International Journal of Engineering & Technology, 7 (3) (2018) 1388-1399

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET
doi: 10.14419/ijet.v7i3.14113
Research paper

Developing a graphical package for sensor arrays design, optimization and maintenance

Ahmed N. Jabbar [1] *, Ibrahim A. Murdas [1]

[1] University of Babylon, College of Engineering, Department of Electrical Engineering
**Corresponding author E-mail: Ahmed_AlJafari@yahoo.com*

Abstract

The sensor arrays are now an essential part of any communication, medical and remote sensing system. These arrays should be designed with utmost performance to ensure the maximum link efficiency. The available commercial sensor arrays design packages are expensive, complicated and cannot be easily modified to accommodate the users' needs. This work suggests a solution that is to design an open source specialized application to serve the ever-changing needs of the users. This package is called Sensor Array Design and Optimization (SADO) and it is developed to allow the unexperienced users and the researchers to design, test and optimize their sensor arrays using effi-cient optimization algorithms. The optimization is trying to reduce the sidelobe levels to reduce the interference. The application is simple and friendly to use, with professional graphical results. The predesigned arrays configurations supplied with this package are uniform and random arrays. The built in optimization algorithms are: Artificial Bee Colony (ABC), Biogeography-Based Optimization (BBO) and Teaching Learning Based Optimization (TLBO). The results for various designs and optimization results are also given and compared to indicate the best settings for the user. Some optimization ratios might reach about 50% that represent -3dB reduction in sidelobe level.

Keywords: *5G Technology; Uniform Sensor Arrays; Random Sensor Arrays; Evolutionary Algorithms; Sidelobes Reduction; Matlab Guide; Digitally Controlled Arrays.*

1. Introduction

The following sections provide the necessary theoretical knowledge related to the arrays configurations and analysis.

1.1. Uniformly distributed sensor array (UDSA)

The sensor array could be defined as a collection of sensors grouped together using a certain configuration such that they can interact with each other. This interaction is called the Array Factor (*AF*). The *AF* is simply the Fourier transform of the sampling window because each sensor will sample the incoming signal depending on its location [1], [2]. Therefore, the arrays are sometimes called spatial filters. To understand and derive the necessary mathematical models for the *AF*, consider the one dimensional sensor array shown in Fig. 1. a. The incident signal can be resolved into a vertical component that can be discarded and a parallel effective component propagates along the *x*-axis. The parallel component can be collected from the sensors and then directed to the next stage. The one-dimensional array can be extended into a planar array as shown in Fig. 1. b. In the case of planar array, the signal direction is measured by the azimuth and elevation angles θ and φ respectively. Assume that there are *M* elements along the *x*-axis and *N* elements along the *y*-axis then; the collected signal is as shown in Eq. 1. Equation 1 shows that the planar *AF* is the product of the AF_x and AF_y that represent the one dimensional *AF* along the *x*-axis and *y*-axis respectively [3 - 5].

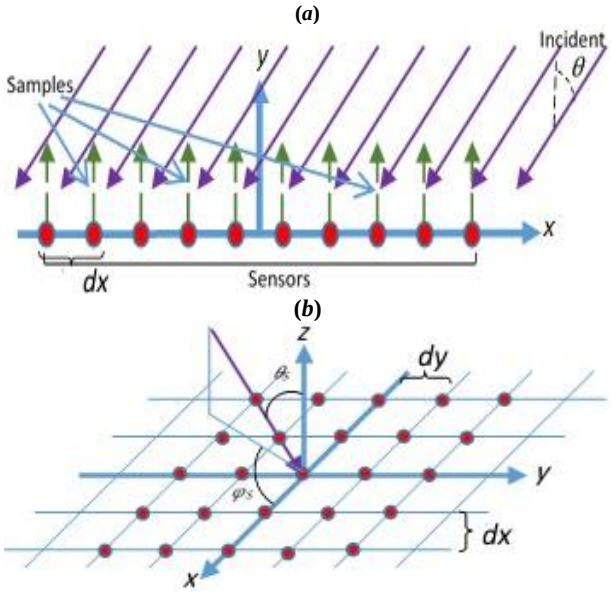


Fig. 1: The Sensor Array Types: (a) One Dimensional Array, (a) Planar Array.

$$AF = \sum_{m=1}^{M} w_m \exp\left(-jkd_m\left(\sin(\theta)\cos(\varphi)-\sin(\theta_s)\cos(\varphi_s)\right)\right)\cdot \sum_{n=1}^{N} w_n \exp\left(-jkd_n\left(\sin(\theta)\cos(\varphi)-\sin(\theta_s)\cos(\varphi_s)\right)\right) \quad (1)$$

Where

θ_s, φ_s : are the source elevation and azimuth angles.

Copyright © 2018 Ahmed N. Jabbar, Ibrahim A. Murdas. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

d_m, d_n: are the spacing along the x and y axis respectively.

k: is the propagation constant $\frac{2\pi}{\lambda}$, λ: the wavelength.

If the array is excited with an impulse function, Eq. 1 becomes

$$AF = \frac{\sin\left(d_x \Psi_x/2\right)}{\sin\left(\Psi_x/2\right)} \frac{\sin\left(d_y \Psi_y/2\right)}{\sin\left(\Psi_y/2\right)} \tag{2}$$

Where

$$\Psi_x = kd_m\left(\sin(\theta)\cos(\varphi) - \sin(\theta_S)\cos(\varphi_S)\right)$$
$$\Psi_y = kd_n\left(\sin(\theta)\cos(\varphi) - \sin(\theta_S)\cos(\varphi_S)\right)$$

Equation 2 is a 2D sinc function and it is analogous to Fourier transform of a 2D uniform window. Therefore, solving Eq. 2 to find its roots yields the locations of the nulls that identify the sidelobes locations. To avoid the grating lobes, d_x and d_y are set to 0.5λ [6].
The Uniformly Distributed Sensor Arrays (UDSA) are the basic and the simplest arrangement for the sensors in an aperture, therefore, it was widely used in the early days of the sensor arrays. Nevertheless, the UDSA are tuned to a one frequency only because all the elements' spacing are calculated with respect to a single λ. Hence, any change in the frequency will degrade the array response. Furthermore, it is susceptible to any element loss [7, 8]. For these reason, UDSA are not adequate for harsh environments nor unreachable places, as they need constant maintenance and/or replacement.

1. 2. Non-uniform or randomly DSA (RDSA)

To overcome the limitations of UDSA, another type of arrays was developed where the elements are not placed at equidistance. These new generations of arrays provided multi frequency tuning and robust performance. However, this came with a price, which is the analysis of the arrays became no longer analytic but numerical [9 - 11]. Equation1 can be modified for the RDSA such that

$$\tilde{AF} = \sum_{m=1}^{M} w_m \exp\left(-jk\tilde{d}_m\left(\sin(\theta)\cos(\varphi) - \sin(\theta_S)\cos(\varphi_S)\right)\right) \cdot \sum_{n=1}^{N} w_n \exp\left(-jk\tilde{d}_n\left(\sin(\theta)\cos(\varphi) - \sin(\theta_S)\cos(\varphi_S)\right)\right) \tag{3}$$

Where the symbol ~ means a random variable. The only method to analyze such type of arrays is by using the computer simulation or practical measurement because there are infinite number of arrangements depending on the random location distributions.
The RDSA can be categorize according to the type of their elements' spacing into two types that are, completely random arrays RDSA where every element has a unique distance from the origin and Semi-random arrays SRDSA where some elements have equal distance from the origin and some are not.

2. The need for optimization

When the array is designed and manufactured, it is highly unlikely to be considered optimized due to manufacturing errors and equipment defects. Therefore, the array needs to be recalibrated to insure optimum performance. Even after the recalibration, the aging and deterioration in the array materials will deviate its response from the optimum point. Performing optimization is carried out inside high-tech laboratories with expensive equipment and requires high level of training. Some of these arrays are installed out of man's reach hence; they cannot be taken for repair and recalibration. Despite that, the arrays have a very powerful feature comes from the usage of the phase shifters that are connected to every array element as shown in Fig. 2. The AF can be shaped and modified, as needed, by changing the phase shifters weights matrix [12 - 14].

