Published in IET Microwaves, Antennas & Propagation
Received on 2nd February 2012
Revised on 25th August 2012
doi: 10.1049/iet-map.2012.0068

ISSN 1751-8725

Self-recoverable antenna arrays

M. Joler

Faculty of Engineering, University of Rijeka, Rijeka 51000, Croatia
E-mail: mjoler@riteh.hr

Abstract: An approach is proposed for a self-recovery algorithm (SRA) to be embedded in an autonomous and reprogrammable controller device that would be used to monitor a system, sense failures of the system components and find a solution to recover the functionality of the system as much as possible. The approach presented here is on the scenario of self-recoverable antenna arrays, perceived as one of the vital interests in today's world of ubiquitous communication. The SRA is presented when run on a desktop computer and an field programmable gate arrays (FPGA) chip, to show the effectiveness of the approach on a few examples.

1 Introduction

In the present day, antenna arrays can be broadly classified into three major categories in terms of their level of sophistication. Non-adaptive arrays are typically present in terrestrial broadcasting systems. Once mounted, these arrays are characterised by an invariant radiation pattern. Unlike them, 'smart antennas', which can either be referred to as a 'switched beam'– or an 'adaptive beam' – type, are intended for deployment in mobile phone communications [1]. The third category in this broad classification is referred to as 'reconfigurable antennas' [2–4], which are meant to change some of their parameters during operation, such as resonance frequency, radiation pattern, or polarisation.

Whatever the category of an antenna array in a particular case is, current solutions are made on the premise that all elements of an antenna array will remain fully functional during operation. However, if any element of the antenna array fails to maintain its normal operation because of any possible cause (e.g. moisture in the feed cables or space debris hitting an antenna array on a satellite system), the original radiation pattern may degrade, possibly severely. The subscribers are then affected by a substantial loss of signal and the antenna array repairment can either be too costly or too time-consuming, especially for spaced-based systems or time-critical operations as in the battlefield.

The motivation for this work is to create a solution that will enable self-recovery of the flawed radiation pattern of an antenna array in the case of failure of any array element(s) on the basis that 'healthy' array elements will compensate as much as possible for the loss of some array elements. This idea is analogous, for example, to the way the human brain works: when one part of the brain is damaged because of some accident or stroke, the undamaged part of the brain takes over some functions that were originally processed by the now-damaged part of the brain, thus partly compensating for the loss of normal brain capacity and reducing the overall damage. This functional compensation is done by some process of learning and training of the healthy part of the brain – of the system, in general.

It is known from the antenna array theory [5] that a radiation pattern depends on the excitation magnitude and phase as well as the locations of the antenna array elements. Owing to arbitrariness of the array layout (especially in the case of a random element failure), it is a challenging problem to tackle, even when numerical approaches are utilised.

A handful of researchers have addressed similarly defined problems in the past. In [6], a conjugate gradient-based algorithm was proposed to minimise the sidelobe level by reconfiguring the amplitude and phase distribution of the properly working array elements. Mailloux [7] proposed a method of replacing the signals from the failed array elements, which was applicable only to a receiving array. In [8], three mating schemes were examined within a genetic algorithm (GA)-based method proposed for an array failure correction. In [9], a GA was used to 'find' defective elements in a planar array. In [10], a degradation of the radiation pattern caused by the excitation coefficient error was analysed. In [11], an artificial neural network was used to 'find' faults in antenna array, whereas in [12], a support vector machine was used to detect the location and the level of failure of failed elements within a planar array. Recently, FPGA was utilised to control switches in a single reconfigurable antenna [13, 14] in such a manner that a control signal is generated to change the switch state.

Unlike the pertinent works cited above, this paper primarily focuses on developing a self-recovery algorithm (SRA) that is suited for embedding into a controller device which would be able to monitor, analyse and correct malfunctioning of the given system, such as an antenna array.

The rest of the text is structured in the following order: Section 2 describes the antenna theory that is pertinent to this work, then a discussion on compensation for the effects of mutual coupling (MC) between the array elements, followed by the principal idea of the SRA and the GA that is created for this problem. Section 3 presents a few test cases that show the results obtained by testing the SRA on the desktop computer and discusses embedding of the SRA into an FPGA chip, with a goal to serve as an on-site controller of an antenna array. The paper concludes with a discussion on possible future developments and a summary of the work.

2 SRA

2.1 Pertinent antenna array theory

The radiation pattern of an arbitrarily configured antenna array of N antenna elements can be calculated knowing the (x, y, z) coordinates and the excitation of each array element $a_i = |a_i|e^{j\alpha_i}$, where $|a_i|$ and α_i are the magnitude and the electrical phase, respectively. In practice, arrays typically comprise of equal antennas, which partly simplifies the calculation since the radiation pattern can be calculated as the product of the radiation pattern of a single array element, E_0, and the so-called array factor (AF), which contains the vector sum of the individual contributions to the overall radiation. The total radiation pattern of an array placed in a spherical coordinate system is calculated as [5]

$$E(\theta, \phi) = E_0(\theta, \phi)\ \mathrm{AF}(\theta, \phi) \qquad (1)$$

where

$$\mathrm{AF}(\theta, \phi) = \sum_{i=1}^{N} a_i e^{j\psi_i(\theta,\phi)} \qquad (2)$$

where ψ is the phase shift because of the position of an ith array element with respect to the given direction of observation and the phase centre of the array.

If an ith array element is located at (x_i, y_i, z_i) and the far point of observation has the coordinates (x_0, y_0, z_0), the phase ψ of the ith element (i.e. the geometrical phase) is evaluated by

$$\begin{aligned}\psi_i &= \beta\vec{R}_i \cdot \vec{R}_0 \\ &= \beta(x_i \sin\theta\cos\phi + y_i \sin\theta\sin\phi + z_i\cos\theta)\end{aligned} \qquad (3)$$

where

$$\beta = \frac{2\pi}{\lambda}$$

$$\vec{R}_i = x_i\hat{x} + y_i\hat{y} + z_i\hat{z}$$

$$\vec{R}_0 = x_0\hat{x} + y_0\hat{y} + z_0\hat{z}$$

$$x_0 = \sin\theta\cos\phi$$

$$y_0 = \sin\theta\sin\phi$$

$$z_0 = \cos\theta$$

where β is the phase constant of the medium and λ is the wavelength of the signal. Thus, substituting (3) into (2), the radiation pattern of an arbitrarily laid out array comprising equal antenna elements can be computed using (1).

