Optimization of Phased Arrays Integrated with FSS and Feeding Elements based on Parametric Models

D.J. Bekers*, S. Monni*, S.M. van den Berg†, A.M. van de Water†, B.J. Morsink†, C. Alboin#, V. Ducros†, M. Celikbas†, J. Blanche*, N. Fiscante*, G. Gerini*, J-P. Martinaud#, M. Rochette†, G.H.C. van Werkhoven†

*TNO Defence, Security, and Safety, P.O. Box 96864, 2509 JG, Den Haag, The Netherlands
†Thales Nederland, Haaksbergerstraat 49, P.O. Box 42, 7550 GD Hengelo, The Netherlands
#Thales Airborne Systems, 2, Avenue Gay-Lussac, 78851 Elancourt Cedex, France
+Ansys France, Immeuble Einstein 11, Boulevard Einstein, F-69100 Villeurbanne, France
!Università degli Studi del Sannio, Piazza Guerrazzi, 1, 82100 Benevento, Italy

Keywords: Optimization, Phased Arrays, Frequency Selective Surfaces, Parameterization, Interpolation

Abstract

In this paper, we introduce a general framework to efficiently optimize phased array antennas integrated with Frequency Selective Surfaces (FSS) and feeding elements. The framework consists of two main parts: the first being parametric models that enable fast evaluation of the full-wave problem for a range of frequency, scan angle and geometry parameters. The second part consists of a number of optimization methods to generate an optimum set of geometry parameters by minimization of a penalty function. Results are shown for parametric models and the optimization of various examples.

1 Introduction

In the design of phased arrays with integrated FSS and feeding elements, the challenge is to synthesize a structure with a desired response in terms of reflection and transmission over specific ranges of frequency and scan angles. To meet this challenge, a CAD tool should be able to optimize the structure over the ranges of specified geometry parameters, such as the dimensions of antenna and FSS elements, thicknesses and material properties of dielectric layers, etc. The optimization process requires the evaluation of the response for many different settings of geometry parameters, frequencies and scan angles. Due to the complexity of the structure, each evaluation requires a general-purpose full-wave simulator. However, such a simulator usually requires far too much computation time per parameter point to allow an efficient optimization process. Therefore, we introduce a different strategy in this paper.

2 General Outline of the Optimization Strategy

In our optimization strategy we first construct a database of parametric models for components that can be used as building blocks for the configurations that we consider. These models facilitate the rapid evaluation of the scattering matrix of the corresponding building block in a specified range of frequency, scan angles, and geometry parameters by means of Kriging interpolation. The interaction between the building blocks is described by means of scattering matrices, which contain the accessible modes of a building block, i.e., the propagating or evanescent modes that have a significant magnitude at the input/output ports of that building block. To compute the parametric models, we equipped three different full-wave simulators with the same parametric techniques, which are based on Singular Value Decomposition and Reduced Order Modelling. Two simulators (Luxaflex and Twodim), based on different integral equation formulations, are dedicated to the analysis of infinite periodic layered structures. One formulation results in a Generalized Scattering Matrix representation [1] and the other one results in a Generalized Impedance / Admittance Matrix representation [2-3]. For a multilayer structure, consisting of several FSSs sandwiched between dielectric layers, these simulators can either address the scattering problem by solving the corresponding system of coupled integral equations or by resorting to an equivalent network representation of each layer and then cascading all the networks. Because of its modularity, the latter approach is particularly convenient as basis for the design. The third simulator (ADR2) is based on a boundary element method and can handle both periodic arrays and feeding elements. It must be emphasized that the availability of different parameterized tools allows an efficient computation, since we can choose the most suitable simulator instead of relying on a single general-purpose full-wave simulator.

