Low-cost Shaped Beam Synthesis for Semi-smart Base Station Antennas

Zhijiao Chen[1,2*], Clive Parini[2]

[1]School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing, China
[2]School of Electronic Engineering and Computer Science, Queen Mary, University of London, Mile End Road, London E1 4NS, United Kingdom
[*]z.chen@bupt.edu.cn

Abstract: We present a low-cost shaped beam synthesis method for semi-smart base station antennas. Compared with the conventional array synthesis for smart antenna applications, the proposed circular/conformal array shaped beam synthesis for semi-smart antenna has much lower system complexity and instrumentation implementation cost. This is because only few elements are involved and the circular array factor is simplified for coarse granularity beam. Adding to this, a fast Quasi-Newton method is used by the linear/conformal array for the beam optimization. By coding the shaped beam synthesis methodology in C/C++, the shaped beam synthesis for 12-element circular array is implemented for the semi-smart Macrocell, showing its advantages of low consumed time and stable mean square errors. Furthermore, a semi-smart sectored Picocel has been designed with 4-element conformal array, which was demonstrated to be superior to the linear one by avoiding the blind angle, exhibiting better accuracy and less consumed processing time.

1. Introduction

The ever increasing growth of user demand has triggered researchers and industries to come up with the 4th generation (4G) mobile communication system. The 4G network is less expensive and data transfer is much faster, with higher spectral efficiency [1]. With the aim of increasing the spectral efficiency of smart antenna schemes, the semi-smart antenna concept was developed by researchers at Queen Mary University of London by simplifying the system of a smart antenna in a cellular network [2-6].

The network deployments of semi-smart antenna, cellular network antenna and smart antenna (adaptive array antenna) are presented in Fig.1. Semi-smart antenna techniques increase user capacity by changing BS coverage patterns to equalise the number of users in each cell, and is best achieved by deploying the Bubble Oscillation Algorithm (BOA) to shape cellular coverage according to the traffic need. The BOA [5] is a simple, low-cost, seamless load balancing algorithm that equates cells in the same network to bubbles, as shown in Fig.1a and is compared with the conventional cellular network in the Fig.1b. Each bubble contains a certain amount of air and its shape can be distorted by the pressure from adjacent bubbles. If one

bubble suffers from high/low air pressure (heavy/light traffic load) and changes its shape, the adjacent bubbles will try to eliminate this vacuum and reach a new balanced state by oscillating their boundaries. After a few oscillations, an overall balance can be reached for all bubbles, and hence the coverage pattern for each BS is set without any central control system, but within a localised approach to optimisation via the intelligent negotiation between BSs. BS radiation patterns are updated at the rate that traffic load changes so, typically of order 40 seconds between updates [5]. Utilising the BOA, the semi-smart scheme balances user capacity by reducing coverage in heavy traffic-loaded cells, with the lost coverage being taken up by adjacent cells. This results in an overall increase in system capacity. Semi-smart antennas keep the system simple, and inexpensive, while also being easily integrated into an existing (dumb) cellular network and providing moderate capacity enhancement. This scheme is different from the conventional smart antenna, or adaptive array antennas shown in Fig.1c, whose capacity and data rate rely heavily on expensive RF systems and complex beam-forming software [7].

A real-time, low-complexity shaped beam synthesis is the key enabler for the semi-smart antenna scheme, which requires a small number of elements to synthesize a coarse granularity beam to track the user clusters rather than the individual user. Shaped beam synthesis of semi-smart antennas has been implemented using linear arrays in the ADAMANT project [2] for outdoor BS applications. In the ADAMANT project, 12-element 3-sector linear antenna arrays (Fig. 2a) were implemented, where each of the 4-elements of a sector covers 180 °pattern synthesis for 120 °coverage, and an overlapping three sector power pattern is implemented to form one cellular pattern. Power pattern synthesis is selected to achieve the shaped radiation pattern of cellular base station antennas, since it can give a better approximation to the desired pattern, compared with the far-field pattern synthesis. This method however, exhibits several drawbacks, such as the blind angles of the linear array and the overlap between the sector patterns should be considered.

This paper investigates a simple, low-cost, circular/conformal array shaped beam synthesis methodology, applying to 12-element circular array for a Macrocell and 4-element conformal array for a sectored Picocell. In section 2, the fundamentals and the investigation process for circular/conformal array shaped beam synthesis is presented in detail. The proposed methodology is verified by C/C++ codes and further modified after applying to a 12-element circular array (Fig. 2b) in section 3. In section 4, a 4-element conformal array shown in Fig. 2d is used to synthesize 180 °radiation patterns for a wall-mounted Picocell sector antenna. Its synthesis performance is compared with a 4-element linear array in Fig. 2c to show the advantage of the proposed conformal array for the sectored Picocell application.

2. Shaped Beam synthesis for circular/conformal array

Shaped beam synthesis can be regarded as a real time process of minimising the distance between the synthesised pattern and the desired pattern. The amplitude and phase values (i.e. $[\mathrm{A}_1 \cdots \mathrm{A}_\mathrm{n}, \mathrm{P}_1 \cdots \mathrm{P}_\mathrm{n}]$) that are used to feed the array are each optimised through a step by step process, until the synthesised pattern reaches the desired pattern. The desired pattern of each BS is determined by the semi-smart multi-agent system platform based on the aforementioned bubble algorithm, aiming at an overall system balance for user coverage and capacity.

An example of the power pattern synthesis by a conformal array is presented in Fig.3, where a 4-element conformal antenna array is excited by amplitude and phase controls in a matrix $\boldsymbol{x} = [\mathrm{A}_1 \cdots \mathrm{A}_4, \mathrm{P}_1 \cdots \mathrm{P}_4]$, resulting in the synthesis pattern (dotted line). In order to better approach the desired pattern (solid line), the gap between the synthesised and desired pattern must be minimised as much as possible. This gap is therefore represented by the weighted Mean Square Error (MSE), $\varepsilon(\boldsymbol{x})$, which is a commonly used error criterion evaluated over the entire visible region. The MSE is minimised by employing optimisation algorithms to optimise the values in the $\boldsymbol{x}$ matrix. When the minimum value of MSE is reached, the corresponding $\boldsymbol{x}$ matrix is passed to the antenna array to synthesise the desired pattern.

Based on this procedure, the whole process of the shaped beam synthesis for a conformal array can be divided into the following three steps.

2.1 Define the circular array factor and power pattern

The circular array factor can be defined by referring to Fig.4. Under the assumption that N isotropic elements are all equally spaced on the x-y plane along a circular ring of radius a, the normalized field of the array observed from the far-field point P can be written as [8],

$$AF(\theta, \phi) = \sum_{n=1}^{N} I_n exp\{j[ka\, sin\theta\, cos(\phi - \phi_n) + \alpha_n]\} \quad (1)$$

Where I_n and α_n are the amplitude and phase excitation of the nth element, respectively, while $\phi_n = 2\pi(n/N)$ is the angular position of the nth element on the x-y plane.

For the low-cost semi-smart antenna scheme application, the circular array factor in (1) is simplified by removing the synthesis on the elevation plane: θ being fixed to 90˚ to reduce calculation complexity. With $\lambda/2$ element spacing being set to avoid grating lobes, the original circular array factor in (1) is reduced and specified as,

$$AF(\phi, \boldsymbol{x}) = \sum_{n=1}^{N} A_n \exp\{j[ka\cos\left(\phi - 2\pi\left(\frac{n}{N}\right)\right) + P_n]\} \quad (2)$$

Where A_n and P_n are the amplitude and phase (real number) of the nth element, which can be assembled into a vector $\boldsymbol{x} = [\mathrm{A}_1 \cdots \mathrm{A}_\mathrm{N}, \mathrm{P}_1 \cdots \mathrm{P}_\mathrm{N}]$. N is the number of elements, which in our examples we have set to 4 for 180 °coverage, and 12 for 360 °coverage.

Based on the simplified array factor $\mathrm{AF}(\phi, \boldsymbol{x})$ in (2), the synthesized power pattern can be defined as,

$$S(\phi, \boldsymbol{x}) = E(\phi) \cdot |AF(\phi, \boldsymbol{x})|^2 \quad (3)$$

where $\mathrm{E}(\phi)$ is the element azimuth radiation pattern.

The array factor in (2) and the power pattern in (3) are both drawn to validate the arrays' scanning ability, as shown in Fig. 5. Here the value of A_n is fixed to 1 and P_n is fixed to 0, and $\mathrm{E}(\phi)$ is set to be 1 while $\mathrm{E}(\theta)$ is set to be $\sin(\theta)$ as we assumed that each element has cardioid radiation pattern to mitigate the mutual coupling effect for shaped beam synthesis. θ is fixed to 90˚ as the synthesis on the elevation plane is removed, while ϕ changes between 30˚-60 °-90˚ to point its beam to 30˚-60 °-90˚ respectively. It can be seen that the pattern in the elevation plane is fixed to 90˚ while the beam is sweeping in the azimuth plane for both array pattern and power pattern, which validated the proposed simplified array factor in (2) and power pattern in (3).