2.2 Mutual coupling effects between array elements

It is known that there also exists an additional effect of MC between the array elements [5, p. 478, 481], [15–17]. The effect of MC can be presented by the MC matrix $\boldsymbol{C}$, and compensated for by multiplying the vector of (uncompensated) array excitations $\boldsymbol{a}$ by the inverse of the MC matrix. For transmitting arrays, that are of primary concern for this paper, it can be explained this way: in an ideal case with no MC effect, the array is fed the excitation vector $\boldsymbol{a}$. When the MC effects are included, they essentially modify the original excitation to obtain the acting excitation vector $\boldsymbol{a}^{\mathrm{act}} = \boldsymbol{Ca}$, which alters the originally designed radiation pattern. To compensate for the effect of MC, one is to adjust the original excitation $\boldsymbol{a}$ by multiplying it by the inverse of $\boldsymbol{C}$, that is $\boldsymbol{a}^c = \boldsymbol{C}^{-1}\boldsymbol{a}$. The acting excitation vector will then effectively be $\boldsymbol{a}^{\mathrm{act}} = \boldsymbol{Ca}^c = \boldsymbol{C}(\boldsymbol{C}^{-1}\boldsymbol{a}) = \boldsymbol{a}$, which is equivalent to the ideal excitation vector with no MC effects present. It is generally difficult to evaluate the MC of antenna arrays analytically, but $\boldsymbol{C}$ can be determined by measuring either the radiation pattern of the array elements or the generalised scattering matrix of the array [15, 16, 18–20], or by making use of a full-wave solver to compute either the scattering matrix $\boldsymbol{S}$ or the mutual impedance matrix $\boldsymbol{Z}$ of the array, either of which can then serve to compute $\boldsymbol{C}$. Since $\boldsymbol{C}$ depends only on the radiating and scattering properties of the array elements and their locations within the array, it can be computed only once and be valid for any set of array excitations [15].

As it would be expensive for the purpose of this analysis to assemble all the array elements with feed cables, digitally controlled attenuators and phase shifters, the MC effects were analysed, and $\boldsymbol{C}$ evaluated, by the use of the CST Microwave Studio (MWS) full-wave electromagnetic solver, as illustrated in Fig. 1 for the case of a 4 × 4 array. Computation of all S-parameters was performed and the most characteristic values chosen along the first row and the major diagonal are shown in Table 1. One can see that for the current interelement spacing set at $d = \lambda/4$, the coupling between adjacent row elements in terms of S-parameters is about −7 dB and decays for about 7 dB for every next element in a row, if it is an outer row, whereas the inner rows have weaker coupling, as seen from the value of $|S_{11,\ 3}| = -21.52$ dB. The first array element along the major diagonal ($|S_{6,\ 1}|$) has −15.77 dB coupling, which further decays as the distance between the two observed elements increases. The overall coupling effect of an array can be compensated for by using the compensation procedure explained next.

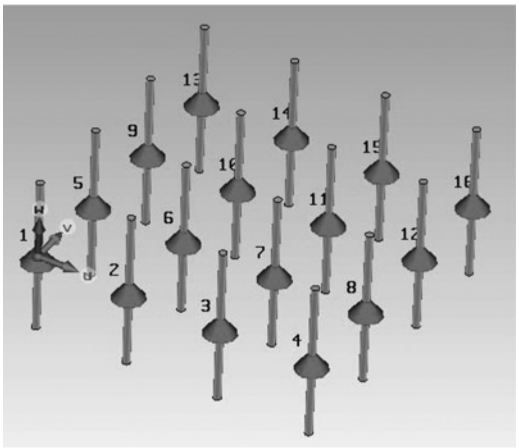

Fig. 1 *Layout of a 4 × 4 array modelled in CST MWS*

Table 1 A 4 × 4 dipole array: coupling values of some row and diagonal elements at 523 MHz

$\|S_{i,\,j}\|$	$\|S_{2,\,1}\|$	$\|S_{3,\,1}\|$	$\|S_{4,\,1}\|$	$\|S_{6,\,1}\|$	$\|S_{11,\,1}\|$	$\|S_{16,\,1}\|$	$\|S_{11,\,3}\|$	$\|S_{3,\,2}\|$	$\|S_{4.2}\|$
dB	−8.6	−15.04	−21.54	−15.77	−22.86	−26	−21.52	−7.21	−15.01

For an N-element array, $\boldsymbol{C}$ is a full $N \times N$ matrix, which is computed by [20]

$$\boldsymbol{C} = (Z_A + Z_L)(\boldsymbol{Z} + Z_L\boldsymbol{I})^{-1} \tag{4}$$

where Z_L is the generator impedance in each element (load impedance if in the receiving mode), Z_A is the antenna impedance, $\boldsymbol{Z}$ is the mutual impedance matrix and $\boldsymbol{I}$ is the identity matrix. If the S-parameter matrix is available instead of $\boldsymbol{Z}$, it can be coverted to $\boldsymbol{Z}$ using [21]

$$\boldsymbol{Z} = (\boldsymbol{I} + \boldsymbol{S})(\boldsymbol{I} - \boldsymbol{S})^{-1} \tag{5}$$

Matrix $\boldsymbol{Z}$, calculated by the MWS solver, is then plugged into (4) to obtain the value of $\boldsymbol{C}$.

Finally, the corrected values of the excitation vector that account for the effects of MC are calculated by

$$\boldsymbol{a}^c = \boldsymbol{C}^{-1}\boldsymbol{a} \tag{6}$$

where $\boldsymbol{a} = [a_1, \ldots, a_N]^T$ is the vector of uncompensated (i.e. original excitations) and $\boldsymbol{a}^c$ is the vector of array excitations that are corrected for the MC effects. An example of the implementation of the above procedure is given in Section 3.