To optimize a specific array, we identify first the corresponding building blocks. Next, we introduce a penalty function that describes the performance of the array with respect to the desired response. Different penalty functions are defined for typical FSS and antenna applications. Subsequently, we minimize the penalty function over the range of the geometry parameters by means of a chosen optimization algorithm. During the optimization process, the
penalty function is rapidly evaluated by first calculating the scattering matrices of the building blocks from their parametric models (by means of interpolation) and then by cascading these matrices.

In this paper, we focus in particular on the application of parameterization techniques in the optimization loop. For several optimization algorithms we demonstrate that the parameterization leads to an efficient optimization process.

3 Parameterization Techniques and their Application

The three different full-wave simulators have the following common aspects. Given a large parameter space of frequency $f$, scan angles $\phi$ and $\theta$, and shape parameters $p_1, \ldots, p_N$ for an FSS or a radiating structure, these simulators first compute a discretized quantity, e.g., the electric or magnetic current density, by calculating a matrix $L(f, \theta, \phi, p_1, \ldots, p_N)$ and a vector $v(f, \theta, \phi, p_1, \ldots, p_N)$, and by solving a linear system of the form

$$ \begin{align*}
L(f, \theta, \phi, p_1, \ldots, p_N) & \cdot u(f, \theta, \phi, p_1, \ldots, p_N) = \\
v(f, \theta, \phi, p_1, \ldots, p_N)
\end{align*} $$

(1)

Subsequently, the scattering matrix $S(f, \theta, \phi, p_1, \ldots, p_N)$ of the structure is computed from the solution $u$ of Eq. (1). If the structure is composed of different blocks that can be handled separately, the scattering matrices of these blocks are computed by calculating and solving matrix systems of the form Eq. (1) and, subsequently, by cascading these scattering matrices to obtain the scattering matrix of the complete structure. We emphasize that for certain blocks it may be efficient to solve a system like Eq. (1) analytically, e.g., for dielectric layers in a periodic structure or waveguide.

For all three simulators, the calculation of the scattering matrices from the solutions of the matrix systems and the cascading of these matrices is carried out relatively fast. The time-consuming step involves calculating and solving the matrix systems of the form Eq. (1). More precisely, for the two simulators for infinite periodic layered structures, calculating the (moment) matrix $L$ is computationally expensive, while solving the system Eq. (1) is very cheap. The reason is that, to compute the elements of $Z$, spectral sums need to be computed, which are slowly convergent. Although applied acceleration techniques have reduced the complexity significantly [1, Ch. 4], [2, Ch. 2-3], [4], the computational time is too large to consider many variations of the parameters. In contrast, for the simulator based on the boundary element method, calculating the matrix $L$ is relatively cheap, while solving the system Eq. (1) is computationally expensive. Therefore, to reduce the computational time of this simulator, we apply parametric and interpolation techniques to the solution $u(f, \theta, \phi, p_1, \ldots, p_N)$, while for the other two simulators, we apply these techniques to the matrix $L(f, \theta, \phi, p_1, \ldots, p_N)$. The advantage of the second alternative is that it allows for a greater geometrical flexibility. We can alter the geometry of the FSS elements and the radiating structures by removing basis functions from the moment matrices. Such an operation is impossible in case we apply the techniques to the solution $u$. For this reason we have also decided not to apply the techniques directly to the scattering matrix $S(f, \theta, \phi, p_1, \ldots, p_N)$, by which we would not only reduce the number of matrix evaluations, but also the number of scattering matrix evaluations and the number of cascade operations.