2.2 Calculate the weighted Mean Square Error (MSE)

With $\mathrm{D}(\phi)$ defined as the desired pattern, the weighted Mean Square Error (MSE) is,

$$\varepsilon(\boldsymbol{x}) = \frac{1}{DEG+1} \cdot \sum_{\phi=0}^{DEG} [S(\phi, \boldsymbol{x}) - D(\phi)]^2 \quad (4)$$

where DEG defines the number of degrees that the synthesised pattern covers. i.e. DEG=180 for 180 ° coverage and DEG=360 for the 360 °coverage scenario.

The MSE in (4) is simply supplied to the optimisation algorithm to find its minimisation. The derivation of the MSE is as follows:

$$\frac{\partial \varepsilon}{\partial \boldsymbol{x_i}} = \frac{2}{DEG+1} \cdot \sum_{\phi=0}^{DEG} \frac{\partial S}{\partial x_i} [S(\phi, \boldsymbol{x}) - D(\phi)] \quad (5)$$

The gradient equations of $S(\phi, \boldsymbol{x})$with respect to A_n and P_n are represented as,

$$\frac{\partial S}{\partial A_n} = E(\phi) \cdot 2 \cdot \{cos[kacos\left(\phi - \frac{2\pi n}{12}\right) + P_n] \cdot D_1 + sin[kacos\left(\phi - \frac{2\pi n}{12}\right) + P_n] \cdot D_2\} \quad (6)$$

$$\frac{\partial S}{\partial P_n} = E(\phi) \cdot 2A_n \cdot \{cos[kacos\left(\phi - \frac{2\pi n}{12}\right) + P_n] \cdot D_2 + sin[kacos\left(\phi - \frac{2\pi n}{12}\right) + P_n] \cdot D_1\} \quad (7)$$

Where, $D_1 = \sum_{n=1}^{N} A_n cos[kacos\left(\phi - \frac{2\pi n}{12}\right) + P_n]$

$$\text{And} \quad D_2 = \sum_{n=1}^{N} A_n sin[kacos\left(\phi - \frac{2\pi n}{12}\right) + P_n]$$

2.3 Optimise with Quasi-Newton method

The process of minimising the MSE can be regarded as an unconstrained optimisation of a nonlinear function. The optimised $\boldsymbol{x}$ value (amplitude and phase in matrix) can be approached in a step-by-step method, such as $x_{k+1} = x_k + \propto_k d_k$, where $\propto_k$ is the "step size" of the next point and d_k is the "angle" of direction [9].

The choice of the optimisation algorithm depends on the accuracy and time that is required in the application itself. For example, the Genetic Algorithm (GA) method is widely used for smart antenna beam-forming, as it works with an ensemble of solutions, aiming to the basin of a global minimum [10]. However, GA is expensive and time-consuming as it calculates from the first generation for every synthesised pattern, which is not suitable for real-time applications such as a semi-smart antenna system [11]. In the previous research on the shaped beam synthesis for 4-element linear array [2], several optimisation algorithms, including Downhill Simplex, Powell's method, Conjugate Gradient and the Quasi-Newton method, have been investigated. The Quasi-Newton algorithm was proven to be by far the best in the real-time scenario of semi-smart antenna applications due to its short synthesis time and robustness. Therefore, in this paper, the Quasi-Newton method is directly used as the optimisation algorithm to minimise the MSE for circular/conformal array shaped beam synthesis.

The Quasi-Newton method [12] is evaluated from the Newton method, which requires the knowledge of the first and second gradient. It is well known that the extreme of a function is characterized by its gradients being equal to zero. If the function is quadratic, the gradient descent algorithm, $d_k = -\nabla f(x_k)$ can be used, arriving at the extreme in a single step. However, if the function is non-quadratic, finding an extreme is not as straightforward. For this kind of problem, Newton proposed an iterative solution: first look at a local quadratic approximation to the nonlinear function and find its extreme, and then generate a new local approximation and so on. In the Newton method, the local approximation can be approximated by the Taylor series and its derivative:

Taylor series:

$$f(x_{k+1}) = f(x_k) + \nabla f(x_k)(x_{k+1} - x_k) + \frac{1}{2}(x_{k+1} - x_k)\nabla^2 f(x_k)(x_{k+1} - x_k) \tag{8}$$

Derivative of the Taylor series:

$$\nabla f(x_{k+1}) = \nabla f(x_k) + \nabla^2 f(x_k)(x_{k+1} - x_k) \tag{9}$$

Setting the derivative $\nabla f(x_{k+1})$ to be zero, we obtain,

$$\nabla f(\mathrm{x_k}) = \nabla^2 f(x_k)(x_k - x_{k+1}) \tag{10}$$

from which the local minimum can be found as,

$$x_{\mathrm{k+1}} = x_k - [\nabla^2 f(x_k)]^{-1} \nabla f(\mathrm{x_k}) \tag{11}$$

As the function is not quadratic, it is necessary to solve for the solution iteratively. Let δ_k be the path from the starting point to the solution, and the new local minimum can be represented as,

$$x_{k+1} = x_k - \delta_k \tag{12}$$

$$\text{With} \quad \delta_\mathrm{k} = [\nabla^2 f(x_k)]^{-1} \nabla f(x_k) = H_k^{-1} g_k \tag{13}$$

Here H_k^{-1} is the inverse of the Hessian matrix which determines the "angle" of the direction and the gradient g_k determines the "step size" of the next point in the iteration. After calculating δ_k from (13), the new approximation, $x_{\mathrm{k+1}}$, can be obtained from (11) and used as a source for the next iteration. The iteration will be stopped when $\nabla \mathrm{f}(\mathrm{x_k}) = 0$, where the minimum is achieved.

The Newton method is simple but not practical, as computing the Hessian matrix is very computationally expensive. Therefore, the Quasi-Newton method is introduced by using information from the current iteration to compute the new Hessian matrix. That is, the Hessian matrix $[\nabla^2 \mathrm{f}(\mathrm{x_k})]^{-1}$ is replaced by an approximation, which is much cheaper in computational terms. The Quasi-Newton method has several flavours. Examples include the DFP (Davidon–Fletcher–Powell) formula, the BFGS (Broyden–Fletcher–Goldfarb–Shanno) algorithm [12] and the BHHH (Berndt-Hall-Hall-Hausman) algorithm [13]. The DFP formula was the first Quasi-Newton method, but it was soon superseded by BFGS algorithm, which is considered to be the most effective of all Quasi-Newton updating formulae [12][14]. Let $\mathrm{s_k} = \mathrm{x_k} - \mathrm{x_{k-1}}$ be the change in the parameters in the current iteration, and $\mathrm{y_k} = \nabla \mathrm{f}(\mathrm{x_k}) - \nabla \mathrm{f}(\mathrm{x_{k-1}})$ be the change in gradients, The BFGS update formula is given by [14]:

$$H_{k+1} = H_k + \frac{s_k s_k^T}{y_k^T s_k}\left[\frac{y_k^T H_k y_k}{y_k^T s_k} + 1\right] - \frac{1}{y_k^T s_k}[s_k y_k^T H_k + H_k y_k S_k^T] \tag{14}$$

This Hessian updates will be imported to (12) and (13) to update the x values until the minimised MSE $\varepsilon(\boldsymbol{x})$ is achieved. The Quasi-Newton method in BFGS flavour is coded in C/C++ in order to validate the beam pattern synthesis on the semi-smart antenna scheme. The flowchart of the C/C++ program is displayed in Fig. 6. $\partial\varepsilon_0$ is set by the user to determine the accuracy of the optimisation. That means, the iteration will not stop if the gradient of MSE is lower than $\partial\varepsilon_0$. The new point $\mathrm{x_{k+1}}$ obtained at the end of the iteration will feed as a source to update the array factor. Following this, all of the parameters are replaced by the updated ones. In this case, the volume of the occupied computer memory is under control.

This C/C++ coding for circular array shaped beam synthesis has been successfully executed for a 12-element circular array and a 4-element linear/conformal array, which are detailed in section 3 and section 4, respectively.

The azimuth and elevation radiation patterns of the array element have been taken to be simply omni in azimuth and $\sin(\theta)$ in elevation. In a practical system the array elements would be, for example, a cavity backed patch antenna and so our simple element pattern functions would be replaced by the array elements embedded antenna pattern, either derived numerically from EM modeling of the array or obtained from measured patterns. This use of the embedded pattern takes account of array mutual coupling and insures that the shadowing effects that occur in non-planar arrays is mitigated. As in any array, the effect of mutual coupling will modify the embedded element pattern of the "end elements" which in this case would be the semi-circular arrays (Fig 2d), but this is a secondary effect for such a small array offering low overall gain.

