

Development and Analysis of an Array of Folded Patch Antennas Using Matlab

Blasi Luca¹, Mastrofini Alessandro², and Mucenica Stefan Leonard³

²alessandro.mastrofini@alumni.uniroma2.eu

³stefanleonard.mucenica@alumni.uniroma2.eu

Abstract

The latest developments in Computer Aided Engineering, by means of the remarkable progress that several softwares have been made in terms of computational speed and of enlarged ranges of phenomena that can be simulated and accurately analyzed, represent an immense span to inventiveness. For this particular instance, it is also a great opportunity for who wants to reach or already has an expertise in the Antenna Design field. In this article, among a vast amount of possibilities, some of the Matlab toolboxes dedicated to antenna designing have been adopted in such a way as to create the structure of a coaxial-fed folded patch antenna combined with a low performance dielectric substrate (i.e. FR4). The main purpose was to obtain an antenna resonating at 2.1 GHz and with an input impedance matched at 50 Ohms. Subsequently, this designed radiating object has been introduced in a broader structure, that is a linear array of five antennas, as its representative single element. The Tchebyshev array synthesis has been used in order to reach particular design and radiating behaviour. Specifically, the beam-steering in the broadside and at 45° off the boresight direction have been analyzed by means of the overall gain and the electric and magnetic field behaviour close and far from the array. All the mathematical models which have been used together with the software feedback represent the structure of an overall process of development, optimization and analysis which have been very carefully performed and shown in the following paragraphs.

Introduction

All the parameters necessary to generate the array structure are shown in this paper together with some considerations on the implications of running certain simulations and of the deriving results. The array of folded patch antennas has been designed entirely in Matlab. Some flaws in the tools that have been used emerged in terms of lacking of full compatibility between them and some limitations in creating and analyzing the exact structure intended for this design. For example, the single folded patch antenna couldn't be used as an array element in the Antenna Toolbox (which would have allowed studying the overall array behaviour in terms of gain and electromagnetic fields), while the Phased Sensor Array Analyzer made possible creating the array structure but it couldn't give results describing the electromagnetic field. This puzzling situation made it necessary to reach a compromise by using a similar structure to that of the folded patch antenna: a PCB stack shorting the patch with a series of small cylindrical pins instead of using a rectangular wall of metal as in the original design recommendation. Some results could be obtained both with the original structure and with its approximation: in those cases several comparisons show their radiating behaviour is very similar, thus the approximation represents an acceptable compromise.

Starting from the theoretical modeling based on mathematical formulations [Balanis 1], a set of parameters related to the single

antenna (the reflection coefficient, the size of the patch and the ground, the substrate thickness, the feed location) have been optimized based upon a mesh density refinement process.

Generally, the study of an antenna array starts with the design of the array factor. The array factor is the complex-valued far field radiation pattern obtained by considering the elements to be punctiform sources. The total field is both influenced by the angular filtering effect of the array factor (which disregards the particular structure of the array element, but taking into account informations as spatial arrangement and currents circulating on the array elements) and the single element radiation. This gives an idea of how the behavior of the single antenna is affected by the other antennas of the array. All the elements are participating to those mutual interactions, bringing out a different behaviour in all.

Subsequently, several tests were conducted in order to verify that the overall behavior of the array complies with the project requirements.

A flowchart of all the steps of the project is available at page 15.

Tchebyshev array factor design

A Tchebyshev array factor will be designed in this part, so that the specific structure of the single element of the array is disregarded for the moment. The array is represented by a linear distribution of five elements ($n_{el} = 5$), which are supposed to be uni-

formly spaced and with a non-uniform amplitude for the feed arrangement, that anyway is going to be symmetrical (starting from the element in the middle, located in the origin of the geometrical axes). Also the "mean lobe to side lobet ratio" (R) will be one of the input design variables. Thus, the variables of this design part are the *element inter-spacing* (d_{opt} , which needs to be optimal such that the beamwidth is minimized), the feed amplitude of the single antenna (each one represented by a C_i , with $i \in \{-2, -1, 0, 1, 2\}$), the i^{th} current coefficient - these unknowns are actually just three and not five, due to the symmetrical amplitude distribution hypothesis), the operating *resonant frequency* required for the single antenna (f) and some other quantities related to them, such as the *tapering efficiency* (η_T) and the *beamwidth* (BW_{fn}). Moreover, the difference between the beamwidth of the Tchebyshev array (*Uniform Spacing Non Uniform Amplitude* case, shorten *NUA*) and that of a uniform array having the same number of elements and inter-spacing value but a uniform amplitude feed distribution (*Uniform Spacing Uniform Amplitude*, shorten *UA*) will be discussed. The input design variables related to the Tchebyshev array factor (i.e. n_{el} , R and f) are listed in **table 1**.