We explain the parametric techniques in more detail. Let the moment-matrix equation be given by Eq. (1), where all parameters have prescribed ranges. In case we apply the techniques to the matrix $L$, we cast the matrix and the right-hand side (RHS) $F$ of this equation in a row vector, $F=[L,v]$, by putting first the (transposed) columns of $Z$ one after the other and, next, the (transposed) RHS. In the other case, we set $F=u^T$. To any parameter point $(f, \theta, \phi, p_1, \ldots, p_N)$ in the specified ranges corresponds a row vector $F=(f, \theta, \phi, p_1, \ldots, p_N)^T$. Let $\Omega$ be the space of all row vectors. Then, the problem of generating rapidly either the matrix $L$ and the RHS $v$, or the solution $u$, for any parameter point $(f, \theta, \phi, p_1, \ldots, p_N)$ in the specified ranges is equivalent to generating rapidly any vector in $\Omega$. To accomplish this task, we employ the following strategy. We first construct a basis for $\Omega$ in a gradual way. To this end, we select an initial set of parameter points, typically 5 points, for which we generate the corresponding vectors $F$ by full-wave computations. On the obtained set of vectors, we perform a singular-value analysis. Next, we extend the set of parameter points, typically by twice as many points, and generate the vectors $F$ for the added points. Subsequently, we perform a singular-value analysis on the extended set of vectors. By comparing both singular-value analyses, we determine whether the amount of vectors is sufficient to span $\Omega$. We repeat the process of extending the set of parameter points until the amount of vectors is sufficient. These vectors represent a basis for $\Omega$: $F_m, m=1, \ldots, M$, where $F_m=[L(v_m) v_m]^T$ or $F_m=[u_m]^T$.

After constructing the basis, we determine the subspaces of the space of parameter points in which the Green’s kernel of the radiating structure is differentiable. Only if the kernel is differentiable, interpolation of the row vectors $F(f, \theta, \phi, p_1, \ldots, p_N)$ will make sense. The subspaces are called differentiability zones. In general, they can be determined straightforwardly. For example, for periodic structures, the boundaries of the differentiability zones are described by the zeros of the static modal propagation constants $k_{qf}$ of the modes.

$$ k_{qf} = \sqrt{k^2 - k_{1qf}^2}, $$

(2)
where \( k = \omega \sqrt{\varepsilon \mu} \) is the propagation constant and \( k_{q,r} \) is the transverse wave number, which depends on the mode indices \((q,r)\), and on the frequency and the scan angles. At each boundary, one of the modes turns from evanescent into propagating or vice versa. In each differentiability zone, we compute expansion coefficients \( \alpha_m \) that describe the row vector \( F \) inside the zone in terms of the basis elements \( F_m \),

\[
F(f, \theta, \phi, p_1, \ldots, p_N) = \sum_{m=1}^{M} \alpha_m(f, \theta, \phi) F_m, \quad (3)
\]

The computation of these coefficients is carried out by means of Kriging interpolation over a gradually extended set of parameter points \((f, \theta, \phi, p_1, \ldots, p_N)\) in the zone. For each considered set, the Kriging results are compared with results obtained by reduced order model techniques applied to the minimum norm problem associated to Eq. (3). The extension of the set stops once a required level of accuracy of the expansion coefficients \( \alpha_m \) is achieved.

The described strategy is called the parametric computation. After this computation step, either the moment matrix \( L \) and the RHS \( v \), or the solution \( u \), can be rapidly evaluated for any parameter point in the specified parameter ranges by simply interpolating the expansion coefficients \( \alpha_m \) and by calculating the sum in Eq. (3). This evaluation, which we call the post processing, is much faster than the full-wave computation. Hence, it offers a rapid ‘walk’ through the parameter space and therefore this approach is much more suitable for the optimization of radiator and/or FSS designs than full-wave computations, as we will show in this paper.

With respect to the parametric computation itself, we note that only in the construction of the basis, full-wave computations of the complete moment matrices are required. In an optimization procedure, the number of matrices \( M \) in the basis is in general much lower than the number of parameter points in which one needs to evaluate the scattering parameters of the FSS to compute the penalty function in every iteration of the optimization loop. Moreover, the reduced order model techniques and the Kriging interpolation used in the parametric computation are much less computationally expensive than the calculation of all moment matrices.