3. 12-element array for a Microcell

The synthesis of four $360°$ radiation patterns is displayed in Table 1. These four target radiation patterns include a sector directional radiation pattern with small back lobe (Pattern 1), a directional radiation pattern with big back lobe and side lobes (Pattern 2), a quasi-omnidirectional radiation pattern (Pattern 3) and a quasi-omnidirectional radiation pattern with a null (Pattern 4). In the left column of patterns, the initial patterns (dotted line) are excited by the random amplitudes and phases, producing patterns with high MSE that are far from the desired pattern (solid line). After several iterations (K) and consumed time (T), the minimisation between the desired and synthesised pattern is achieved with low MSE, which are plotted on the right-hand pattern column with a good approximation between the desired and the synthesised patterns.

The MSE determines how well the synthesis pattern approaches the desired pattern, the smaller the MSE is, the closer the synthesis pattern is to approaching the desired one. We set 0.05 as the MSE threshold to separate good synthesis pattern from the poor ones. As compared in Fig. 7, the pattern 2 synthesised with MSE=0.051574 has a tolerable approach to the desired pattern, while the synthesis pattern with MSE=0.112753 is beyond acceptable levels of pattern synthesis. Therefore, MSE=0.05 is set as the threshold for semi-smart antenna pattern synthesis.

To check the stability of the pattern synthesis, these four patterns are synthesised 100 times. The computational time and the optimised MSE for each of the pattern syntheses are displayed in Fig.8, with the data sorted in ascending order of consumed time (a) and MSE (b). Using an Intel Core i5-2500K CPU@3.3GHz computer processor, the time for the $360°$ pattern synthesis (Fig. 8a) is less than 0.4 seconds for more than 70% of the cases, and the longest time is less than 2 seconds over all the cases. By today's

standards this is a CPU of modest performance and such CPU power would easily be found in a typical 4G BS. The 2 second maximum processing time is fast enough for a semi-smart antenna scheme since its BS radiation patterns are updated at the rate of 40 seconds. The MSE values for the 360 ° pattern synthesis is presented in Fig. 8b. It is scaled and re-plotted in the inner figure of Fig. 8b, where the MSE values larger than 0.05 will be set as 0.05.

It can be seen that the synthesised pattern may have a large MSE that goes far beyond the MSE threshold. This means that the iterations stop for a local minimum, despite the synthesis pattern not approaching the desired pattern. It reveals that the power pattern synthesis search area for the circular array is the one that has multiple minima, consisting of a single global minimum, more than 50% quasi-global minimum (MSE<0.05) and local minimum with MSE>0.05. The global minimum can be directly achieved by using other algorithms (i.e. GA) that aim for global minimum, but the time these will consume will be much longer (i.e. more than 100 times) than that of Quasi-Newton method.

This "local minimum" problem can be solved by using a re-synthesis method in the optimisation process. Thus, if the optimised MSE values fall into the local minimum (MSE>0.05), the optimisation will start over automatically, with a new initial point, until the optimised MSE reaches the defined threshold. By adding the re-synthesis method, the consumed time and MSE of the synthesis are re-calculated and updated in Fig. 9. The consumed time only increases by a small amount compared to the case without the re-synthesis. This is because the patterns synthesised with a small MSE value usually take much longer time than those with a large MSE value since good pattern synthesis needs more iterations. Hence, it doesn't take too much time to re-synthesise the pattern when the pattern dropped into the local minimum, which proves the reliability of the proposed re-synthesis methodology.

The computational cost could be further reduced by saving the successfully synthesised patterns in a case database to avoid repeated synthesis of a frequently occurring desired power patterns (very likely in a 4G deployment). The patterns in this case database can be retrieved and reused directly when similar patterns are needed, and revised if a new frequency pattern is found. These artificial intelligence (or machine learning) techniques have been investigated in a semi-smart antenna scheme [2] with the Case Based Reasoning (CBR) and Neural Network approaches. Such an approach would be a trivial image classification problem as the "images" would by just 360 floating point numbers representing the power pattern (1° spacing) and the number of "images" to compare would be unlikely to exceed 100. By using this methodology, it is not important to have a small average MSE and fast calculations for each synthesis, but it is crucial to find out what patterns can be synthesised by different array parameters.

4. 4-element array for a Picocell sector antenna

A sector radiation pattern that points its maximum gain towards the broadside direction is desirable for a wall-mounted Picocell sector antenna. In this section, both the 4-element linear array in Fig. 2c and the 4-element conformal array in Fig. 2d are used to synthesise 180 °sectored radiation patterns. The synthesised patterns are compared in Table 2. The proposed circular array shaped beam synthesis method plus the re-synthesis method (threshold=0.05) is used for the 4-element conformal array. Four patterns are intensively investigated, including a pattern with a maximum edge (Pattern 1), a pattern with an irregular edge (Pattern 2), an asymmetrical pattern with broadside gain (Pattern 3), and a pattern with a deep null (at least -20 dB) at one edge of the pattern (Pattern 4). Using the same computer processor, their MSEs and the consumed time for 100 syntheses are displayed in Fig. 10, with the MSE sorted in ascending order (Fig. 10a) while the consumed time (Fig. 10b) is displaced mapping to the MSE values in Fig. 10a. It is found that in the linear array synthesis, the optimised MSE remains almost unchanged when a different initial point is applied. This proves that the pattern synthesis of linear array has multiple, equally good minima. This is different from the pattern synthesis of the conformal array, which has a global minimum and several quasi-global minima.

The MSE comparison in Fig. 10a shows that the linear array has better synthesis performance with better MSE and less consumed time for Pattern 1. However, for Pattern 2 and Pattern 3 with asymmetrical patterns, the linear array synthesis results in much larger optimised MSEs and higher consumed time than that of the conformal array. Furthermore, the linear array cannot synthesise the "null" at the edge of the patterns' visible region (i.e. Pattern 4). This problem is defined as the "blind angle" problem of the linear array. In practical sector beam synthesis, a deep null (at least -20 dB) at the edge of pattern (±180 °) is sometimes desirable to avoid the cross coupling to the next sector. The pattern synthesised by the conformal array can avoid this "blind angle" problem, which presents this arrays advantages over the linear array synthesis. The other advantages of the conformal array can be seen from providing a simpler and smaller array structure, with better approximation accuracy for asymmetrical pattern synthesis. The consumed time shown in Fig. 10b for both linear array and conformal array are less than 1 second, which is acceptable for the semi-smart antenna scheme.

5. Conclusion

This paper considers the shaped beam synthesis of circular/conformal arrays for a semi-smart antenna scheme, where only a few antenna elements are involved in the synthesis of beams with low angular accuracy and gain. Hence, the low computational cost Quasi-Newton method is employed as the

optimisation algorithm for the shaped beam synthesis process, and it is found that the synthesis accuracy can be improved by adding the re-synthesis method. The comparisons between the 4-element conformal array and the 4-element linear array are made to show the advantages of the conformal array, i.e. low complexity, better MSE and avoidance of the blind angle.

Compared with the conventional circular array synthesis for smart antenna applications, the proposed shaped beam synthesis involves only few elements and utilizes low-cost Quasi-Newton method for a coarse granularity beam synthesis. This results in a much lower system complexity and instrumentation implementation cost, which is suitable for low-cost semi-smart base station antenna applications for 4G.

6. Acknowledgement

This work was supported in part by the Fundamental Research Funds for the Central Universities of China.

7. References

[1] Amit Kumar, Yunfei Liu, Jyotsna Sengupta and Divya, “Evaluation of Mobile Wireless Communication Networks: 1G to 4G”, IJECT Vol.1, Issue 1, 2010.

[2] Petrit Nahi, “Development of semi-smart antennas for use with cellular BSs employing agent control of radiation pattern coverage for resource management” PhD thesis, Electronics Engineering, Queen Mary University of London, 2004.

[3] L. Cuthbert, Y. Wang, “Resource Management Design Document”, ADAMANT IST-2001-39117,(2003)

[4] Clive Parini, Xiaodong Chen, Yasir Alfadhl, “Final Report on Semi-smart Antenna Technology Project”, Ofcom Contract No.830000081, 2006.

[5] Lin Du, “Intelligent Geographic Load Balancing for Mobile Cellular Networks”, PhD thesis, Electronics Engineering, Queen Mary University of London, 2004.

[6] Z. Chen, M. Candotti, C. G. Parini, "Pattern Reconfigurable Dielectric Loaded Antenna (DLA) for Polarisation Diversity Wi-Fi Access Point," IEEE International Symposium on Antennas and Propagation, pp.1240-1241, 2013.

[7] Ahmed EI Zooghby, "Smart Antenna Engineering", ©2005, ARTECH HOUSE, INC, ISBN-10: 1-58053-515-1, 2005.

[8] Constantine Balanis, "Antenna Theory, Analysis and Design", ©2005, John Wiley and Sons, ISBN: 978-0-471-91547-5, 2005.

