Proposal Defense: Synthesis of Non-uniformly Spaced Antenna Arrays Using Data-driven Probabilistic Models

Nicholas Misiunas
Center for Advanced Computation and Telecommunications
Department of Electrical and Computer Engineering
University of Massachusetts Lowell

February 2017

Outline

- Introduction
  - Thesis Objective
  - Motivation
  - Prior Work
- Proposed Approach
- Problem Formulation for Linear Array
  - Optimization of Element Locations: Firefly Algorithm
  - Analysis of Positional Vectors
  - Hypothesized Probability Models
  - A Regression Model for Assignment of Element Positions
  - Performance Results
- Formulation for Planar Array
  - Mapping to a Parallel Program (GPU Implementation)
- Conclusion
- Timeline

Thesis Objective

Investigate the design and performance of beam forming from randomly spaced antenna elements on linear and planar arrays.

To accomplish this:

- Determine the probability distributions for element locations
- Design a model for rapid configuration of inter-element spacings that minimizes variance in the random beam forms
- Characterize the robustness of the system to positional errors

Motivation

Most antenna arrays consist of uniformly spaced elements that require complex current distributions across the elements to generate beams with required beamwidth (BW) and sidelobe levels (SLL).

With the advent of dense arrays formed with low-cost sensors, this is an opportunity to optimize spatial location.

Spacing elements non-uniformly will:

- Reduce complexity in eliminating element specific excitation currents for elements in a uniformly spaced array.
- Allow a reduced number of elements to be configured for beamforming.
- Design of non-uniformly spaced arrays particularly relevant to:
  - Minimal activation of sensors in a random distribution with fixed positions [Wong (2012) *et al.* [**?**]].
  - UAV swarm acting as a relay. Uniform spacings meant positional errors were significant [Palat *et al.*(2005) [**?**]].
  - Grating lobe reduction in micro-UAV swarm [Namin(2012)[**?**]].

Prior Work

- Unz (1960) [?] investigated non-uniformly spaced arrays for element reduction by using a matrix inversion method to determine the required currents to match a target pattern.
- King (1960) [?] investigated different non-uniform spacing schemes with uniform amplitudes with the goal of eliminating grating lobes and reducing the required number of elements. Noted that SLL was lower than uniform spacing for some schemes.
- Ishimaru (1962) [?] built upon Taylor (1955) [?] and derived a source position function using an infinite series (including only the first term) where non-uniform spacings with equal amplitude reduced the SLL. However, the SLL was not constant, and rose with distance from main beam.
- Au and Thompson (2013) [?] also built on Taylor to obtain a position function through contour integration, yielding constant SLL matching a modified Dolph-Chebychev.

Prior Work (Cont'd)

Recently, optimization algorithms have seen large application to array synthesis.

- Metaheuristic algorithms provide a flexibility that is attractive for antenna array synthesis.
- Firefly algorithm applied to optimizing element positions for a chosen SLL and designing a null [Zaman *et al.* (2012) [**?**], Basu *et al.* (2011) [**?**]]. Obtains one solution, potentially large run time.
- Particle swarm optimization for optimizing element positions to minimize SLL [Bevelacque (2008) [**?**]]. Used MATLAB, many hours required for low element count.

Lack of models for fast non-uniformly spaced array synthesis. Little work towards using optimization algorithms to create a distribution of element positions whose beam pattern matches a target.

Proposed Work

Methodology:

- Use a metaheuristic algorithm to generate an ensemble of position vectors that generate a target beam pattern.
- Analysis of this data to hypothesize probabilistic model for element positions and element spacings.
- Application of the model as a prior to accelerate optimization.

Technical considerations:

- Parallel implementation to allow fast calculation of position vectors.
- Sensitivity to positional variations.

Preliminary Results

- Linear Array Formulation
- Optimization model: Firefly Algorithm
- Computational complexity
- Analysis of $\underline{x}^*$ for the linear array.
- Parallel GPU implementation for planar array.

Geometry of a Linear Array

- Direction of incoming source signal: ϕ_d.
- x_n: position of the n^{th} element

./plots/arraygeo-eps-converted-to.pdf

Figure 1: Geometry of Linear Array

Overview

Directivity or beam pattern from linear array of aperture length L is generated from this expression:

$$R(u) = \frac{1}{N} \sum_{n=0}^{N-1} \cos\left(x_n(u - u_d)\right)$$

- Directional cosine: $u = kL\cos\phi$, $u_d = kL\cos\phi_d$
- ϕ: azimuthal angle.
- k: wave number $\left(\frac{2\pi}{\lambda}\right)$.
- λ: wavelength.
- $\underline{x}$: $[x_0, x_1, ..., x_{N-1}]$.