Starting from the optimal inter-element spacing, it requires some secondary variables in orderer to be evaluated. Firstly, a coefficient (γ , depending on R and indirectly on the number of array elements) needs to be calculated, among with the wavelength in the free space (λ , as the ratio between the light speed in the free space - c - and the operating frequency f). Having those quantities, d_{opt} minimizing the BW_{fn} can be computed by using the **eq. (1)** group.

The array factor evaluation through the Tchebyshev polynomial comes as the next step. The Tchebyshev polynomial approximation to the second order ($T_2(x)$) is sufficient so the current coefficients can be obtained, remembering that a symmetrical amplitude distribution hypothesis has been made. In this case the current coefficients C_i , $i \in \{-2, -1, 0, 1, 2\}$ follow the rule $C_i = C_{-i}$, thus their symbolic representation can be simplified as follows: C_n , $n \in \{0, 1, 2\}$ (or just $n = \overline{0, 2}$). Since the *Riblet variation* to the *Dolph-Tchebyshev synthesis model* has been used, the vari-

able of $T_2(x)$ becomes $x = a + b \cos(u)$ and the formulation for $d_{opt} \in (\lambda/2, \lambda]$ differs from the standard model. The coefficients a and b are related to the maximum value selected in the sub-domain of $T_2(x)$ (the window of visible radiation lobes). This maximum, called for example x_1 , corresponds to the main lobe in which it can be convertend by evaluating the array factor $|T_2(x_1)|$ (some additional references to the specific formulas that need to be used in order to compute can be found in **[Balanis1]** and **[ewa]**). That said, the current coefficients can be extracted from $T_2(x)$ (so by using **eq. (2)**). Resulting uniquely out of a C_n dependence, the tapering efficiency is obtained by using **eq. (3)**.

$$d_{opt} \approx \min\{BW_{fn}\}$$

$$d_{opt} = \lambda \left[1 - \frac{\arccos\left(\frac{1}{\gamma}\right)}{\pi} \right] \quad (1)$$

$$\gamma = \cosh \left[\frac{1}{2N} \ln \left(R + \sqrt{R^2 - 1} \right) \right]$$

Next, both non uniform amplitude (Tchebyshev array, *NUA*) and uniform amplitude (*UA*) cases are compared. The comparison shows how the BW_{fn} in the *UA* case (i.e. $BW_{fn}^{[UA]}$) is narrower than that of the *NUA* case (i.e. $BW_{fn}^{[NUA]}$). This result (see **eq. (4)** and **table 2**) is generally effective and the comparison has been made to show that this particular design case confirms the general condition.

All the numerical results related to the Tchebyshev array design are gathered together in **table 2**. After these quantities are calculated, some important design considerations can be made about the array efficiency. A Non-Uniform Amplitude Array Factor has been designed by using the Riblet variation of the Dolph-Tchebyshev array synthesis model. Considering the *maximum to minimum feed ratio*:

$$r_{\max/\min} = \frac{C_{\max}}{C_{\min}}$$

the less $r_{\max/\min}$ is, the more efficient distribution of current is reached. In this particular design, the requirement was to to find the d_{opt} which minimizes the beamwidth, starting from the input design variables. Thus, the $r_{\max/\min}$ is a straight consequence of the current coefficients and its optimal value has not been the seek of this project. Anyway, for this design, $r_{\max/\min} \cong 4.39$ meaning that if a damage of the element with the C_{\max} (i.e. C_0) level of feed occurs, most part of the efficiency will be lost. In any case, the tapering efficiency shows how it will not be possible to take advantage of 21 % of the array in an ideal situation, remembering that this design model can be discerned by the real circumstance in terms of the Tchebyshev error (see **[Balanis1]**).

$$\begin{aligned} T_2[x = a + b \cos u] &= \dots \\ &= C_0 + 2C_1 \cos u + C_2 \cos 2u \quad (2) \\ &= (2a^2 + b^2 - 1) + 4ab \cos u + b^2 \cos 2u \end{aligned}$$

$$\eta_T = \frac{1}{2N+1} \frac{\|C_0 + 2C_1 + 2C_2\|^2}{C_0^2 + 2C_1^2 + 2C_2^2} \quad (3)$$

Parameter	Value
# elements	$n_{el} = 2N + 1 = 5$
Main lobe/side lobe ratio	$R = 120 \cong 41.58 \text{ dB}$
Frequency	$f = 2.1 \text{ GHz}$

TABLE 1: Array factor input design variables

Parameter	Value
Feed coefficients [A]	$C_0 = 41.2$ $C_1 = C_{-1} = 29.8$ $C_2 = C_{-2} = 9.6$
Feed coefficients normalized to C_{\max}	$C_0^* = 1.000$ $C_1^* = C_{-1}^* = 0.7215$ $C_2^* = C_{-2}^* = 0.2336$
Tapering efficiency	$\eta_T = 79\%$
Beamwidth	Tchebyshev 50.6° Uniform 34.8°

