UNIVERSITY
OF TRENTO

DIPARTIMENTO DI INGEGNERIA E SCIENZA DELL'INFORMAZIONE

38123 Povo – Trento (Italy), Via Sommarive 14
http://www.disi.unitn.it

SYNTHESIS OF LARGE MONOPULSE LINEAR ARRAYS THOUGH A TREE-BASED OPTIMAL EXCITATIONS MATCHING

P. Rocca, L. Manica, A. Martini, and A. Massa

January 2007

Technical Report # DISI-11-059

Synthesis of Large Monopulse Linear Arrays through a Tree-Based Optimal Excitations Matching

P. Rocca, L. Manica, A. Martini, and A. Massa, *Member, IEEE*

Department of Information and Communication Technologies,

University of Trento, Via Sommarive 14, 38050 Trento - Italy

Tel. +39 0461 882057, Fax +39 0461 882093

E-mail: *andrea.massa@ing.unitn.it,*

{*paolo.rocca*, *luca.manica*, *anna.martini*}@*dit.unitn.it*

Web-site: *http://www.eledia.ing.unitn.it*

Synthesis of Large Monopulse Linear Arrays through a Tree-Based Optimal Excitations Matching

P. Rocca, L. Manica, A. Martini, and A. Massa

Abstract

In this paper, the synthesis of large arrays for monopulse tracking applications is addressed by means of a simple and effective sub-arraying technique. Towards this purpose, the synthesis problem is recast as the search of an optimal path in a non-complete binary tree by exploiting some relationships between independently-optimal sum and difference excitations. Because of a suitable reduction of the solution space and the implementation of a fast path-searching algorithm, the computational issues arising in dealing with large array aperture are properly addressed. Some numerical experiments are provided in order to assess the feasibility and the computational effectiveness of the tree-based approach.

Key words: Large Linear Arrays, Monopulse Antennas, Sum and Difference Pattern Synthesis, Tree-Searching Algorithm.

1 Introduction

The synthesis of monopulse antennas is not a trivial task because of the need of generating two different patterns (namely, a sum and a different pattern) by means of the same array structure. Such a difficulty further increases for large arrays. As a matter of fact, the optimal solution of implementing two independent feed networks is almost impracticable due to the required costs, the architecture complexity, and the spatial extension especially when dealing with large structures. As an indication regarding the practical aspects of the actual realization on large arrays, let us consider that when dealing with phased systems for beamforming purposes, time delay units are typically placed at the output of a sub-array due to the high cost of placing a time delay unitn at each element of the array [1].

Alternatively, some methods based on sub-arraying techniques have been proposed [2]-[5]. As a reference, *McNamara* presented in [2] the so-called excitation matching method aimed at obtaining the "*best compromise*" difference pattern starting from a set of optimal sum excitations. Unfortunately, the reliability of such an approach reduces when large arrays are taken into account due to the inversion of the arising ill-conditioned solution matrix. Optimization approaches [3]-[5] based on the minimization of a suitable cost function overcome this drawback. In such a framework, Ares *et al.* proposed in [3] a Simulated Annealing-based technique where, starting from a fixed configuration of sub-arrays, the sub-arrays weights are determined through the minimization of a suitable cost function that penalizes side-lobe levels (SLLs) exceeding a prescribed threshold. Unlike [3], the simultaneous optimization of sub-arrays partitions and their weights has been addressed in [4] and [5] by applying a Genetic Algorithm (GAs) and a Differential Evolution (DE) algorithm, respectively.

However, although optimization methods are not affected by the ill-conditioning issue and perform very well in facing with both multi-constrained and multi-variable (i.e., real/discrete/binary unknowns) problems, they are usually time-consuming when dealing with large arrays. In such a case, even though the solution space is efficiently sampled, its dimension is very large and several iterations are needed for reaching a reliable and satisfactory solution. In order to avoid such an event, this letter deals with a simple and computationally-effective resolution method. By considering that an optimal sub-arraying can be obtained by exploiting the similarity properties between independently-optimum sum and difference excitations, the problem is recast as the search of an optimal path in an incomplete binary tree carried out by a simple swapping algorithm.

The paper is organized as follows. The mathematical formulation of the tree-based approach is described in Sect. 2, whereas some representative results from a set of numerical experiments are presented and discussed in Sect. 3. Eventually (Sect. 4), some conclusions are drawn.

2 Mathematical Formulation

Let us consider a linear uniform array of $N = 2M$ elements where the sum and difference patterns are obtained staring from symmetric $\underline{A} = \{a_m = a_{-m};\ m = 1, ..., M\}$ and anti-symmetric $\underline{B} = \{b_m = -b_{-m};\ m = 1, \ldots, M\}$ real excitations, respectively. Thanks to the symmetries, only one half of the elements of the array $\underline{S} = \{\xi_m;\ m = 1, ..., M\}$ is considered dealing with the monopulse synthesis problem.