[9] R. Fletcher, "Practical Methods of Optimization" Second Edition, ©1987, John Wiley and Sons, ISBN: 978-0-471-66782-7, 2005.

[10] Mantawy, A.H, Abdel-Magid, Y.L, Selim, S.Z, "Integrating genetic algorithms, tabu search, and simulated annealing for the unit commitment problem," IEEE Transactions on Power Systems, vol.14, no.3, pp.829,836, Aug 1999.

[11] Dapeng Zhang, “Radio Resource Management based on Genetic Algorithms for OFDMA Networks”, PhD thesis, Electronics Engineering, Queen Mary University of London, 2012.

[12] Ronald Schoenberg, "Optimization with the Quasi-Newton Method", Aptech Systems, Inc. Maple Valley, WA, 2001.

[13] William Gould，Jeffrey Pitblado，William Sribney, “Maximum Likelihood Estimation with Stata”, Third Edition, Stata Press, 2010.

[14] Jorge Nocedal and Stephen J. Wright, Chapters 6 and 7 from “Numerical Optimization, Second Edition, Springer Verlag, 2006.

Figure Captions:

Fig. 1. The system deployment of smart antenna, cellular network and semi-smart antenna.

a Semi-smart antenna

b Cellular network antenna

c Smart antenna (adaptive array antenna)

Fig. 2. Antenna array layouts for 12-element Macrocell and 4-element Picocell and sectored Picocell.

a 12-element linear array with 360 °coverage for Macrocell

b 12-element circular array with 360 °coverage for Macrocell

c 4-element linear array with 180 °coverage for sectored Picocell

d 4-element conformal array with 180 °coverage for sectored Picocell

Fig. 3. Example for shaped beam synthesis with 4-element linear antenna array.

Fig. 4. Geometry of an N-element circular array

Fig. 5. The array pattern and power pattern synthesised by 12-element circular array with λ/2 spacing and $\theta = 90°$

a Array pattern in azimuth plane

b Power pattern in azimuth plane

c Array pattern in elevation plane

d Power pattern in elevation plane

Fig. 6. Flowchart of Quasi-Newton method BFGS algorithm in C/C++ coding

Fig. 7. The radiation pattern plots of synthesised Pattern 2 with different MSE

a Pattern with MSE=0.058574

b Pattern with MSE=0.112753

Fig. 8. The synthesis performance of four different patterns synthesised by 12-element circular array over 100 times with random starting points, with data sorted in ascending order

a Consumed time

b MSE in dB; the inner figure is the scale of MSE in dB (the MSE values larger than 0.05 will be set as 0.05)

Fig. 9. By setting the MSE threshold as 0.05 and using auto-restart, the synthesis performance for four patterns over 100 times with random starting points and data sorted in ascending order

a Consumed time in ms

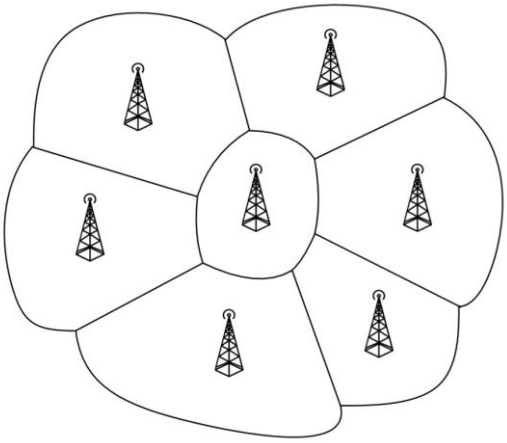
b MSE in dB

Fig.10. Synthesis performance comparison of the 4-element linear array and 4-element conformal array for four different patterns

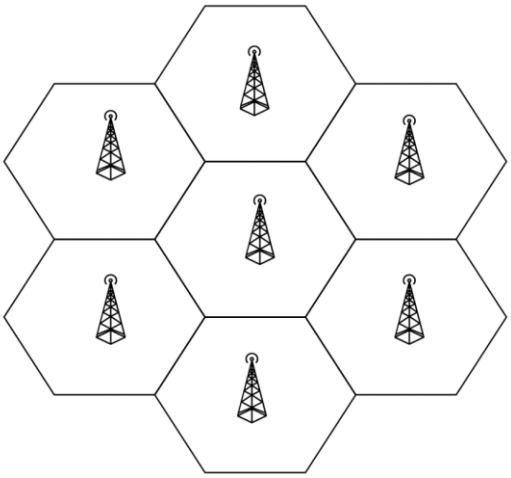
a MSE in dB, with data sorted in ascending order

b Consumed time in ms, mapping to the MSE values in Fig. 10a.

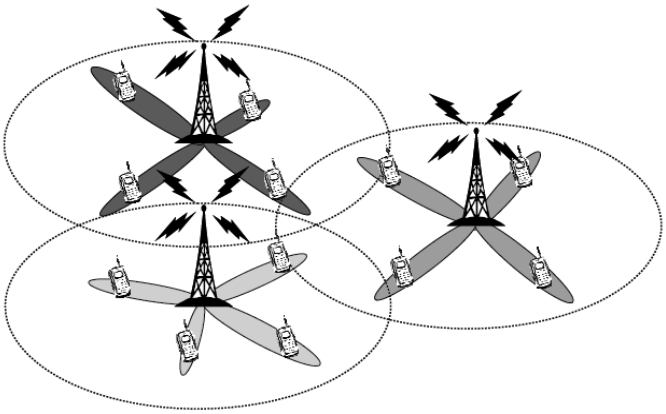
Fig. 1:



(a)



(b)



(c)

Fig. 2:

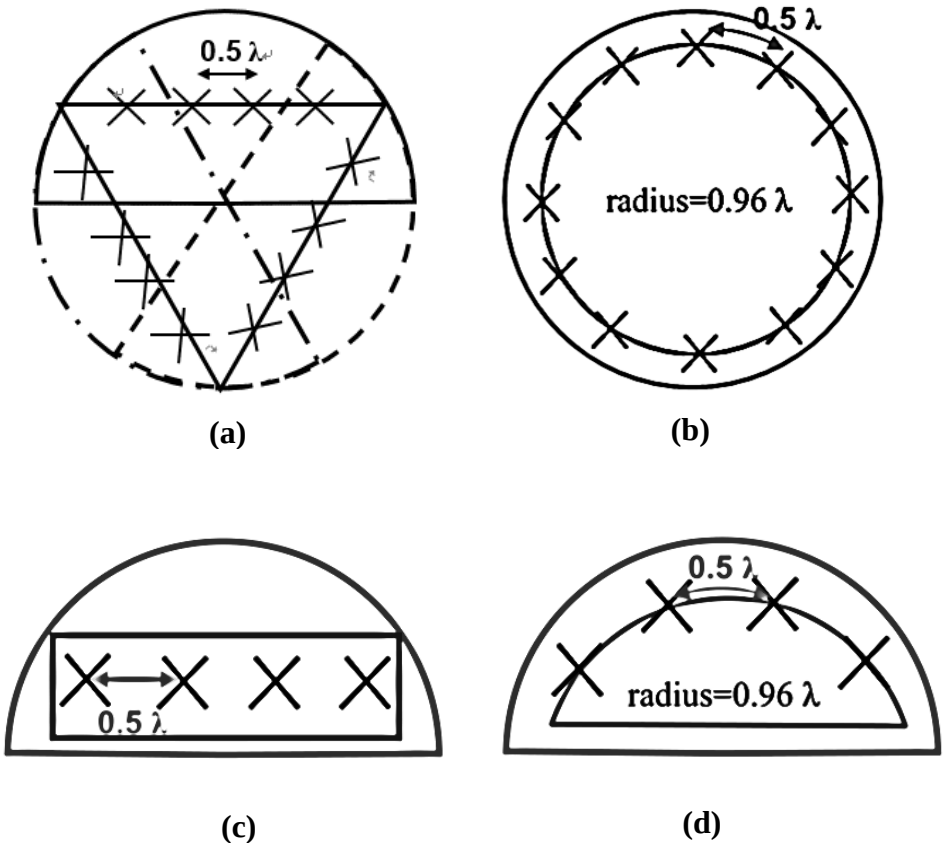


Fig. 3:

