

ACOUSTIC ARRAY BEAMFORMING ANALYSIS THROUGH MODELING, SIMULATION & RESPONSE ESTIMATION

Moussas V.C.^{1,2}, Tsahalis D.¹, and Tsahalis H.-T.¹

¹ Paragon S.A.

Protopapadaki 19, Galatsi, GR-11147, Athens, Greece
e-mail: htsahalis@paragon.gr, web page: <http://www.paragon.gr>

² School of Technological Applications, Tech. Educ. Inst. (TEI) of Athens,
Ag. Spyridonos Str., Egaleo 12210, Athens, Greece
e-mail: vmouss@teiath.gr, web page: <http://users.teiath.gr/vmouss/>

Keywords: Acoustic arrays; beamforming; far field noise; array simulator; antenna response patterns; MatLab

Abstract. *This paper presents the development and application of a phased-array simulator software package, for the design, simulation and response evaluation of different acoustic arrays. The developed tool offers the capability to select specific or random array configuration, frequency ranges, microphones, directivity, etc, and, calculates the 2D & 3D response patterns of the array as well as performance specific metrics such as the angular resolution or the dynamic range, that are often required for comparing different array designs. The paper also addresses an assessment of the simulator, and presents some array designs and simulations to demonstrate the main capabilities of the tool.*

1 INTRODUCTION

Far-field phased array antennas are increasingly being used in almost any engineering and research field from aerospace to structural health and from medicine to geophysics. A large part of these are used in the acoustic field applying microphone arrays and beamforming techniques for sound/noise source localization. The microphone antenna array was introduced by Billingsley in 1974 and has seen dramatic improvements due to the availability of better data acquisition and computing hardware. [1]

Acoustic beamforming techniques [2] are indispensable for the localization of sound sources on moving objects, on flying aircraft, on high-speed trains, on motor cars in motion, on open rotors like helicopter and wind turbine rotors. As real applications require the ability to follow the motion of the sources, a very challenging task, they are often replaced by static source localization experimental tests, inside the test sections of open and closed wind tunnels. [3, 4]

Depending on the application in hand, a different design, configuration and parameterization of the acoustic array may be required for each case. The complex configuration of a phased array antenna presents the designer with the challenge to select the best solution among a very large set of potential ones. The antenna designer needs to know at design stage the response of each acoustic array under consideration, therefore, the support of a software simulation tool is needed that will automatically calculate the corresponding antenna array pattern & characteristics for any design parameters set under consideration.

Several tools may be found either from manufacturers of electronic equipment (Agilent TD-LTE Beamforming, Bruel & Kjaer Acoustic software, LMS Test.Lab, etc.) [5, 6] or from researchers (APAS, BeamformIt, etc.) [7, 8], but most of them are constrained with specific hardware/equipment, tailored to a specific application, or, not incorporating all the features required, thus forcing often the designer to develop its own tool.

In this paper, a versatile Acoustic Array Beamforming Analysis tool (AABA) is presented, specially designed for acoustic phased-arrays, that is intended to support the design and analysis of far-field acoustic array antennas. The tool is implemented in MatLab code [9], a matrix-oriented computer language designed for efficient numerical large-scale computation and data analysis, and it also makes use of the accompanying Phased Array System toolbox. The MatLab was selected because of its ease of use, its powerful library functions, its visualization tools and, most of all, because most signal processing researchers use it today to develop their algorithms.

The AABA tool offers accurate computation of the far field response patterns of planar phased-array

antennas, including two-dimensional (2D) and three-dimensional (3D) visualization of the acoustic response. The response patterns can be calculated in one or more planes in azimuth or elevation, for various steering angles. It may also calculate additional pattern/lobe characteristics such as dynamic range and angular resolution. The developed tool AABA was tested for a number of representative acoustic array design configurations, first as stand-alone program and then as a sub-module called by larger applications, developed for array design automation.