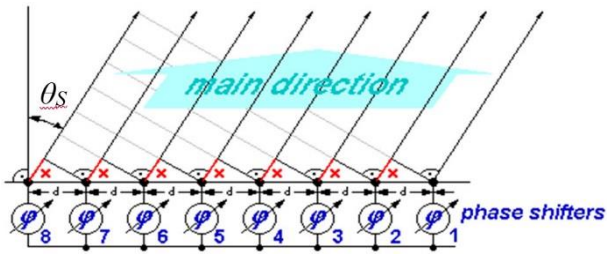


Fig. 2: The Array Phase Shifters.

The shifter values are derived from solving Eq. 1 at every element location. This yields a complex matrix called the shifters weights matrix [15]. When the weights matrix values are changed, the main beam will be steered towards the source at θ_S, and φ_S. In addition, we can solve Eq. 1 for different coordinate values for d_{xi} and d_{yi} tricking the array to think it has new element distributions rather than the actual distribution.

3. Evolutionary optimization algorithms (EOA)

The complicated and diverse configuration of the arrays require very powerful optimization algorithms. The EOA is the latest trend in the field of optimization, which tries to exploit the accumulated intelligence of the organic entities. These entities are driven by the need of survival hence; they have to adapt to fit in their environment. This collective intelligence inspired the scholars to develop the EOA. Consequently, if we assume the solutions are the organic individuals and the fitness function is the environment that decides which one is to continue and which one is to be eliminated depending of their fitness, then the last survivor is the best solution as it was able to modify and evolve to fit itself as the best as possible [16].
The EOA essentially go through three main steps, which are, generate random potential solutions, find which one is the closest to the solution, eliminate the failed solutions and upgrade the remaining for the next round. The EOA algorithms differ in the elimination and upgrade steps. Some of them follow complicated procedures by adding more layers of intelligence to their algorithms to be qualified to solve complicated objective functions. Others follow simple procedures to provide speed. The best of these is the one that joins the speed and intelligence during its operation.

3.1. The artificial bee colony (ABC)

This ABC is trying to mimic the bee behavior during foraging and search for honey. Thus, the bees are divided into forgers and lookers. The forgers are responsible for collecting the honey while the lookers are assessing the amount of nectar and trying to find new sources. The nectar sources are the solution and the amount of nectar represents the closeness to the best solution. Hence, if the source is rich with nectar then higher number of forgers are gathered around it. The process is described as follows, assume that there are n workers at every nectar source i such that $X_i = \left[x_{i,1} \quad x_{i,2} \quad \cdots \quad x_{i,n}\right]$. Each $x_{i,j}$ is a potential solution to the fitness function. The lookers keep watching the forgers to estimate the amount of nectar and decides to look for a new solution as given by Eq. 4 [17 - 19]

$$S_{i,k} = X_{i,k} + \Theta \times \left(X_{i,k} - X_{j,k}\right) \tag{4}$$

Where $X_{j,k}$ is a randomly selected solution ($i \neq j$) and k is a randomly selected nth worker and Θ is a random number within [−1 to 1]. When S_i is better than X_i then replace the new solution with the older one. The lookers choose a source randomly to estimate the amount of nectar using the roulette wheel probability, which is

$$P_i = \frac{Cst_i}{\sum_j Cst_j} \tag{5}$$

Where
Cst_i represents the value of the fitness for that solution.
The discard is based on the amount of nectar at each source. If the amount is not increasing then it will be abandoned using the following equation

$$X_{i,k} = lb_i + \text{rand}(\min,\max)\times(ub_i - lb_j) \quad (6)$$

The rand function is governed by the min and max range provided by the user. The *ub* and *lb* are the upper and lower boundaries of the solution dimensions respectively.

3.2. Biogeography-based optimization (BBO)

The BBO was motivated by the appearance and disappearance of multi species along the history of earth. It describes the migration, inhabitation, flourishing and extinct of the living animals based on their geographic surrounding conditions. Consider Fig. 3,

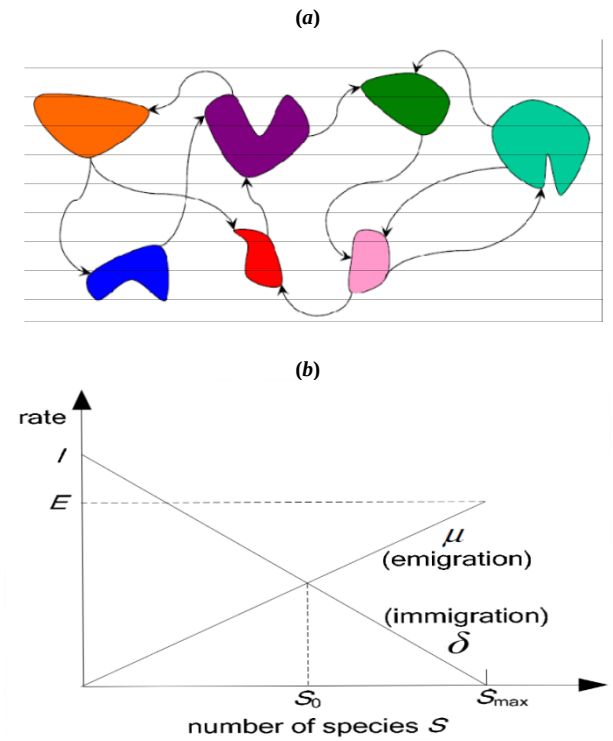


Fig. 3: (*a*) Adjacent Islands, (*b*) The Model of Migration to an Island.

Each of these islands in Fig. 3. *a* is habitats with a Habitat Suitability Index (HSI) which is an independent variable and Suitability Index Variable (SIV) that depends on HSI. The HSI defines the maximum number of species welcomed by that habitat. The lower the index the lesser the number of species. Therefore, each island has its own maximum number of species S_{max}. Now generate a number of species with their own probability to immigrate or emigrate δ and μ respectively. When a new habitat is discovered, the species immigrate with maximum rate due to the abundant resources. The rate will decrease over time and the species are now emigrating from it. Assume the maximum immigration and emigration rates are *I* and *E* respectively. The equilibrium point S_0 is reached when *I*=*E* as shown in Fig. 3. *b*. Let n=S_{max} then [20], [21].

$$\mu = E\frac{S}{n} \quad \text{and} \quad \delta = I\left(1-\frac{S}{n}\right) \quad (7)$$

When S_0 is reached, $E=I=\delta+\mu$. To apply the BBO to any objective function, generate a number of solutions (habitats) and each solution has its own SIV according to its closeness to the solution. Generate species with their own δ and μ then distribute them along the habitats. When the HSI is lower than the number of existing species, the species start to emigrate according to their μ or when the HSI is higher than the species, the habitat will accept the new comers. These habitats are changing with time such that they are either improving or degrading according to the following update equation

$$\text{SIV}_{i,k}^{\text{new}} = \text{SIV}_{i,k} + \alpha\left(\text{SIV}_{j,k} - \text{SIV}_{i,k}\right) \quad (8)$$

The new SIV value depends on the improvement at the kth period and *j*th solution and α is a random variable defined by the user. During the history of the habitat, it will either prosper, approaching the final solution, or destroyed and abandoned due to natural disaster.