TABLE 2: Tchebyshev array design results

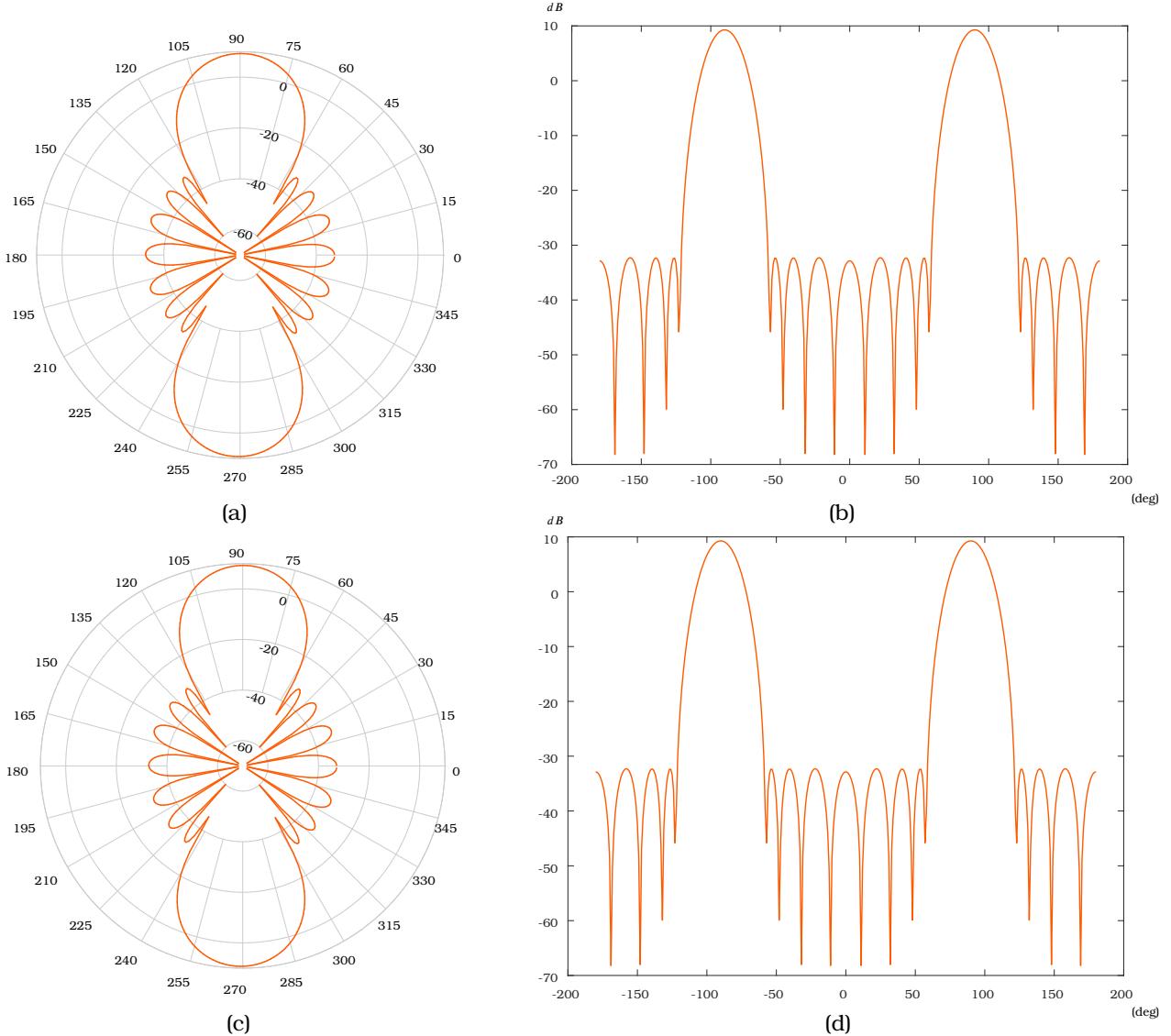


FIGURE 1: Array factor polar, in azimuth (a) and elevation (c) cut, and rectangular in azimuth (b) and elevation (d) cut, diagrams

$$BW_{fn}^{[UA]} < BW_{fn}^{[NUA]}$$

$$BW_{fn}^{[NUA]} = 2 \frac{180}{\pi} \left[\frac{\pi}{2} - \arccos \left(\frac{\arccos \left(\frac{\cos \left(\frac{\pi}{2N} - a \right)}{b} \right)}{k_0 d} \right) \right] \quad (4)$$

$$BW_{fn}^{[UA]} = \frac{2\lambda}{N d} \frac{180}{\pi}$$

In the end, the 2D array patterns resulting by the use of the calculated parameters are shown in **fig. 1**. Two polar patterns and their corresponding rectangular ones have been plotted (in the azimuth cut plane and in the elevation cut one).