2 THE PHYSICAL PROBLEM

A linear sensor array is shown in the Figure 1 where the distance between adjacent sensors (microphones) is d and a sound source is at an angle φ . The response of the array $y(f)$ is the weighted sum of the responses of its microphones x_n . The simplest of all array beamforming techniques is the delay-sum beamforming, where by applying phase weights to the input channels, we can steer the main lobe of the directivity pattern to a desired direction. So, if the delay for the n^{th} sensor is given by r_n , the array output in time $y(t)$ & frequency $y(f)$ domain will be respectively:

$$y(t) = \frac{1}{N} \sum_{n=1}^N x_n(t - \tau_n), \quad \tau_n = \frac{(n-1)d \cos \phi'}{c} \quad \& \quad y(f) = \frac{1}{N} \sum_{n=1}^N x_n(f) e^{j \frac{-2\pi f}{c} (n-1)d \cos \phi'} \quad (1)$$

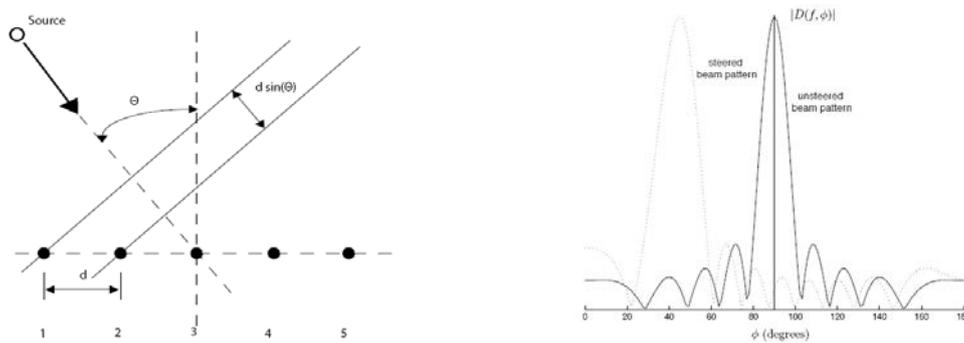


Figure 1. Sensor Array schematic, and directivity pattern

The delay-sum beamformer belongs to a more general class known as filter-sum beamformers, in which both the amplitude and phase weights are frequency dependent $w_n(f)$. In practice, most beamformers are a class of filter-sum beamformer and their general output is given as:

$$y(f) = \sum_{n=1}^N w_n(f) x_n(f) \quad (2)$$

The main advantages of using microphone arrays are: the directionality of the lobe, the reduced interference, background noise & higher SNR, and, the acoustic source localization, DOA & tracking. By combining the properties of the sensor arrays and adjusting the direction of the lobe, one can obtain the direction of the sound source (Direction of Arrival - DOA), or its localization (by combining more arrays) or even can track a moving sound source. Typical examples of sound source localization are shown in Figure 2.

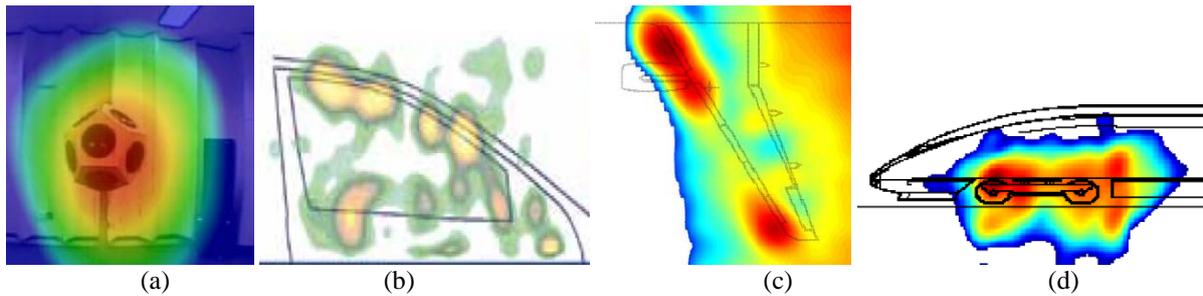


Figure 2. Sample images of sound source localization from the open literature:

a) acoustic room, b) car door, c) airplane wing, d) hi-speed train

The accuracy of the above results will depend on the performance of each acoustic array and it may be improved by selecting the appropriate parameters and a different array configuration. The selection task is not

trivial and it needs to be repeated many times to reach a better solution. In order to address this problem and facilitate/accelerate the array design task, a special tool (AABA) for calculating the performance of acoustic arrays was developed and is presented in the following sections.