A parameter-sweep analysis was performed to examine how MC decays with an increase in interelement spacing. The interelement spacing d of a 3-element linear array was swept in six steps, from λ/4 to λ/2, and the results are presented in Table 2. The S-parameter values show that the MC has a minor effect on arrays with larger interelement spacing of about λ/2, while it is also known that MC has a weak effect on larger arrays [5, p. 481], [18].

However, although determining the MC matrix $\boldsymbol{C}$ of the 'pure' antenna array (i.e. free of the tower construction, cables and other components such as power splitters) is a nice laboratory task, the ultimately true MC matrix could be determined only when the array is mounted in a realistic place, that is—on the broadcasting tower with all the feed cables and power splitters, which is then a subtle task to either model in a full-wave solver or measure at the Tx/Rx site.

In the author's opinion, the advantage of having a solution that can self-recover the radiation pattern of any array in case of a partial array failure during its run-time, before a crew is able to arrive and repair the system, especially for the systems that are on remote locations, surpasses the imperfection of exactly compensating for the excitation vector for the effects of MC.

2.3 The Principal idea of SRA

Considering a possible approach to provide a self-recovery solution, one's early idea may be to use the look-up table (LUT) as the fastest response. Such an LUT would have prescribed solutions to any possible failure pattern. Although an LUT would provide the fastest recovery response to some failure scenarios, there are some drawbacks to it, such as:

- a matrix of all possible recovery solutions should be computed in advance for 'every' particular antenna array, that is, for 'every' antenna site where the recovery solution is to be implemented for 'every' failure pattern;
- a complete LUT requires storage of a large number of data sets in the memory of a controller device that is assumed to be mounted on the site of the antenna array.

Another viable approach may be to have an algorithm that will be generally applicable and the characteristics of this approach will be that:

- One can use the 'same' algorithm for 'any' particular antenna array (i.e. at 'any' site) and 'any' failure pattern;
- In the memory of the on-site controller device, one has to store only the algorithm itself and the initial parameters of the healthy array, as a reference to be used during the recovery-solution-finding process.

A GA is found to be beneficial to this scenario.

From (1) to (3), it is evident that the radiation pattern of an array can be modified by three parameters: the locations (x_i, y_i, z_i) of the array elements, the excitation magnitudes $|a_i|$ and the electrical phases α_i of the excitation. In this work, it is assumed that, irrespective of the array layout, the array elements have fixed positions, which is the case for most arrays in practice. Thus, two degrees of freedom remain to affect the radiation pattern: the magnitude and the electrical phase. Throughout this work, it is assumed that the information on the failed elements is given from a failure detector (whose implementation is outside the scope of this paper) that would have N input ports being fed small portions of the signals feeding the array elements (e.g. through directional couplers). Converting those N input signals into a binary sequence where the negligible signal level of the failed element is interpreted as binary '0' and the signals from the healthy elements are interpreted as '1,' the detector can then convert the input binary sequence into a position of the failed element within the array and feed that information through its output port into the FPGA board for further processing. If any array element stops delivering power because of some malfunction, the

Table 2 MC analysis as a function of interelement spacing in a 3-element array of dipoles at 529 MHz

d/λ	0.25	0.3	0.35	0.4	0.45	0.5
$\|S_{2,\,1}\|$, dB	−7.33	−8.06	−9.33	−10.61	−11.96	−13.21
$\|S_{3,\,1}\|$, dB	−13.76	−15.24	−16.75	−17.92	−19.07	−19.85

doi: 10.1049/iet-map.2012.0068

radiation pattern of the array will change, possibly severely. Upon the current scenario, even if a faulty element works at some reduced power, it is deemed entirely faulty and its magnitude is taken to be zero, that is, $|a_k| = 0$, where k is the number of the faulty array element.

The initial set of excitation magnitudes and phases is stored. When a flawed element is detected, the SRA first computes the radiation pattern of the flawed array by (1) and compares it with the original radiation pattern. If the average error between the two patterns is greater than some preset amount that is tolerable, the SRA starts searching for a new set of magnitudes and phases that will feed the remaining fully functional array elements and create a radiation pattern that will be as close as possible to the original radiation pattern. The SRA in (2) works with the excitation vector that is uncompensated for by the MC effects ($\boldsymbol{a}$), but if $\boldsymbol{C}$ is available, the correction of the newly found excitation vector for the MC effects is simply performed by applying (6).

2.4 Use of GA for self-recovery

Following the approach of having a generally valid algorithm, GA was chosen to support the task (see, e.g. [22]). Each array element is presented here by two genes—one for the excitation magnitude and another for the phase. A single chromosome, that represents one possible recovery solution, is thereby composed of $2N$ genes. Since it is convenient to do the binary arithmetic in digital circuits and in GA, it had to be decided on resolution of the excitation magnitude and the electrical phase. Since it is generally impractical to adjust the power in a wide range, 16 levels of magnitude were estimated to suffice. The magnitude information is therefore presented with 4 bits. The phase is a more effective parameter to shape the radiation pattern because the pattern is more responsive to variations in the phase. It is hence desirable to have a finer resolution of phase, because of which it was chosen to encode the phase with 8 bits, which corresponds to 2^8 discrete levels in the span of 360°. This provides a phase resolution of 1.4°, which is very satisfactory for most practical purposes and it can be realised by off-the-shelf digitally controlled phase shifters [23, 24].