As far as ideal conditions are concerned, the optimal excitations sets (i.e., optimal sum $\underline{A}^{opt} = \{\alpha_m;\ m = 1, ..., M\}$ and optimal difference $\underline{B}^{opt} = \{\beta_m;\ m = 1, ..., M\}$ coefficients) are computed by using analytical methods based on Chebyshev's [6] and Zolotarev's [7][8] polynomials, respectively.

On the other hand, sub-arraying techniques [2], starting from the optimal sum pattern (by assuming $a_m = \alpha_m;\ \ m = 1, ..., M$), generate a sub-optimal difference pattern by means of a suitable grouping of the M array elements in Q different sub-arrays. Mathematically, the sub-arrays solution is described in terms of the grouping vector $\underline{C}$ [5] of M positive integers $c_m \in [1, Q]$, which identifies the membership of each element to a sub-array, and by the difference excitations given by $b_m = w_{mq}\alpha_m$ $(m = 1, ..., M;\ q = 1, ..., Q)$ where $w_{mq} = \delta_{c_m q} w_q$ ($\delta_{c_m q} = 1$ if $c_m = q$, $\delta_{c_m q} = 0$ otherwise) is the weight coefficient associated to the m-th array element belonging to the q-th sub-array.

The proposed synthesis approach is aimed at defining a sub-array configuration $\underline{C}_{opt}$ such that the compromise difference excitation set $\underline{B}$ is as close as possible to $\underline{B}^{opt}$. Towards this end, let us observe that, although the total number of sub-array configurations is equal to $U = Q^M$, the number of "*allowed*" solutions reduces to $U^{(tree)} = \left(\begin{array}{c} M-1 \\ Q-1 \end{array}\right)$ by avoiding "*empty*" or "*equivalent*"[(1)] sub-arrays configurations. Such a set of solutions can be usefully represented by means of a binary tree (or "*solution tree*") of depth M by properly sorting the array elements. More in detail, let us define a set of reference parameters $\underline{V} = \{v_m;\, m = 1, ..., M\}$ called "*optimal gains*"

$$v_m = \frac{\beta_m}{\alpha_m} \qquad m = 1, ..., M \tag{1}$$

in correspondence with each element of $\underline{S}$. Then, the values of the v_m parameters are ordered in a list $\underline{L} = \{l_m;\ m = 1, ..., M\}$, where $l_i \leq l_{i+1},\ i = 1, ..., M-1$, $l_1 = min_m \{v_m\}$, and $l_M = max_m \{v_m\}$, in order to build the *solution tree* shown in Fig. 1 ($M = 5$ and $Q = 3$) where the positive integer q inside a node at the l_m-th level indicates that the array element identified by l_m is a member of the q-th sub-array. Such an architecture guarantees that elements grouped in the same sub-array have close v_m values. Moreover, according to this representation, it is possible to recast the problem solution (i.e., $\underline{C}_{opt}$) as the search of an optimal path inside the tree. Towards this end, let us define a suitable cost function (or *metric*) Ψ that quantifies the closeness of each candidate/trial solution $\underline{C}$ to the optimal one

$$\Psi\left\{\underline{C}\right\} = \sum_{m=1}^{M} \left(v_m - d_m \left\{\underline{C}\right\}\right)^2, \tag{2}$$

where the estimated parameters $d_m\left(\underline{C}\right)$ are computed as follows

$$d_m\left(\underline{C}\right) = \frac{\sum_{s=1}^{M} \delta_{c_s q} v_s}{\sum_{s=1}^{M} \delta_{c_s q}} \qquad m = 1, \ldots, M. \tag{3}$$

Consequently, $\underline{C}_{(opt)}$ is identified as the result of a sequence of trial solutions that min-

(1) As an example, $\underline{C}_i = \{3, 1, 1, 2, 1, 3, 2, 2\}$ is *equivalent* to $\underline{C}_j = \{1, 2, 2, 3, 2, 1, 3, 3\}$.

imizes the cost function Ψ (i.e., $\underline{C}_{opt} = arg\left[min_{k=1,...,K}\Psi\left\{\underline{C}_k\right\}\right]$, k being the iteration index) and the sub-array weights are assumed to be equal to the optimal values of the "*computed gains*" $d_m\left(\underline{C}_{opt}\right)$

$$w_q = \delta_{c_m q} d_m \left\{\underline{C}_{opt}\right\} \qquad q = 1, ..., Q. \tag{4}$$

It should be noticed that the sub-arrays weights $\{w_q;\ q = 1, ..., Q\}$ are analytically-computed once the sub-array membership of each element is determined and they are not involved in the optimization process.