3 THEACOUSTIC ARRAY BEAMFORMING ANALYSIS (AABA) TOOL DESIGN

The AABA software tool aims at helping the array designer to calculate and visualize quickly the expected performance of the acoustic array under consideration. The main requirements regarding tool inputs, outputs & code structure follow.

3.1 Tool Structure

The AABA tool is composed by two distinct sub-modules: the array Configuration module and the array Performance module. The tool structure is shown in Figure 3.

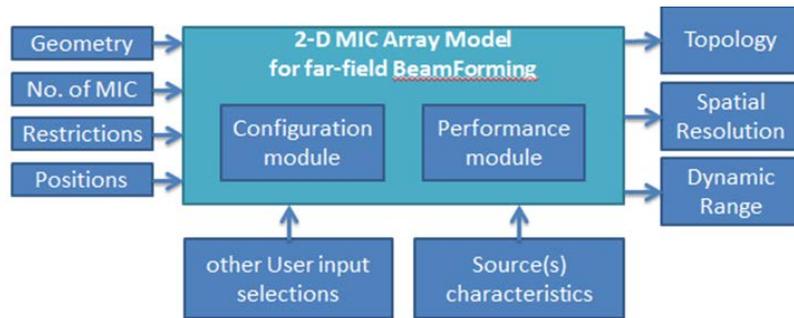


Figure 3. General AABA Tool Structure

The array Configuration module is responsible for collecting the array-related inputs and creating the topology and the configuration of the array. The array Performance module is responsible for collecting the source-related inputs and simulating the array behavior, i.e., to calculate the pattern or the metrics of the array performance. The detailed structure is shown in Figure 4.

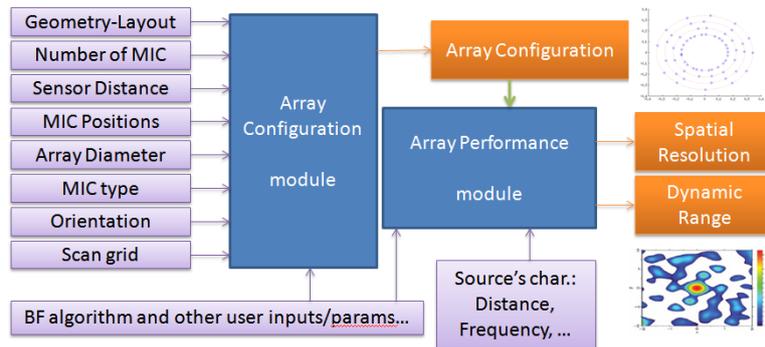


Figure 4. Detailed Module connections and main I/O

This modular structure enables AABA to be executed either as a standalone MatLab program, or, to be called by another program or procedure. These two distinct modes can also be invoked separately as two different functions, one to create an acoustic array and the other to estimate its performance. It also gives to AABA the possibility to play the role of the objective (pay-off) function, when array optimization is required and it has to be invoked automatically by other applications. The basic working flow of AABA is shown in Figure 5.