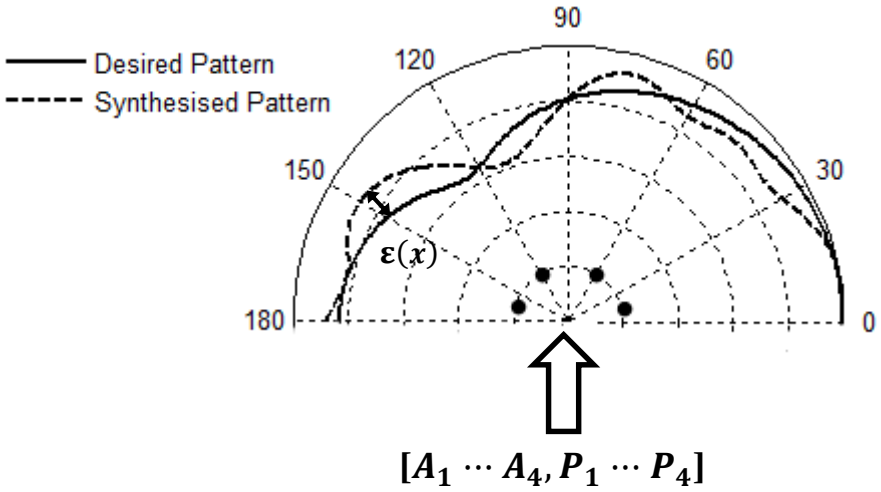

Desired Pattern
Synthesised Pattern
90
120
60
150
30
180
0
$\varepsilon(x)$
$[A_1 \cdots A_4, P_1 \cdots P_4]$

Fig. 4:

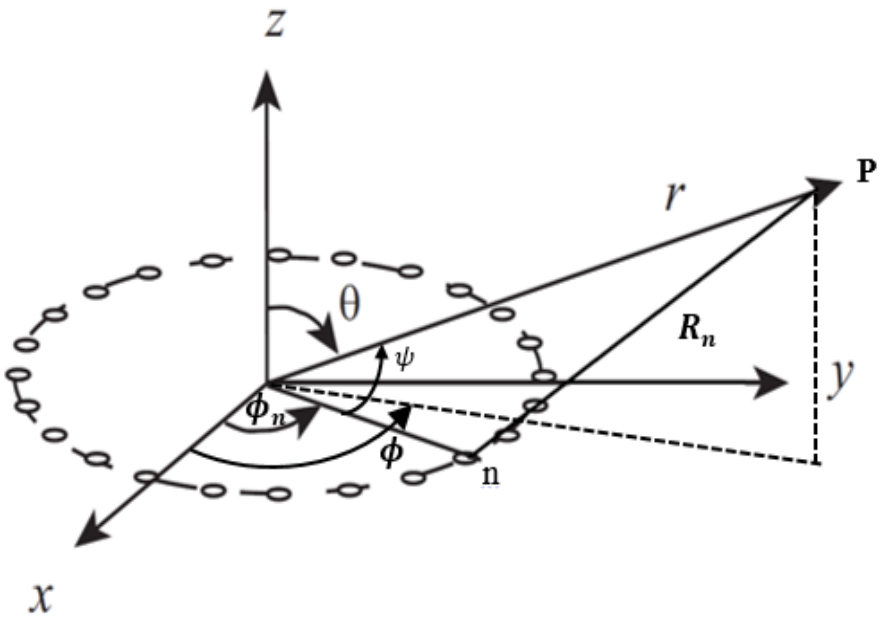
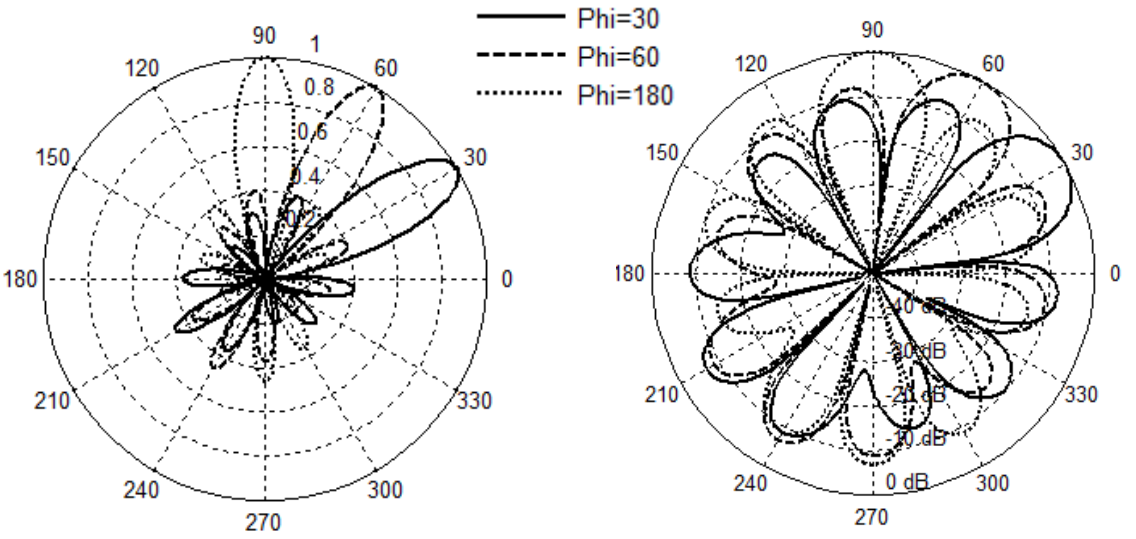
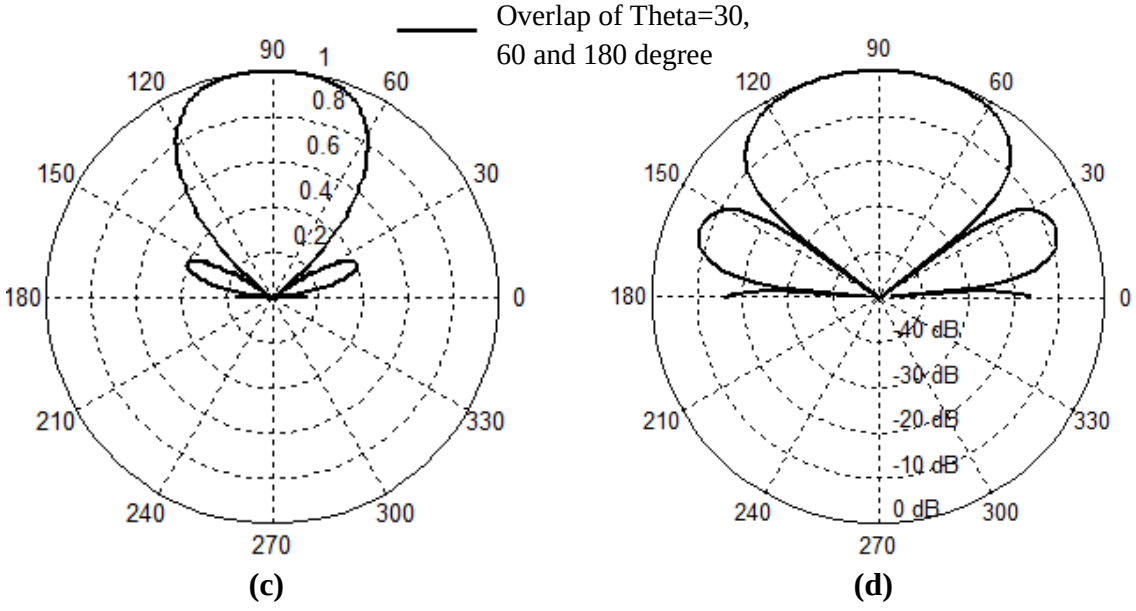

z
P
r
θ
R_n
ψ
y
ϕ_n
ϕ
n
x

Fig. 5:


Phi=30
Phi=60
Phi=180
90
120
60
150
30
180
0
210
330
240
300
270
1
0.8
0.6
0.4
0.2
-40 dB
-30 dB
-20 dB
-10 dB
0 dB

(a) **(b)**


Overlap of Theta=30,
60 and 180 degree
90
120
60
150
30
180
0
210
330
240
300
270
1
0.8
0.6
0.4
0.2
-40 dB
-30 dB
-20 dB
-10 dB
0 dB

(c) **(d)**

Fig. 6:

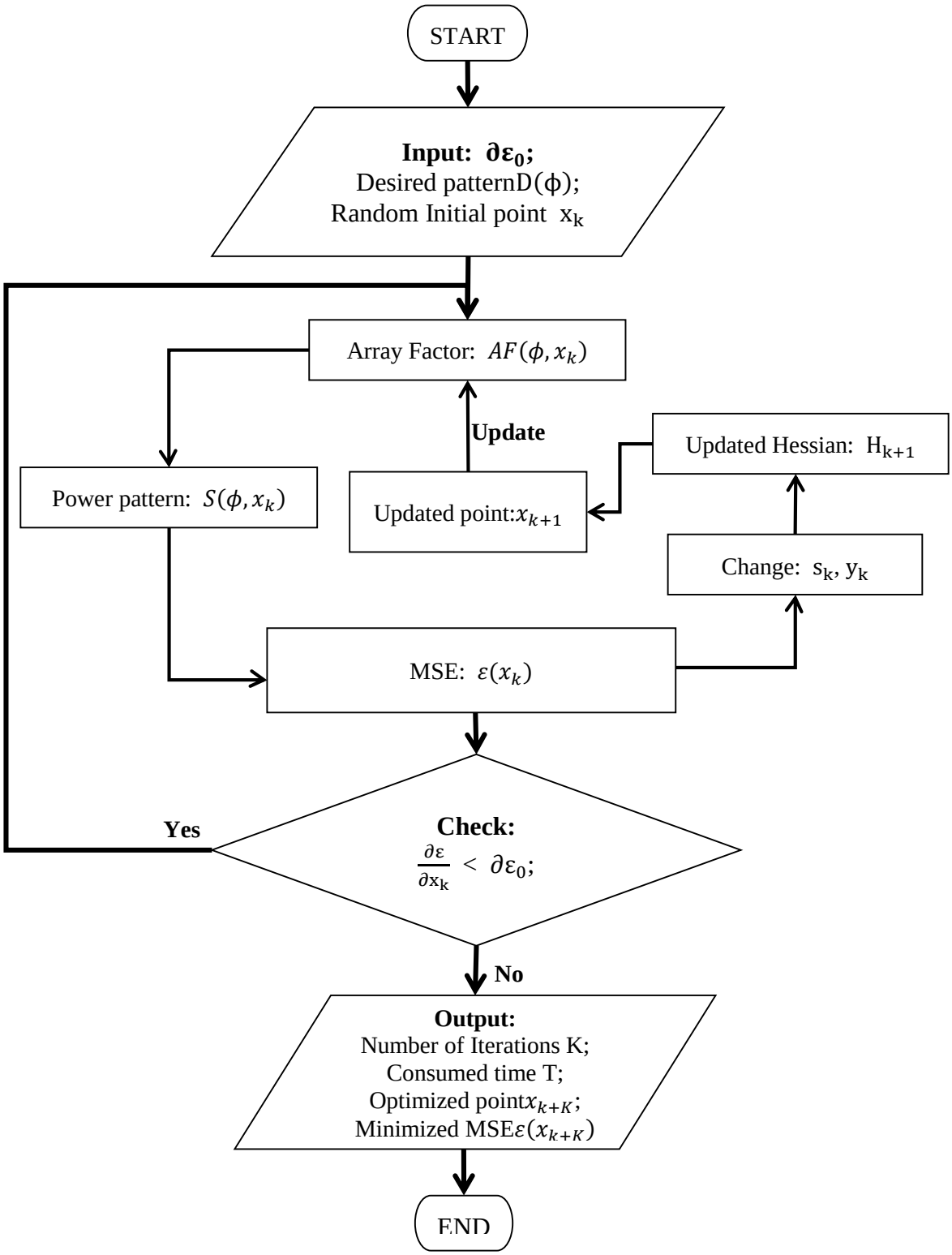
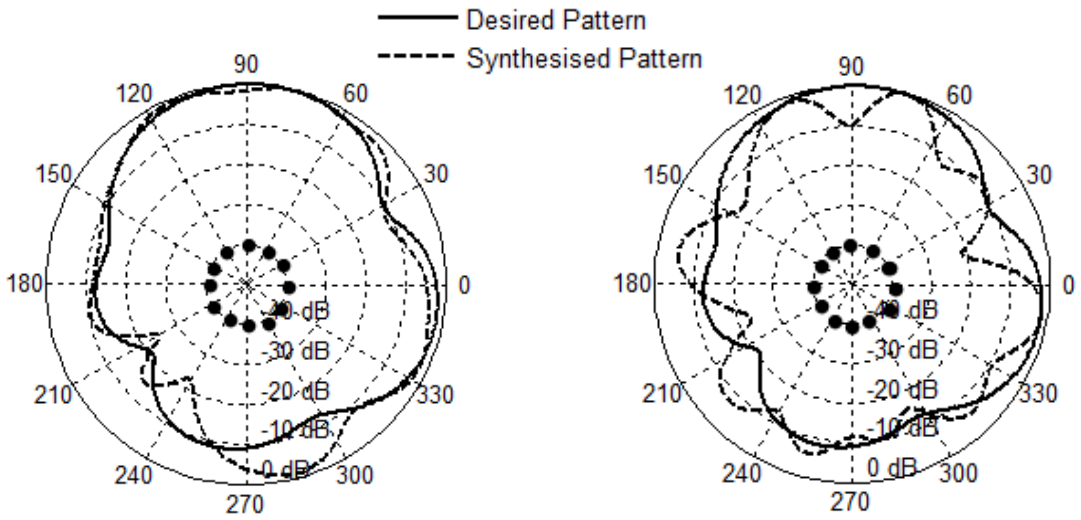

START
Input: $\partial\varepsilon_0$;
Desired patternD(ϕ);
Random Initial point x_k
Array Factor: $AF(\phi, x_k)$
Update
Updated Hessian: H_{k+1}
Power pattern: $S(\phi, x_k)$
Updated point:x_{k+1}
Change: s_k, y_k
MSE: $\varepsilon(x_k)$
Yes
Check:
$\frac{\partial\varepsilon}{\partial x_k} < \partial\varepsilon_0$;
No
Output:
Number of Iterations K;
Consumed time T;
Optimized pointx_{k+K};
Minimized MSE$\varepsilon(x_{k+K})$
END

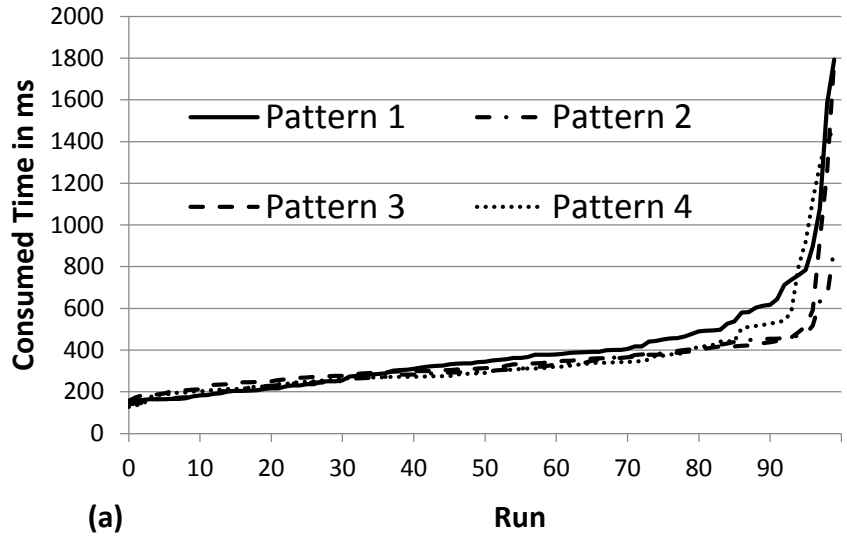
Fig. 7:



(MSE=0.051574) **(MSE=0.112753)**

(a) **(b)**

Fig. 8:


Consumed Time in ms
2000
1800
1600
1400
1200
1000
800
600
400
200
0
Pattern 1
Pattern 2
Pattern 3
Pattern 4
0 10 20 30 40 50 60 70 80 90
(a)
Run

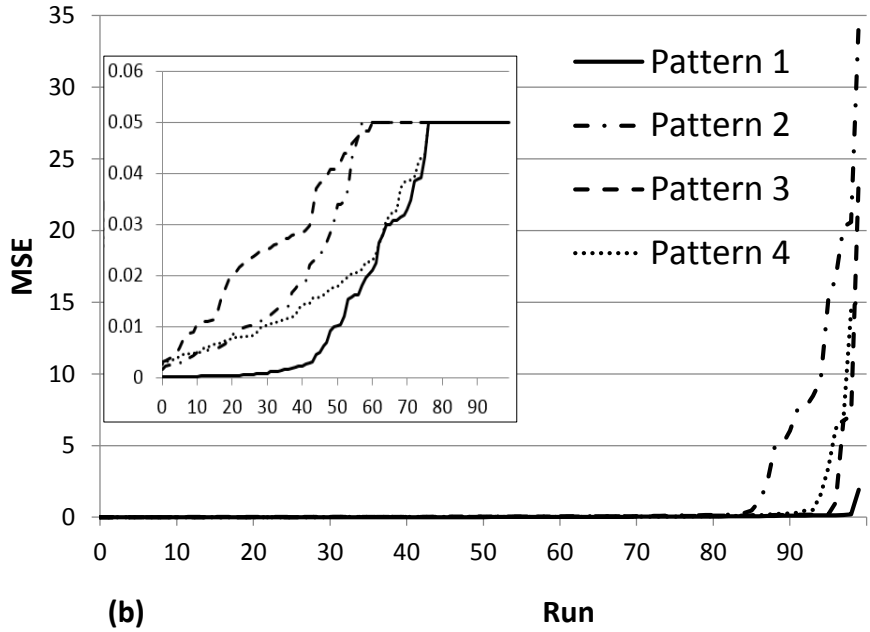
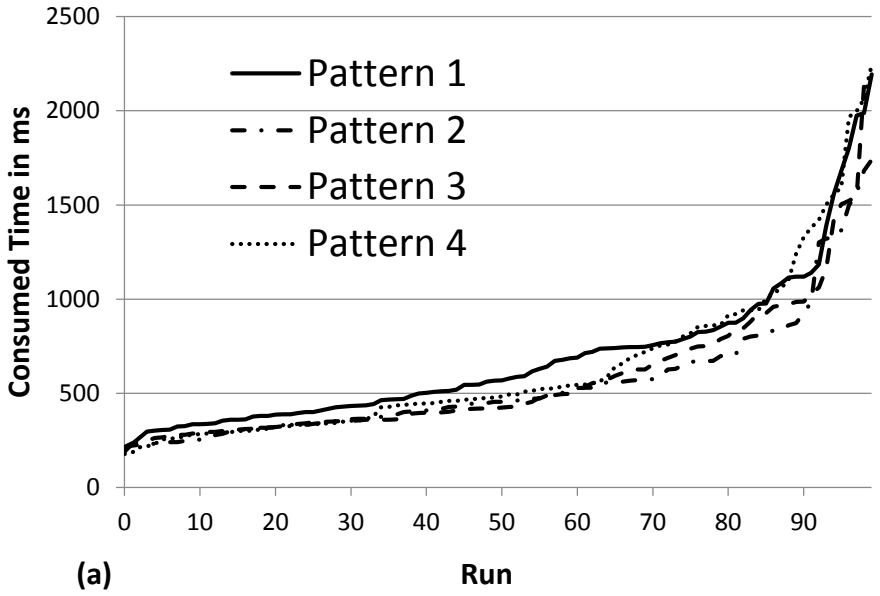