3.3. Teaching learning based optimization (TLBO)

The TLBO is derived from the behavior of human academic society. This algorithm is divided into two phases that are Teacher Phase (TP) and Learner Phase (LP). The TP depends on the abilities of the teacher and his/her scientific level to graduate highly educated learners with high grades. Each teacher is initialized by a factor called the Teaching Factor TF as follows

$$\text{TF} = \text{round}\left(1-\text{rand}\left(0,v\right)\right) \quad (9)$$

Where v is any value assumed by the user. At any iteration *i*, assume that there are *m* subjects, *n* learners and $M_{i,j}$ means (averages) at the *i*th iteration and *j*th subject (*j*=1, 2, …, *m*). The best student *kbest* represents the student with the highest grades (*k*=1, 2… *n*). The difference in the mean can be calculated as in Eq. (10) [22]

$$DF_{j,k,i} = r_i\left(X_{j,kbest,i} - \text{TF}\times M_{j,i}\right) \quad (10)$$

Where $X_{j,kbest,i}$ is the results of the best learner in subject *j* and the r_i is a random variable in the range of [0, 1]. Depending on *DF* value, the learners will shift gradually towards the best learner among all of them. Based on *DF*, the TP is updated as follows

$$X'_{j,k,i} = X_{j,k,i} + DF_{j,k,i} \quad (11)$$

When TP is over, the LP starts by selecting random learners and allow them to exchange knowledge. For example, select two students *P* and *Q* with different averages and let them share knowledge. The update in the *P* average is

$$\begin{aligned} X''_{j,P,i} &= X'_{j,P,i} + r_i\left(X'_{j,P,i} - X_{j,Q,i}\right) \quad if\ X'_{total-P,i} < X'_{total-Q,i} \\ X''_{j,P,i} &= X'_{j,P,i} + r_i\left(X'_{j,Q,i} - X_{j,P,i}\right) \quad if\ X'_{total-Q,i} < X'_{total-P,i} \end{aligned} \quad (12)$$

The new average $X''_{j,P,i}$ is accepted only if it gave better results. At the end of the LP, the students go back to their classes with new improved skills. The students represent possible solutions to the objective function and their skills is the closeness to the best solution. The constant update in the average will transfer the population towards the solution at the best class. Other solutions will be discarded, as their classes will remain empty.

4. The settings of the sensor array design and optimization (SADO)

The SADO is developed using the Matlab-GUIDE because it can operate on different platforms and supports parallel processing [23]. The SADO is an open source that makes it adequate for upgrading and development by users and programmers. The graphics of the results are displayed using the open-GL that is equipped with Matlab [24]. The design criteria of SADO is simplicity, accuracy and versatility. The software comes with pre-programmed array generating algorithms and three powerful EOA (ABC, BBO and TLBO). The idea behind the SADO was inspired by the imminent

jump in the communication technology designated for the 5G technology. The 5G intends to implement massive MIMO to provide parallel channels to increase the data rate transmission. These MIMO are placed close to each others such that they can be considered as an array. Besides, the 5G plans to exploit the available users' resources to route the data through their personal device.

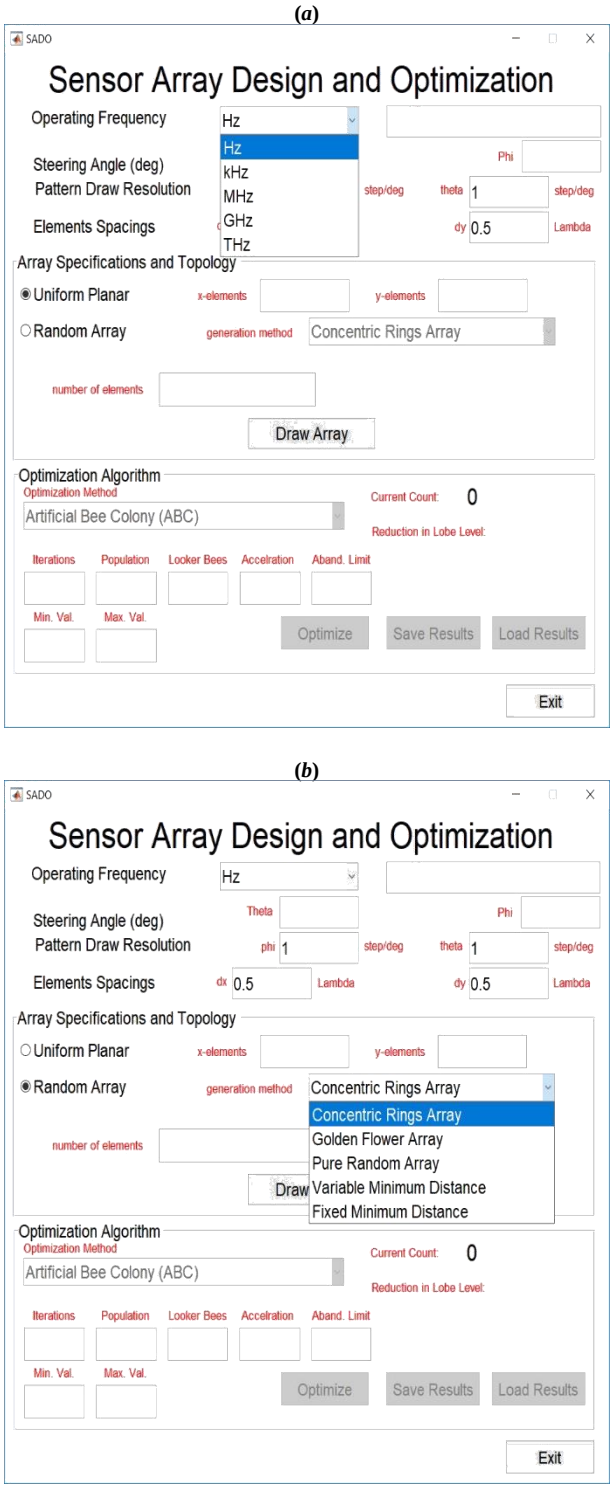


Fig. 4: SADO Main Settings: (*a*) Frequency, (*b*) RDSA, (*c*) EOA, (*d*) EOA Iterations Counter.

Hence, each device, in the future, will be equipped with an array of antennas [25 - 27]. When this happened, most of the users would not have the necessary trainings to operate their arrays optimally. Therefore, we present SADO as a solution or as part of it. The interface of SADO is very simple as shown in Fig. 4. When the SADO is launched, its default setting is set to design a uniform array. The user selects the operating frequency of the array using the frequency scale popup menu in SADO ranging from Hz to THz to calculate λ, see Fig. 4. *a*. After selecting the scale, the user types the exact operating frequency in the edit box. If the user skipped or accidentally typed a wrong entry, the SADO will alert the user, deletes the erroneous entry and return back to the location were the user should provide the required value. Then, the user selects the resolution for the *AF* testing sensor.

To explain this point, assume that there is an array and it is required to find the shape of its *AF* as given in Fig. 5. In Fig. 5, the number

of elements is M along the x-axis and N along the y-axis. Each element is separated from its adjacent elements by a distance d_x and d_y in the x and y directions respectively. The RF sensor will rotate around the array in on a hemisphere ($0 \le \theta_S \le 180°$, $0 \le \varphi_S \le 360$) recording the field intensity at every point with the coordinates θ_s^i and φ_S^i. Assume the radiated field form any element is $E_{m,n}$ then the total field at the sensor is:

$$E_T = \sum_{m=1}^{M}\sum_{n=1}^{N} E_0 \exp\left(-jkd_{m,n}\left(\sin(\theta)\cos(\varphi) - \sin(\theta_s^i)\cos(\varphi_S^i)\right)\right) \quad (13)$$

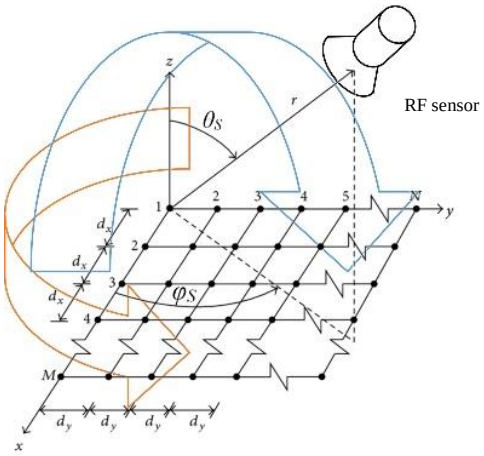


Fig. 5: Recording the Radiated Field of the Sensor Array.

After recording E_T at position i, the sensor will move to the next position i+1. The difference between these two positions (i, i+1) is called the sensor resolution. When the difference is small, then this means that the number of steps is high and vice versa.