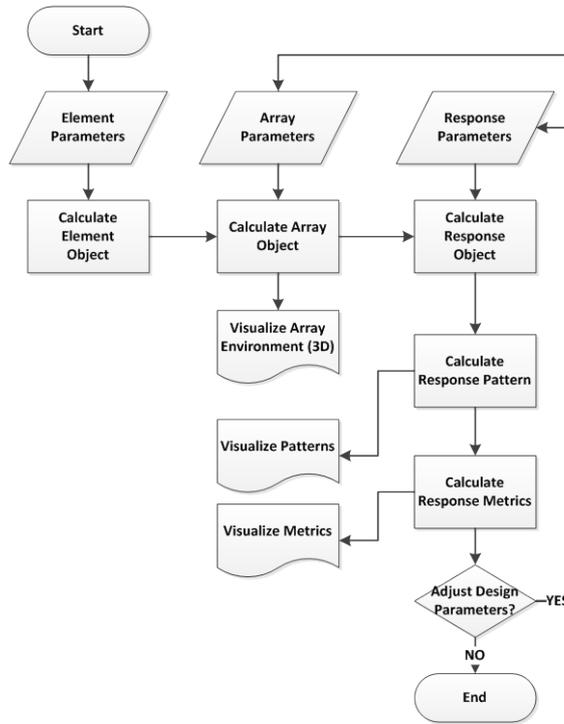


Figure 5. The AABA tool working flow

3.2 I/O Requirements

Based on the array designer’s needs, a list (Table 1) of the required inputs and returned results was compiled. The AABA tool design has incorporated these requirements to maximize its usefulness. The tool allows the designer to define linear, rectangular, circular, and conformal or arbitrary element position arrays specified by x, y, z coordinates. The designer is allowed to select between fixed or random configurations and also set the microphone type, the operating frequency or frequency range, the steering direction, and other parameters of interest.

Tool	Sub-modules	Inputs	Outputs
AABA	Configuration	Elements-MIC (pattern, orientation, number, freq.-range) Geometry (shape, size, distance, diameter) Fixed/Random positioning (if any) Restrictions (if any)	Array topology (in Global Coordinates) Array topology (in Local Coordinates)
	Performance	Array topology Source characteristics (if known: freq.-range, focus area, direction, distance, etc.)	Response Patterns (a number of different views) Response Metrics (Angular resolution, Dynamic range)

Table 1. List of Required I/O

The returned results should include the calculation of the response pattern, in one, two & three dimensions, for various directions or azimuth/elevation cuts, accompanied by some pattern characteristics/metrics of interest such as the angular resolution and dynamic range, or, an overall cost function (Table 2).

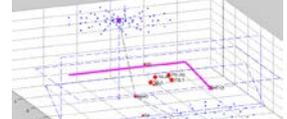
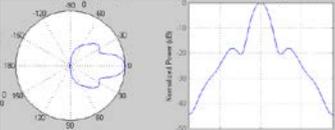
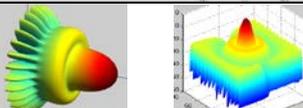
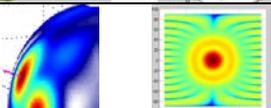
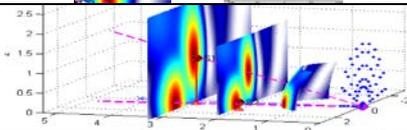
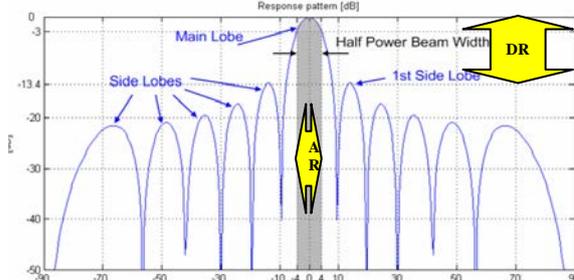
Description of the Output	Sample Outputs
3-D Representation of the area of the array in Global view (i.e., using absolute/global coordinates)	
3-D Representation of the area of the array in Local view (i.e., using local coordinates, from the array origin)	
1-D Response Pattern (polar & linear) in azimuth or elevation & various cuts	
3-D Response Pattern (spherical & plane)	
2-D Response Pattern (spherical & plane)	
Response Projections to a specific planes inside the array-source area.	
<p>Response Metrics: Dynamic Range, Ang. Resolution</p> 	<p>E.g. from the response shown at left, the textual output or the function return of the module should be:</p> <p>Dynamic Range = -13.4 (dB) Angular Resolution = 8 (°)</p>

Table 2. List of Required Outputs

4 THE AABA TOOL FUNCTIONING

For improved compatibility the tool was developed in MatLab and it makes extensive use of the Phased Array Toolbox for MatLab [9] that provides essential functions for phased array antenna systems.