### 3 Test Cases and Validation

In this section, we describe the test cases for the optimization combined with parameterization. In all test cases, parametric models are computed for the ‘building blocks’ that comprise the test case. After that, the optimization loop is carried out. A penalty function is evaluated at each iteration of the optimization scheme by invoking the post processing of the simulator by which the corresponding parametric model was calculated.

#### 3.1 Test case 1: Rectangular Waveguide Integrated with a Dipole Array

The first test case that we consider has the same geometry as the one discussed in [5], where we demonstrated that the application of parameterization techniques results in fast-executable simulations and that the results obtained from post processing match very well the full-wave results. For completeness, we present the structure with its dimensions in Figure 1. The waveguide is excited by the \( \text{TE}_{10} \) mode. Because of the large distance between the FSS and the array, only one accessible mode (the fundamental Floquet mode) was sufficient to perform the analysis. The parametric models of the structure are computed with both Luxaflex and Twodim, which consider a unit cell of the infinite structure.

The dipole is discretized in terms of 8x3 piece-wise linear functions (directed in the length direction of the dipole), which results in a 24x24 matrix and a 24x1 RHS. Consequently, \( F = [L, v] \) is a row vector of 24x24 + 24x1 = 600 elements.

![Figure 1: Geometry of the first test case, a dipole FSS cascaded with a waveguide array.](image)

The parametric models computed with Twodim and Luxaflex have two varying parameters: the frequency \( f \) with range 9GHz - 13GHz and the sizes of the dipole with ranges 6mm - 9mm and 1mm - 1.5mm.

To test optimization algorithms, we choose the penalty function and its threshold in such a way that at least one dipole size satisfies the threshold. We define first upper and lower bounds for the \( S_{11} \) such that the \( S_{11} \) for a strip with size 9mm x 1mm satisfies all these bounds, see Figure 2. Next, we define the penalty function by

\[
F = \frac{\sum S_n}{\sum T_n}, \quad (4)
\]

where \( S_n \) is the number of prescribed frequency points in the range of (upper or lower) bound \( n \) that do not satisfy this bound and \( T_n \) is the total number of frequency points in this
bound. We prescribe 40 frequency points uniformly distributed over the frequency range 9 - 13GHz. The threshold value for the penalty function is set to 0.0035.

Figure 2: The $S_{11}$ (solid red curve) as a function of the frequency for a strip size of 9mm x 1mm in test case 1, together with the upper and lower bounds (solid blue curves).

For test cases 1 and 2 we consider three different algorithms: a Genetic Algorithm (GA), a Particle Swarm Optimization (PSO), and Adaptive Simulated Annealing (ASA). The first two algorithms are run on the parametric model generated by Twodim, while the third algorithm is run on the parametric model generated by Luxaflex. We emphasize that the parametric model that is used influences in particular the total simulation time, since the time to invoke and run the post processing of the simulators may be different. The parametric model will not influence the number of iterations or the accuracy of the result, provided that the parametric models are equally accurate.

Table 1: Number of penalty-function evaluations for two runs of GA and PSO with different population sizes (indicated between brackets) and one run of ASA, together with the determined dipole sizes (mm).

<table>
<thead>
<tr>
<th></th>
<th>GA</th>
<th>PSO</th>
<th>ASA</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of penalty</td>
<td>2500 (50)</td>
<td>1000 (20)</td>
<td>40 (40)</td>
</tr>
<tr>
<td>function evaluations</td>
<td>1500 (100)</td>
<td>41</td>
<td></td>
</tr>
<tr>
<td>Determined dipole</td>
<td>8.99 x 1.45</td>
<td>8.47 x 1.36</td>
<td>8.77 x 1.33</td>
</tr>
<tr>
<td>size (mm)</td>
<td>8.99 x 1.04</td>
<td>8.95 x 0.99</td>
<td></td>
</tr>
</tbody>
</table>

