitude and phase adjustments are applied to groups of array elements.

A. Integrated IFT (iIFT)

The first proposed modified IFT algorithm, herein referred to as the integrated IFT, or iIFT algorithm, enhances the general IFT method by using a "mean pooling" strategy to group the elements into subarrays and assign each element in the subarray the average phase value of the elements in the subarray. The average phase value is calculated using the complex number form of the element excitations to account for the phase being a circular function. These averaged values are used as the new element excitations and the next AF in the iterative process is generated using these element excitations. The iIFT process is shown in block diagram form in Figure 2. The detailed process for the iIFT method is shown in

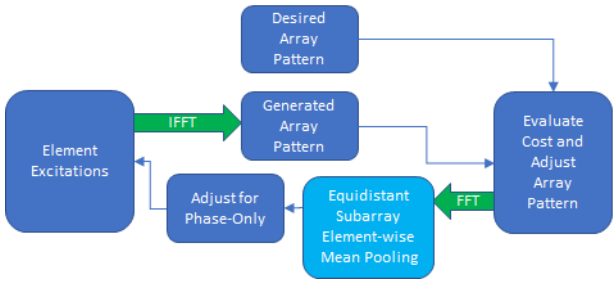


Fig. 2. Block diagram of the iIFT process. Note that the mean pooling phase is integrated into the iterative loop.

Algorithm 1.

Algorithm 1 Integrated Iterative Fourier Transform (iIFT)

$i \leftarrow 0$
$W(i) \leftarrow 1$ ▷ Initialize element excitations
$C \leftarrow 1 \times 10^{30}$ ▷ Initialize the cost value
Define AF_{des} ▷ Normalized desired pattern
while $i < i_{\max}$ **do**
  $W_{\text{zp}} \leftarrow W(i)+$ zero padding
  $AF_{\text{gen}}(i) \leftarrow \text{IFFT}\{W_{\text{zp}}\}$ ▷ Generate the pattern
  Compute C_{new} using (14)
  if $C_{\text{new}} < C$ **then**
    $C \leftarrow C_{\text{new}}$
    $i_{\text{best}} \leftarrow i$ ▷ Save index of best pattern
  end if
  $|AF_{\text{gen}}(i, \theta_{\text{mb}})| \leftarrow |AF_{\text{des}}(\theta_{\text{mb}})|$
  $|AF_{\text{gen}}(i, \theta_{\text{sl}})| \leftarrow |AF_{\text{des}}(\theta_{\text{sl}})|$
  $W_{\text{zp}} \leftarrow \text{FFT}\{AF_{\text{gen}}(i)\}$ ▷ Generate new excitations
  $W(i+1) \leftarrow W_{\text{zp}}(1:\text{N})$
  for $s = 0 : S$ **do** ▷ Adjust subarray excitations
    Calculate element ϕ_{mean} of equidistant subarrays.
    for $n_s = 0 : N_s$ **do** ▷ Set element excitations
      $|W_{(N_s s:N_s s+n_s)}(i+1)| \leftarrow 1$
      $\angle W_{(N_s s:N_s s+n_s)}(i+1) \leftarrow \phi_{\text{mean}}$
    end for
  end for
  $i \leftarrow i+1$
end while
$W_{\text{best}} \leftarrow W(i_{\text{best}})$
$AF_{\text{best}} \leftarrow AF_{\text{gen}}(i_{\text{best}})$

B. Decoupled IFT (dIFT)

The second proposed modified IFT algorithm, herein referred to as the decoupled IFT, or dIFT algorithm, creates an additional set of element excitations, V_n, to decouple the grouping of the elements into subarrays from the iterative process. After the element excitations, W_n, are found via the IFT process for the next iteration, they are grouped into subarrays to form a new set of element excitations, V_n. V_n is then used to generate a subarray influenced array factor, AF_{genv}, which is compared to AF_{des} using (14). The dIFT process is shown in block diagram form in Figure 3. The

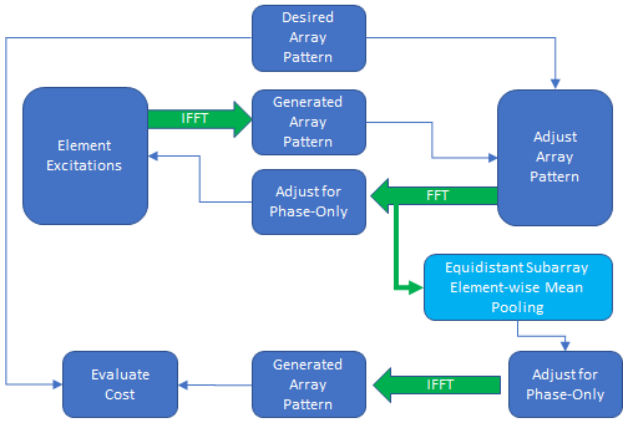


Fig. 3. Block diagram of the dIFT process. Note that the mean pooling phase is decoupled from the iterative loop.

detailed process for the dIFT method is shown in Algorithm 2.

C. Subarray Spaced IFT (ssIFT)

The third proposed modified IFT algorithm, herein referred to as the subarray spaced IFT, or ssIFT algorithm, enhances the general IFT method by reducing the problem to focus on the subarray substructure pattern instead of focusing on the full array pattern. With the ssIFT algorithm there is some additional complexity added with the transition from the full pattern domain to the subarray substructure domain. From (10), given a desired overall array pattern, the desired overall array pattern can be divided by the subarray pattern which results in the pattern of the phase weights at the center of each subarray referred to herein as the desired subarray structure pattern ($AF_{\text{des}_{(\text{ss})_{(\text{elem})}}}$). This subarray substructure pattern then needs to be converted from element-wise FFT bins to subarray-wise FFT bins. A new number of FFT bins (M) is calculated as the nearest power of two greater than $10 \times S$. The angular extent of the pattern (θ_{ss}) also needs to be limited to $\arcsin\left[-M/2, M/2-1\right](\lambda/(Md_{\text{sub}}))$ due to the subarray spacing. Then interpolation can be used to arrive at the desired subarray spaced subarray substructure pattern ($AF_{\text{des}_{(\text{ss})_{(\text{sub})}}}$). While there is some added single-use complexity added with this method, the complexity inside the iterative loop can be reduced using this algorithm by reducing the size of the FFT and using a simplified "mean pooling" strategy to ensure that subarrays that are equidistant from

Algorithm 2 Decoupled Iterative Fourier Transform (dIFT)