4.1 Main Interfaces of AABA

The tool's **Graphical User Interface (GUI)** was developed using the MatLab GUIDE tool. The main input dialog pages are the array setup and the output setup, shown in Figure 6.

A **Command prompt** interface is also available in order to run the tool in a non-graphical environment where also the outputs are textual and graphic results are directly stored in files.

Finally a **Function version** of the AABA tool was developed to enable calls from other programs such as a design optimization tool. The function receives all the required inputs via a common 'options' variable and returns an output vector with the results (metrics or array performance indices).

A **Web Service** tool is under development in order to make the tool accessible remotely through the network as a service.

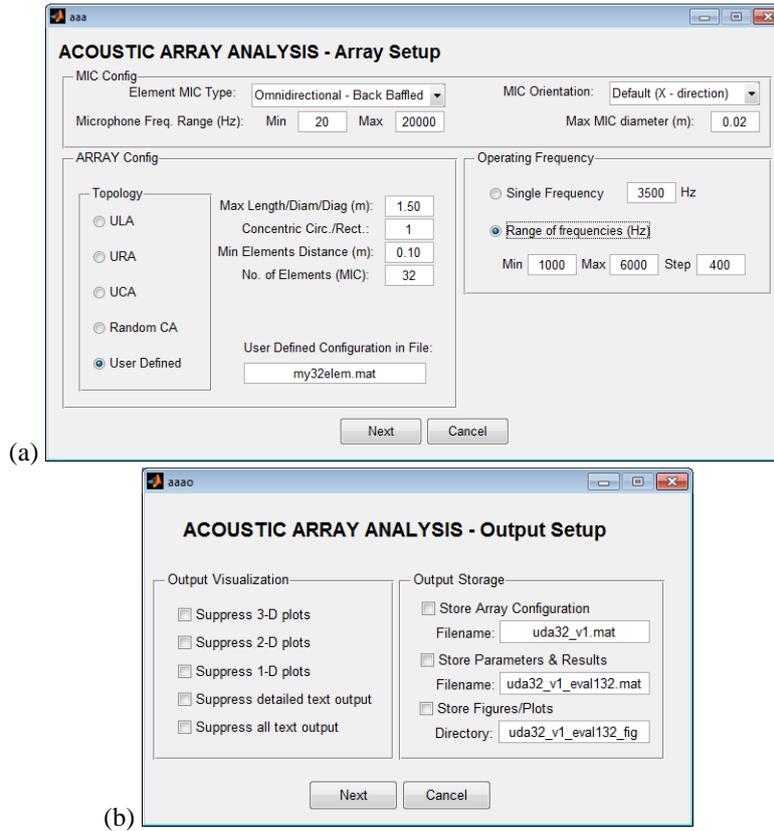


Figure 6. The AABA GUIs for: a) array setup, where the user sets the element & array information, a) output setup, where the user selects the desired outputs to view/store.

4.2 Visualization Capabilities of AABA

Visualization of the array area: The tool can visually represent the array area with any objects of interest or reference points either in global or local coordinates. It is possible to mark a focused area of interest or add several reference and projection levels in the array area. Examples are shown in Figure 7.

