GPU-accelerated power pattern synthesis of aperiodic linear arrays

A. Capozzoli, A. Breglia, C. Curcio, and A. Liseno

Università di Napoli Federico, Dipartimento di Ingegneria Biomedica, Elettronica e delle Telecomunicazioni, via Claudio 21, I 80125 Napoli (Italy). Tel.: +39 081 7683358. Email: a.capozzoli@unina.it

Abstract

We deal with the development of a computationally effective approach for the synthesis of equivalently tapered, aperiodic linear arrays, i.e. arrays matching the requirements on the power pattern by acting only on the element positions and excitation phases.

The computational effectiveness of the algorithm is reached by the development of a parallel Non Uniform Fast Fourier Transform (NUFFT) routine running on a Graphics Processing Units (GPU). The numerical results point out the computational and synthesis performance of the technique. For an array of 3000 elements, the parallel approach has resulted to be about 10 times faster than the sequential one.

1 Introduction

In the design of high performance antennas, equivalently tapered arrays [1] represent a valid alternative to uniform arrays and reflectors, since they allow combining flexibility, power efficiency, beam shaping, electronic reconfigurability, polarization purity and bandwidth enlargements [1].

In the synthesis of aperiodic arrays, many design parameters are involved and complexly, and non-linearly, interrelated. This translates in the optimization of multimodal functionals which, for a successful search, should be performed using multi-stage iterative approaches [2], possibly involving global techniques [3]. Such iterative optimizations are computationally burdened since they require the evaluation of the far-field, and the functional gradient, at each iteration step which are moreover complicated by the need of dealing with an irregular element lattice. Finally, the synthesis procedure should be properly driven to account for design specifications dictated by minimum and maximum acceptable interelement spacings, to comply with superdirectivity issues, mutual coupling, or maximum acceptable antenna size.

Recently, the authors have developed a new, computationally efficient approach to the synthesis of equivalently tapered, conformal aperiodic arrays, with phase excitation control [4, 5]. The approach exploits proper representations of the element positions and excitation phases limiting the number of design parameters, enabling progressive enlargements of the unknowns to mitigate the false solution issue and a simple procedure to enforce the constraints. To cope with computational complexity, the approach has introduced, for the very first time, the computation of the array factor by means of Non Uniform Fast Fourier Transforms (NUFFTs) [6] having the same asymptotic complexity of standard FFTs.

Purpose of this paper is to show how the approach in [4, 5] can be even more fruitfully exploited if the NUFFT-based array factor calculation is implemented on innovative, intrinsically parallel, off-the-shelf hardware provided by Graphics Processing Units (GPUs) [7, 8]. Indeed, the recent interest in GPUs for scientific computing has prompted the development of programming frameworks, as the NVIDIA Compute Device Unified Architecture (CUDA) extension of the ANSI C, strongly simplifying the implementations as compared to the direct use of graphical APIs.

2 The problem and the synthesis procedure

We consider a linear, non-uniform array made of M elements, located on the x axis, having positions $\underline{x} = (x_0, ..., x_{M-1})$ (see Fig. 1(left)). The array elements are assumed uniformly excited in amplitude, while their respective excitation phases are $\underline{\gamma} = (\gamma_0, ..., \gamma_{M-1})$. The array factor F can be written as

$$F(u) = \sum_{i=0}^{M-1} a \cdot e^{j\gamma_i + jux_i} \tag{1}$$

where $u = \beta \cos \psi$, $\beta = 2\pi/\lambda$, λ is the wavelength, and a is the excitation amplitude.

The problem of interest is determining the element control phases γ_i and positions x_i , given the design specifications on $|F|^2$ (power pattern synthesis) [4, 5], which are provided by means of two mask functions

 $M_l(u)$ and $M_u(u)$, lower and upper bounding $|F|^2$, respectively [9]. The array synthesis is performed by optimizing the cost function

$$\Phi(\underline{x},\underline{\gamma}) = \left\| \left| F \right|^2 - P_{\mathcal{U}}(\left| F \right|^2) \right\|^2$$
(2)

where $P_{\mathcal{U}}$ is the projector onto the set \mathcal{U} (defined by mask functions [9]) that contains all the power patterns satisfying the specifications, and $\|\cdot\|$ is a properly chosen (\mathcal{L}^2 in this paper) norm.

The synthesis algorithm is appointed to determine the 2M - 2 parameters represented by element locations and excitation phases (for all the elements except for one) minimizing Φ and to satisfy the contraints on the element positions in terms of minimum interelement spacings and of maximum array dimensions [4, 5].

However, for large arrays, a large number of unknowns can make the synthesis procedure computationally very demanding, especially when global optimization techniques are employed [3], or can make the procedure to be very sensitive to the trapping problem when local minimizations are adopted. Furthermore, an efficient and effective way of enforcing the design constraints needs to be devised.