The SRA is based on a 'steady-state' GA [22] with the chromosomes encoded from real-value parameters to spare the memory resources and reduce the number of operations that are needed to encode the parameters. Moreover, Grey coding of the genes is used in the SRA to improve the solution-search process. In that, the chromosomes are encoded from real values into binary coded decimals and then into Grey-coded binaries. Before evaluating the fitness function of the GA, the reverse process takes place. On the other hand, the 'crossover' and 'mutation' operations are performed directly on the binary words. The selection of the parents for the operation of GA mating is based on the 'tournament scheme' and the mating is realised as a random two-point crossover. The crossover in the SRA is always done with 100% probability as it was found to be beneficial for the quality of the solution (a similar experience was reported in [8]). The mutation, as a means of exploring the solution space in a better manner and preventing the premature convergence of the algorithm to a suboptimal solution, is here realised by the two common parameters—the 'probability of mutation' and the 'rate of mutation', complementing the bit values at random bit positions within the randomly selected chromosomes. If a slow convergence towards a better solution is observed during the execution of the SRA, the 'rejuvenation' operation is applied in the GA to genetically refresh the GA population. It is done by replacing most of the population with new chromosomes, thus injecting fresh genetic material that may help exploration of the solution space. The 'elitist' strategy is also used, which guarantees that the fittest individual in each GA generation is preserved and inserted into the next GA generation, to ensure at least as good a fitness as the current generation had. An additional GA tactic that was applied was to encode the state of the flawed array in the initial GA population, whereas the rest of the population was then filled with randomly generated chromosomes. Choosing the initial GA generation this way has shown speeding up of algorithm convergence and leads to better solutions.

Although the specific values of the SRA parameters that were used will be listed with every case discussed in Section 3, from a number of computer runs that were conducted, the following values were found to achieve best solutions on average: 100% crossover probability, 2% mutation probability with 15% mutation rate, population size of 100 chromosomes (for the 4×4 array size used here), and 50% population replacement rate.

Having received information about a flawed array element (from the sensing circuitry that is assumed to be part of the system), the SRA computes the radiation pattern of the flawed array and calculates the 'average error' of the flawed radiation pattern with respect to the original radiation pattern as

$$e = \sum_{j=\phi_{\text{start}}}^{\phi_{\text{end}}} w_j |\dot{E}_{jo} - \dot{E}_{jf}| \tag{7}$$

where j takes the values of the scanning angle ϕ from its start value ϕ_{start} to end value ϕ_{end} in the xy-plane at every 1°, w_j is the weight factor whose purpose is to discriminate between the more important sectors of the radiation pattern from the less important ones, $\dot{E}_{jo}$ is the complex value of the electric far-field of the 'original' pattern at angle j, whereas $\dot{E}_{jf}$ is the complex value of the electric far-field of the 'flawed' pattern at the same angle. Then, if $e >$ tol, where tol is the acceptable tolerance (e.g. 1.5 dB), the SRA starts searching for a new set of array excitations that will produce e to be within the tolerance. On the contrary, if $e <$ tol, the remaining original excitations are maintained because the flaw in the array did not corrupt the radiation more than the acceptable standards. That logic stems from the fact that not every element in an array has an equal impact on the overall radiation pattern and in spite of the flaw of some element(s) in the array, the error e may still be smaller than the tolerable level.

As each chromosome in the GA population represents a set of the excitations for the entire array, by decoding of the genes of each chromosome, (1) can be computed to evaluate the radiation pattern that would be created by each chromosome in GA generation. The radiation pattern of each chromosome is compared with the radiation pattern of the healthy array and the discrepancy (7) between the two arrays is calculated. The chromosomes are then ranked according to their fitness to the original array, which constitutes the base from which the next GA generation is formed. The SRA stops if there is a chromosome which satisfies $e <$ tol and the difference between the smallest e's in the two consecutive GA generations is smaller than some user-defined margin (the latter ensures that the SRA will not stop when the very first solution that satisfies $e <$ tol comes up, but keep searching for the best possible solution and stop only when the difference in the solution quality in two consecutive generations is negligible to be worth running the SRA any longer or the maximum number of GA generations is reached).

3 Self-recovery test cases

3.1 SRA testing on the computer

The results obtained by such an SRA are now discussed on a few array cases and failure scenarios (first without a compensation for the MC, and then with it). The common setting for all examples here is that a planar array is placed in the *xy*-plane and comprises of vertically polarised half-wave dipoles equidistantly separated by λ/4. Fully operational dipoles are each displayed as a vertical line with a small circle in the centre of the dipole and its location number in the array, whereas at the flawed dipoles there is an empty spot and the dipole number is displayed, to indicate its position in the array, but suggest it is not radiating. The radiation pattern in each case is computed in the horizontal plane, that is, at $E(90^\circ, \phi)$, where ϕ is the scanning angle. In the following graphs that compare the radiation patterns of the 'original'-, 'flawed'- and 'recovered'-array, the last two are each normalised with respect to the 'original' radiation pattern. That way, one can track the effects that element failure and the SRA solution have on the radiation pattern with respect to the original condition. The flawed array element(s) is/are randomly selected within the SRA and then the SRA starts the analysis as described in Section 2.4.

1. *Uniform array:* The first example presented is a 4 × 4 'uniform' array (i.e. comprising two uniform broadside arrays along the *x*- and *y*-axes). The faulty element was randomly selected to be at position #6, as illustrated in Fig. 2, according to the convention for graphing that was outlined above. The array settings are

$$|a_i| = \begin{cases} 1 & \text{for } i = 1, 2, \ldots, 16 \quad \text{and} \quad i \neq k \\ 0 & \text{for } i = k = 6 \end{cases}$$
$$\alpha_i = 0 \text{ for } i = 1, 2, \ldots, 16$$

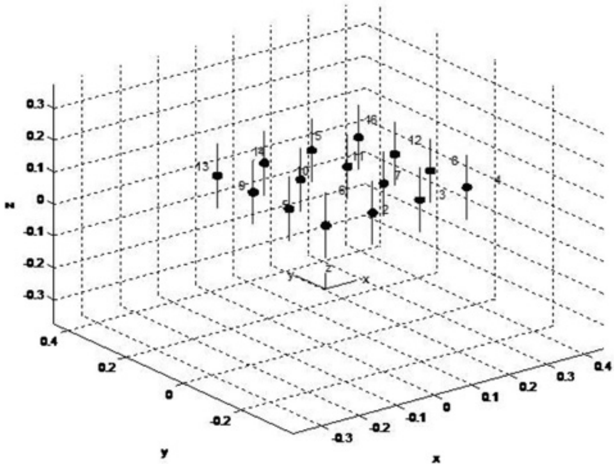

Fig. 2 *Layout of a 4 × 4 array with element #6 failed (represented as an empty space on location #6)*