$i \leftarrow 0$
$W(i) \leftarrow 1$ ▷ Initialize element excitations
$C \leftarrow 1 \times 10^{30}$ ▷ Initialize the cost value
Define AF_{des} ▷ Normalized desired pattern
while $i < i_{\max}$ **do**
  $W_{\text{zp}} \leftarrow W(i)+$ zero padding
  $AF_{\text{gen}}(i) \leftarrow \text{IFFT}\{W_{\text{zp}}\}$ ▷ Generate the pattern
  $|AF_{\text{gen}}(i, \theta_{\text{mb}})| \leftarrow |AF_{\text{des}}(\theta_{\text{mb}})|$
  $|AF_{\text{gen}}(i, \theta_{\text{sl}})| \leftarrow |AF_{\text{des}}(\theta_{\text{sl}})|$
  $W_{\text{zp}} \leftarrow \text{FFT}\{AF_{\text{gen}}(i)\}$
  $W(i+1) \leftarrow W_{\text{zp}}(1:\text{N})$ ▷ Generate new excitations
  for $s = 0 : S$ **do** ▷ Adjust subarray excitations
    Calculate element ϕ_{mean} of equidistant subarrays.
    for $n_s = 0 : N_s$ **do** ▷ Set element excitations
      $|V_{(N_s s:N_s s+n_s)}(i+1)| \leftarrow 1$
      $\angle V_{(N_s s:N_s s+n_s)}(i+1) \leftarrow \phi_{\text{mean}}$
    end for
  end for
  $V_{\text{zp}} \leftarrow V(i)+$ zero padding
  $AF_{\text{genv}}(i) \leftarrow \text{IFFT}\{V_{\text{zp}}\}$
  Compute C_{new} using (14)
  if $C_{\text{new}} < C$ **then**
    $C \leftarrow C_{\text{new}}$
    $i_{\text{best}} \leftarrow i$ ▷ Save index of best pattern
  end if
  $i \leftarrow i+1$
end while
$V_{\text{best}} \leftarrow V(i_{\text{best}})$
$AF_{\text{best}} \leftarrow AF_{\text{genv}}(i_{\text{best}})$

the center of the array receive the same excitation to force symmetry in the array pattern. These averaged values are used as the new excitations and the next generated subarray substructure pattern, $AF_{\text{gen}_{(\text{ss})}}$, in the iterative process is generated using these subarray excitations. The ssIFT process is shown in block diagram form in Figure 4. The detailed process for the ssIFT method is shown in Algorithm 3.

D. Genetic Algorithm (GA)

A genetic algorithm (GA) is inspired by the process of natural selection and relies on bio-inspired operators such as mutation, crossover and selection to evolve an initial population of candidate solutions toward better solutions. Each candidate solution has a set of properties (its chromosomes or genotype) which can be mutated and altered. The evolution usually starts from a population of randomly generated individuals, and is an iterative process, with the population in each iteration called a generation. In each generation, the fitness of every individual in the population is evaluated; the fitness is usually the value of the objective function in the optimization problem being solved. The more fit individuals are stochastically selected from the current population, and each individual's genome is modified (recombined and possibly randomly mutated) to form a new generation. The new generation of candidate solutions is then used in the next iteration of the algorithm. The algorithm

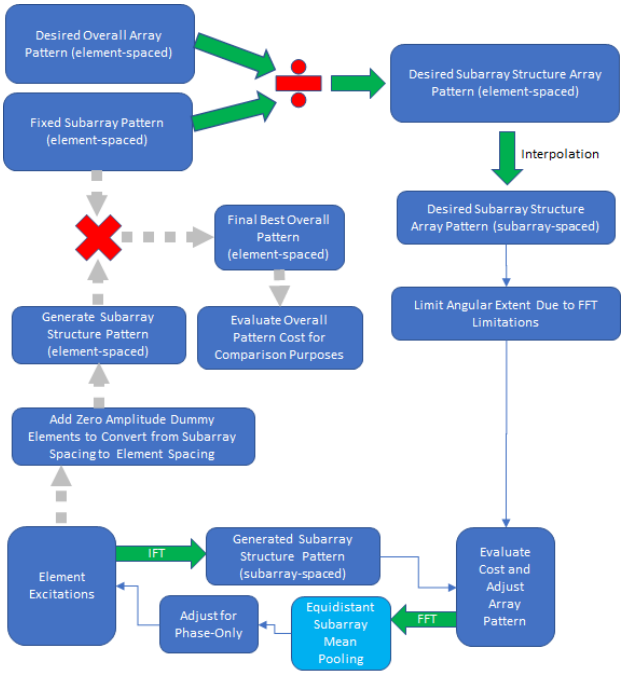


Fig. 4. Block diagram of the ssIFT process. Note that the mean pooling phase is simplified from the iIFT and dIFT approaches since it is not averaging the grouped element phases but only averaging at the subarray level for those subarrays that are equidistant from the center of the array.

terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population.

For our beam broadening array pattern optimization problem the population includes candidate sets of element excitations. Therefore, in genetic algorithm terminology, a set of element excitations for each element in the array is the chromosome and each individual element citation is the gene.

The GA process used in this paper is derived from the implementation described in [17] but minimizes a cost function instead of maximizing a fitness function. In addition, elite members of the population are introduced due to analysis in [18]. The variables description of the GA process are defined as:

- U: A candidate set of element excitations.
- P: The population of candidate sets of element excitations.
- P_{new}: The new population of candidate sets of element excitations derived from the previous population.
- C: A function that assigns an evaluation score to a given candidate set of element excitations.
- G: The maximum number of generations.
- Z: The number of candidate sets of element excitations to be included in the population.
- r: The fraction of the population to be replaced via the crossover technique at each step.
- m: The mutation rate.

The implemented GA process can be described as:

1) Randomly generate a population, P, of size Z candidate sets of subarray excitations, U.

Algorithm 3 Subarray Spaced Iterative Fourier Transform (ssIFT)