Table 1 shows the number of penalty-function evaluations for the three algorithms together with the determined dipole sizes for the GA and the PSO, two different population sizes are considered. For our research, the most important result is the large reduction of the computational load that we have achieved by the parameterization. For example, the calculation of the parametric model with Twodim takes 1h 20min 12s, while the post processing takes only 0.15s per parameter point. On the other hand, a full-wave simulation takes 148s per parameter point. As mentioned in [5], the parametric-computation time seems high, but having carried out this computation, we need only 0.15s per parameter point to evaluate the scattering parameters, which is a factor 1000 less than the full-wave computation time. Moreover, we can use the parametric model for more than one run of the optimization algorithm. For the ASA algorithm, the time to compute the $S_{11}$ for all frequency and shape settings is 41*40*0.15s = 4min 6s by post processing and more than 67 hours by full-wave computation. For the GA with 1500 function evaluations, these times are 2h 30min and 5488 hours, which is more than 10 days. These results clearly illustrate the benefit of parametric models in an optimization process (note that a larger parameter space favours our approach further). We note that we have not accounted here for the time to invoke the post processing of the simulators and to transfer their results for the scattering parameters to the optimization routine to compute the penalty function. This time is at most of the same order as the post processing time.

The three most important observations regarding the performance of the optimization algorithms are:

1) The GA requires much more function evaluations than the ASA and the PSO, provided that its population size is sufficiently large.
2) The width of the dipole size varies considerably in the obtained result.
3) The GA reaches the maximum number of iterations (50) without convergence for a population size of 50.

The first observation can be due to our specific implementations of the algorithms and choices of algorithm parameters, e.g., population size, cross-over and mutation probabilities. Further investigation is necessary to clarify this point. The second observation is explained by the weak dependence of the $S_{11}$ on the width of the strip.

3.2 Test case 2: Three-Layered FSS for MM-Wave Applications

The second test case concerns a multilayer FSS with rectangular slots for mm wave applications, as depicted in Figure 3. The FSS consists of a cascade of three equal FSSs with rectangular slots at a distance of 0.3575mm, as described in [6]. The only difference with [6] is that the metal planes with the slots are infinitely thin and the Ohmic losses are not taken into account. The slots are arranged on a rectangular lattice with $d_x = 0.49$mm and $d_y = 0.5$mm. The excitation is represented by a plane wave with incidence angle $\theta = 45^\circ$ and incidence plane $\phi = 90^\circ$. Only one accessible mode is considered.

Figure 3: Multilayer dipole FSS with rectangular slots.
The parametric models are computed with Luxaflex and Twodim, where the varying parameters are the frequency (250 - 400GHz), the length of the slots (0.4mm - 0.5mm), and the width of the slots (0.01mm - 0.02mm). The slot in the unit cell is discretized with 10x1 piecewise linear functions directed in the length direction.

We choose a similar penalty function as in the previous test case, where we define the upper and lower bounds as indicated in Figure 4. Again, we make sure that there is one geometry setting for which the \( S_{11} \) satisfies bounds. The threshold on the penalty function is set to 0.012.

![Figure 4: The \( S_{11} \) (solid red curve) as a function of the frequency for a slot size of 0.46mm x 0.015mm in test case 2, together with the upper and lower bounds (solid blue curves).](image)

Table 2 shows the number of penalty function evaluations. We observe that in this case the GA and PSO require much less evaluations than in the previous test case for relatively small population sizes.

<table>
<thead>
<tr>
<th></th>
<th>GA</th>
<th>PSO</th>
<th>ASA</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of penalty function evaluations</td>
<td>100 (50)</td>
<td>60 (20)</td>
<td>40</td>
</tr>
<tr>
<td>Determined dipole size</td>
<td>0.46x0.015</td>
<td>0.46x0.015</td>
<td>0.46x0.015</td>
</tr>
</tbody>
</table>

Table 2: Number of penalty-function evaluations for a single run of GA, PSO (population size indicated between brackets), and ASA, together with the determined dipole sizes (mm).