As far as the generation of the sequence of trial solutions $\{\underline{C}_k;\ k = 1, ..., K\}$ is concerned, let us observe that only some elements of the list L are candidate to change their sub-array membership without violating the sorting condition of the *allowed* sub-array configurations. These array elements (called "*border elements*") are identified by the l_m indexes whose adjacent list values l_{m-1} or/and l_{m+1} belong to a different sub-array. Therefore, $\underline{C}_{opt}$ is found starting from an initial path $\underline{C}^{(0)}$, randomly-chosen among the set of paths of the *solution tree*, and iteratively updating the candidate solution $\underline{C}_k \leftarrow \underline{C}_{k+1}$ just modifying the membership of the border elements, until a maximum number of iterations K (i.e., $k > K$) or a stationary condition for the fitness value (i.e., $\frac{\left|K_{window}\Psi_{opt}^{(k-1)} - \sum_{j=1}^{K_{window}}\Psi_{opt}^{(j)}\right|}{\Psi_{opt}^{(k)}} \leq \eta$, K_{window} and η being a fixed number of iterations and a fixed numerical threshold, respectively) is satisfied.

3 Numerical Results

In order to assess the effectiveness of the proposed method in dealing with large arrays, some numerical simulations have been performed. Since, likewise [2], such an approach belongs to the class of synthesis techniques aimed at determining the "*best compromise*" difference pattern, let us define some indexes for allowing a quantitative evaluation of the closeness of sub-optimal compromises to optimal patterns. In particular, the pattern matching Δ

$$\Delta = \frac{\int_0^{\pi} \left| |AF(\psi)|_n^{opt} - |AF(\psi)|_n^{rec} \right| d\psi}{\int_0^{\pi} |AF(\psi)|_n^{opt} d\psi}, \tag{5}$$

where $\psi = (2\pi d/\lambda) \, sin\theta$, $\theta \in [0, \pi/2]$, (λ and d being the free-space wavelength and the inter-element spacing, respectively), $|AF(\psi)|_n^{opt}$ and $|AF(\psi)|_n^{rec}$ are the optimal and synthesized difference pattern, respectively. Moreover, the *beamwidth* B_W and the *power slope* P_{slo} that numerically "describe" the pattern slope on the boresight direction

$$P_{slo} = 2 \times \left[\max_{\psi} \left(|AF(\psi)|_n \right) \times \psi_{max} - \int_0^{\psi_{max}} |AF(\psi)|_n \, d\psi \right], \tag{6}$$

ψ_{max} being the angular position of the maximum of the array pattern.

In the numerical assessment, a linear array of N equally-spaced ($d = \frac{\lambda}{2}$) elements has been considered. Concerning the optimal reference setup, sum and difference excitations have been chosen to generate a Dolph-Chebyshev pattern [6] with $SLL = -25 \, dB$ and a Zolotarev pattern [8] with $SLL = -30 \, dB$. Moreover, in order to evaluate the performance of the tree-based method versus the array dimension, N has been varied from 20 (small/medium arrays, i.e. $M < 50$) up to 500 (large arrays, i.e. $M \geq 50$) and different array partitions ($Q \in [3, \, 10]$) have been considered.

The plot of Δ versus M for different values of Q is shown in Figure 2. As it can be observed, for a fixed number Q of sub-arrays, the distance between the optimal difference pattern and the compromise one does not significantly vary as the number of elements M increases ($M > 50$) ranging from $\Delta \cong 0.15$ ($Q = 10$) up to $\Delta \cong 0.36$ ($Q = 3$). Moreover, as expected, for each array aperture (i.e., $M = cost$), the synthesized difference patterns get closer and closer to the optimal one when the value of Q grows ($Q \rightarrow M$).

As a representative result, sum and compromise difference patterns when $N = 500$ and $Q = 3$ are shown (Fig. 3) as well as the corresponding sets of excitations (Fig. 4). For completeness, the number of elements of each sub-array, n_q, and the sub-arrays weights are reported in Tab. I. Notwithstanding the value of $\frac{M}{Q} \simeq 83$ (i.e., a limited number of large sub-arrays), it turns out that the differences in terms of P_{slo} and B_w between compromise pattern and optimal difference target are less than 1.5% and 2% (being $\Delta <$

0.4), respectively.

As far as the computational issues are concerned, let us firstly analyze the dimension of the solution space $U^{(tree)}$ of the tree-based method as shown in Fig. 5. As stated in Sect. 2, $U^{(tree)}$ behaves as a binomial function of M and Q, while the total number of configurations U (i.e., the dimension of the solution space sampled by optimization algorithms) grows exponentially with M ($U = Q^M$). Thanks to the reduced dimension of $U^{(tree)}$ and because of the computational simplicity of the swapping algorithm, the number of iterations k_{opt} needed to reach the final solution turns out to be acceptable whatever the array aperture ($max_{M,Q}\left\{k_{opt}\rfloor_{M,Q}\right\} < 90$) especially taking into account that the CPU-time t_k for evaluating a trial solution (on a $3\,GHz$ Pentium 4 and $512\,MB$ of RAM) is lower than $max\left\{t_k\right\} = 0.81$ $[sec]$ ($M = 250$ and $Q = 10$ - Fig. 7) and, as an example, it reduces to 0.12 $[sec]$ when $M = 100$.