Define $AF_{\mathrm{des}_{(\mathrm{full})}}$ ▷ Full desired pattern (element-spaced)
$W(1:N_{\mathrm{s}}) \leftarrow 1$ ▷ Initialize subarray element excitations
$W_{\mathrm{zp}} \leftarrow W(1:N_{\mathrm{s}})+$ zero padding
$AF_{\mathrm{subarray}} \leftarrow \mathrm{IFFT}\{W_{\mathrm{zp}}\}$ ▷ Generate subarray pattern
$AF_{\mathrm{des}_{(\mathrm{ss})_{(\mathrm{elem})}}} \leftarrow AF_{\mathrm{des}_{(\mathrm{full})}}/AF_{\mathrm{subarray}}$
$\theta_{\mathrm{ss}} \leftarrow \arcsin\left[-M/2, M/2-1\right](\lambda/(Md_{\mathrm{sub}}))$
$AF_{\mathrm{des}_{(\mathrm{ss})_{(\mathrm{sub})}}} \leftarrow \mathrm{interp}(AF_{\mathrm{des}_{(\mathrm{ss})_{(\mathrm{elem})}}}, \theta_{\mathrm{ss}})$
$i \leftarrow 0$ ▷ Set iteration count variable
$W_{\mathrm{sub}}(i) \leftarrow 1$ ▷ Initialize element excitations
$C \leftarrow 1 \times 10^{30}$ ▷ Initialize the cost value
while $i < i_{\max}$ **do**
  $W_{\mathrm{zp}} \leftarrow W_{\mathrm{sub}}(i)+$ zero padding
  $AF_{\mathrm{gen}_{(\mathrm{ss})_{(\mathrm{sub})}}}(i) \leftarrow \mathrm{IFFT}\{W_{\mathrm{zp}}\}$ ▷ Generate pattern
  Compute C_{new} using (14)
  if $C_{\mathrm{new}} < C$ **then**
    $C \leftarrow C_{\mathrm{new}}$ ▷ Update cost value
    $i_{\mathrm{best}} \leftarrow i$ ▷ Save index of best pattern
  end if
  $|AF_{\mathrm{gen}_{(\mathrm{ss})_{(\mathrm{sub})}}}(i, \theta_{\mathrm{mb}})| \leftarrow |AF_{\mathrm{des}_{(\mathrm{ss})_{(\mathrm{sub})}}}(\theta_{\mathrm{mb}})|$
  $|AF_{\mathrm{gen}_{(\mathrm{ss})_{(\mathrm{sub})}}}(i, \theta_{\mathrm{sl}})| \leftarrow |AF_{\mathrm{des}_{(\mathrm{ss})_{(\mathrm{sub})}}}(\theta_{\mathrm{sl}})|$
  $W_{\mathrm{zp}} \leftarrow \mathrm{FFT}\{AF_{\mathrm{gen}_{(\mathrm{ss})_{(\mathrm{sub})}}}(i)\}$
  $W_{\mathrm{sub}}(i+1) \leftarrow W_{\mathrm{zp}}(1:\mathrm{S})$
  for $s = 0 : S$ **do** ▷ Adjust subarray excitations
    Calculate element ϕ_{mean} of equidistant subarrays.
    $|W_{\mathrm{sub}_{(s)}}(i+1)| \leftarrow 1$
    $\angle W_{\mathrm{sub}_{(s)}}(i+1) \leftarrow \phi_{\mathrm{mean}}$
  end for
  $i \leftarrow i+1$
end while
$W_{\mathrm{sub}_{\mathrm{best}}} \leftarrow W_{\mathrm{sub}}(i_{\mathrm{best}})$
$AF_{\mathrm{gen}_{(\mathrm{ss})_{(\mathrm{sub})_{\mathrm{best}}}}} \leftarrow AF_{\mathrm{gen}_{(\mathrm{ss})_{(\mathrm{sub})}}}(i_{\mathrm{best}})$
Zero pad subarray excitations to elem spacing ($W_{\mathrm{elem}_{\mathrm{best}}}$)
$AF_{\mathrm{gen}_{(\mathrm{ss})_{(\mathrm{elem})}}} \leftarrow \mathrm{IFFT}\{W_{\mathrm{elem}_{\mathrm{best}}}\}$
$AF_{\mathrm{gen}_{(\mathrm{full})}} \leftarrow AF_{\mathrm{gen}_{(\mathrm{ss})_{(\mathrm{elem})}}} \times AF_{\mathrm{subarray}}$

2) For each U, synthesize the array pattern and evaluate the generated pattern against the desired pattern via a cost function, C.
3) For G generations of solutions
   a) Create a set of parents by selecting rZ members of P in a probabilistic manner. The probability $Pr(U_z)$ of selecting candidate set U_z from P is given by

$$Pr(U_z) = \frac{C_{\max} - C(U_z)}{\sum_{j=1}^{Z}(C_{\max} - C(U_j))}, \qquad (18)$$

   where $C_{\max} = \max(C(U_1), C(U_2), \ldots C(U_Z))$. If all the values of $C(U_z)$ are the same, $Pr(U_z) = \frac{1}{Z}$.
   b) For each pair of candidate sets from P, $\langle U_1, U_2 \rangle$, produce two offspring by applying the crossover operator. Add all offspring to P_{new}.
   c) For each of the members of P_{new}, perform the following mutation operator from [18]: for each gene, if $\mathrm{rand}() < m$, mutate it.
   d) Add the $(1-r)Z$ members of P with the lowest cost into P_{new} as elites.
   e) Update the population: $P \leftarrow P_{\mathrm{new}}$.
   f) For each candidate set of subarray excitations, synthesize the array pattern and evaluate the generated pattern against the desired pattern via a cost function, C.
4) Return the candidate set from P that has the lowest cost function value.

IV. Results

The results presented in this paper were generated using a computer simulation that synthesized an approximation of the array pattern using (1) where each element was assumed to be an isotropic radiator thus making EF equal to 1. The AF portion of the array pattern approximation was implemented with an FFT-based approach that utilized (2) as its foundation. This simulation contained adjustable parameters that allowed arbitrary arrays to be modeled based on the number of elements in each subarray, the number of subarrays in the array, the spacing between elements, and the number of bits for each phase shifter. The desired antenna pattern was described using parameters for width of the broadened main beam and maximum sidelobe level. The simulation also constrained the generated element/subarray excitations to be symmetric based on their distance from the center of the array to ensure that the synthesized array pattern was symmetric about the center of the array. The iIFT, dIFT, ssIFT, and GA methods are iteratively executed until the maximum number of iterations is achieved.

The simulation was configured to model X-band (10 GHz transmit frequency) linear arrays of 40 and 80 elements and rectangular arrays of 40x40 and 80x80 elements. The linear arrays were grouped into contiguous subarrays of five elements while the rectangular arrays were grouped into subarrays of 5x5 elements. The simulation was configured to synthesize an array pattern that closely approximates a flat-top pattern with a beamwidth of 12 degrees and maximum sidelobes of -13 dB. To better understand the impact of subarrayed architectures, array patterns were synthesized for arrays with and without subarrays. Since it is the relative phase of the elements and subarrays to each other that is important, the innermost subarrays were fixed to zero phase to reduce the solution space and thus decrease the convergence time.

Since the GA method is a not a deterministic process, the GA method was performed 100 times with the same settings to obtain statistical significance. The constants used in the GA algorithm are given in Table I. Using a quarter of the population as elites in the parent selection was determined to improve convergence. Due to the use of elitism preserving genes in the GA, r and m were both set higher to increase the rate of testing new solutions. The selection of population size was based on the rule-of-thumb that population sizes between 20 - 50 are usually sufficient. For the smaller 40x40 array with a smaller solution space, a population of size 25 was used. However, in the case of the 80x80 radar, due to the

increased solution space size, a larger population size of 50 was used. The generation size was set to ensure that 20000 arrary patterns were generated.

TABLE I
CONSTANTS USED IN GA APPROACH

	r	m	N	G
1x40	1	0.07	25	800
1x80	1	0.07	50	400
40x40	1	0.07	25	800
80x80	1	0.07	50	400

A. A Note on Complexity

The iIFT, dIFT, ssIFT, and GA approaches are compared based on their computational efficiency and their effectiveness. The efficiency of each approach is compared based upon the number of array patterns and/or sets of element excitations that need to be generated per iteration times the number of iterations that were performed. Each iteration of the proposed iIFT and ssIFT methods calculate one array pattern and one set of array element excitations. Thus, the number of generated array patterns and sets of element excitations for iIFT is twice the number of iterations of the algorithm. The dIFT method creates an additional set of element excitations and generates an array pattern with this additional set for each iteration. The number of generated array patterns and sets of element excitations for dIFT is three times the number of iterations of the algorithm. The GA method calculates an array pattern for each new candidate set of the population at each generation.