Rectangular folded patch antenna design

The main components of a rectangular folded patch are: the patch, the substrate (generally accessory, but used in this project), the ground, the rectangular shorting wall between the patch and the ground, and the feed. More details about them will be presented in a short while. Before that, some other remarks are necessary: this antenna will be the element of the array, which will be designed starting actually from a PIFA (*Planar Inverted F Antenna*). A general PIFA realized with a dielectric substrate is shown in **fig. 2**. A particular condition is imposed to the PIFA, by which the width of the rectangular shorting wall (w_{sc}) equals the patch width (W_{patch}), so that the PIFA and the folded patch antenna will be two equivalent structures (see **eq. (5)**). This remark on the PIFA is necessary because generally its structure is not equivalent to that of the folded patch antenna because of the possible variability of the shorting width (w_{sc}), which doesn't always satisfy the above imposed condition. This condition on w_{sc} represents one of the input

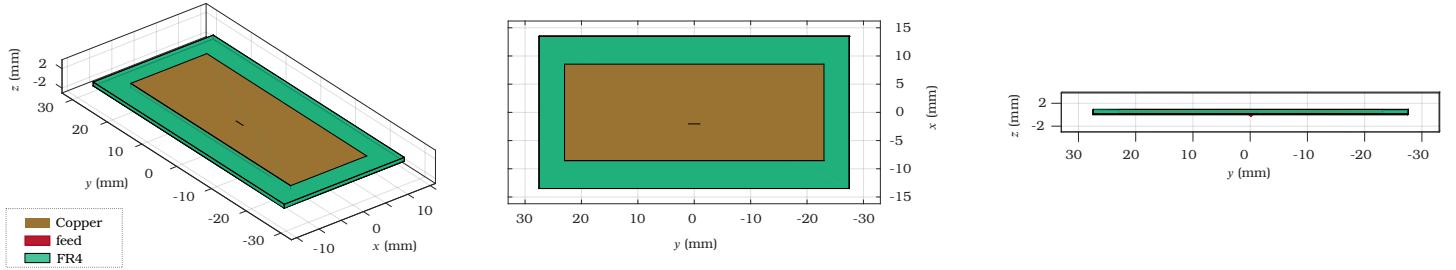


FIGURE 2: PIFA realized with a dielectric substrate

variables of the PIFA design, such as the matched input resistance (R_{in}), the resonant frequency (f) and the substrate characteristics (see **table 3**). A preliminary evaluation of the patch parameters have been realized by leaning on a theoretical set of formulas (see **[Balanis 1]**). That's the reason why the characteristics of the PIFA, shown in **table 3**, are distinguished between pre-optimal and post-optimal (same thing applies to the ground component). Thus, an optimization process of all these and other parameters will be performed in some following steps. Just before that, now the theoretical model will be shortly pointed out: it furnishes a provisional set of results (that will be adjusted afterward). Firstly, the length and the width of the patch starting from the $w_{sc} = W_{patch}$ imposed condition, the substrate wavelength and thickness (i.e. λ_{FR4} and h_{FR4}) are calculated (**eq. (6)**). Then, other parameters, such as the effective permittivity ϵ_{eff} and length (L_{eff}) in **eq. (7)**, the radiation resistance (R_r) in **eq. (8)**, the half-power beamwidth in the E-cut (Θ_E) and the H-cut (Θ_H) in **eq. (9)** and eventually the feed location, defined by ℓ_{feed} in the patch length direction with respect to the free edge (**eq. (10)**), have been calculated.

$$\begin{aligned}\epsilon_{eff} &= \frac{\epsilon_{FR4} + 1}{2} + \frac{\epsilon_{FR4} - 1}{2} \left(1 + 12 \frac{h_{FR4}}{W_{patch}} \right)^{-\frac{1}{2}} \\ L_{eff} &= \frac{\lambda_{FR4}}{4} \\ \Delta L &= 0.412 h \left[\frac{(\epsilon_{eff} + 0.3) \left(\frac{W_{patch}}{h_{FR4}} + 0.268 \right)}{(\epsilon_{eff} - 0.258) \left(\frac{W_{patch}}{h_{FR4}} + 0.8 \right)} \right] \\ L &= L_{eff} - 2\Delta L\end{aligned}\quad (7)$$

$$R_r = \frac{120 \lambda}{W_{patch}} \left[1 - \frac{1}{24} \left(2\pi \frac{h_{FR4}}{\lambda} \right)^2 \right]^{-1} \quad (8)$$

$$\begin{aligned}\Theta_E &= 2 \arccos \sqrt{\frac{7.03 \lambda^2}{4(3L_e^2 + h_{FR4}^2)\pi^2}} \\ \Theta_H &= 2 \arccos \sqrt{\frac{1}{2 + 2\pi \frac{W_{patch}}{\lambda}}}\end{aligned}\quad (9)$$

$$\ell_{feed} = \frac{L_{patch}}{\pi} \arccos \sqrt{\frac{R_{in}}{R_r}} \quad (10)$$