4 Conclusions

In this paper, the synthesis of large array monopulse antennas has been dealt with a tree-based sub-arraying method. In order to reach an optimal matching between synthesized and independently-optimum sum and difference patterns and by exploiting some relationships among admissible aggregations, the synthesis problem has been recast as the search of the minimum-cost path in a non-complete binary tree. Towards this purpose, a simple and affective swapping algorithm that considers the presence of border elements more suitable to change sub-array membership, has been used. Some representative results from an exhaustive set of numerical experiments confirmed the potentialities of the approach in dealing with large arrays both in terms of computational costs and accuracy.

Acknowledgments

The authors wish to thank Dr. M. Donelli and Prof. M. Pastorino for useful discussions and suggestions. This work has been partially supported in Italy by the "*Progettazione di un Livello Fisico 'Intelligente' per Reti Mobili ad Elevata Riconfigurabilità*," Progetto di Ricerca di Interesse Nazionale - MIUR Project COFIN 2005099984.

References

[1] R. L. Haupt, "Optimized weighting of uniform subarrays of unequal size," *IEEE Trans. Antennas Propagat.*, vol. 55, no. 4, pp. 1207-1210, Apr. 2007.

[2] D. A. McNamara, "Synthesis of sub-arrayed monopulse linear arrays through matching of independently optimum sum and difference excitations," *IEE Proc. H*, vol. 135, no. 5, pp. 371-374, Oct. 1988.

[3] F. Ares, S. R. Rengarajan, J. A. Rodriguez, and E. Moreno, "Optimal compromise among sum and difference patterns through sub-arraying," *Proc. IEEE Antennas Propagat. Symp.*, pp. 1142-1145, Jul. 1996.

[4] P. Lopez, J. A. Rodriguez, F. Ares, and E. Moreno, "Subarray weighting for difference patterns of monopulse antennas: Joint optimization of subarray configurations and weights," *IEEE Trans. Antennas Propagat.*, vol. 49, no. 11, pp. 1606-1608, Nov. 2001.

[5] S. Caorsi, A. Massa, M. Pastorino, and A. Randazzo, "Optimization of the difference patterns for monopulse antennas by a hybrid real/integer-coded differential evolution method," *IEEE Trans. Antennas Propagat.*, vol. 53, no. 1, pp. 372-376, Jan. 2005.

[6] C. L. Dolph, "A current distribution for broadside arrays which optimises the relationship between beam width and sidelobe level," *Proc. IRE*, vol. 34, pp. 335-348, 1946.

[7] D. A. McNamara, "Discrete $\bar{n}$-distributions for difference patterns," *Electron. Lett.*, vol. 22, no. 6, pp. 303-304, Jun. 1986.

[8] D. A. McNamara, "Direct synthesis of optimum difference patterns for discrete linear arrays using Zolotarev distribution," *IEE Proc. H*, vol. 140, no. 6, pp 445-450, Dec. 1993.

FIGURE CAPTIONS

- **Figure 1.** *Solution-Tree* structure ($M = 10$, $Q = 3$).
- **Figure 2.** *Large Arrays* ($d = \frac{\lambda}{2}$) - Behavior of Δ versus M for various values of Q.
- **Figure 3.** *Large Arrays* ($M = 250$, $d = \frac{\lambda}{2}$, $Q = 3$) - (a) Optimal sum pattern (*Dolph-Chebyshev pattern* [6] - $SLL = -30\,dB$) and (b) compromise difference pattern.
- **Figure 4.** *Large Arrays* ($M = 250$, $d = \frac{\lambda}{2}$, $Q = 3$) - Excitations coefficients.
- **Figure 5.** *Computational Analysis* - Behavior of U versus M for various values of Q.
- **Figure 6.** *Computational Analysis* - Behavior of k_{opt} versus M for various values of Q.
- **Figure 7.** *Computational Analysis* - Behavior of t_k versus M for various values of Q.

TABLE CAPTIONS

- **Table I.** *Large Arrays* ($M = 250$, $d = \frac{\lambda}{2}$, $Q = 3$) - Number of elements per sub-array, n_q, and sub-array weights, w_q.