Objective: Find optimal $\underline{x}^*$ to match given $R_T(u)$.

Example Beam Form

- Steered to $\phi_d = \frac{\pi}{2}$.
- $R_T(u)$ consists of a side lobe level (SLL_T) and beamwidth (BW_T).

./plots/meandb-eps-converted-to.pdf

Figure 2: Example Beam Form and Target Function

Firefly Algorithm [Yang (2009) [?]]

$$\underline{x}_i^{t+1} = \underline{x}_i^t + \sum_{j \in \hat{N}_{i,ff}} e^{-\gamma r_{ij}^2} \left(\underline{x}_j^t - \underline{x}_i^t\right) + \alpha \left(1 - \frac{t}{T}\right) \underline{\epsilon}_i^{t+1}$$

- $\underline{x}_i^{t+1}$: vector of positions for i^{th} firefly at $(t+1)^{th}$ iteration.
- N_{ff}: set of fireflies. $\hat{N}_{i,ff}$: more attractive fireflies.
- r_{ij}: distance between i^{th} and j^{th} fireflies.
- γ: governs visibility region.
- $\underline{\epsilon}_i^{t+1}$: vector of random perturbations.
- α: governs random step size.
- T: total number of iterations

Firefly Algorithm, Cont'd

The fitness of a firefly is computed by comparing its beam form with the target:

$$f_i^t = -\sum_u \left[R_i^t(u) - R_T(u)\right]^2 \; I(u)$$

where $I(u) = 1$ for $R_i^t(u) > R_T(u)$. The target function is:

$$R_T(u) = \begin{cases} 1, & -\frac{BW_T}{2} < \cos^{-1}\left(\frac{u}{kL}\right) < \frac{BW_T}{2} \\ SLL_T, & \text{else} \end{cases}$$

where BW_T is the beam width and SLL_T is the side lobe level.

Computational Complexity

The evaluation function scales as $\mathcal{O}(N \cdot N_u)$, where N_u is the angular resolution.

The firefly algorithm scales as $\mathcal{O}\left(N_{ff}^2 \cdot N_{ss} + N_{ff} \cdot N_{eval}\right)$, where N_{ss} is the size of the search space and N_{eval} is the scaling of the evaluation function.

CPU serial profiling results:

Function	Instruction Fetches
cos	34%
$R_i(\phi_j)$	47%
log	2%
e	2%

For linear array synthesis: parallelize fireflies when computing beam form.

Parameters

Use values as in work by Au and Thompson [**?**].

- $N = 10$
- $BW_T = 54^\circ$
- $SLL_T = -24$ dB
- *ensembles* : 10^3
- $u = [0 : 5\pi]$
- Error tolerance: 0.1

Sensitivity analysis of the firefly algorithm:

- $\gamma = \frac{1}{N}$
- $\alpha = 0.5\frac{1}{N}$

These values found to yield fast convergence rates with majority of fireflies resolving a high fitness.

Linear Array Results

./plots/comparisonWPI-eps-converted-to.pdf

Figure 3: Distribution of Elements for Analytic and Firefly Results

Correlation

Correlation matrix of inter-element spacings for $d_0 : d_6$.

	d_0	d_1	d_2	d_3	d_4	d_5	d_6
d_0	**1.00**	**-0.58**	-0.34	0.29	0.18	-0.17	-0.16
d_1	**-0.58**	**1.00**	**-0.53**	0.06	0.25	-0.05	-0.24
d_2	-0.34	**-0.53**	**1.00**	**-0.61**	-0.28	0.25	0.29
d_3	0.29	0.06	**-0.61**	**1.00**	**-0.50**	-0.09	0.24
d_4	0.18	0.25	-0.28	**-0.50**	**1.00**	**-0.51**	-0.33
d_5	-0.17	-0.05	0.25	-0.09	**-0.51**	**1.00**	**-0.56**
d_6	-0.16	-0.24	0.29	0.24	-0.33	**-0.56**	**1.00**

Correlation matrix of inter-element spacings for $d_6 : d_9$.

	d_6	d_7	d_8	d_9
d_6	**1.00**	**-0.91**	**0.67**	-0.42
d_7	**-0.91**	**1.00**	**-0.88**	**0.64**
d_8	**0.67**	**-0.88**	**1.00**	**-0.88**
d_9	-0.42	**0.64**	**-0.88**	**1.00**

Paired Adjacent Spacings

./plots/scatteradj-eps-converted-to.pdf

Figure 4: Scatter Plot of Paired Adjacent Spacings (d_i, d_{i+1})

Paired Spacing with Summed Successive Pair

./plots/scatteradjsum-eps-converted-to.pdf

Figure 5: Summed Successive Pair $(d_i, d_{i+1} + d_{i+2})$

Regression Model

Considering importance of the boundary of the aperture, synthesize array beginning with the last spacing, d_9.