The Antenna Toolbox gives specific information about the mesh density level that should be adopted for the design of the patch antenna components. The only issue is that these details are given only for particular ranges of the ratio indicator called *relative thickness* or *electrical thickness* h_λ (see **[makarov]** and **[meshing]**). The electrical thickness depends on the ratio between the substrate thickness (h_{FR4}) and the wavelength related to the substrate medium (λ_{FR4}). When a mesh is configured in the Antenna Toolbox environment, a specific parameter needs to be adjusted: the maximum edge length of the generic triangle covering the geometry of the antenna (e_{max}). In the case of a relative length h_λ comparable to $1/10$, it's recommended to select an $e_{max} \cong \lambda/10$. A substrate thickness respecting this relationship is called a *thick substrate*. None of the available substrates verifies this condition. Among them, only the thinnest substrate and the second to last one (thus $h_{FR4} = 0.8 \text{ mm}$ and $h_{FR4} = 1.0 \text{ mm}$) are part of a range which the Antenna Toolbox provides instructions of. It's the *thin substrate range*: the automatic mesh mode should be adopted for a thin substrate, namely having a relative thickness less or equal than one fiftieth ($h_\lambda \leq 1/50$). This leads to a further explanation about the actual substrate thickness chosen for this project. The thinner substrate choice rationale starts with the consideration of its quality factor (Q) depending on the loss tangent ($\tan\delta$) and being generally low in the FR4 substrate case (being the two quantities inversely proportional, i.e. $Q \propto 1/\tan\delta$,

Refinement with MatLab Method of Moments

Substrate thickness selection

Three thickness levels were available for the *FR4* substrate required in this project (see **table 3**). In this part it will be explained why choosing a thinner substrate (if the *FR4* is used) is more convenient. This choice will be motivated both in relation to the physical behaviour (radiation efficiency) and to the limits of the Matlab tools that have been used (in terms of the mesh density level selection in a range that gives more reliable results).

$$W_{patch} = w_{sc} \quad (5)$$

$$\begin{aligned}L_{patch} + W_{patch} - w_{sc} &= \frac{\lambda_{FR4}}{4} + h_{FR4} \\ W_{patch} &= \frac{\lambda}{2} \sqrt{\frac{2}{\epsilon_{FR4} + 1}}\end{aligned}\quad (6)$$

Folded patch design parameters	
Frequency	2.1 GHz
Matched input resistance	$R_{in} = 50\Omega$
Substrate	FR4
Relative permittivity	$\epsilon_{FR4} = 4.8$
Relative permeability	$\mu_{FR4} \cong 1$
Loss tangent	$\{\tan(\delta)\}_{FR4} = 0.0260$
Available thicknesses	
(1)	$h_{FR4} = 0.8\text{ mm}$ (optimal choice)
(2)	$h_{FR4} = 1.0\text{ mm}$
(3)	$h_{FR4} = 1.6\text{ mm}$
Patch	Copper
Conductivity	$\kappa_{copper} = 5.96 \cdot 10^7 \text{ S/m}$
Thickness	$h_{patch} = 3.556 \cdot 10^{-5} \text{ m}$
Length (pre-opt) (post-opt)	$L_{patch} \cong \frac{\lambda_{FR4}}{4} = 0.0171 \text{ m}$ $L_{patch} = 0.0158 \text{ m}$
Width (pre-opt) (post-opt)	$W_{patch} \cong 0.0419 \text{ m}$ $W_{patch} \cong 0.0358 \text{ m}$
Half-power beamwidth (pre-optimization)	$\Theta_E = 360^\circ$ (E-cut) $\Theta_H = 121^\circ$ (H-cut)
Half-power beamwidth (after optimization)	$\Theta_E = 109^\circ$ (E-cut) $\Theta_H = 325^\circ$ (H-cut)
Ground (pre-optimized features)	Copper (same conductivity listed above)
Thickness	$h_{GND} = h_{patch}$
Length (pre-opt) (post-opt)	$L_{GND} = 0.04 \text{ m}$ $L_{GND} = 0.0296 \text{ m}$
Width (pre-opt) (post-opt)	$W_{GND} = 0.06 \text{ m}$ $W_{GND} = 0.0580 \text{ m}$
Feed	Coaxial cable
Position across L_{patch} (pre-opt) (post-opt)	0.0066 m 0.0067 m
Position across W_{patch} (pre-opt) (post-opt)	0.0000 m 0.0010 m

TABLE 3: Project input parameters (frequency, matching and substrate features), theoretical (pre-optimized, shorted "pre-opt") and optimized ("post-opt") features of the patch and ground PIFA components

and being $(\tan \delta)$ very high compared to that of other more efficient substrates). This means the FR4 is a big power disporser. Since increasing h_{FR4} will provoke just more losses in terms of a radiation efficiency drop and since the only thickness values of 0.8 mm and 1.0 mm would give reliable/accurate results in the Antenna Toolbox simulations, the 0.8 mm thickness level will be adopted.