To summarize, the computational efficiency is measured by counting the number of array patterns and sets of element excitations that were generated to achieve a desired minimum cost value. The effectiveness of each approach is measured by its ability to achieve the lowest cost value.

After the IFT and GA methods calculate their best solutions, the calculated phase values were constrained to values that a 6-bit phase shifter can utilize. Constraining the phase values during the IFT process was found to overly constrain the solution space and hinder the IFT process. In other words, there is no advantage to enforcing the resolution constraint before running the algorithms.

B. 1x40 Results

The convergence curves for the synthesis of a 40 element linear array (subarrayed into 8 subarrays of 5 element each) are shown in Figure 5. The iIFT approach requires 50 generated array patterns/excitations to find its best solution (C = 6.38) and 90 generated array patterns/excitations to find its steady-state solution (C = 6.41). The dIFT approach requires 78 generated array patterns/excitations to find its best solution (C = 6.47) and 114 generated array patterns/excitations to find its steady-state solution (C = 6.54). The ssIFT approach requires 30 generated array patterns/excitations to find its best solution (C = 6.36) and 70 generated array patterns/excitations to find its steady-state solution (C = 6.39). In comparison the GA approach on average needs to generate greater than 20000 array patterns to reach the same best cost function value as the ssIFT approach (3000 array patterns to reach the ssIFT steady-state value), 14000 array patterns to reach the same best cost function value as the iIFT approach (1500 array patterns to reach the iIFT steady-state value), and 600 array patterns to reach the same best cost function value as the dIFT approach (400 array patterns to reach the dIFT steady-state value). The best (lowest cost, most effective) synthesized amplitude patterns are shown in Figure 6. The phase excitations for each element in the 40 element array that are associated with the best generated patterns are presented in Figure 7.

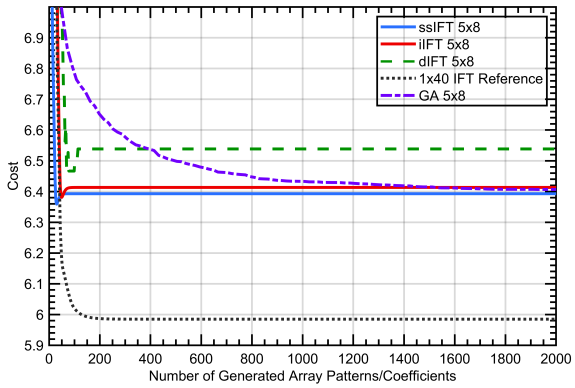


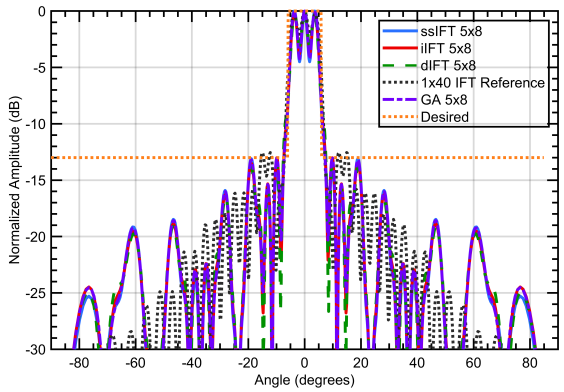
Fig. 5. The cost convergence curves for the synthesized 40 element linear array. Notice that the iIFT, dIFT, and ssIFT methods achieve their best solutions before their iterative process finds the steady state. The convergence curve for the non-subarrayed array is given as a reference. The full synthesis was executed for 20000 generated array patterns/coefficients.

C. 1x80 Results

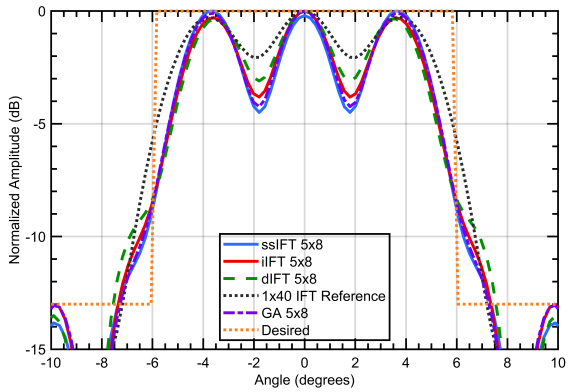
The convergence curves for the synthesis of an 80 element linear array (subarrayed into 16 subarrays of 5 element each) are shown in Figure 8. The iIFT approach requires 120 generated array patterns/excitations to find its best and also steady state solution (C = 7.21). The dIFT approach requires 315 generated array patterns/excitations to find its best solution (C = 7.28) and 381 generated array patterns/excitations to find its steady state solution (C = 7.36). The ssIFT approach requires 100 generated array patterns/excitations to find its best and steady state solution (C = 7.19). In comparison the GA approach on average needs to generate 1240 array patterns to reach the same best cost function value as the ssIFT approach, 1195 array patterns to reach the same best cost function value as the iIFT approach, and 1160 array patterns to reach the same best cost function value as the dIFT approach. The best synthesized amplitude patterns are shown in Figure 9. The phase excitations for each element in the 80 element array that are associated with the best generated patterns are presented in Figure 10.

D. 40x40 Results

The convergence curves for the synthesis of a 40x40 element rectangular array are shown in Figure 11. The iIFT method requires 50 generated array patterns/excitations to provide its



(a) Full Visible Region



(b) Zoomed

Fig. 6. The full (a) and zoomed (b) amplitude patterns for the synthesized 40 element linear array. Notice that the patterns are very similar in the main beam region, and the effectiveness is comparable among the iIFT, dIFT, and ssIFT, and GA methods. Also notice that the sidelobes in the visible region are below the desired level.

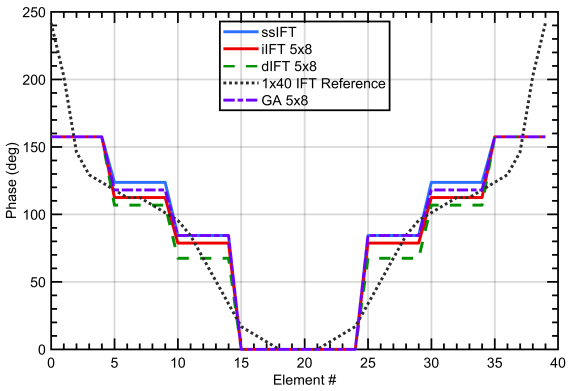


Fig. 7. The element phase excitations for the synthesized 40 element linear array that generate the most effective AF for each synthesis method. Note: Each method achieves its most effective AF at a different iteration number. Therefore, the displayed subarray phases of the dIFT method are not, and should not be, the average of the 1x40 phase excitations.