Usually, the companies and RF Labs. use electric field sensor placed at the far field from the array to record the intensity of radiation [28], [29]. The labs walls should be coated with anti-reflection materials. The sensor's motion is controlled using high precision stepper motors. Hence, such arrangements are beyond the ordinary user to provide and use. In our package this process is simulated to provide a no-cost simple solution for the users. The resolution in SADO resembles this operation and the values entered in the Pattern Draw Resolution represent the number of steps between the adjacent degrees (180° for elevation and 360° for azimuth). Setting the value of resolution to 2, for example, yields (360 steps in the elevation and 720 in the azimuth). Hence, caution should be taken as the generated matrix size will increase exponentially. Then, the user should provide the angles of the sources (θ_S and φ_S). The user can choose the configuration of the array that is either UDSA or RDSA. The RDSA selection gives the user versatile choices for either random or semi random arrays from the popup menu as shown in Fig. 4. b. After completing the basic settings; the user choice is UDSA, then the user should provide the number of elements along the x and y axes or when the user choses RDSA then the number of element of the array elements should be provided directly. Then after, the user clicks the Draw Array button to see the *AF* of the designed array according to the settings.

When the SADO is launched, the optimization section is disabled. After completing the design, it will be enabled letting the user to optimize the array. The popup menu contains three EOA as shown in Fig. 4. *c*. The edit boxes are changed interactively on the same panel according to the parameters of the selected EOA. The optimization section is also provided with a counter indicator to allow the user to see the current iteration and the ratio of reduction in the sidelobe levels as shown in Fig. 4. *d*.

The buttons (Save Results and Load Results) can be used for either upgrading or maintaining the array. After the design and optimization steps are completed, the results can be saved to a text file for deployment or send to the remote control console to upgrade the array. In modern digital arrays, the phase shifters are connected to a DSP control unit as shown in Fig. 6.

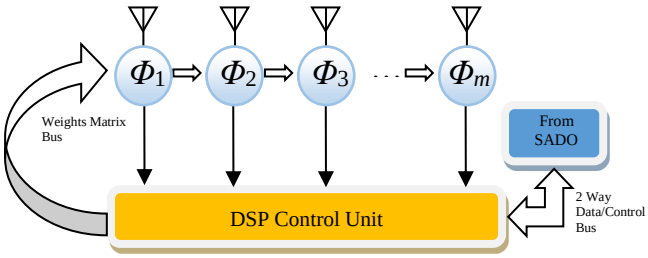


Fig. 6: DSP Controlled Array.

The SADO will send or store the newly calculated weight values on the DSP and the DSP will upgrade the phase shifters (Φ) values. The maintenance can be performed by saving the incoming data from the array and comparing them with the previously stored values. If they coincide then the array is working as designed otherwise an element(s) are defected.

5. SADO tests and results

The SADO is tested using an array with 256 elements operating at 12 GHz with θ_S=70° and φ_S=80°. This array is generated using the UDSA, SRDSA using concentric rings and RDSA using the min/max distance algorithm. The min/max distance is an algorithm used to distribute the elements randomly such that the distance between two adjacent elements falls within the min/max range measured with respect to λ. The optimization procedure searches for the sidelobe peaks and then averages the peaks level as in Eq. (13), which represents the fitness function

$$\min(P_{av}) = \min\left(\frac{\sum_{i=1}^{N_{Lo}} Lo_i}{N_{Lo}}\right) \quad (14)$$

Where

Lo: is the peak level.

N_{Lo}: is the number of peaks.

The optimization is trying to minimize P_{av} to improve the array response. The average lobe level to the original average lobe level is displayed on SADO GUI. The tests include the time required to complete the optimization and it is used to calculate the relative speed for every EOA. The relative speed is calculated as in Eq. (15)

$$S_i = \frac{T_i}{T_{\max}} \quad (15)$$

Where

T_i: is the time taken by the current EOA.

$T_{\max}$: is the time required by the slowest EOA.

The interfaces of SADO for each test with the relative speed to complete the optimization are shown in Table 1. The design and optimization results are tabulated in Table 2. The red dots in the array configurations represents the optimized physical locations for the array under design or the weights matrix values that should be sent to the phase shifters.

6. Conclusions

From this work, we conclude that the arrays need a constant maintenance. Hence, simple and effective tools are required to perform this task. The need for the arrays is rising with the need for more efficient communication links. Henceforth, such simple tools will become a pressing necessity in the near future. The SADO can aid to solve this problem. Naturally, the user needs a simple, clear and accurate interface to be encouraged to use the application and that was the pivot of the design for the application. Despite the EOA time delays, they have enough flexibility to cope with any objective function. The BBO has the best time to reach the solution with

almost the best performance. Therefore, it can be recommended as the first choice to optimize the sensor arrays. In addition, the results of the array configuration for the BBO shows clearly the emigration out of the bounding region. The optimization results can trick the array by changing the phase shifter circuits' values to think it has a different physical configuration. This is a very important feature for the array. This feature also provide a mean to repair a partially defected array without the need for replacement. Clearly, the designed and manufactured arrays require additional optimization step. The UDSA require less tweaking than the RDSA, but, overall, each configuration can be improved even further. The SDRA could be the best choice to design the sensor arrays because they combine the best features of the UDSA and RDSA and the BBO has the fastest rate with almost the best results. Deploying the application as an open source gives a strong chance for it to be updated, upgraded and evolved according to the users' experience.

References

[1] C. Balanis, *Antenna Theory Analysis and Design*, John Wiley & Sons, Inc., Third Edition, USA, 2005.

[2] R. Haupt, *Antenna Arrays a Computational Approach*, John Wiley & Sons, USA, 2010. https://doi.org/10.1002/9780470937464.

[3] J. Hilbertsson and J. Magnusson, *Simulation and Evaluation of an Active Electrically Scanned Array (AESA) in Simulink®*, Chalmers University of Technology, 2009.

[4] H. Cher, *Two-Way Pattern Design for Distributed Subarray Antennas*, Master of Science in Engineering Science (Electrical Engineering), Naval Postgraduate School, Monterey, California, 2012.

[5] V. Madisetti Editor in Chief, Wireless, *Networking, Radar, Sensor Array Processing, and Nonlinear Signal Processing*, CRC Press, USA, 2010.

[6] A. Brown, Electronically Scanned Arrays MATLAB® Modeling and Simulation, CRC Press, USA, 2012. https://doi.org/10.1201/b12044.

[7] A. N. Jabbar, "New Elements Concentrated Planar Fractal Antenna Arrays for Celestial Surveillance and Wireless Communications", *ETRI Journal*, Volume 33, Number 6, December 2011, pp. 849–856. https://doi.org/10.4218/etrij.11.0111.0036.

[8] P. Gorman, *Wideband Aperiodic Antenna Array Design with CMA-ES*, M. Sc. Thesis in Electrical Engineering, Pennsylvania State University, 2012.

[9] A. O'Donnell and R. McGwier, "Review of Modern Thinned Array Methods for Optimizing Randomly Scattered Elements", *United States National Committee of URSI National Radio Science Meeting (USNC-URSI NRSM)*, Boulder, CO, 2018, pp. 1–2.

[10] H. Jung and I.-H. Lee, "Secrecy Rate of Analog Collaborative Beamforming with Virtual Antenna Array Following Spatial Random Distributions", *IEEE Wireless Communications Letters*, 2018. https://doi.org/10.1109/LWC.2018.2804389.

[11] H. Jung and I. H. Lee, "Analog Cooperative Beamforming with Spherically-Bound Random Arrays for Physical-Layer Secure Communications," *IEEE Communications Letters*, Vol. 22, No. 3, March 2018, pp. 546–549. https://doi.org/10.1109/LCOMM.2017.2782807.

[12] J. Benveniste, *Design and Development of a Single Channel RSNS Direction Finder*, M. Sc. in Electrical Engineering, Naval Postgraduate School, Monterey, California, 2009.