Parameter	Value
Γ_{final}	-54.94 dB
R_{in}	49.86Ω
Y_{in}	0.11 S

TABLE 4: Final Γ and impedance matching values after last optimization choice: the w_{feed} change

Mesh density refinement

Although a mesh density choice has already been made by selecting the best maximum edge length e_{max} , the accuracy achievable by using the mesh automatic mode in the case of substrates belonging to the *thin substrate range* will be proved hereafter. An initial study of the mesh density level influence on the reflection coefficient (Γ in dB) evaluated at the resonant frequency ($f = 2.1\text{ GHz}$) has been realized, thus a $\Gamma_{2.1\text{ GHz}} = F(e_{max})$ function has been plotted with an initial step of $\Delta e_{max} = 2.5 \cdot 10^{-4} \text{ m}$ between every two mesh densities related to their specific e_{max} . This first simulation considered a broader range of e_{max} variation: $[2.5 \cdot 10^{-4} \text{ m}, 6.0 \cdot 10^{-4} \text{ m}]$.

Since the resulting plot (fig. 3) has shown big uncertainty of the reflection coefficient value at the resonant frequency ($\Gamma_{2.1\text{ GHz}}$) at almost every mesh e_{max} level (primarily due to the big step selected between one density level and another), some more detailed tests have been run, by considering a slightly narrower range ($[2.5 \cdot 10^{-4} \text{ m}, 5.0 \cdot 10^{-4} \text{ m}]$) and a thicker evaluation of the maximum edge values (so that the mesh variation step has been remarkably reduced to $1.0 \cdot 10^{-4} \text{ m}$). Specifically, the step between two mesh density levels in terms of the maximum edge length of each one has been reduced from a $\Delta e_m = 2.5 \cdot 10^{-4}$ to $\Delta e_m = 1.0 \cdot 10^{-4}$. In all those simulations, an important fact needs to be noted. Even very small variations on the maximum edge length value involved considerable inconsistencies in almost every part of the mesh range in terms of considerable variations of the frequency at which Γ reaches its minimum ($\min(\Gamma)$, therefore the resonant frequency of the antenna changes very easily). Thus, considering the frequency f^* at which $\min(\Gamma)$ is actually obtained, instead of evaluating it at the theoretical resonating frequency value at every mesh level, not only the standard test comparing e_{max} and Γ has been run, but also some mesh refinement plots representing the relationship between f^* and e_{max} (fig. 8c), Δf^* and e_{max} (fig. 8b) and also $\min(\Gamma)$ and e_{max} (fig. 8a) have been taken into account (where Δf^* is the difference between f^* and the resonant frequency $f = 2.1\text{ GHz}$). More parameter relationships have been collected and this led to the setting of the mesh density choice in terms of e_{max} that has been selected inside the most stable region (i.e. showing the smallest deviation of the reflection coefficient minimum from the resonant frequency). In the end it's been specifically taken the 'automatic' e_{max} ($= 3.5 \cdot 10^{-4} \text{ m}$) suggested by the Antenna Toolbox, since this value belongs to the stable region and seems to give the most accurate results. The e_{max} values belonging to the stable region ($[3.1 \cdot 10^{-4} \text{ m}, 3.7 \cdot 10^{-4} \text{ m}]$) exhibit slight deviations from the resonant frequency ($\Delta f^* \in [0.01\text{ GHz}, 0.03\text{ GHz}]$) and the minimum of the reflection coefficient varies in the range $[-24\text{ dB}, -33\text{ dB}]$.

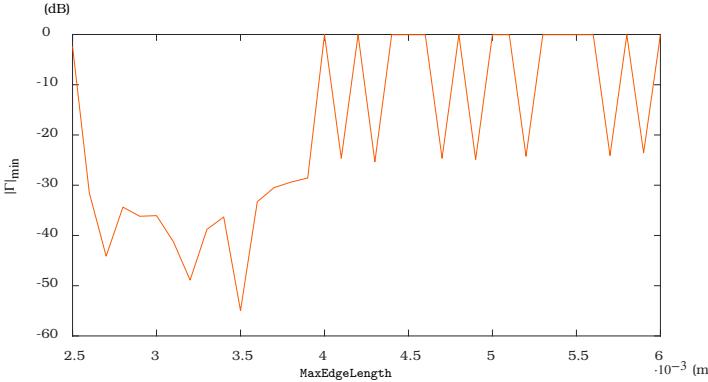


FIGURE 3: Minimum of the reflection coefficient Γ [dB] in the frequency range $2.0 \div 2.2$ GHz depending on the varying mesh density level