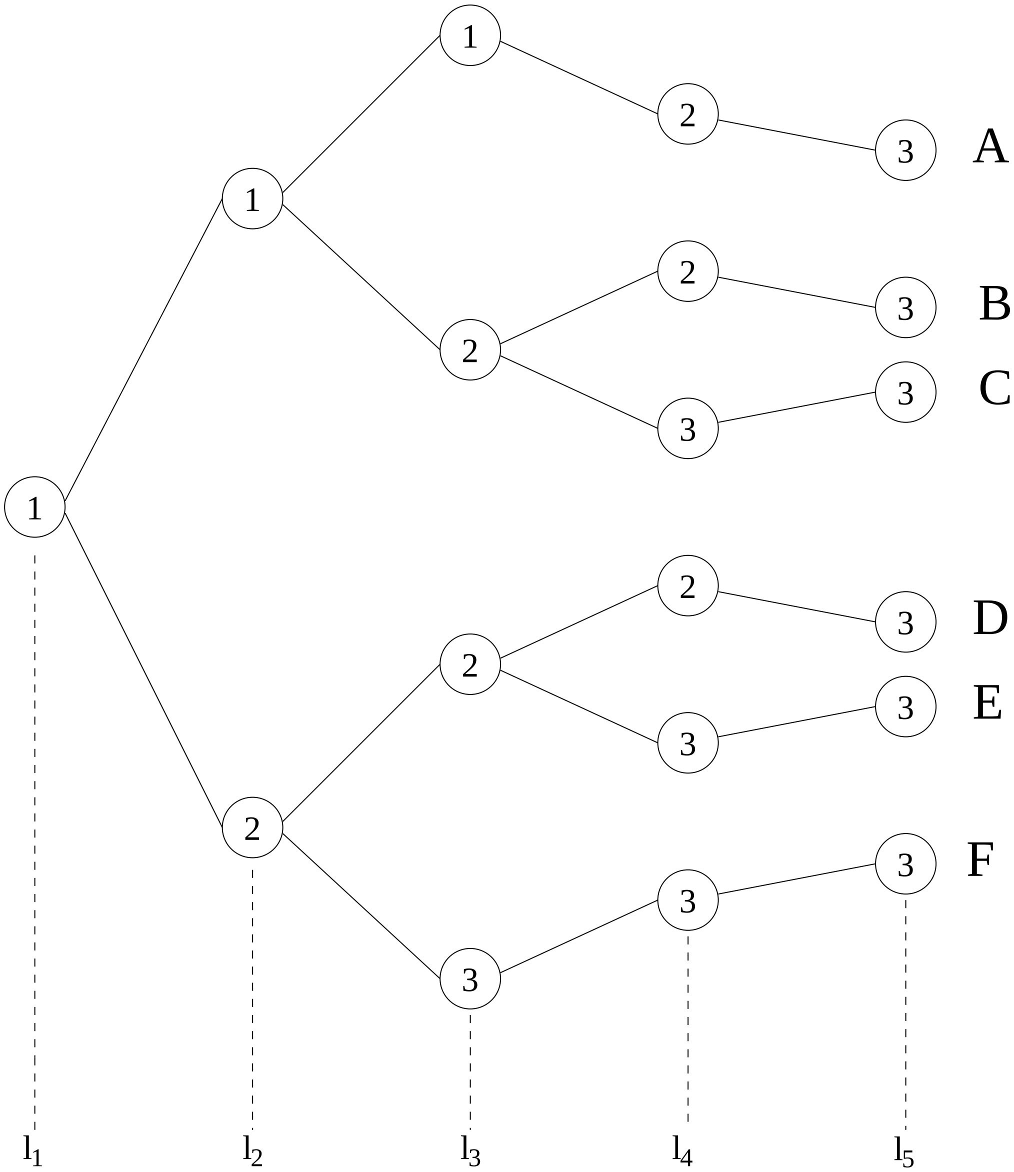


Fig. 1 - P. Rocca *et al.*, "Synthesis of large monopulse linear arrays ..."