To control the number of unknowns, a mapping function transforming a uniform grid into a non-uniform one (see Fig. 1(right)) is employed, namely:

$$f: \xi \in [-1,1] \to f(\xi) \in \Re.$$
(3)

Accordingly, the sampling locations x_m 's are obtained by uniformly sampling f at the points $\xi_m = -1 + m\Delta\xi$, with $\Delta\xi = 2/(M-1)$ and $m = 0, \ldots, M-1$, so that $x_m = f(\xi_m)$. Furthermore, f is chosen so to depend on $L \ll M-1$ parameters $\underline{c} = (c_0, \ldots, c_{L-1})$ only and is expressed as

$$f\left(\xi\right) = af_0\left(\xi\right) + b\xi\tag{4}$$

where

$$f_0(\xi) = \sum_{l=0}^{L-1} c_l T_l(\xi),$$
(5)

the T_l 's are properly chosen basis functions, e.g., Legendre polynomials of degree l in this paper, while a and b are parameters depending on $(c_0, ..., c_{L-1})$, determined to satisfy the synthesis constraints [4, 5]. Similarly, a mapping function g depending on $R \ll M - 1$ coefficients $\underline{w} = (w_0, ..., w_{R-1})$ is also introduced to represent the element excitation phases, namely

$$g: \xi \in [-1,1] \to g(\xi) \in \Re, \tag{6}$$

so that $\gamma_m = g(\xi_m)$. The function g can be likewise expressed by a modal expansion as

$$g(\xi) = \sum_{r=0}^{R-1} w_r S_r(\xi),$$
(7)

where the S_r 's are properly chosen functions, again Legendre polynomials in this paper.

According to eqs. (3-7), the number of parameters is strongly reduced since now $\Phi = \Phi(\underline{c}, \underline{w})$. It is now possible to modulate the number of unknowns to enable a progressive enlargement thereof (mitigating the trapping problem [2]), and the synthesis constraints can be easily enforced at each iteration step [4, 5, 10]. In this paper, the minimization of Φ is performed by an iterative, gradient-based, local optimization algorithm. The initial point of the iterative procedure is obtained by considering the result of an aperture synthesis based on prolate spheroidal functions, that provides a continuous aperture field [4].

3 GPU-based, NUFFT evaluation of the array factor

During the iterative optimization of Φ , the array factor is evaluated a large number of times. To mitigate the computational effort, let us denote by F_k the discrete values of F at the uniform sampling locations $u = k2\beta/K$, with k = -K/2, ..., K/2 - 1. Then

$$F_k = \sum_{l=0}^{M-1} z_l e^{-2\pi j \hat{x}_l \frac{k}{K}},$$
(8)



Figure 1: Left: Geometry of the problem. Right: The mapping function to represent the array positions.

where $z_l = \exp(j\gamma_l)$ and $\hat{x}_l = -[(2\beta)/(2\pi)]x_l$. Eq. (8) is the expression of a Non Uniform Discrete Fourer Transform of Non-Equispaced Data (NED) type, which can be evaluated by a NED-NUFFT [6], having a $N \log N$ asymptotic computational complexity. It should be mentioned that the most general situation concerns the case when also the sampling locations of F are non uniform. This circumstance can be dealt with by a "type-3" NUFFT [11] and will be subject of future extensions.

The fast evaluation of the NUDFT in eq. (8) relies on the following representation of the exponential kernel

$$e^{-2\pi j \hat{x}_l \frac{k}{K}} = \frac{(2\pi)^{-1/2}}{\phi\left(\frac{2\pi k}{cK}\right)} \sum_{m \in \mathbb{Z}} \hat{\phi}(cx_l - m) e^{-2\pi j m \frac{k}{cK}},\tag{9}$$

where c is an oversampling factor (usually equal to 2), ϕ is the Kaiser-Bessel window and $\hat{\phi}$ is its transform. Accordingly, eq. (8) can be rewritten as

$$F_k \simeq \frac{1}{\phi_k} \sum_{i=-cK/2}^{cK/2} u_i e^{-2\pi i j \frac{k}{cK}}$$
(10)

where $u_i = \sum_{l \in \mathbb{Z}} \sum_{m \in \mathbb{Z}} z_l \hat{\phi}_{l,j+cmK-\mu_l}, \phi_k = \phi(2\pi k/(cK)), \hat{\phi}_{lm} = (1/\sqrt{2\pi})\hat{\phi}(cx_l - (\mu_l + m)), \mu_l$ is the closest integer to cx_l and $\hat{\phi}$ has been assumed to be negligible outside [-P, P].