Patch parameters refinement

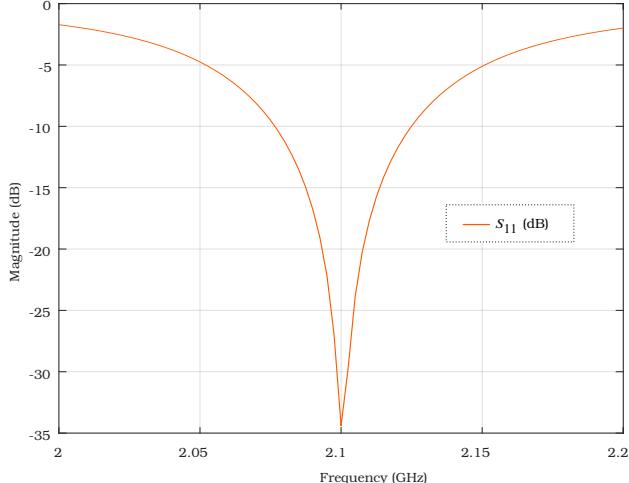
After the selection of the maximum edge length of the mesh (consequently, of its density level), a more refined computation of the reflection coefficient will be made, depending on the patch size (i.e. on its length and width), but also on the feed location. Firstly, only a parametric variation of the feed position across the patch length direction has been considered, depending on variations of L_{patch} and W_{patch} . This means that the first refinement of the feed position has been evaluated starting from its theoretical equation (depending indirectly by W_{patch}). The change of the feed location has been taken into account in the computation of every step of the simulation, thus in every evaluation of the reflection coefficient, keeping e_{\max} as a constant (the previously selected length). The patch size variations provoked wide modifications of the reflection coefficient, which values depending on that have been represented by an initial contour plot (with variations of the patch size in a broader range and with a larger step between one value and another, see **fig. 5a**). After that, another simulation (**fig. 5b**) in a narrower range of the patch size variation has been run in order to choose from there a set of Γ values (related to a set of coupled values $(L_{\text{patch}}, W_{\text{patch}})$) that should put the patch antenna in the best resonant condition (which means that a Γ as close as possible to zero in the linear scale and as negative as possible in the dB, logarithmic scale is sought). The logarithmic (dB scale) was used for the contour plots because this way the resulting Γ values are easier to distinguish from one another even graphically and mathematically.

A set of 20 values of Γ and respective coupled values $(L_{\text{patch}}, W_{\text{patch}})$ has been selected from the second simulation range so that a more specific simulation could be run. In this third case (**fig. 15**), the reflection coefficient has been plotted in a range around the resonant frequency ([2.0 GHz, 2.2 GHz]) in order to find which is the best combination for the patch size that makes actually resonate the antenna at the project frequency. Another determining and discriminating factor was the input impedance ($Z_{in} = R_{in} + jY_{in}$, where the real part of Z_{in} is the input resistance, while the imaginary one is the input reactance), because a impedance matching (at 50Ω) needed to be achieved for the project. In the ideal case, of course, a reflection coefficient $\Gamma^{(id)} = 0.00 \rightarrow -\infty$ dB would be required in order to reach the perfect impedance matching (perfect matching with input resistance at 50.00Ω and null reactance). As an actual result, before

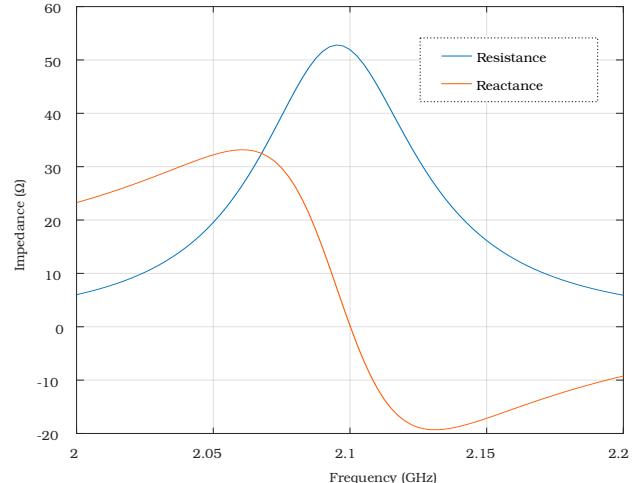
seeking a better matching, the Γ value related to all the couple candidates $((L_{\text{patch}}, W_{\text{patch}}))$ ranged from -24 dB to -30 dB. The two best candidates of the Γ value related to the coupled size $(L_{\text{patch}}, W_{\text{patch}})$ are plotted in **fig. 15**. Two refined plots (with narrower frequency steps in the same range) of Γ and Z for one of the candidates are represented in **fig. 4**. An additional design strategy contributed to the final patch size choice: the feed location varying across the patch width direction (w_{feed}). Thus, a parametric impedance matching study depending on that has been run on the best couple candidates and a few others from the set of 20. The resulting values are listed in **table 4** and the plots are shown in **fig. 6**