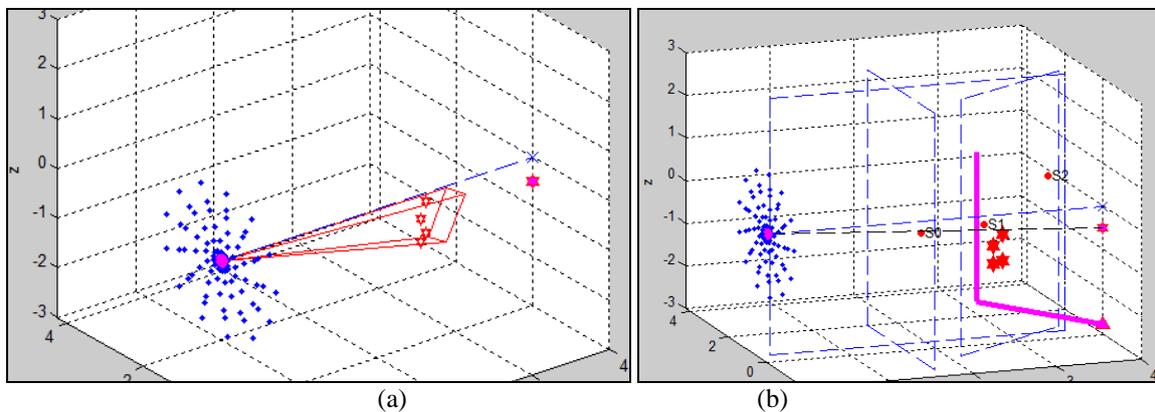


Figure 7. Array Area visualization: a) array view with the focus area for the beamforming b) array view with some sources and reference planes to project the responses

Visualization of the array response pattern: All the visualizations requested by the array designers can be implemented in the tool and they are controlled by a specific on/off flag to improve performance during extensive iterative runs. Some visualization are shown in figure 8.

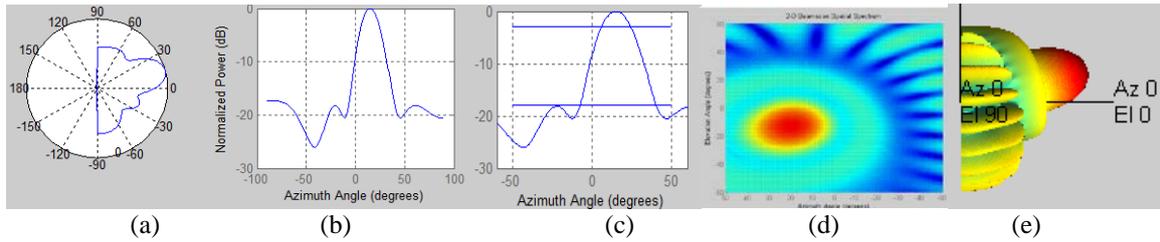


Figure 8. Array Response visualization: a) 1-D response (Azimuth cut) in polar plot, b) 1-D response (Azimuth cut) in linear plot, c) Same as (b) with metrics' reference lines, d) The array response in a 2-D plane, e) The array response in 3-D.

Projection of the array response pattern: All 2-D visualizations can be projected to a reference plane of interest. It is often useful to project a part of the response sphere at a larger distance towards the direction of the source, or even on the source surface itself. The tool can project the response results on a plane at a given distance and inclination (Figure 9).

Visualization of the array response metrics: Several metrics and performance indices can be evaluated in order to describe and quantify the array performance. These metrics are also essential for other tools such as for design optimization. They are mostly used as variables but often their visualization is required on a plot of a response pattern plot (Figure 10).