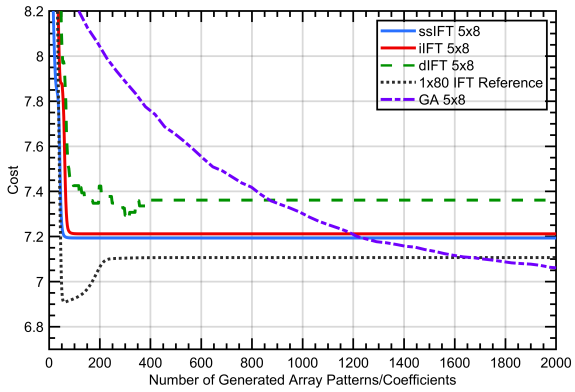
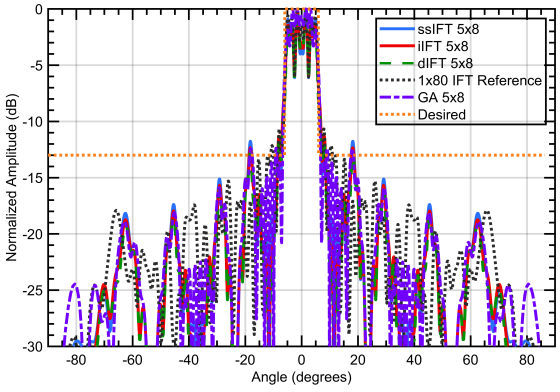
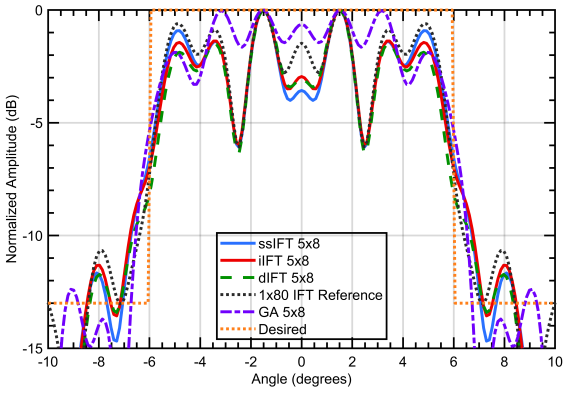


Fig. 8. The cost convergence curves for the synthesized 80 element linear array. Notice that the dIFT method achieves its best solution before its iterative process finds the steady state. The convergence curve for the non-subarrayed array is given as a reference. The full synthesis was executed for 20000 generated array patterns/coefficients.



(a)



(b)

Fig. 9. The full (a) and zoomed (b) amplitude patterns for the synthesized 80 element linear array. Notice that patterns are similar for the iIFT, dIFT, and ssIFT methods, but the GA pattern has noticeably lower main beam ripple. Also notice that the sidelobes in the visible region are below the desired level.

best and steady state solution (C = 11.52). The dIFT method requires 60 generated array patterns/excitations to provide its best and steady state solution (C = 11.59). The ssIFT method requires 70 generated array patterns/excitations to provide its best and steady state solution (C = 11.39). In comparison the GA approach on average needs to generate 1039 array patterns to reach the same best cost function value as the ssIFT approach, 440 array patterns to reach the same best cost function value as the iIFT approach, and 270 array patterns to reach the same best cost function value as the dIFT approach.

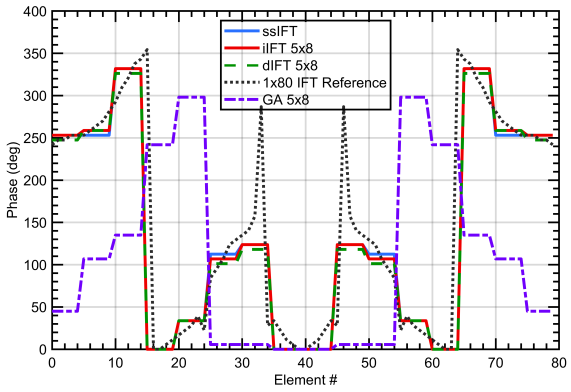


Fig. 10. The element phase excitations for the synthesized 80 element linear array that generate the most effective AF for each synthesis method.

The best synthesized amplitude patterns are shown in Figure 12 and Figure 13. The phase excitations for each element in the 40x40 element array that are associated with the best generated patterns are presented in Figure 14.

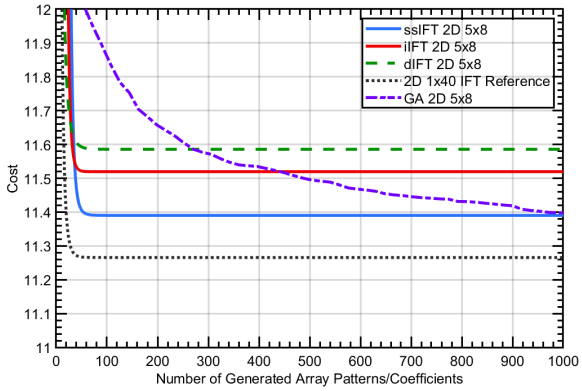
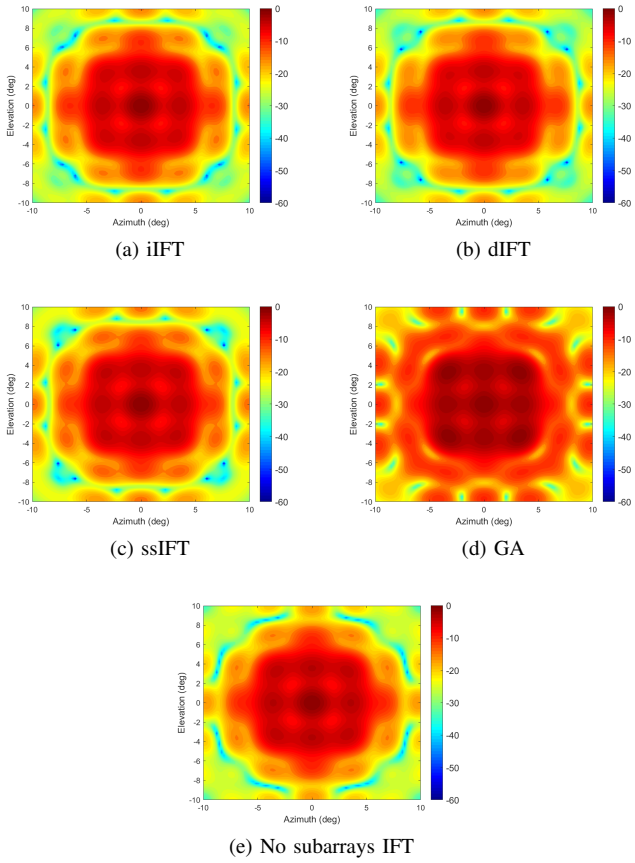


Fig. 11. The cost convergence curves for the synthesized 40x40 array. The convergence curve for the non-subarrayed array (40x40 IFT Reference) is given as a reference. Notice that the iIFT, dIFT, and ssIFT methods converge significantly faster than the GA method, but the GA method can eventually achieve a lower cost value. The full synthesis was executed for 20000 generated array patterns/coefficients.