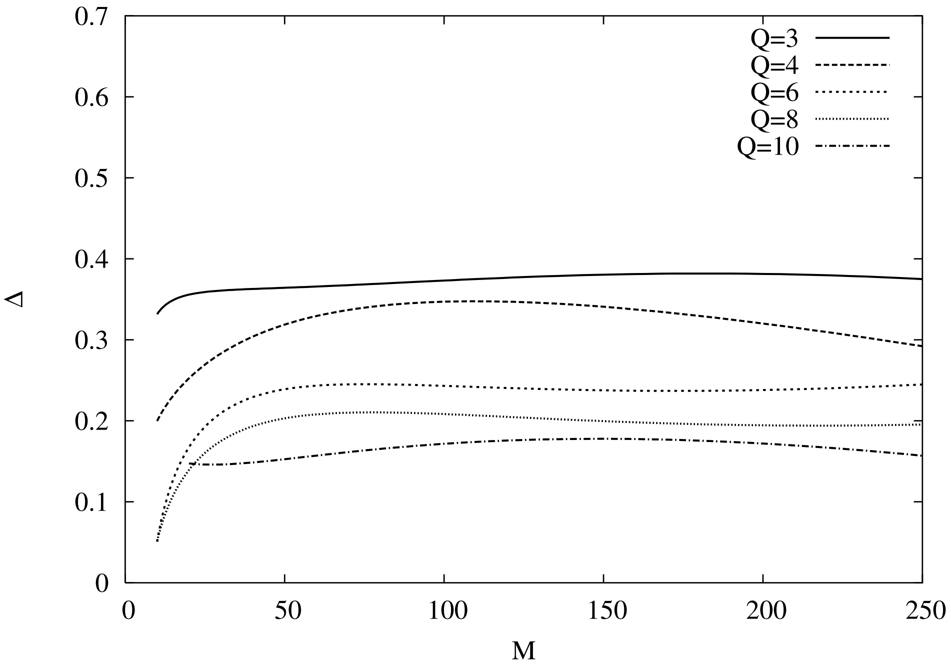
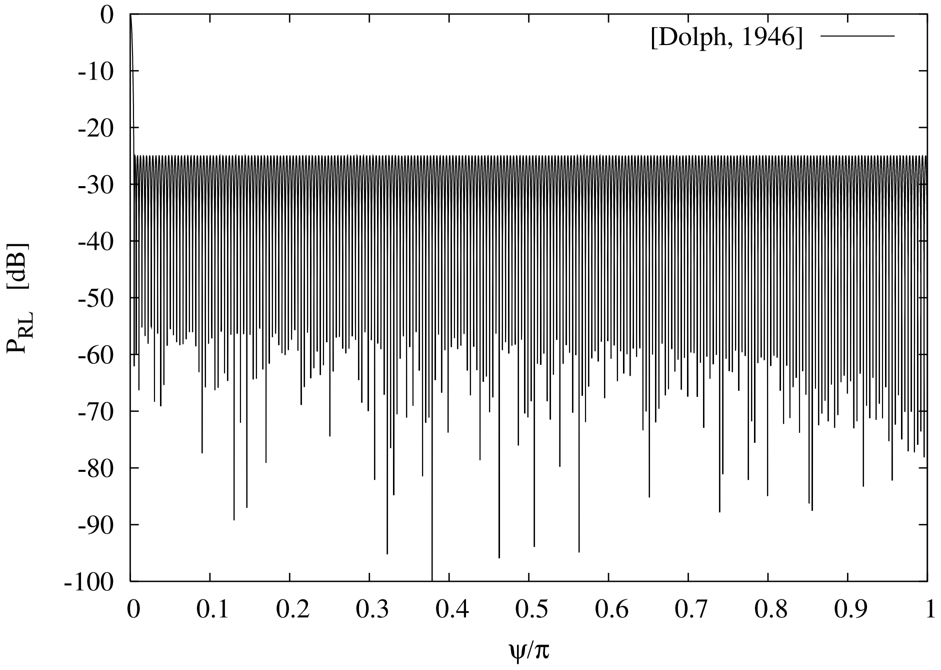
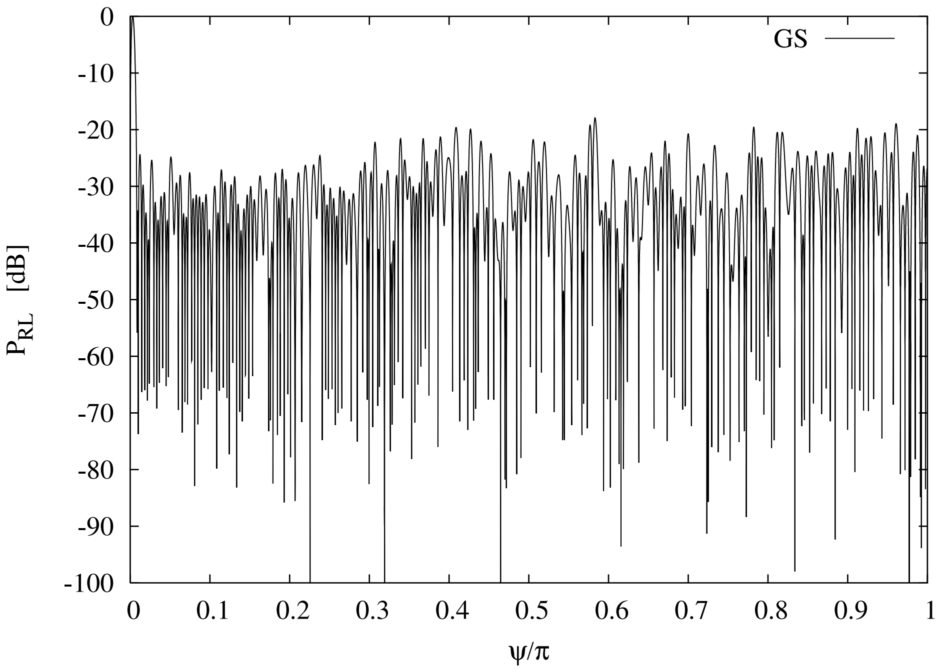


Fig. 2 - P. Rocca *et al.*, "Synthesis of large monopulse linear arrays ..."