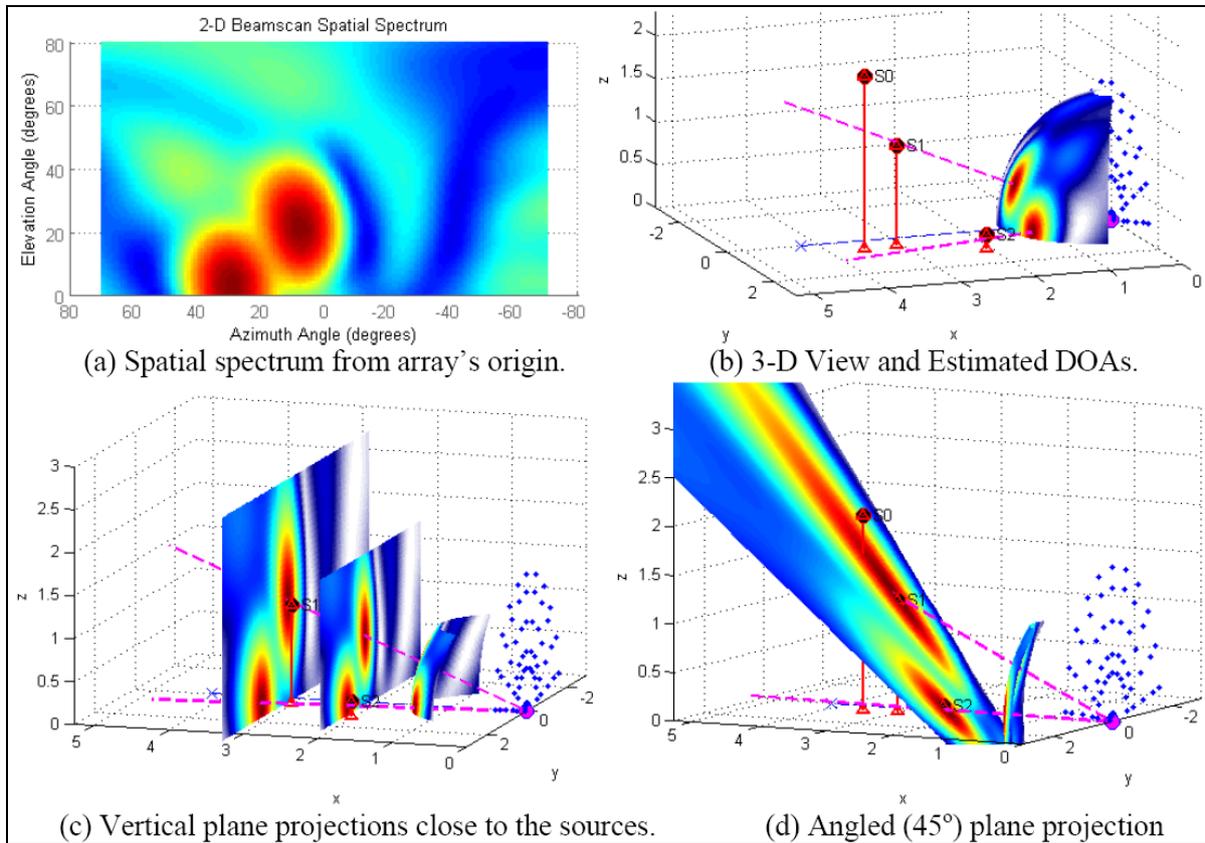


Figure 9. Array Response projections: a) the array response matrix, b) shown as a sphere of diameter 1.5m, c) projected on two vertical planes at 2-3m, d) projected on a 45° inclined plane.

4.3 Performance of AABA

The AABA tool computational and prediction performance depends mainly on the user settings and selections. The angular step, for instance, affects both accuracy and speed in an inverse manner. Simple azimuth/elevation response cuts at a specific direction are calculated very fast, but, 3-D responses for the entire sphere (360°) are more demanding. A compromise should be found by the user, when numerous iterated runs are

required.

In order to improve speed, the tool provides the option to suppress the heavier or all of the graphics and visualize only the final ones after completion. In addition, the MATLAB code used was built according to MathWorks® guidelines to enable the parallel use of processor cores, therefore, when running the AABA tool on a multicore machine, its speed may increase up to an order of magnitude.