E. 80x80 Results

The convergence curves for the synthesis of a 80x80 element rectangular array are shown in Figure 15. The iIFT method requires 195 generated array patterns/excitations to provide its best and steady state solution (C = 12.85). The dIFT method requires 480 generated array patterns/excitations to provide its best and steady state solution (C = 12.94). The ssIFT method requires 360 generated array patterns/excitations to provide its best solution (C = 11.64) and 700 generated array patterns/excitations for its steady state solution (C = 12.94). In comparison the GA approach needs to generate 5000 array patterns to reach the same best cost function value as the dIFT approach, 6600 array patterns to reach the same best cost function value as the iIFT approach, and greater than 20000 array patterns to reach the same best cost function value as the ssIFT approach. The best synthesized amplitude patterns are shown in Figure 16 and Figure 17. The phase excitations for each element in the 80x80 element array that are associated with the best generated patterns are presented in Figure 18.



(a) iIFT (b) dIFT

(c) ssIFT (d) GA

(e) No subarrays IFT

Fig. 12. The amplitude patterns for the synthesized 40x40 array. Notice that the iIFT, dIFT, ssIFT, and no subarrays IFT methods produce very similar antenna patterns. Also notice that the GA method produces an antenna pattern with more power and a flatter response in the main beam than the IFT methods.

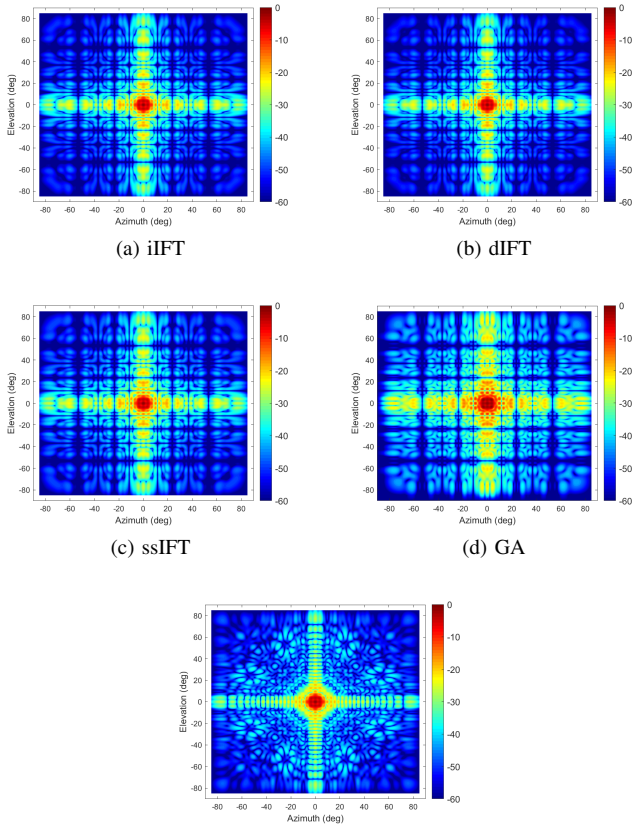
F. Summary Results

Table II summarizes the efficiency comparisons of the three proposed IFT methods with the GA method and identifies the improvement factor (F_{imp}) of each IFT method versus the GA method. Table III compares the best effectiveness of each method for 20,000 generated array patterns/excitations.

TABLE II
NUMBER OF GENERATED ARRAY PATTERNS TO EQUIVALENT COST SOLUTION (IFT METHODS VS. GA)

	$iIFT$			$dIFT$			$ssIFT$		
	$iIFT$	GA	F_{imp}	$dIFT$	GA	F_{imp}	$ssIFT$	GA	F_{imp}
1x40	50	140000	280	78	600	7.7	30	>20000	>667
1x80	120	1195	10	315	1160	3.7	100	1240	12.4
40x40	50	440	8.8	60	270	4.5	70	1039	14.8
80x80	195	6600	33.8	480	5000	10.4	360	>20000	55.6

The provided efficiency results summarized in Table II show that the three proposed modified IFT methods are between four

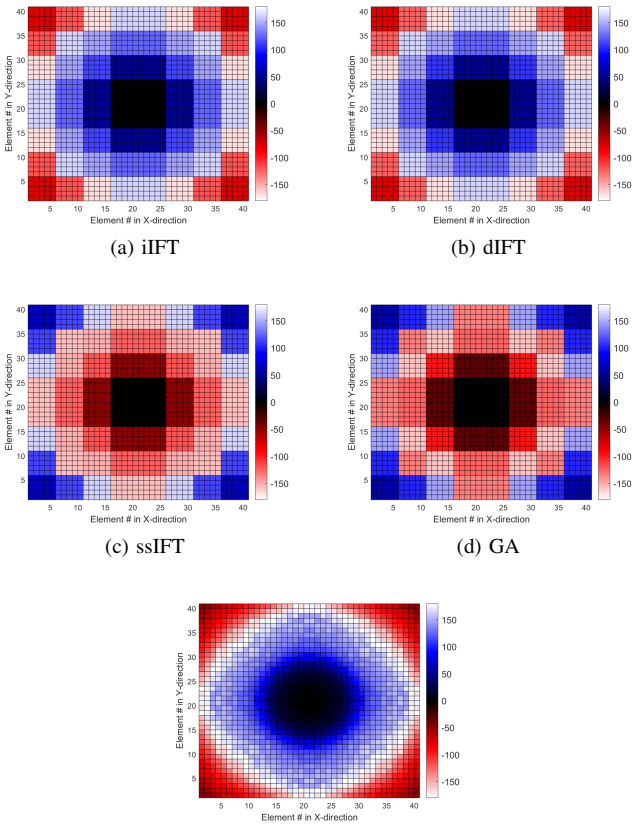


(a) iIFT (b) dIFT

(c) ssIFT (d) GA

(e) No subarrays IFT

Fig. 13. The amplitude patterns including the full visible region for the synthesized 40x40 array. Notice that the extended sidelobe patterns are below the desired sidelobe level.



(a) iIFT (b) dIFT

(c) ssIFT (d) GA

(e) No subarrays IFT

Fig. 14. The element phase excitations for the synthesized 40x40 array.