Fig. 3 shows the comparison of the original-, flawed- and recovered-radiation pattern, improvement in the fitness function during the solution-finding process and the magnitudes and phases of the recovery solution, respectively. The original pattern has a maximum at $\phi = 45^\circ$, which is expected when two broadside arrays are placed orthogonally to each other, as is the case here. When element #6 failed, the original radiation pattern, on average, changed more than the tolerable level of 1.5 dB that was allowed here. With the crossover of 100%, mutation probability of 0.7%, population replacement rate of 50% and using rejuvenation, but without the use of Grey coding,

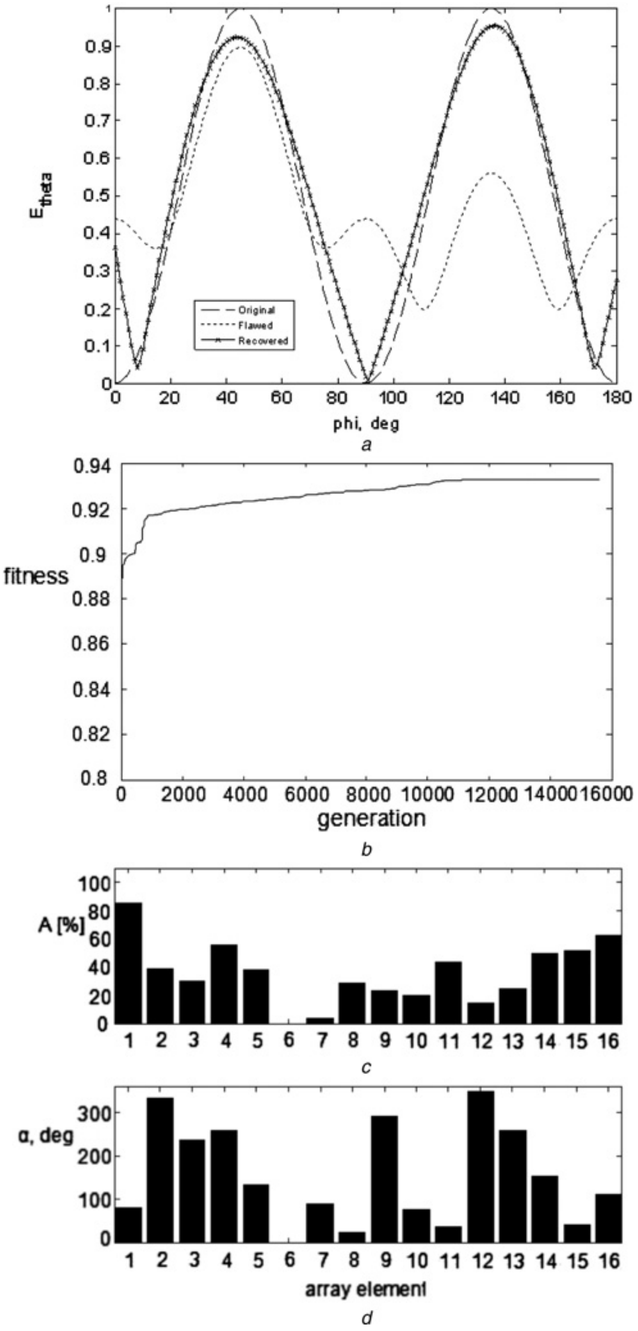

Fig. 3 *Recovery of a 4 × 4 uniform array with element #6 failed*

a Radiation pattern comparison
b Convergence of fitness
c Recovery solution excitation magnitudes
d Recovery solution excitation phases.
GA parameters: crossover, 100%; mutation probability, 0.7%; population replacement rate, 50%; use of rejuvenation, but no use of Grey coding

doi: 10.1049/iet-map.2012.0068

a very close fit to the original pattern was achieved when the SRA was run over almost 16 000 generations to reach a fitness value of 0.93 (Fig. 3*b*). Although a GA can produce a different solution every time it is run, because of its random nature, and it is hard to reproduce the exact result because of that particular magnitudes and phases for this 'recovered' pattern are shown in Figs. 3*c* and *d*. The magnitude is shown in percentages of the original magnitude, whereas the phases are directly in electric degrees. The magnitude of element #6 is 0 as it is the flawed element, whereas its phase is actually irrelevant after that, but also set to zero, for simplicity. It can be seen that a healthy array elements takes very different phases in order to recover the radiation and that kind of non-intuitive solution is inherent in optimisation algorithms that have a stochastic character, such as GA.

Although typical optimal values of GA parameters that will lead to satisfactory solutions for most problems are known from practice, GAs still leave a lot of room to adjust the value of any parameter to values that are advantageous for the particular application of interest. Thus, to reduce the number of GA generations that were needed to achieve the solution in Fig. 3*a* and still obtain a satisfactory recovery solution, the following adjustments were applied: the Grey coding is now used and 50% of the weight factors w_j are set to be within $\phi = 0^\circ, \ldots, 90^\circ$, thus directing the GA to find a solution that will ensure a good fit within the sector of the major lobe. With this adjustment, an excellent recovery solution, shown in Fig. 4, was found in 4500 generations on a population size of 100, for the same case of the array and element failure as in Fig. 3. This shows a good effect of Grey coding on solution convergence and algorithm time efficiency. This reduction in the number of generations is not that critical when the SRA is run on a desktop PC, since the solution is found in a matter of seconds even for larger arrays; however finding a solution with a smaller number of GA generations is of great interest when it comes to embedding such a recovery algorithm into an autonomous controller that does not have so much processor power and memory resources. Testing the SRA within a reprogrammable controller was the reason to pick moderate-size arrays in this work and searching the best combination of the GA parameters was further pursued in order to see how much the number of GA generations could be further reduced without losing any or most of the quality of the recovery solution.