Moving towards the origin, predict each spacing based on the sum of the previous two.

Fit the regression model to the firefly data set.

$$
\begin{aligned}
\hat{d}_9 &\sim f_{D_9}(d_9) \\
\hat{d}_8 &= a_8 + b_8 \hat{d}_9 \\
\hat{d}_i &= a_i + b_i \left(\hat{d}_{i+1} + \hat{d}_{i+2} \right), i = 7, 6, ..., 0
\end{aligned}
$$

10^4 random samples from $f_{D_9}(d_9)$.
R^2 values:

- $d_0 : 0.67$, $d_1 : 0.31$
- $d_2 : 0.77$, $d_3 : 0.41$
- $d_4 : 0.72$, $d_5 : 0.47$
- $d_6 : 0.70$, $d_7 : 0.87$, $d_8 : 0.78$

Performance

./plots/costcdfnolegend-eps-converted-to.pdf

Figure 6: CDF of Cost of Arrays Generated via Model

Performance (Cont'd)

./plots/meanplusstd-eps-converted-to.pdf

Figure 7: μ and $\mu + \sigma$ of Firefly and Model Beam Forms

Standard Deviation of Side Lobe Level

./plots/comparison_std_allWPI-eps-converted-to.pdf

Figure 8: Standard Deviation for Analytical, Firefly, and Model Results

Planar Array Complexity

The additional angular dimension introduces a second cosine to compute.

To maintain angular resolution, number of discrete angles to consider is squared.

Scaling is now $\mathcal{O}\left(N_{ff}^2 \cdot N + N_{ff} \cdot N \cdot N_u \cdot N_v\right)$.

Infeasibly slow run-time.

Parallel Implementation

Consider the three different parallelization models[**?**].

Extremely expensive evaluation function - choose solution-level parallel model.

- Fine-grained parallel approach.
- Concatenate firefly arrays.
- Element positions are fine-grained dimension for memory coalescing.
- CPU synchronization for all required synchs.
- Data transfers only at initialization/finalization stages.
- CUB library for parallel key-sort for unconstrained movement.
- Online algorithm[**?**] for statistics, done in parallel and merged in finalization stage.

Parallelization Performance

Parameters	CPU (s)	GPU (s)	Speedup
N_{ff} : 20 N : 100 *ensembles* : 10	2648.22	8.63	306x
ensemble : 1	265.20	3.11	85x
ensemble : 1 N_{ff} : 100	1327.18	5.96	222x
ensemble : 1 N : 400	1052.17	4.79	219x
N : 400	10499.71	26.11	402x

Other Results

- Verification of algorithm performance against analytically derived performance of uniformly distributed array.
- Analytic ordered statistics using distribution from Au and Thompson (2013)[**?**].
- Data generation for a main beam steered to $\phi_d = \frac{5\pi}{8}, \frac{6\pi}{8}, \frac{7\pi}{8}$.
- Sensitivity analysis of firefly results for $N = 7, 10, 13$.

Conclusion

- The parallel firefly algorithm allowed generation of a large data set of $\underline{x}^*$.
- Summed successive spacings posses a more linear relationship over paired spacings.
- The data-driven probabilistic model demonstrated here showed good performance, with the variance approaching that of the firefly algorithm.
- The parallel planar array implementation resulted in significant speedups and will allow generation of required data sets.

Timeline

- Explore robustness of probabilistic models for prediction. 2/10/17
- Extend preliminary model to beam steering problem. 3/10/17
  - Three different u_d.
  - Determine model specifications.
- Design predictive model for broadside planar array. 4/14/17

Bibliography I

Symmetric Planar Antenna Array Synthesis (Cont'd)

plots/2Darrayseg-eps-converted-to.pdf

Figure 9: Symmetric 2D Antenna Array with Bounded Elements

Symmetric Planar Antenna Array Synthesis

plots/2Dgeometry-eps-converted-to.pdf

Figure 10: Symmetry and Geometry of 2D Planar Array