The GPU-based evaluation of eq. (10) has been implemented in CUDA and is performed in three steps:

- 1. Evaluation of ϕ_k , μ_l , u_i and $\hat{\phi}_{lm}$. These are independent of each other, but have been implemented in a single kernel to exploit the shared memory and reduce time consuming global memory accesses. On the other side, the evaluation of u_i can be performed in parallel, provided to properly dealing with multiple, pseudo-random memory accesses, leading to *race condition* occurrences. To save the data integrity, atomic instructions have been exploited [8].
- 2. Evaluation of an FFT of size cK. This step has been easily implemented by cuFFT library [7].
- 3. Scaling of the result. This is an embarassingly parallel step which can be easily parallelized.

4 Computational and synthesis performance

To point out the computational benefit arising from the introduced parallel approach, a sequential ANSI C implementation of the NUFFT algorithm has been also worked out. The sequential code has been run on a Genesis Tesla I-7950 workstation, with a 8-core In- tel CPU i7-950, working at 3.06GHz and with 6Gbytes of RAM. On the other side, the parallel CUDA code has been executed on the same workstation used for the sequential tests, but powered with a NVIDIA Tesla C2050, whose architecture consists of 14 streaming multiprocessors, each containing 32 streaming processors running at 1.15GHz, with a memory, 2.8GB sized.

Fig. 2(left) illustrates the speedup (sequential over parallel computing times) for different values of M. As it can be seen, for an array having 1000 elements, the parallel code has resulted to be about 10 times faster than the sequential one.

On the other side, Fig. 2(right) shows the pattern synthesized by requiring the equivalently tapered array to radiate a field according to an uniform Chébyshev array having the same number of M = 128 elements. The minimum and maximum allowed inter-element spacing have been 0.3λ and 0.8λ , respectively. Fig. 2(center) shows the synthesized element positions. The minimum and maximum synthesized inter-element spacings have been 0.46λ and 0.8λ , respectively.



Figure 2: Left: Speedup of the parallel approach as compared to the sequential one. Right: Synthesized field pattern for an equivalently tapered Chébyshev array. Center: Synthesized array positions for an equivalently tapered Chébyshev array.

References

- G. Toso, P. Angeletti, "Sparse and thinned array tracking," *European Patent Application*, no. 08290154.7, filed on 18.02.2008, published on 19 Aug. 2009 with number EP 2090995.
- [2] A. Capozzoli, C. Curcio, G. D'Elia, A. Liseno, "Fast phase-only synthesis of conformal reflectarrays", IET Microw. Antennas Prop., vol. 4, n. 12, pp. 1989-2000, Dec. 2010.
- [3] A. Capozzoli, G. D'Elia, Global optimization and antennas synthesis and diagnostics, part one: concepts, tools, strategies and performances, *Progr. Electromagn. Res.*, vol. 56, pp. 195-232, 2006.
- [4] A. Capozzoli, C. Curcio, G. D'Elia, A. Liseno, P. Vinetti, FFT & aperiodic arrays with phase-only control and constraints due to super-directivity, mutual coupling and overall size, *Proc. of the 30th ESA Antenna Workshop* on Antennas for Earth Observ., Science, Telecomm. and Navig. Space Missions, Noordwijk, The Netherlands, May 27-30, 2008, CD ROM.
- [5] A. Capozzoli, C. Curcio, G. D'Elia, A. Liseno, P. Vinetti, FFT & equivalently tapered aperiodic arrays, Proc. of the XXIX General Assembly of the Int. Union of Radio Sci., Chicago, IL, Aug. 7-16, 2008, CD ROM.
- [6] K. Fourmont, Non-equispaced fast Fourier transforms with applications to tomography, J. Fourier Anal. Appl., vol. 9, n. 5, pp. 431-450, 2003.
- [7] D.B. Kirk, W.W. Hwu, Programming massively parallel processors, Morgan Kaufmann, Burlington, MA, 2010.
- [8] A. Capozzoli, C. Curcio, G. D'Elia, A. Liseno, P. Vinetti, Fast CPU/GPU pattern evaluation of irregular arrays, Applied Comput. Electromagn. Soc. J., vol. 25, n. 4, pp. 355-372, Apr. 2010.
- [9] O.M. Bucci, G. D'Elia, G. Mazzarella, G. Panariello, Antenna pattern synthesis: a new general approach, Proc. of the IEEE, vol. 82, n. 3, pp. 358-371, Mar. 1994.
- [10] A. Capozzoli, A. Colella, C. Curcio, G. D'Elia, A. Liseno, P. Vinetti, Constrained phase-only synthesis for reflectarray antennas, *Proc. of the Adv. Lightweight Structures and Refl. Antennas*, Tbilisi, Georgia, Oct. 14-16, 2009, CD ROM.
- [11] J.-Y. Lee, L. Greengard, The type 3 nonuniform FFT and its applications, J. Comput. Phys., vol. 206, n. 1, pp. 1-5, 2005.