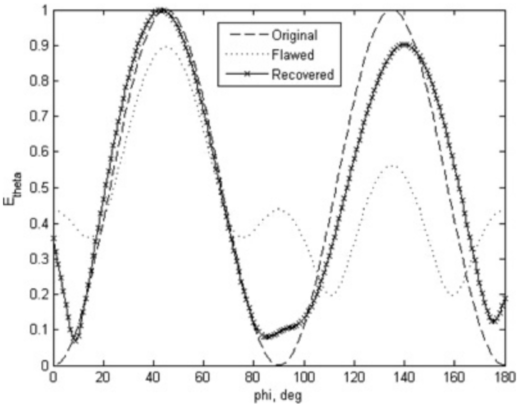

Fig. 4 *Faster recovery solution to a 4 × 4 uniform array with element #6.*

GA parameters: crossover, 100%; mutation probability, 0.9%; mutation rate, 15%; population replacement rate, 50%; population size, 100; generations, 4500; and Grey coding used

When the GA parameters were further set to values of crossover at 70% (though not the typical value here), mutation probability at 2%, and population replacement rate at 70%, the recovery solution shown in Fig. 5 was found in only 60 generations, which is very encouraging in the context of embedding the SRA into an external controller.

2. *Binomial array:* The SRA was also tested for a 4 × 4 'binomial' broadside array. The magnitudes $|a_{mn}|$ of a planar binomial array are esentially a product of the magnitudes of the linear arrays lying along the x- and y-axes, respectively. The magnitudes of the two linear arrays are

$$\begin{array}{lcll} |a_m| & = & 1331 & \text{for } m = 1, 2, 3, 4 \text{ along } x - \text{axis} \\ |a_n| & = & 1331 & \text{for } n = 1, 2, 3, 4 \text{ along } y - \text{axis} \\ |a_{mn}| & = & |a_m| \cdot |a_n| & \text{for } m, n \text{ as above} \\ \alpha_{mn} & = & 0 & \text{for } m, n \text{ as above} \end{array}$$

Thus, the four central elements of the array have the highest impact on the overall radiation, followed by the two inner elements on the edges and then the corner elements. Consequently, the largest damage to the radiation is done if any of the four central elements fail. This time, the recovery solution was searched for the case that two of the central elements failed, in particular: elements #6 and #7 (i.e. 2nd and 3rd element in the second row, according to our enumeration in Fig. 2). The comparison of the radiation patterns is shown in Fig. 6. This recovered pattern is close to the original pattern, especially given the fact that two of the four major elements were flawed and that the solution was found in only 15 GA generations. The following GA parameters were used: population replacement rate at 50%, crossover probability at 100%, mutation probability at 2% and mutation rate at 15%.

The effect that the recovered pattern exceeds the field value above 1 seems implausible at first, but since all the patterns here are normalised with respect to the original pattern, it is possible for some degrees of the scanning angle that the recovered pattern may provide a higher field than the original pattern, while having a lower field at most of the other range of the scanning angle. In Fig. 6, the recovered pattern has a higher value than the original in the sector $\phi \in [110^\circ, 150^\circ]$ and slightly in $\phi \in [50^\circ, 70^\circ]$, while it is lower than the original elsewhere, especially in the sector $\phi \in [0^\circ, 50^\circ]$.

Overall, the success of the SRA depends on the number – and locations – of the failed elements, but also on the type of

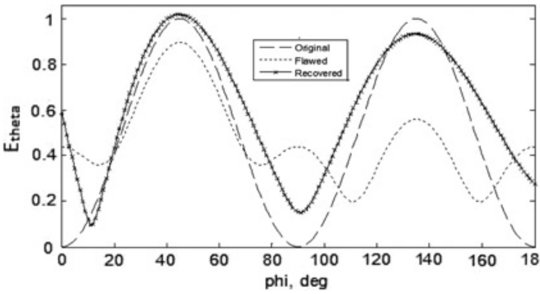

Fig. 5 *Very close-fit recovery solution for a 4 × 4 uniform array was achieved in only 60 generations*

GA parameters: crossover, 70%; mutation probability, 2%; population replacement rate, 70%

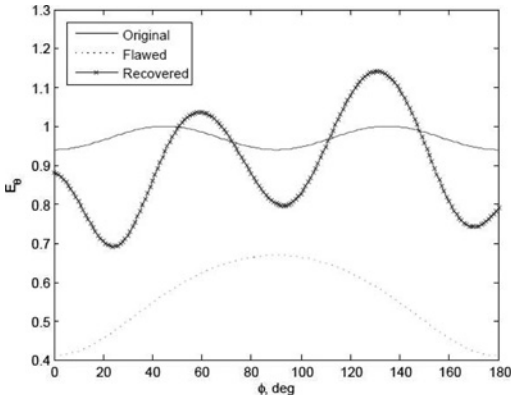

Fig. 6 *Comparison of the radiation patterns for a 4 × 4 binomial array with elements (#6, #7) being flawed*

GA parameters: 15 generations, crossover, 100%; population replacement rate, 50%; mutation probability, 2%; and mutation rate, 15%

the array. It was found that a good recovery solution was harder to find for the uniform linear arrays than for the binomial or planar arrays as the GA tends to keep the original excitations of the uniform linear array with only minor changes in the phase values, which does not affect the radiation pattern as much as preferable.