TABLE III
EFFECTIVENESS COMPARISON

	C				P_{eff}			
	$iIFT$	$dIFT$	$ssIFT$	GA	$iIFT$	$dIFT$	$ssIFT$	GA
1x40	6.38	6.47	6.36	6.38	78.10	77.22	78.29	78.10
1x80	7.21	7.28	7.19	6.9	76.75	76.03	76.95	79.72
40x40	11.52	11.59	11.39	10.92	86.70	86.26	87.48	89.97
80x80	12.85	12.94	11.64	12.52	87.04	86.49	92.70	88.90

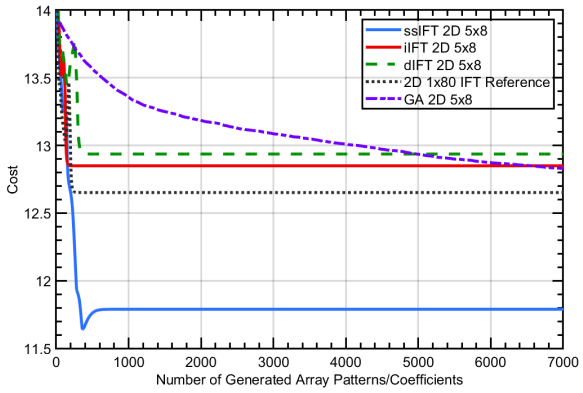
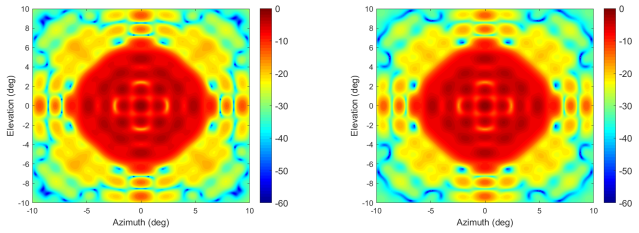


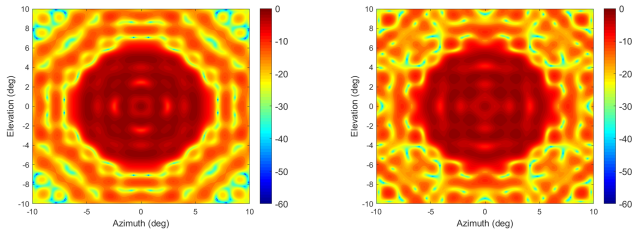
Fig. 15. The cost convergence curves for the synthesized 80x80 array. The convergence curve for the non-subarrayed array (80x80 IFT Reference) is given as a reference. Notice that the iIFT, dIFT , and ssIFT methods converge significantly faster than the GA method. The full synthesis was executed for 20000 generated array patterns/coefficients.

and greater than 667 times more efficient in synthesizing array patterns than GA which is a more common approach in the published literature. The GA approach can be more effective in generating a lower cost pattern if given enough computational resources and time. However, Table III shows that even after 20000 iterations the effectiveness of the GA method is only a few percentage points higher than the iIFT, dIFT, and ssIFT methods which take substantially less computational time. The pattern efficiency value, P_{eff}, in Table III describes how close the generated pattern is to the desired pattern.

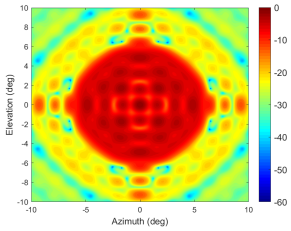
A comparison of the proposed methods reveals that the ssIFT approach does limit the effectiveness of the synthesis by inhibiting the iterative process from jumping out of local optima to find a more effective global optima, but this approach is the most efficient approach overall. Among the IFT methods, the performance of the ssIFT is followed by the iIFT and then the dIFT. Although the ssIFT adds some initial complexity in its transition between the element spaced domain and the subarray spaced domain, it is able to utilize smaller FFT sizes and a simpler mean pooling technique than the other two presented IFT methods inside the iterative loop. This makes the ssIFT more computationally efficient than the other presented IFT methods.



(a) iIFT (b) dIFT

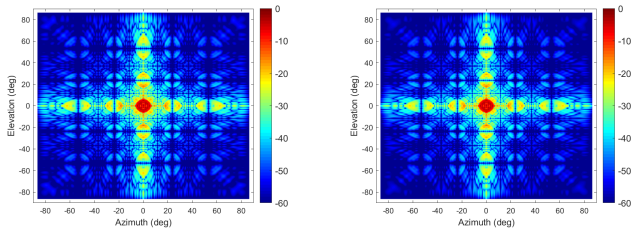


(c) ssIFT (d) GA

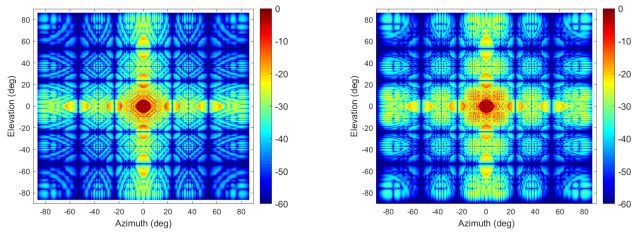


(e) No subarrays IFT

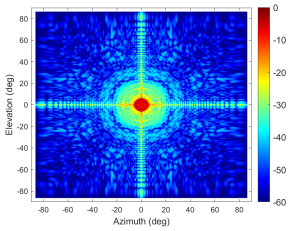
Fig. 16. The amplitude patterns for the synthesized 80x80 array. Notice that the iIFT, dIFT, and no subarrays IFT methods produce very similar antenna patterns. The extended sidelobe patterns, not shown in this figure, are all below the desired sidelobe level.



(a) iIFT (b) dIFT



(c) ssIFT (d) GA



(e) No subarrays IFT

Fig. 17. The amplitude patterns including the full visible region for the synthesized 80x80 array. Notice that the extended sidelobe patterns are below the desired sidelobe level.

V. Conclusions

This paper has presented four methods (three modified IFT methods and one GA method) that have been found to be efficient and effective in generating array patterns for contiguous uniform subarrayed arrays. The modified IFT methods are significantly (up to 667 times) more efficient than the average performance of the GA method and are only slightly outperformed (a few percentage points in power efficiency in the main beam) in effectiveness by the GA.

This paper has extended the impact of the IFT and GA methods and has shown that they can be efficiently and effectively utilized for synthesis of subarrayed architectures. Future work in this area includes an extension of this work that utilizes the iterative projection method (IPM) to give more freedom to the transition regions between the main beam and the sidelobe region.

References

[1] J. C. Kerce, G. C. Brown, and M. A. Mitchell, "Phase-only transmit beam broadening for improved radar search performance," in *2007 IEEE Radar Conference*, 2007, Conference Proceedings, pp. 451–456.