[13] J. C. Chen, "Low-Complexity Constant Envelope Precoding Using Finite Resolution Phase Shifters for Multiuser MIMO Systems with Large Antenna Arrays," *IEEE Transactions on Vehicular Technology*, (Early Access), 2018.

[14] Y. Xiong and G. Wang, "An X-Band 6-Bits Highly-Accurate Digital-Stepped Phase Shifter MMIC for Phased Array System," *3rd IEEE International Conference on Computer and Communications (ICCC)*, Chengdu, 2017, pp. 826–829.

[15] T. Biedka, *Analysis and Development of Blind Adaptive Beamforming Algorithms*, Ph. D. Thesis in Electrical Engineering, Blacksburg, Virginia, 2001.

[16] J. Brownlee, *Clever Algorithms Nature-Inspired Programming Recipes*, Lulu.com, 2011.

[17] Home Page, https://abc.erciyes.edu.tr/.

[18] R. Parpinelli, C. Benitez and H. Lopes, *Parallel Approaches for the Artificial Bee Colony Algorithm Handbook of Swarm Intelligence: Concepts, Principles and Applications*, Series: Adaptation, Learning, and Optimization; Springer; 2011, pp: 329–346, Berlin, Germany.

[19] Y. Boudouaoui, H. Habbi and F. Harfouchi; *Swarm Bee Colony Optimization for Heat Exchanger Distributed Dynamics Approximation With Application to Leak Detection*, Handbook of Research on Emergent Applications of Optimization Algorithms, IGI Global, 2017, pp: 557–578, , Hershey, PA, USA.

[20] D. Simon, "Biogeography-Based Optimization", *IEEE Transactions on Evolutionary Computation*, Vol. 12, No. 6, 2008, pp: 702– 713. https://doi.org/10.1109/TEVC.2008.919004.

[21] H. Ma and D. Simon, *Evolutionary Computation with Biogeography-based Optimization Vol. 8*, Wiley & Sons, USA, 2017. https://doi.org/10.1002/9781119136507.

[22] R. Rao, *Teaching Learning Based Optimization Algorithm and Its Engineering Applications*, Springer, USA, 2016. https://doi.org/10.1007/978-3-319-22732-0.

[23] Parallel Computing Toolbox™ User's Guide, The MathWorks, Inc., USA, 2018.

[24] MATLAB® Graphics, the MathWorks, Inc., USA, 2018.

[25] Jameel, F. et. al., "Massive MIMO: A Survey of Recent Advances, Research Issues and Future Directions", *International Symposium on Recent Advances in Electrical Engineering (RAEE)*, 2017. https://doi.org/10.1109/RAEE.2017.8246040.

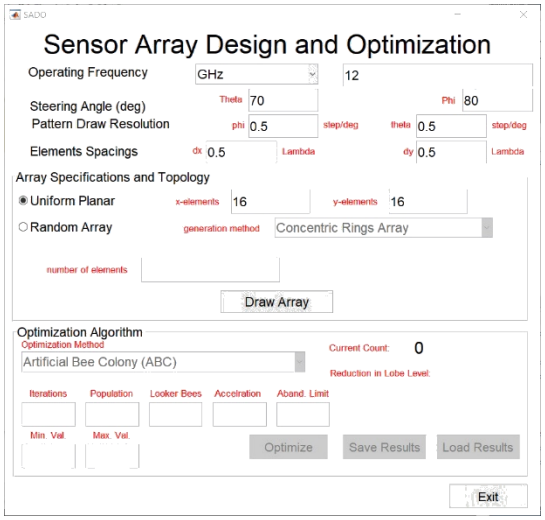
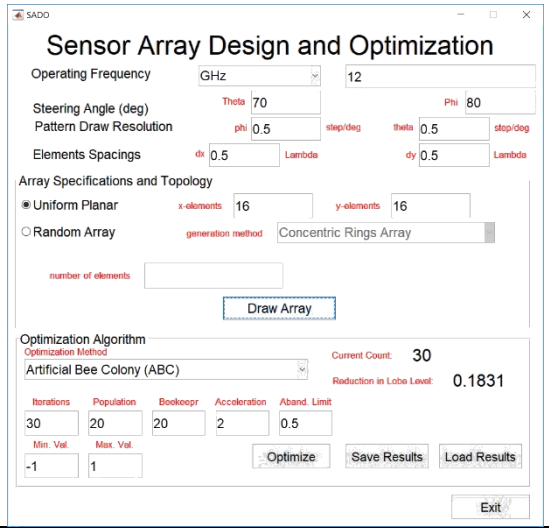
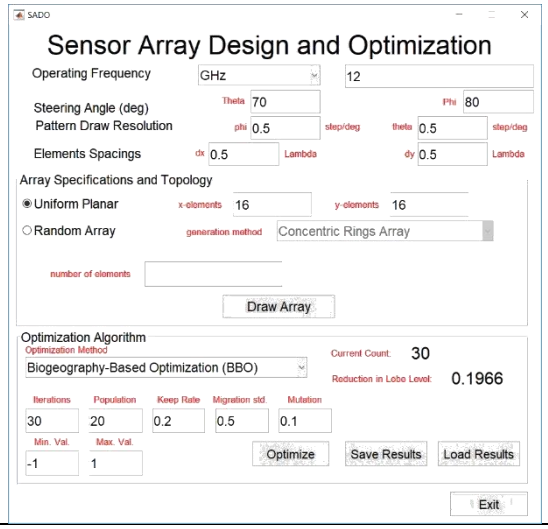
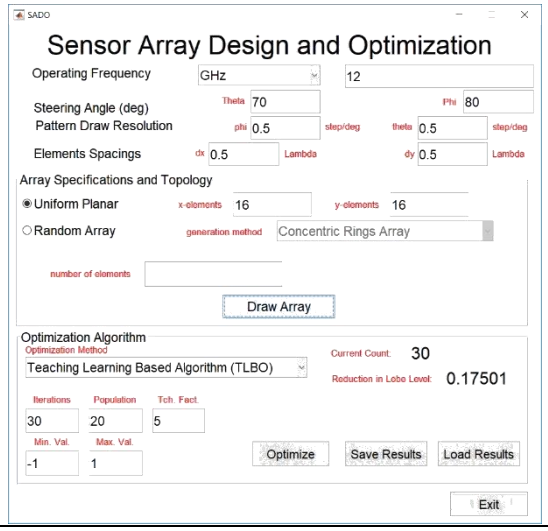
[26] Ancansa, G. et. al., "Spectrum Considerations for 5G Mobile Communication Systems", *Procedia Computer Science 104*, Elsevier, 2017, pp. 509 – 516. https://doi.org/10.1016/j.procs.2017.01.166.

[27] *IMT traffic estimates for the years 2020 to 2030 IMT*. Report M.2370-0; 2015. p. 10–14.

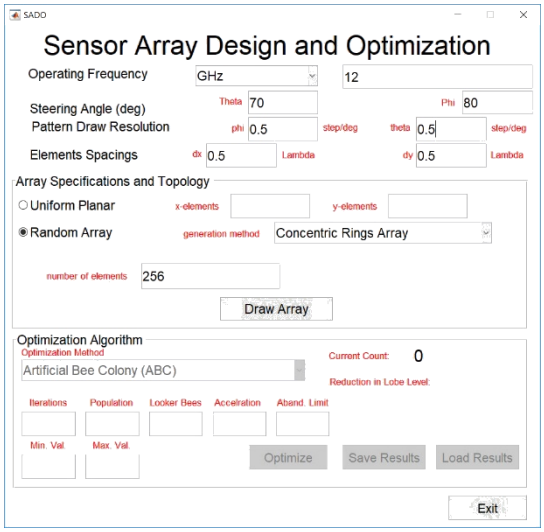
[28] S. Orfanidis, *Electromagnetic Waves and Antennas*, Rutgers University. http://www.ece.rutgers.edu/~orfanidi/ewa/, 2014.

[29] M. Bansal, *Digital Control Board for Phased Array Antenna Beam Steering In Adaptive Communication Applications*, M. Sc. in Electrical Engineering, California Polytechnic State University, USA, 2013.