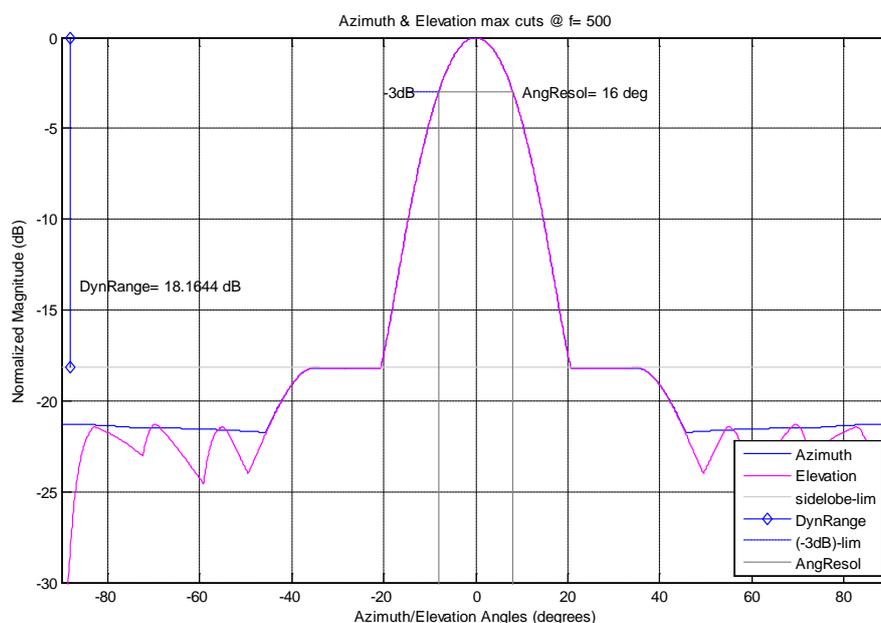


Figure 10. Array response and performance metrics: AR & DR

5 CONCLUSIONS

In this paper, the AABA tool is presented for Acoustic Array Beamforming Analysis. The tool was developed for the design, simulation and response evaluation of different acoustic arrays. The paper presented an assessment of the program, and some array simulations to demonstrate the main capabilities of AABA. The tool user interfaces, output capabilities and performance metrics were also shown. A main advantage of the tool is its additional implementation as a function in order to be called by other tools, e.g., for array design optimization, as it can play the role of a fitness function that offers the capability to select array configurations, frequency ranges, microphones number, directivity, or other parameters, and, calculate specific metrics such as angular resolution or dynamic range as required by the caller. Further development and improvement of the tool is ongoing in order to expand its capabilities and its application in other fields.

REFERENCES

- [1] Michel Ulf, (2006) "History of Acoustic Beamforming", *1st Berlin Beamforming Conference, 21-22 Nov. 2006*
- [2] B.D. Van Veen and K.M. Buckley "Beamforming – A Versatile Approach to Spatial Filtering", *IEEE ASSP Magazine*, pages 4-21, April 1988.
- [3] Lars Koop, et al., (2012) "Microphone-Array Measurements in Wind Tunnels: Challenges and Limitations", *Berlin Beamforming Conference, 2012*
- [4] Jeroen Lanslots, Filip Deblauwe and Karl Janssens "Selecting Sound Source Localization Techniques for Industrial Applications", *Sound & Vibration (www.SandV.com)*, June 2010, p. 6-9
- [5] Agilent Technologies, "Verify and Visualize Your TD-LTE Beamforming Signals", Application Note, www.agilent.com (1/a: 2013).
- [6] Bruel & Kjaer, Acoustic software – Noise source identification, www.bksv.com (1/a: 2013).
- [7] Keizer Will P.M.N., (2010) "APAS: An Advanced Phased-Array Simulator", *IEEE Antennas and Propagation Magazine*, Vol. 52, No. 2, April 2010
- [8] Xavier Anguera, Chuck Wooters and Javier Hernando, (2007) "Acoustic beamforming for speaker diarization of meetings", *IEEE Transactions on Audio, Speech and Language Processing*, September 2007, volume 15, number 7, pp.2011-2023.
- [9] MatLab R2012b, Phased Array System Toolbox, The MathWorks (2011).