[2] A. F. Morabito, A. Massa, P. Rocca, and T. Isernia, "An effective approach to the synthesis of phase-only reconfigurable linear arrays," *Ieee Transactions on Antennas and Propagation*, vol. 60, no. 8, pp. 3622–3631, 2012. [Online]. Available: ⟨GotoISI⟩://WOS:000308435800009http://ieeexplore.ieee.org/document/6204059/

[3] A. Chatterjee and G. K. Mahanti, "Combination of fast fourier transform and self-adaptive differential evolution algorithm for synthesis of phase-only reconfigurable rectangular array antenna," *Annals of Telecommunications*, vol. 69, no. 9-10, pp. 515–527, 2014. [Online]. Available: ⟨GotoISI⟩://WOS:000342412500005https://link.springer.com/content/pdf/10.1007%2Fs12243-013-0396-1.pdf

[4] T.-S. Lee and T.-K. Tseng, "Subarray-synthesized low-side-lobe sum and difference patterns with partial common weights," *IEEE transactions on antennas and propagation*, vol. 41, no. 6, pp. 791–800, 1993.

[5] P. Rocca, L. Manica, R. Azaro, and A. Massa, "A hybrid approach to the synthesis of subarrayed monopulse linear arrays," *IEEE Transactions on Antennas and Propagation*, vol. 57, no. 1, pp. 280–283, 2009.

[6] R. L. Haupt, "Optimized weighting of uniform subarrays of unequal sizes," *IEEE Transactions on Antennas and Propagation*, vol. 55, no. 4, pp. 1207–1210, 2007.

[7] P. Rocca, R. L. Haupt, and A. Massa, "Sidelobe reduction through element phase control in uniform subarrayed array antennas," *IEEE Antennas and Wireless Propagation Letters*, vol. 8, pp. 437–440, 2009.

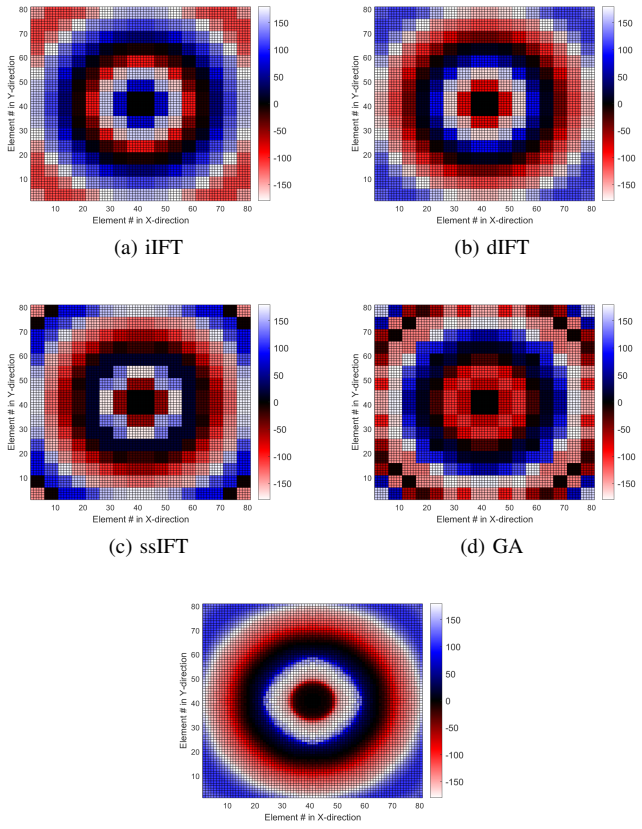
[8] L. Manica, P. Rocca, G. Oliveri, and A. Massa, "Synthesis of multi-beam sub-arrayed antennas through an excitation matching strategy," *IEEE Transactions on Antennas and Propagation*, vol. 59, no. 2, pp. 482–492, 2011.

[9] S. Rajagopal, "Beam broadening for phased antenna arrays using multi-beam subarrays," in *2012 IEEE International Conference on Communications (ICC)*, 2012, Conference Proceedings, pp. 3637–3642.

[10] C. Balanis, *Antenna Theory: Analysis and Design*. Wiley, 2016. [Online]. Available: https://books.google.com/books?id=iFEBCgAAQBAJ

[11] O. Bucci, G. Franceschetti, G. Mazzarella, and G. Panariello, "Intersection approach to array pattern synthesis," in *IEE Proceedings H (Microwaves, Antennas and Propagation)*, vol. 137. IET, 1990, Conference Proceedings, pp. 349–357.

[12] G. Franceschetti, G. Mazzarella, and G. Panariello, "Array synthesis with excitation constraints," in *IEE Proceedings H (Microwaves, Antennas*



(a) iIFT (b) dIFT

(c) ssIFT (d) GA

(e) No subarrays IFT

Fig. 18. The element phase excitations for the synthesized 80x80 array.

and Propagation), vol. 135. IET, 1988, Conference Proceedings, pp. 400–407.

[13] C. Carroll and B. V. Kumar, "Iterative fourier transform phased array radar pattern synthesis," in *Proc. of SPIE*, vol. 827, 1987, Conference Proceedings, pp. 73–84.

[14] W. P. Keizer, "Fast low-sidelobe synthesis for large planar array antennas utilizing successive fast fourier transforms of the array factor," *IEEE Transactions on Antennas and Propagation*, vol. 55, no. 3, pp. 715–722, 2007.

[15] ——, "Low-sidelobe pattern synthesis using iterative fourier techniques coded in matlab [em programmer's notebook]," *IEEE Antennas and Propagation Magazine*, vol. 51, no. 2, 2009.

[16] Y.-Q. Wen, B.-Z. Wang, and X. Ding, "A wide-angle scanning and low sidelobe level microstrip phased array based on genetic algorithm optimization," *IEEE Transactions on Antennas and Propagation*, vol. 64, no. 2, pp. 805–810, 2016.

[17] T. M. Mitchell, *Machine learning*. McGraw-Hill Boston, MA:, 1997.

[18] G. Rudolph, "Convergence analysis of canonical genetic algorithms," *IEEE transactions on neural networks*, vol. 5, no. 1, pp. 96–101, 1994.

Barry K. Daniel (S'95, M'16, SM'19) received both a B.S. degree in Electrical Engineering and a B.S. degree in Computer Science in 1995 from Tennessee Technological University. He received a M.S. degree in Electrical Engineering in 2000 from the University of Alabama, Huntsville. He is currently pursing a Ph.D. degree in Electrical Engineering from Tennessee Technological University.

At the beginning of his career, he worked in Huntsville, Alabama for 10 years developing simulations of radar and optical portions of missile threat systems and performing signal processing analysis. Since 2005, he has been working for Oak Ridge National Laboratory in the areas of antenna array synthesis, efficient digital processing algorithms, track/data fusion, machine learning, real-time simulation test beds to analyze new waveforms, evaluation of prototype radar designs, and environment models that propagate RF and EO/IR energy.

Adam L. Anderson (S'00, M'10, SM'15) received the B.S. and M.S. degrees from Brigham Young University and the Ph.D. degree from the University of California at San Diego in 2008. He is currently a scientist at MRSL Real-Time Systems Laboratory. Dr. Anderson was the winner of the 2014 DARPA Spectrum Challenge, received the 2014 Leighton E. Sissom Award for Creativity and Innovation, and was recently awarded as a finalist in the 2019 DARPA Spectrum Collaboration Challenge.