MSE
35
30
25
20
15
10
5
0
0.06
0.05
0.04
0.03
0.02
0.01
0
0 10 20 30 40 50 60 70 80 90
Pattern 1
Pattern 2
Pattern 3
Pattern 4
0 10 20 30 40 50 60 70 80 90
(b)
Run

Fig. 9:


2500
2000
1500
1000
500
0
Consumed Time in ms
Pattern 1
Pattern 2
Pattern 3
Pattern 4
0
10
20
30
40
50
60
70
80
90
(a)
Run

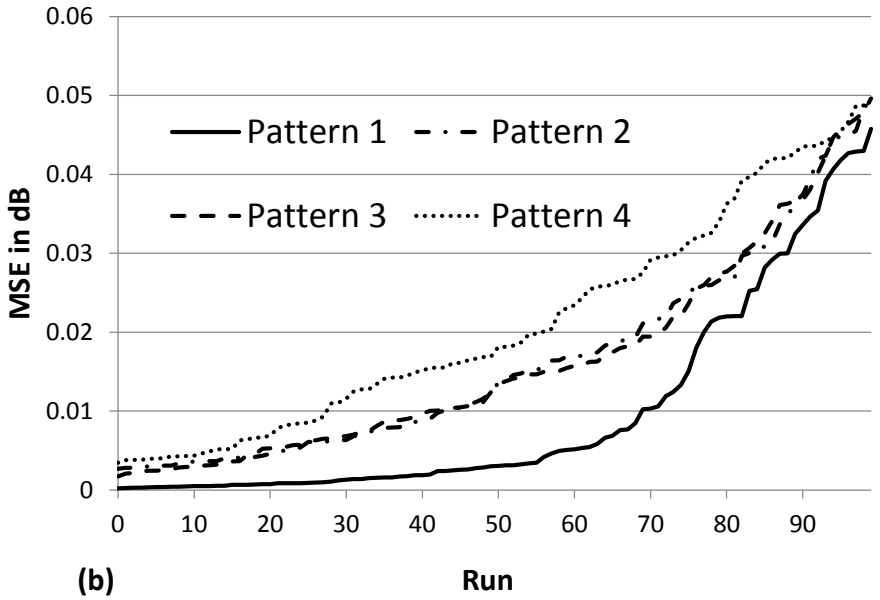
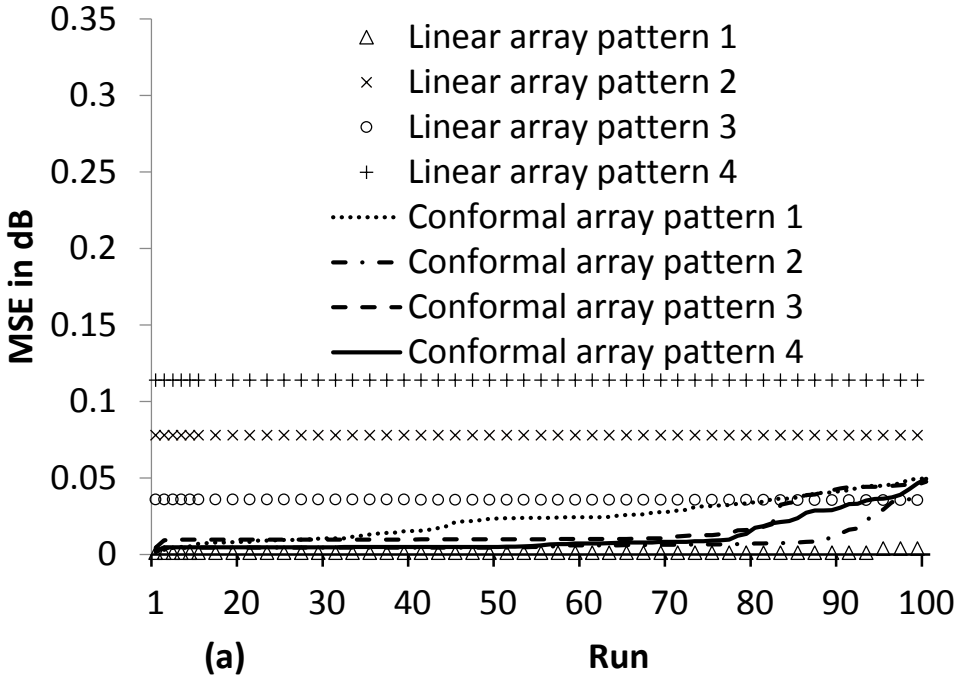

0.06
0.05
0.04
0.03
0.02
0.01
0
MSE in dB
Pattern 1
Pattern 2
Pattern 3
Pattern 4
0
10
20
30
40
50
60
70
80
90
(b)
Run

Fig. 10:


0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
MSE in dB
Linear array pattern 1
Linear array pattern 2
Linear array pattern 3
Linear array pattern 4
Conformal array pattern 1
Conformal array pattern 2
Conformal array pattern 3
Conformal array pattern 4
1
20
30
40
50
60
70
80
90
100
(a)
Run

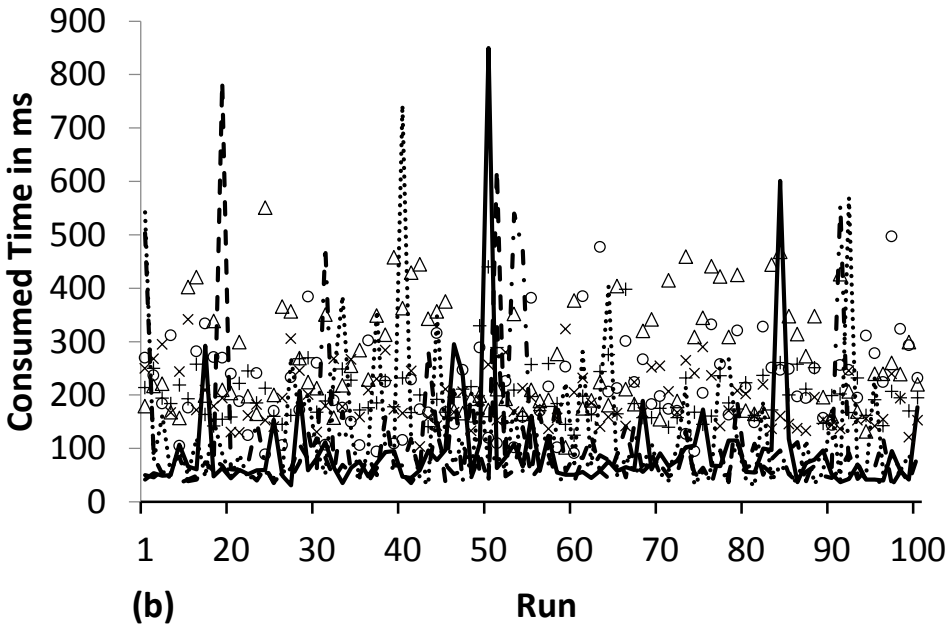

900
800
700
600
500
400
300
200
100
0
Consumed Time in ms
1
20
30
40
50
60
70
80
90
100
(b)
Run

Table Captions:

Table 1. Radiation pattern comparisons for the synthesis patterns and desired pattern by 12 element circular array: the pattern before the synthesis (left column) and after the synthesis (right column)

Table 2. Synthesised radiation pattern comparisons between 4-element linear array and 4-element conformal array: pattern synthesis by the linear array (left column) and by the conformal array (right column).