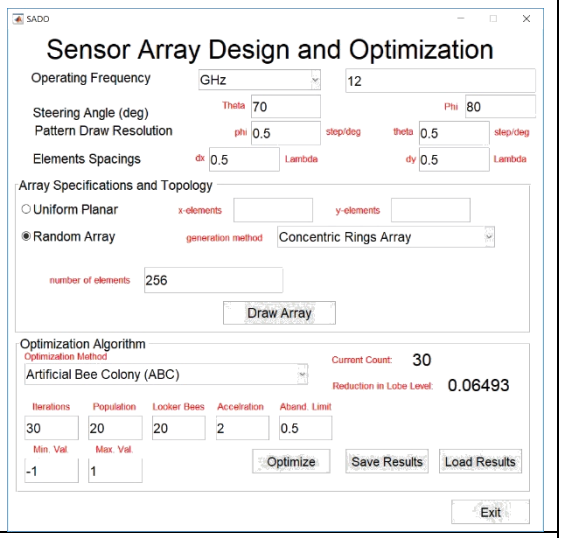
Table 1: The SADO Interface Settings and the Relative Speed of the EOA

UDSA Original Design	EOA Results	
	ABC Results: Relative Speed=1.052 Sidelobe Level Reduction=18.31%	
	BBO Results: Relative Speed=2.095 Sidelobe Level Reduction=19.66%	
	TLBO Results: Sidelobe Level Reduction=17.501% Relative Speed=1	

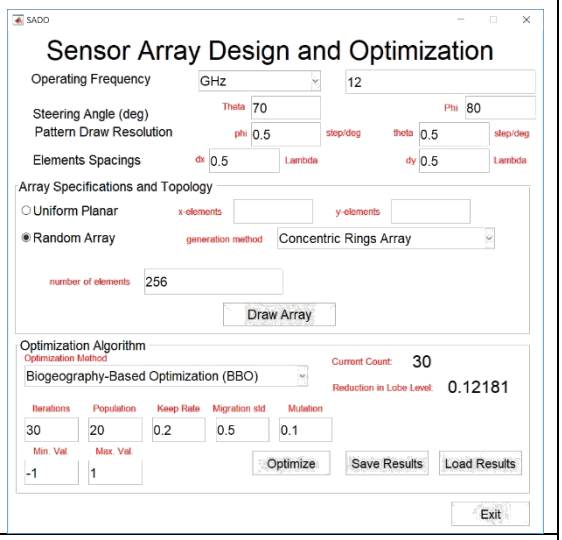
SRDSA (Concentric Ring Array)	EOA Results



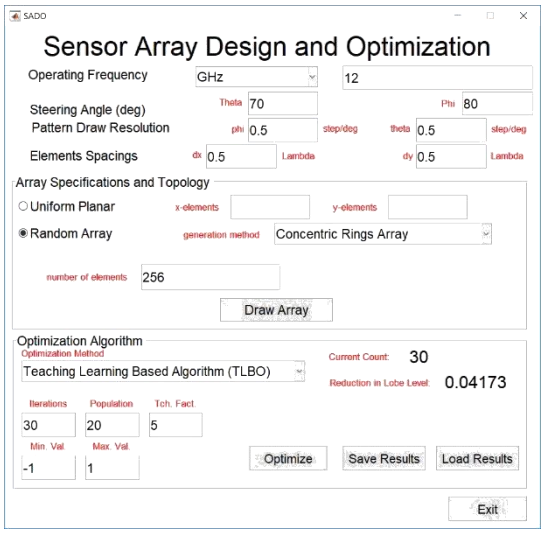
ABC Results: Relative Speed=1.001
Sidelobe Level Reduction=6.493%



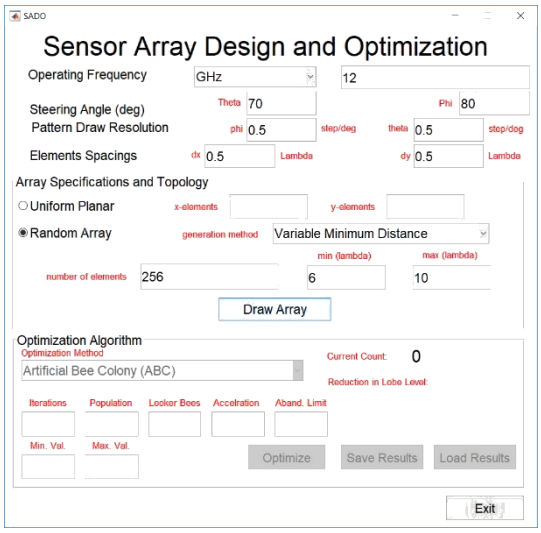
BBO Results: Relative Speed= 1.974
Sidelobe Level Reduction=12.181%



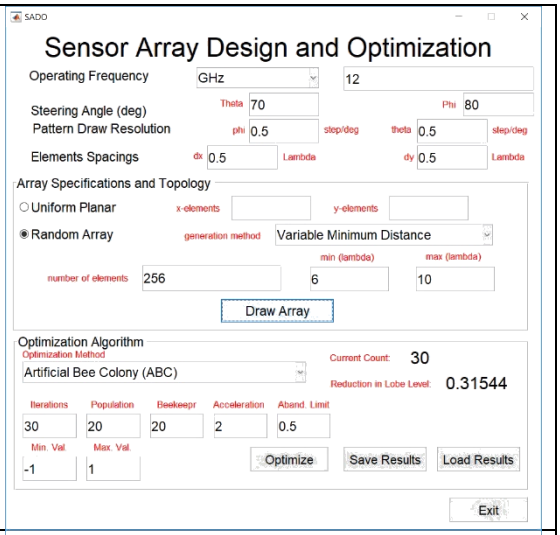
TLBO Results: Relative Speed=1
Sidelobe Level Reduction=4.173%