In this test case, the parametric computation with Twodim takes only 17min. As in test case 1, we observe that the ASA needs less penalty function evaluations and is therefore more efficient than the other algorithms. Further investigation is necessary to clarify the differences. For Twodim, the parametric-computation time is 17min, the post processing time is approximately 1s, and the time of the full-wave computation is approximately 7s. The reason that the post processing is in this case not much faster than the full-wave computation is in particular the appearance of a differentiability boundary. Near this boundary, the Kriging interpolation is not accurate and therefore reduced-order-model computations are used in the post processing. Further research is carried on how to improve the speed of the post processing at this point.

### 3.3 Test case 3: Three-Layered FSS on X-band

For the third test case, we consider the FSS design of a triplanar hybrid radome presented by Munk [7]. An overview of the radome structure is given in Figure 5. The configuration contains a layered space, where the three thick lines indicate the positions of the slots. A slot consists of a four-legged loaded element. Centre frequency of the FSS is 10GHz.

![Figure 5: The configuration of the triplanar radome. The positions of the three slot arrays (b) in the stack (a) are highlighted with thick lines. The grey parts of the slot array are metal.](image)

The parameter setting used for the reference solution presented by Munk is that of Figure 5 with \( L_1 = L_3 = 6.806 \text{mm} \) and \( L_2 = 6.466 \text{mm} \). In an initial indicative optimisation analysis where we have varied only \( L_i \) and the thickness \( h_i \) of the slabs, the optimisation resulted in configurations that are symmetric with respect to the centre FSS layer. To minimize the number of optimisation parameters we therefore restrict ourselves to symmetric cases. With the relative permittivity included in the optimisation process, we have 6 optimisation parameters for the symmetric case (permittivity and thickness of the first dielectric layer, length of the dipole of the first FSS layer, permittivity and thickness of the second dielectric layer and the length of the dipole of the centre FSS layer).

Owing to the symmetry of the four-legged loaded dipole, the response for the TE_{00} and TM_{00} Floquet modes is the same. In Figure 6 we plot the \( S_{11} \) and \( S_{12} \) for the configuration as proposed by Munk, which we will consider to be the reference configuration for comparison of the optimization results. Calculation of the response as function of 101 frequency points and for one parameter setting (geometry and material properties) is approximately 40s, leading to a calculation time per point of 0.4s.
As can be seen from Figure 6, the reflection coefficient has a maximum of –10dB in the pass band and the centre frequency is 10.25GHz (design centre frequency was 10GHz). The grey areas are bounds that we will use for the optimization, as described below (note that they have not been used in the calculation of the reference solution).

For the optimization we introduce a penalty function $F$ that compares the reflection coefficient $S_i$ at each computation point with a reference value $S_i^b$. The reference values are defined in the form of bounds (see Figure 6). The penalty function that has been used is given by

$$ F = \sum w_i \left\| S_i - S_i^b \right\|, $$

where $S_i^b$ is the value of the bound for the $i^{th}$ sample point within that bound. Furthermore, we can define a weight coefficient $w_i$ for each of the bounds to tune the optimization. Note that the bounds are set without a-priori knowledge of a solution that completely satisfies these bounds.

Optimization has been done for this configuration using Adaptive Simulated Annealing (ASA) and fminsearchbnd. The ASA method has been introduced as very fast simulated re-annealing [8]. Simulated annealing is a global optimisation technique that is based on thermodynamics, i.e., the optimisation process is analogous to the cooling of a set of atoms to a state of minimum energy. The free available ASA C-code of L. Ingber [9] has been applied throughout.