Table 1:

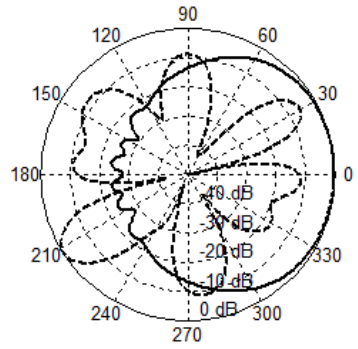
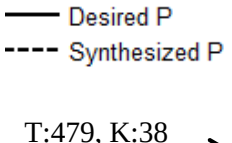
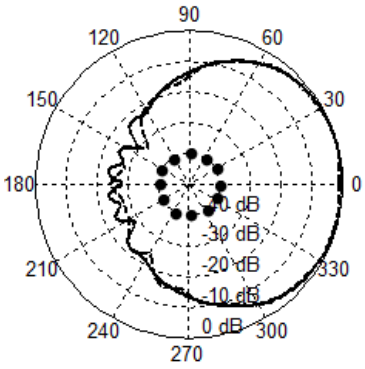
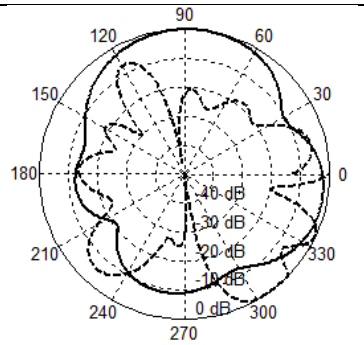
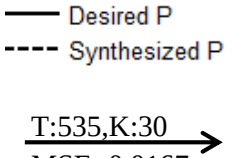
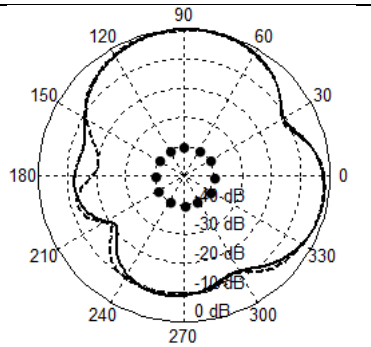
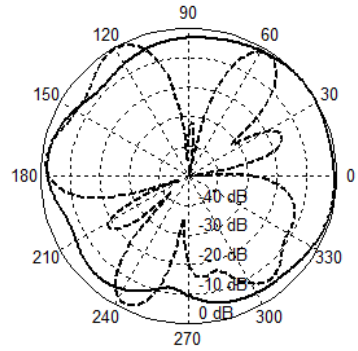
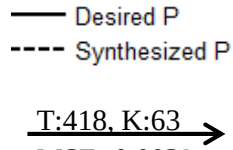
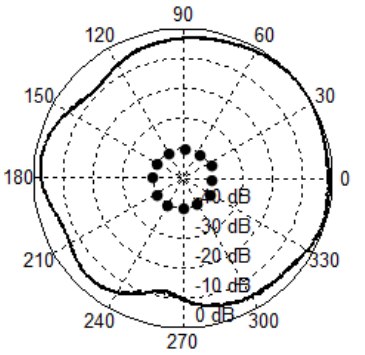
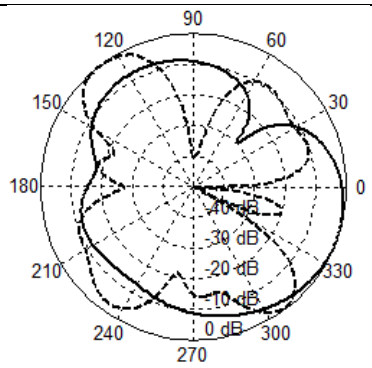
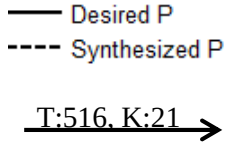
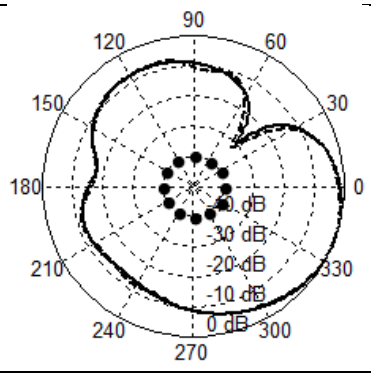
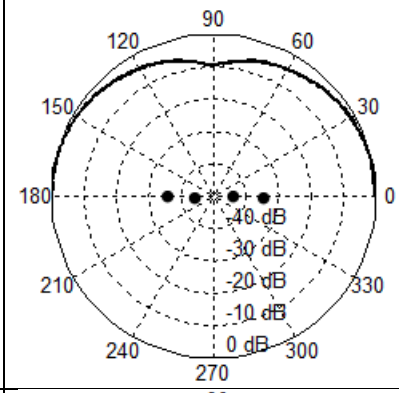
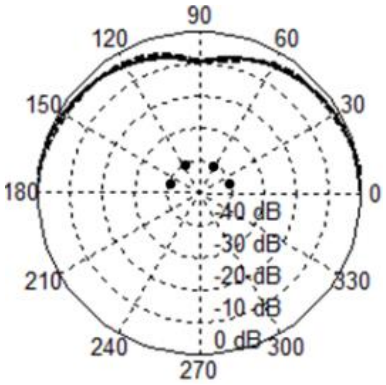
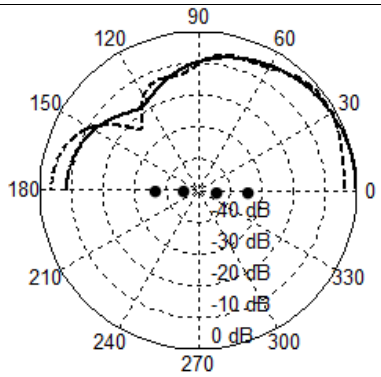
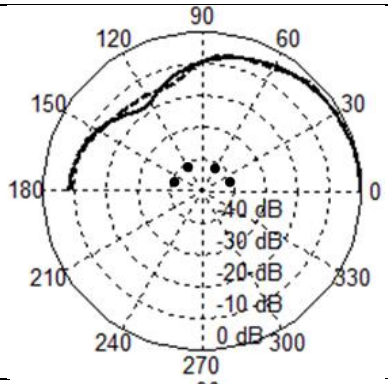
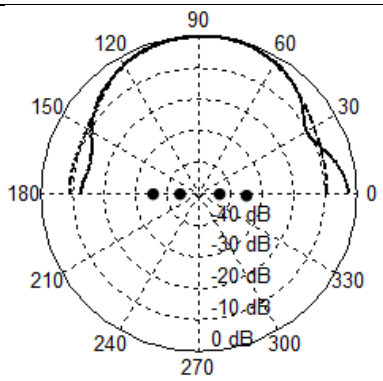
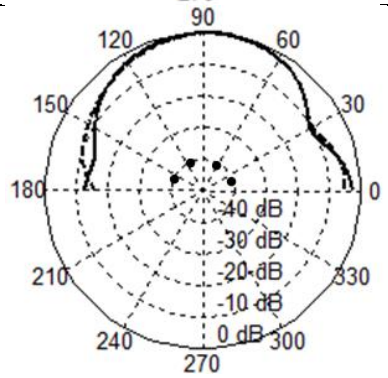
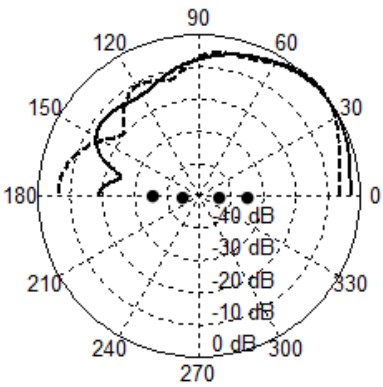
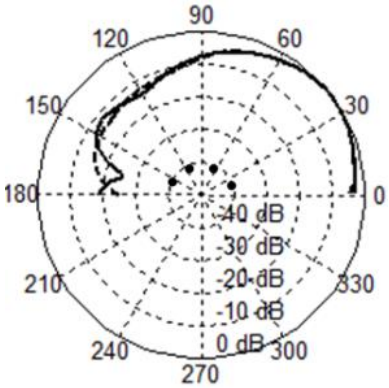
Pattern	Initialled patterns	Consumed time in ms (T); Iterations (K);	Synthesis patterns
1		 T:479, K:38 MSE=0.00032	
2		 T:535,K:30 MSE=0.0167	
3		 T:418, K:63 MSE=0.0032	
4		 T:516, K:21 MSE=0.0075	

Table2:

Desired P / Synthesized P	Linear array (LA)	Conformal array (CA)	MSE Comparison
Pattern 1			LA:0.00136 CA:0.00163
Pattern 2			LA:0.078146 CA: 0.004511
Pattern 3			LA:0.036153 CA: 0.010055
Pattern 4			LA:0.114156 CA: 0.007925