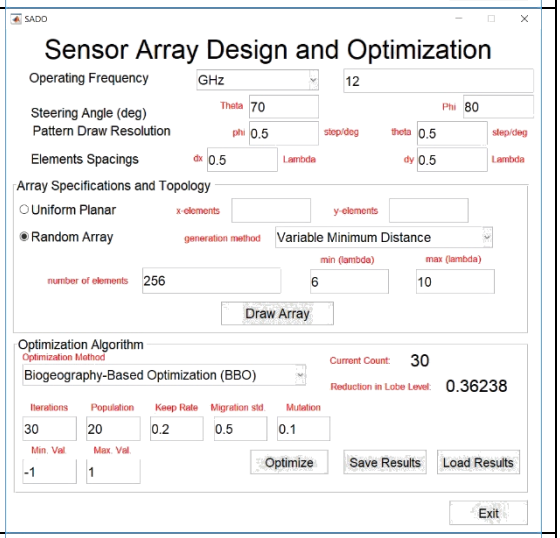
RDSA Min/Max Distance	EOA Results



ABC Results: Relative Speed=1
Sidelobe Level Reduction=31.544%



BBO Results: Relative Speed= 2.225
Sidelobe Level Reduction=36.238%



TLBO Results: Relative Speed=1.123
Sidelobe Level Reduction=51.262%

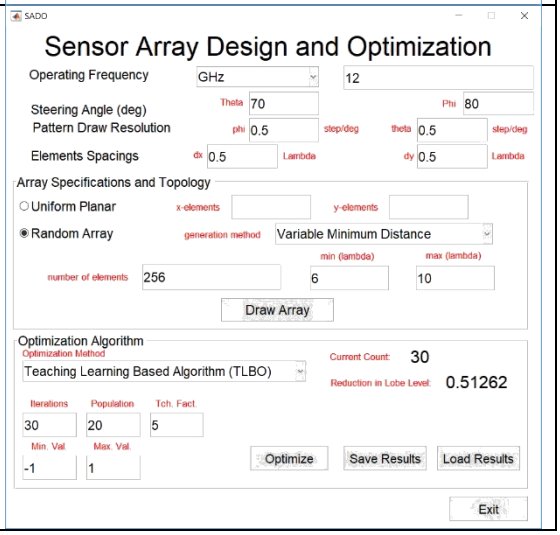
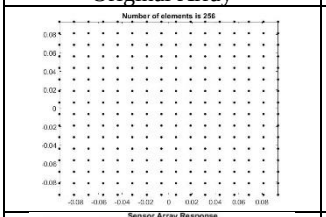
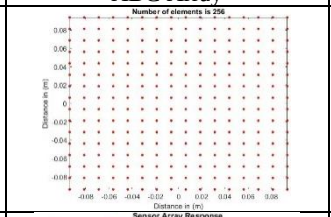
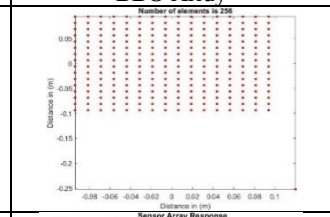
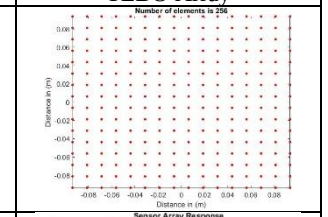
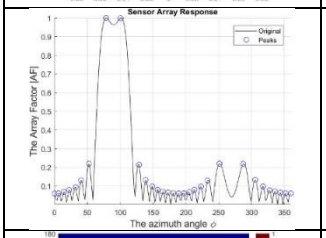
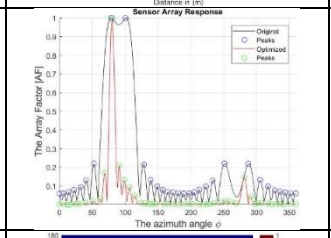
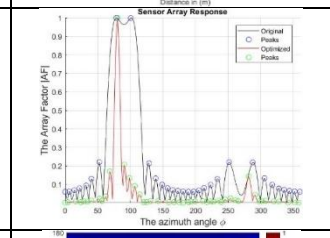
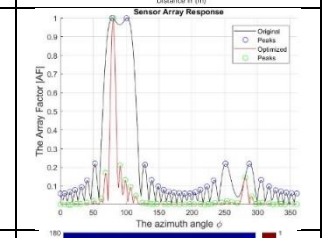
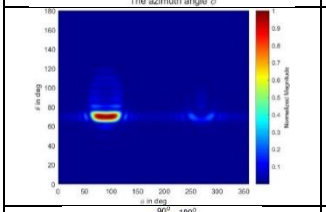
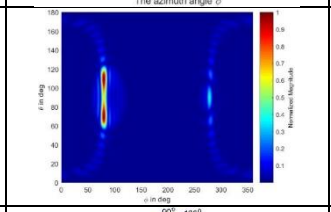
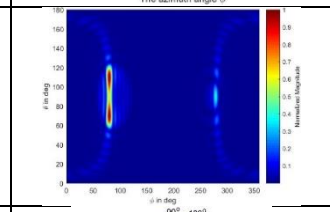
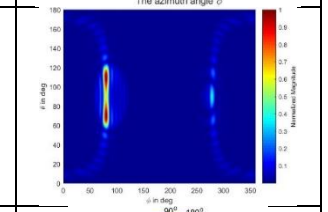
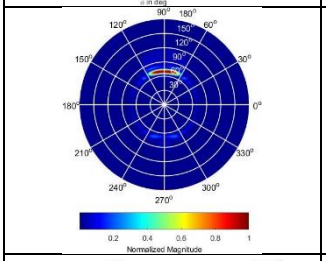
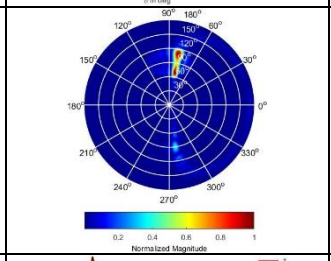
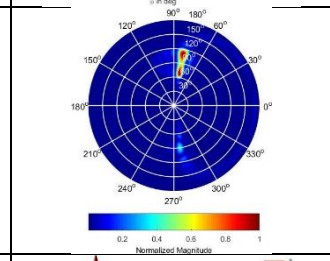
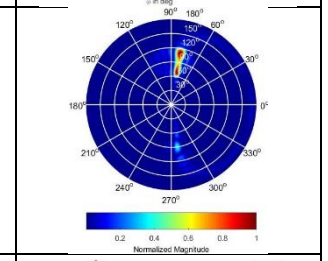
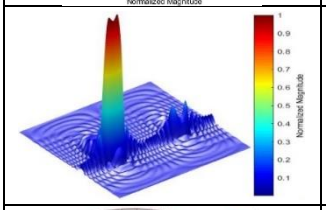
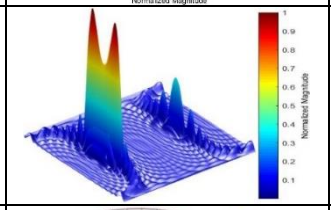
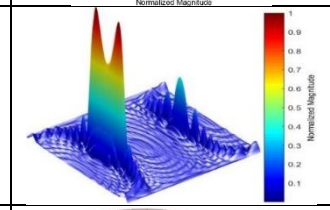
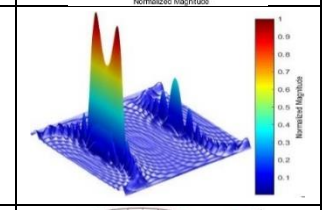
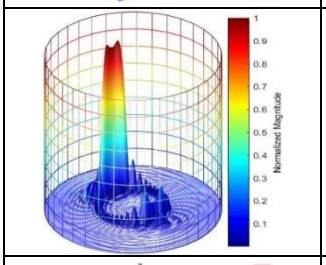
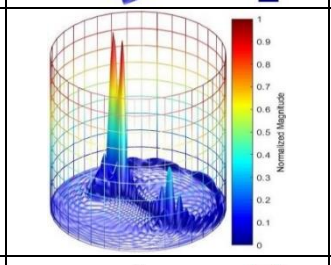
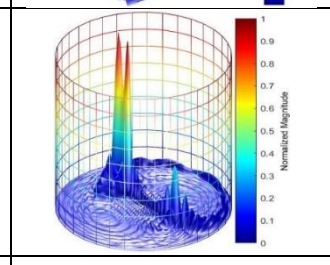
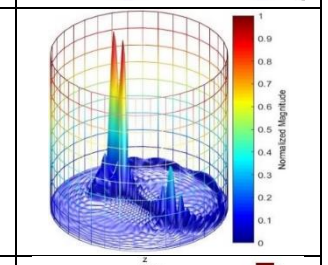
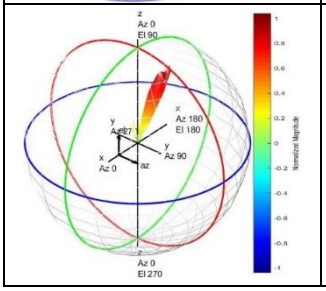
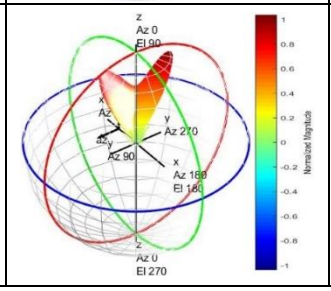
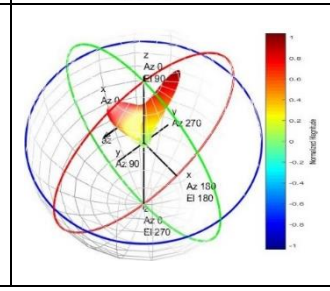
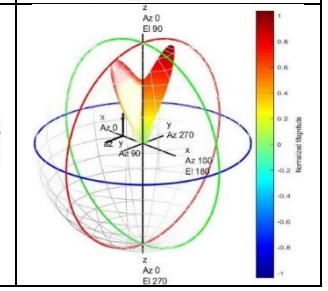
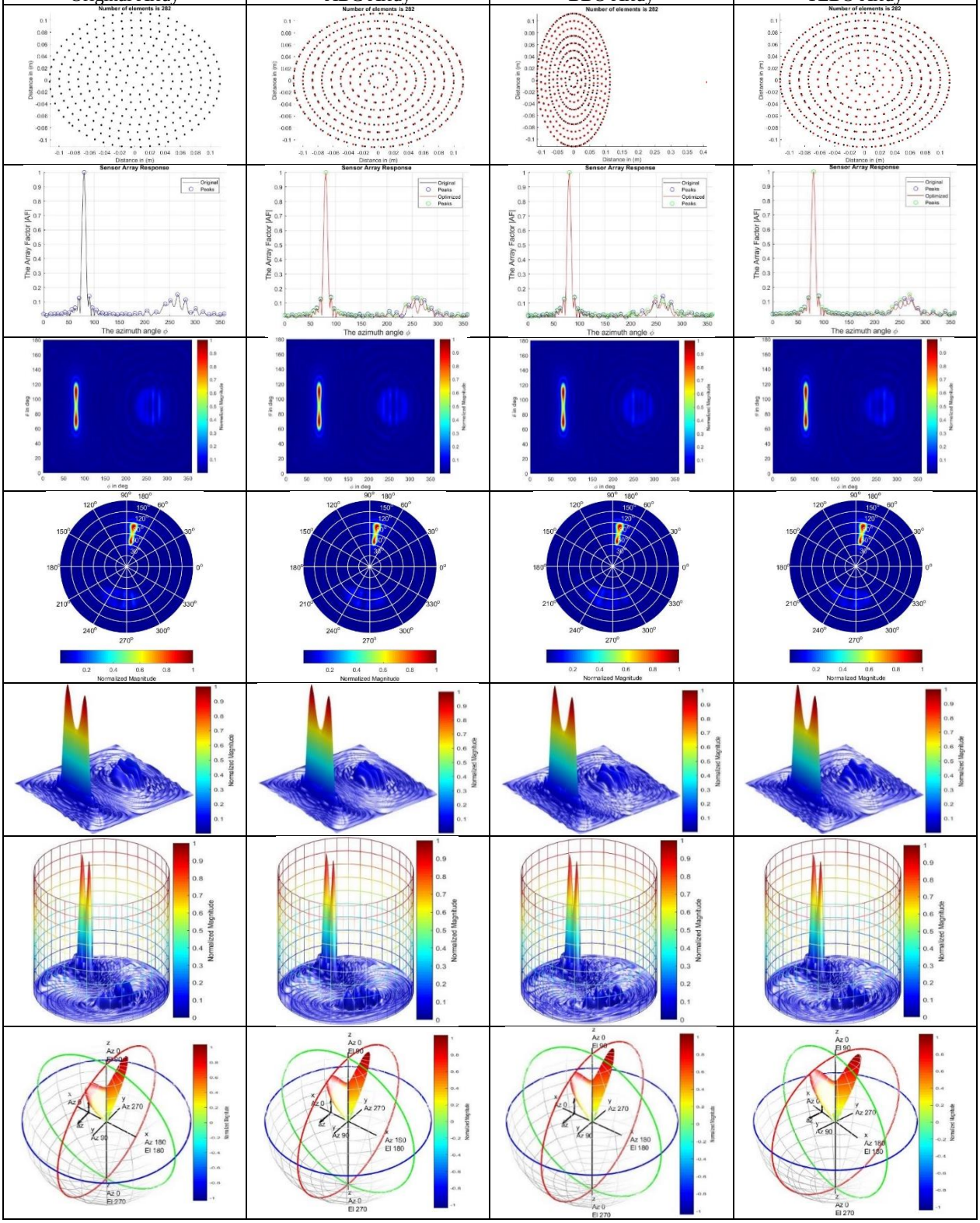