3. *Compensating for the effects of MC:* To illustrate how the effects of MC can be compensated for, as explained in Section 2.2, let us take a 4-element binomial broadside linear array. A numerical computation was run in MWS to obtain $\boldsymbol{Z}$ and store it in SRA for the computation of $\boldsymbol{C}$ using (4), where $Z_L = 50\ \Omega$ and $Z_A = 73.1291 + \text{j}42.5445\ \Omega$ were used. Array element #3 is assumed to eventually fail and then the SRA computes the pattern of the flawed array. Since e, as given by (7), was below the tolerance level, the self-recovery procedure is run to obtain a new value of $\boldsymbol{a}$. Fig. 7 shows the 'original' and 'flawed' radiation patterns simulated in MWS when the array is fed the original excitation vector $\boldsymbol{a}$, the 'recovered' pattern ('SRA uncomp') using the uncompensated excitation a_{rec} and the 'recovered' pattern ('SRA comp') generated by the MC-compensated excitation a^c_{rec} that was computed using (6). All the curves are normalised with respect to the 'original' radiation pattern. Since all the curves were obtained by MWS, the MC coupling effects affect all of them. It is evident that the MC-compensated curve comes closer to the original pattern than the MC-uncompensated curve. Since this is a smaller array with a small interelement spacing d (which is the hardest case), MC coupling has the strongest possible effect on the radiation pattern, affecting both the shape and the level of the pattern. In case of small arrays with larger d and large regular arrays, the MC will preserve its shape, requiring only scaling up or down of the pattern magnitude [5, p.481].

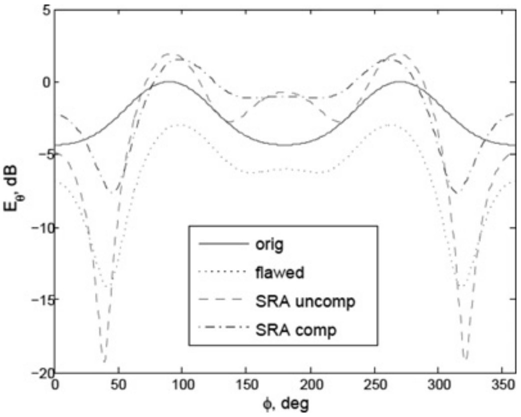

Fig. 7 *Comparison of 4-element binomial broadside linear array radiation patterns, as simulated in MWS: original and flawed array pattern, the uncompensated- and the MC-compensated pattern*

3.2 Embedding the SRA into an autonomous system

As mentioned earlier, the motivation for this work was to investigate the means of creating an intelligent and autonomous controller device that would monitor the condition of an antenna array (with the help of some sensing circuitry), find and apply a recovery solution in the case that the array radiation degraded because of a failure of any array element. In fact, such a solution could then not only be used in cases with no failure, but also to reconfigure the radiation characteristics according to a change in the characteristics of the communication channel and so on.

To test the SRA within one such controller, we embedded the SRA into an Altera DE2 development board [25] containing the Cyclone II FPGA shown in Fig. 8.

By running the SRA on the FPGA chip for a few array cases, it was possible to establish a relationship for the FPGA-based solution computation time T as

$$T \simeq \frac{K}{60}(2N)cg[\text{min}] \qquad (8)$$

where K is some constant in $\frac{\text{seconds}}{\text{gene}\cdot\text{generation}}$ that depends on the speed of the particular FPGA chip, N is the number of array elements (i.e. $2N$ is the number of genes in one chromosome), c is the GA population length (i.e. the number of chromosomes) and g is the number of GA generations that are necessary to reach a satisfactory solution. For the Cyclone II chip that was used in this work $K = 0.5\frac{\text{s}}{\text{gene}\cdot\text{generation}}$. Although K will be different for the other FPGA boards, it is assumed that (8) will still hold, but we were not in possession of the other FPGA boards to verify its validity. The Cyclone II FPGA chip has

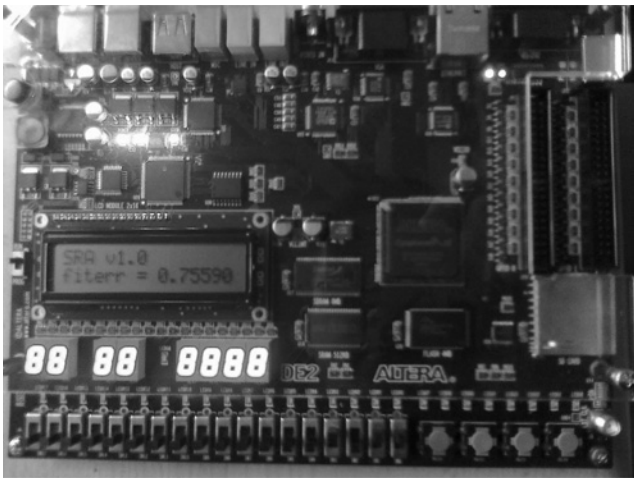

Fig. 8 *Altera DE2 development board with Cyclone II FPGA chip, running the SRA*

meanwhile been superseded by six newer Altera FPGA models and T is expected to be substantially smaller. In any case, (8) indicates that to reduce T, a tradeoff between c and g is possible in a way that is most beneficial to produce a good recovery solution. In the additional experiments to see how much it is possible to lessen c- and g-values and still get a satisfactory solution for the same N, we reached the ($c = 40$, $g = 15$) pair of values. The recovered pattern in Fig. 6 was found based on this setting. In contrast, the pair ($c = 60$, $g = 10$) did not provide good solutions. Although the latter had a larger GA population, ten generations of running the SRA was simply not enough to find a good solution. The experiments also showed that when g is lowered, the mutation probability has to be increased to enable a good enough solution.

4 Discussion and conclusion

In this work, an approach for creating a smart, reconfigurable and autonomous self-reconfigurable system was proposed on the basis of having a recovery-enabling algorithm, such as an SRA developed here, being then embedded in a reconfigurable controller device such as an FPGA.

The approach was presented keeping in mind possible self-recovery of antenna arrays, but with advancement of technology and further development of this concept, we envision it could be implemented in other scenarios where smart and autonomous controllers are desired to respond to an unforeseen failure of any part of the system within a time that is practically short for the particular application. Such situations are anticipated at unattended broadcasting towers at high altitudes or residing far from the nearest service crew, for components in space-based systems, or equipment in the battlefield.