(a)



(b)

Fig. 3 - P. Rocca *et al.*, "Synthesis of large monopulse linear arrays ..."

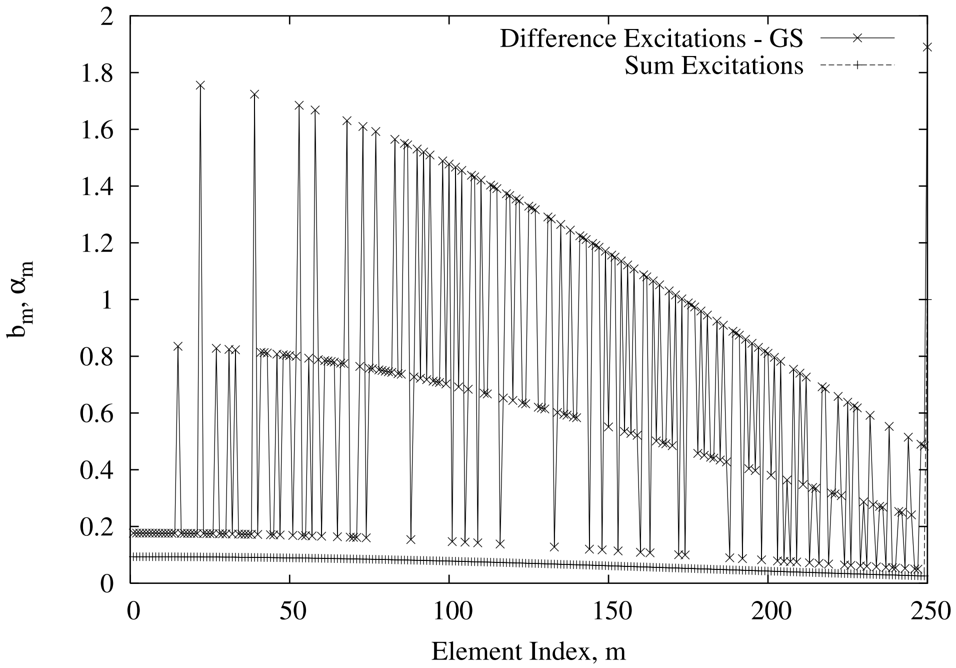


Fig. 4 - P. Rocca *et al.*, "Synthesis of large monopulse linear arrays ..."

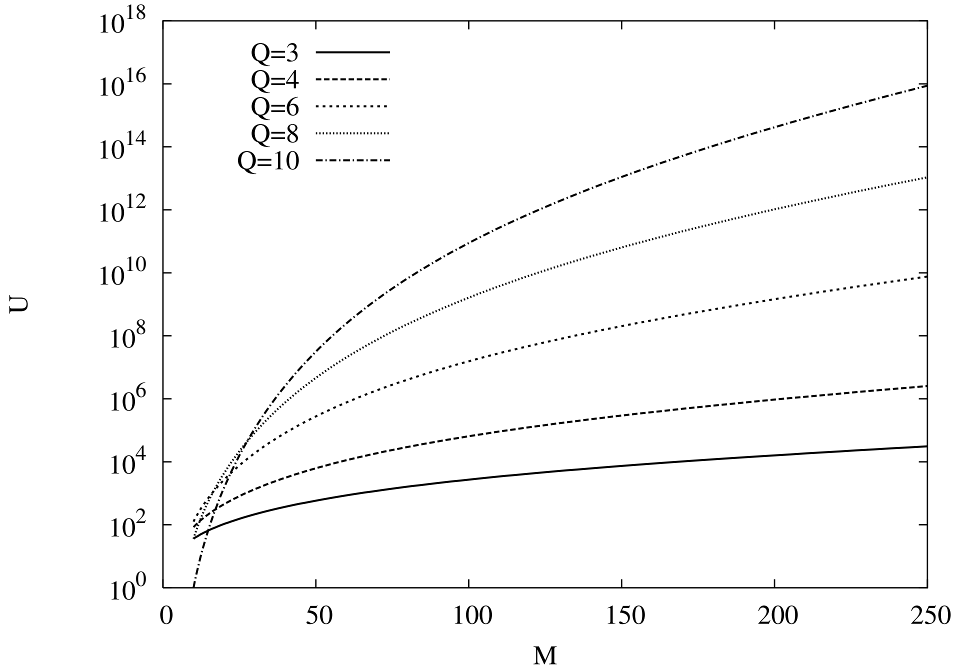


Fig. 5 - P. Rocca *et al.*, "Synthesis of large monopulse linear arrays ..."