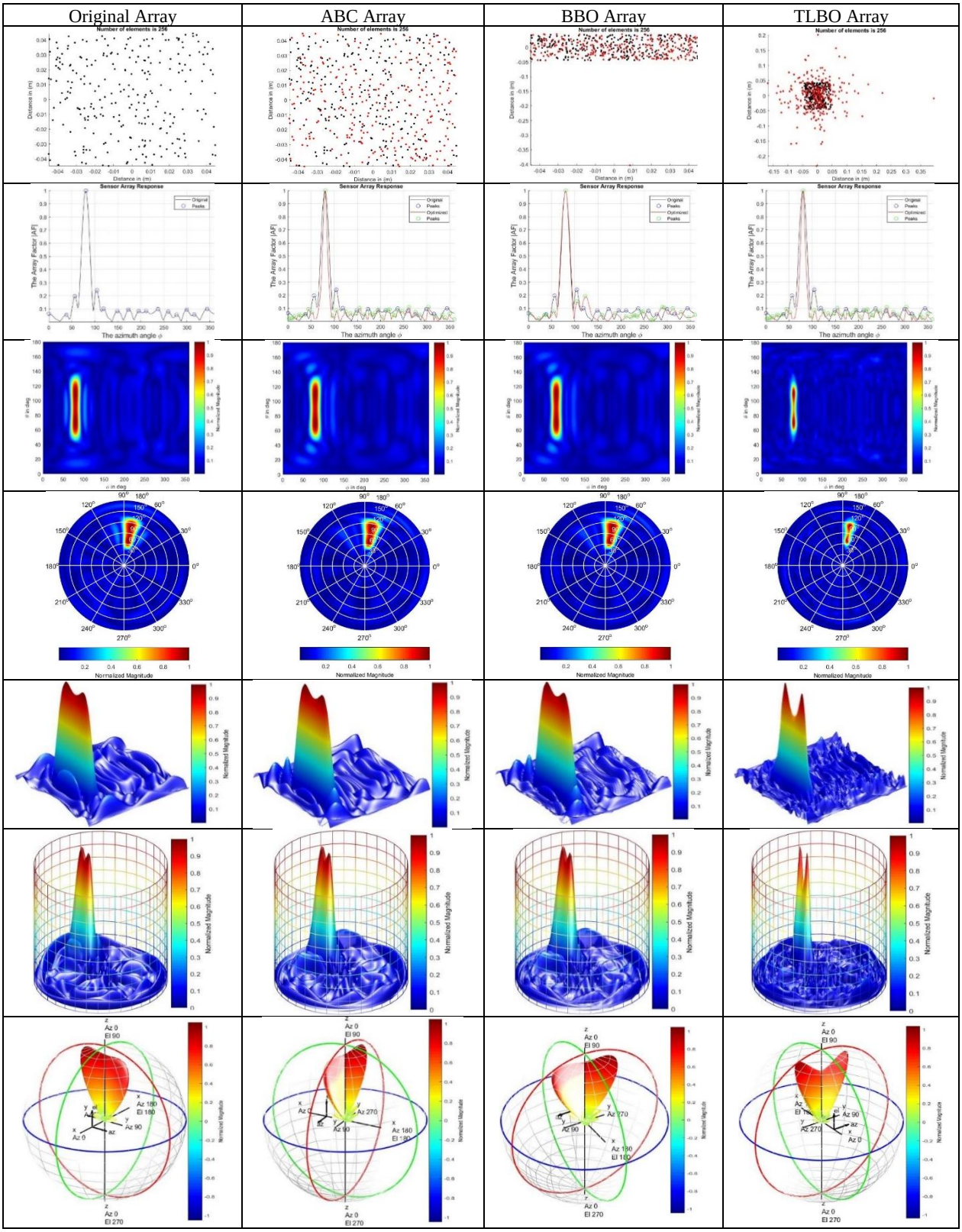
Table 2: The SADO Graphical and Optimization Array Response Results (Red Dots are the Optimized Locations)

UDSA			
Original Array	ABC Array	BBO Array	TLBO Array
			
			
			
			
			
			
			

SRDSA Concentric Ring

Original Array	ABC Array	BBO Array	TLBO Array
			

SRDSA Concentric Ring


Original Array
ABC Array
BBO Array
TLBO Array
Number of elements is 256
Distance in (m)
Sensor Array Response
The Array Factor (AF)
The azimuth angle
Original
Peaks
Optimized
Normalized Magnitude
Az 0
El 90
Az 180
El 180
Az 90
El 270