Successful recovery of some failure cases was illustrated along with embedding of a full-scale SRA into an FPGA and running it to find a recovery solution. Future FPGA devices will be capable of executing the SRA quite faster and provide recovery solutions in a substantially shorter time. Moreover, with more storage capacity, the overall SRA could be developed to include additional recovery algorithms (a LUT, as discussed in Section 1, and, for example, an additional neural network-based algorithm) and thus provide multiple levels of recovery response – from the fastest, but least general, to the most general, yet slower one. Depending on the failure case, the controller logic would decide which recovery algorithm to start (first) to find the (faster) recovery solution. Should the faster algorithm not give a satisfactory response, a smarter algorithm could still be started to achieve a better solution.

Although the quality of the SRA solution depends on the array layout, the original excitation type, the number of failed elements and the original weights of the array elements that failed, satisfactory recovery solutions can be obtained and the results presented in this work showed that SRA was able to achieve significant improvements in the shape of the radiation pattern after tuning the excitations of the healthy array elements, which gives promise for further development.

5 References

1 Rayal, F.: ‘Why have smart antennas not yet gained traction with wireless network operators?’, *IEEE Antennas Propag. Mag.*, 2005, **47**, (6), pp. 124–126

2 Jung, S., Lee, M., Li, G.P., de Flaviis, F.: ‘Reconfigurable scan-beam single-arm spiral antenna integrated with RF-MEMS switches’, *IEEE Trans. Antennas Propag.*, 2006, **54**, (2), pp. 455–463

3 Zhang, S., Huff, G.H., Feng, J., Bernhard, J.T.: ‘A pattern reconfigurable microstrip parasitic array’, *IEEE Trans. Antennas Propag.*, 2004, **52**, (10), pp. 2773–2776

4 Aissat, H., Cirio, L., Grzeskowiak, M., Laheurte, J.-M., Picon, O.: ‘Reconfigurable circularly polarized antenna for short-range communication systems’, *IEEE Trans. Microwave Theory Tech.*, 2006, **54**, (6), pp. 2856–2863

5 Balanis, C.A.: ‘Antenna theory: Analysis and design’ (Wiley-Interscience, 2005, 3rd edn.)

6 Peters, T.J.: ‘A conjugate gradient-based algorithm to minimize the sidelobe level of planar arrays with element failures’, *IEEE Trans. Antennas Propag.*, 1991, **39**, (10), pp. 1497–1504

7 Mailloux, R.J.: ‘Array failure correction with a digitally beamformed array’, *IEEE Trans. Antennas Propag.*, 1996, **44**, (12), pp. 1542–1550

8 Yeo, B.K., Lu, Y.: ‘Array failure correction with a genetic algorithm’, *IEEE Trans. Antennas Propag.*, 1999, **47**, (5), pp. 823–828

9 Rodriguez, J.A., Ares, F.: ‘Finding defective elements in planar arrays using genetic algorithm’, *Prog. Electromagn. Res.*, 2000, **29**, pp. 25–37, PIER

10 Nakazawa, S., Tanaka, S., Murata, T.: ‘Evaluation of degradation of shaped radiation pattern caused by excitation coefficient error for onboard array-fed reflector antenna’. Proc. 2004 IEEE Antennas Propag. Soc. Int. Symp., 2004, vol. 3, pp. 3047–3050

11 Patnaik, A., Choudhury, B., Pradhan, P., Mishra, R.K., Christodoulou, C.: ‘An ANN application for fault finding in antenna arrays’, *IEEE Trans. Antennas Propag.*, 2007, **55**, (3), pp. 775–777

12 Nan, X., Christodoulou, C.G., Barbin, S.E., Martinez-Ramon, M.: ‘Detecting failure of antenna array elements using machine learning optimization’. Proc. 2007 IEEE Antennas Propag. Soc. Int. Symp., Honolulu, HI, USA, 2007, pp. 5753–5756

13 Zuraiqi, E.A., Joler, M., Christodoulou, C.G.: ‘Neural networks FPGA controller for reconfigurable antennas’. Proc. 2010 IEEE Antennas Propag. Soc. Int. Symp. 2010 USNC/CNC/URSI Meeting, Toronto, ON, Canada, 2010, pp. 1–4

14 Shelley, S., Costantine, J., Christodoulou, C.G., Anagnostou, D.E., Lyke, J.C.: ‘FPGA-controlled switch-reconfigured antenna’, *IEEE Antennas Wirel. Propag. Lett.*, 2010, **9**, pp. 355–358

15 Corcoles, J., Gonzalez, M., Rubio, J.: ‘Mutual coupling compensation in arrays using a spherical wave expansion of the radiated field’, *IEEE Antennas Wirel. Propag. Lett.*, 2009, **8**, pp. 108–111

16 Sadat, S., Ghobadi, C., Nourinia, J.: ‘Mutual coupling compensation in small phased array antennas’. Proc. 2004 IEEE Antennas Propag. Soc. Int Symp., 2004, vol. 4, pp. 4128–4131

17 Zamlynski, M., Slobodzian, P.: ‘Verification of the mutual coupling compensation technique in small planar antenna arrays’. 18th Int. Conf. on Microw, Radar and Wireless Comm. (MIKON), 2010, pp. 1–4

18 Steyskal, H., Herd, J.: ‘Mutual coupling compensation in small array antennas’, *IEEE Trans. Antennas Propag.*, 1990, **38**, (12), pp. 1971–1975

19 Rubio, J., González, M., Zapata, J.: ‘Generalized-scattering-matrix analysis of a class of finite arrays of coupled antennas by using 3-D FEM and spherical mode expansion’, *IEEE Trans. Antennas Propag.*, 2005, **53**, (3), pp. 1133–1144

20 Huang, Z., Balanis, C., Birtcher, C.: ‘Mutual coupling compensation in UCAs: simulations and experiment’, *IEEE Trans. Antennas Propag.*, 2006, **54**, (11), pp. 3082–3086

21 Pozar, D.M.: ‘Microwave engineering’ (John Wiley & Sons, Inc., 2005, 3rd edn.)

22 Haupt, R.L., Haupt, S.E.: ‘Practical genetic algorithms’ (Wiley-Interscience, 2004, 2nd edn.)

23 http://www.pulsarmicrowave.com, accessed April 2012

24 http://www.gtmicrowave.com, accessed April 2012

25 http://www.altera.com, accessed February 2012