The fminsearchbnd routine is a modification by John D’Errico of the standard Matlab minimization routine fminsearch [10]. The Matlab routine is based on the simplex method by Nelder and Mead, which is a local optimization method. Constraints are built into the method using a sine-transformation. The parameter ranges that were used for the optimization are listed in Table 3. We have used the centre values of the parameter ranges as starting point for the ASA method, while we have taken the design by Munk as the starting point for the fminsearchbnd method due to the fact that fminsearchbnd is a local optimization method.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Min value</th>
<th>Max value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h_1$</td>
<td>1mm</td>
<td>15mm</td>
</tr>
<tr>
<td>$\varepsilon_{r1}$</td>
<td>1.1</td>
<td>2.6</td>
</tr>
<tr>
<td>$L_1$</td>
<td>6mm</td>
<td>7mm</td>
</tr>
<tr>
<td>$h_2$</td>
<td>1mm</td>
<td>15mm</td>
</tr>
<tr>
<td>$\varepsilon_{r2}$</td>
<td>1.1</td>
<td>2.6</td>
</tr>
<tr>
<td>$L_2$</td>
<td>6mm</td>
<td>7mm</td>
</tr>
</tbody>
</table>

Table 3: Parameter ranges for the optimization.

The response of the total structure was calculated by cascading the scattering matrices of the individual structures (note that the S-matrices of the dielectric layers are calculated analytically). The results for the optimization with the ASA and fminsearchbnd methods are given in Figure 7 and Figure 8, respectively.
Comparing Figures 7 and 8 to Figure 6 we can see that the results of both optimization methods are an improvement over the reference solution taken from Munk [7]. The result obtained with the ASA method has $-20$dB reflection coefficient in-band (10dB improvement), at the expense of a $-5$dB dip in $S_{11}$ around 14GHz. Furthermore the bandwidth is now symmetric around the centre frequency and is 2% larger than the reference solution. The solution obtained with the fminsearchbnd method shows a 5dB improvement in $S_{11}$ in-band, while the bandwidth is increased by 14%. This difference in results is due to the difference in optimization methods and the different starting points that were used. The parameter values for the optimization results are given in Table 4.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>ASA</th>
<th>fminsearchbnd</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h_1$</td>
<td>6.172mm</td>
<td>6.390mm</td>
</tr>
<tr>
<td>$\varepsilon_{r1}$</td>
<td>2.1</td>
<td>1.5</td>
</tr>
<tr>
<td>$L_1$</td>
<td>6.426mm</td>
<td>6.863mm</td>
</tr>
<tr>
<td>$h_2$</td>
<td>8.166mm</td>
<td>3.868mm</td>
</tr>
<tr>
<td>$\varepsilon_{r2}$</td>
<td>1.9</td>
<td>2.0</td>
</tr>
<tr>
<td>$L_2$</td>
<td>6.246mm</td>
<td>6.679mm</td>
</tr>
</tbody>
</table>

Table 4: Parameter values for the optimization results.

4 Conclusions and outlook

In this paper we have presented a framework for the optimization of radiators and/or FSS structures. Using parametric techniques, we are able to strongly reduce the computation time that is required to calculate the response of the structures. The scattering matrix / impedance matrix approach enables us to use multiple tools for the computation of the response. For the optimization, various optimization methods and penalty functions have been implemented and compared. Calculations on various test cases show that the combined approach of parametric computation techniques and optimization is well suited to the synthesis and optimization of FSS and radiating structures.

The future steps in the development of an optimizer for phased arrays integrated with FSS and feeding elements will be particularly focused on the linking of the three simulators in a single optimization routine. We will aim at the optimization of integrated phased arrays composed of building blocks with parametric models generated by different simulation methods. To be able to cascade the scattering matrices of different building blocks, we will take care of a number of aspects, such as determining common modes in the blocks, the mode ordering, and determining common frequency, scan angle, and parameter ranges for the optimization. The optimizer will be validated by considering a number of test cases, which include FSS and feeding elements that have already been manufactured.

Acknowledgements

The work described in this paper was supported by the Dutch and French ministries of defence under the EDA (European Defence Agency) contract RTP No. 1.29 SYNTAS (Synthesis Tooling for Antenna Structures).

References