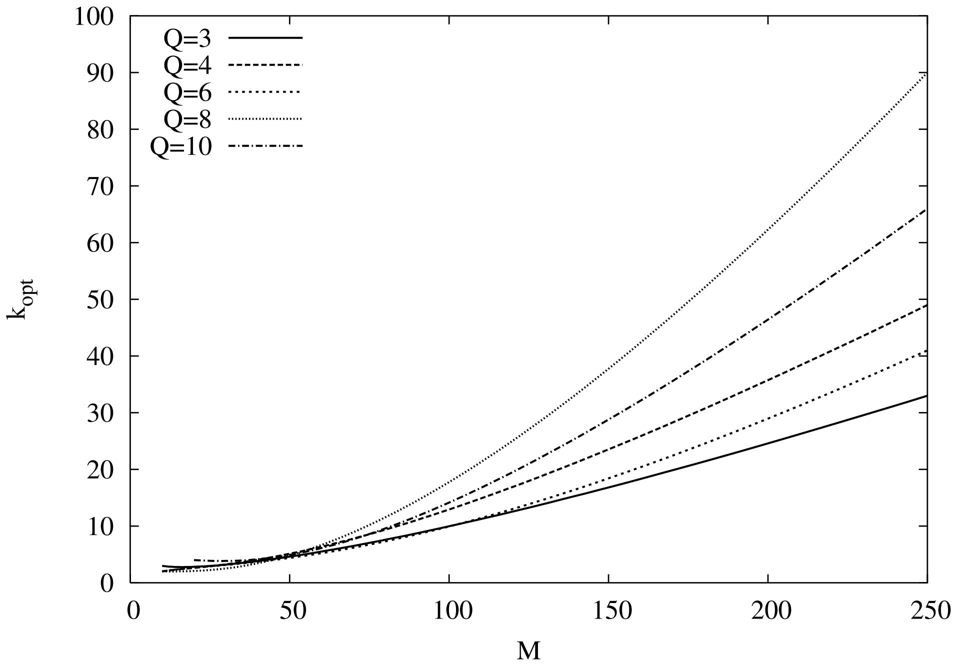


Fig. 6 - P. Rocca *et al.*, "Synthesis of large monopulse linear arrays ..."

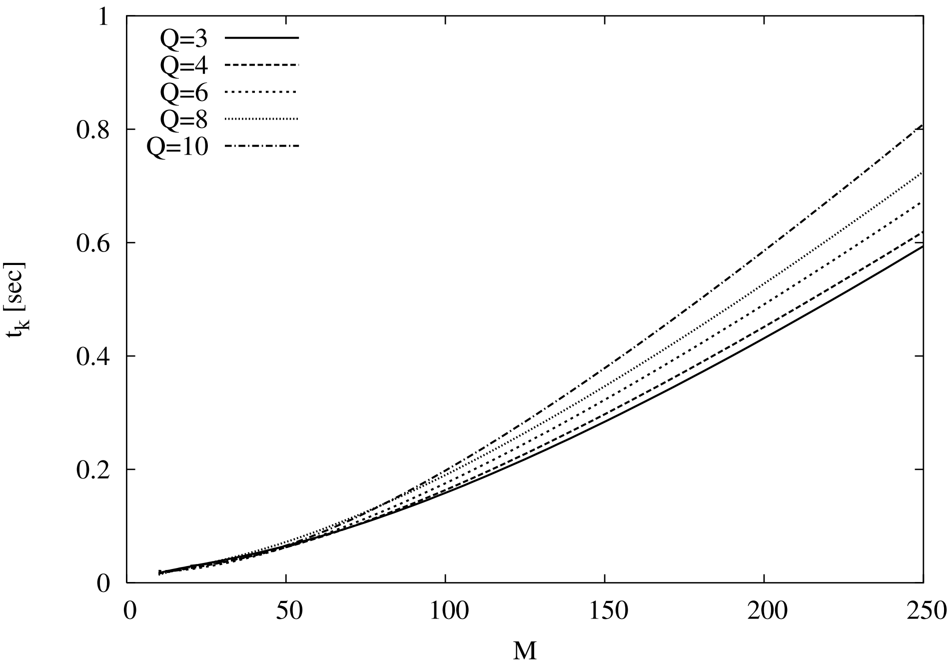


Fig. 7 - P. Rocca *et al.*, "Synthesis of large monopulse linear arrays ..."

q	1	2	3
n_q	93	105	52
w_q	1.59	7.29	14.50

Tab. I - P. Rocca *et al.*, "Synthesis of large monopulse linear arrays ..."