From the PIFA to a PCB stack approximation

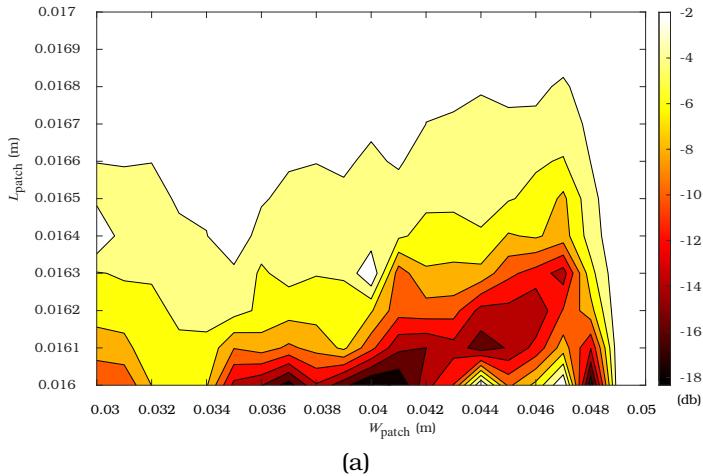
Since the Antenna Array Designer (AAD) is not able to generate an array of PIFAs, the Sensor Array Analyzer (SAA) has been used to create that type of structure, starting from the Tchebyshev array factor design made in the first part and from the previous optimization process of a single PIFA. In the last part the overall array of PIFAs needs to be analyzed by means of beamsteering for particular angles, total gain, electric and magnetic field patterns. Unfortunately, not all of this information can be extracted by using uniquely the SAA tool. This fact led to the use of another tool, the PCB Antenna Designer, so that all the parameters required for the project could be analyzed. With this last tool, a similar structure to that of the PIFA has been realized and introduced as the antenna element of the array distribution. Just before moving to the array analysis, a comparison between the single antenna being a PIFA structure and the antenna created by using a PCB stack was performed. A single limitation in the use of the PCB Antenna Designer has been encountered, but it has been overcome with an approximation strategy. Thus, the PIFA and the PCB stack designed are not perfectly identical, but their single and overall performance inside the array had shown a slight quantifiable difference. The limitation consisted in the absence of a specific option or combination of commands that would allow to realize the rectangular shorting wall between the patch and the ground. The only thing the designer can do is to replace this kind of shorting pin with a series of small diameter (e.g. 0.4 mm) cylindrical shorting pins (see **fig. 16** and **fig. 9** close to one edge of the patch, across the patch width direction). With this design choice, a very similar behaviour to that of the PIFA structure was possible to simulate. To support this results, a comparison of the 2D gain patterns (elevation and azimuth cut) of the PIFA to those of the PCB stack antenna has been realized in terms of the mean squared error values, shorten MSE , related to the two specific two-dimensional patterns considered ($MSE_{el} = 0.55$ dB in the elevation cut directivity pattern and $MSE_{az} = 0.16$ dB in the azimuth cut directivity pattern). Furthermore, since the SAA allows the plotting of some patterns such as the directivity (but not of the gain), a comparison between the 2D patterns of the antenna array of PIFAs and the array of PCBs has been presented. In that case the error is presented as a comparison between the main lobe levels in the PIFA array and the PCB array cases, but also in terms of first side lobe levels of the two arrays (see **table 5**). This discussion was necessary, because moving to the last part of the project required a deeper study of the array of antennas, that will be displayed by using the PCB antenna as the element of the array. A further discussion about the differences between the tools can be consulted in the **Methods**.



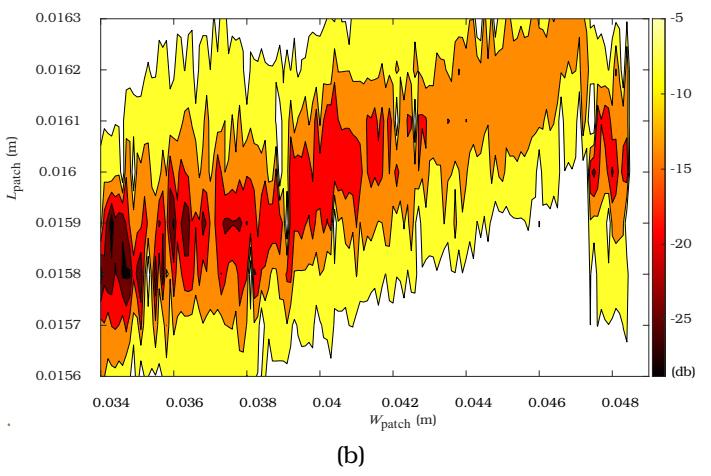
(a)



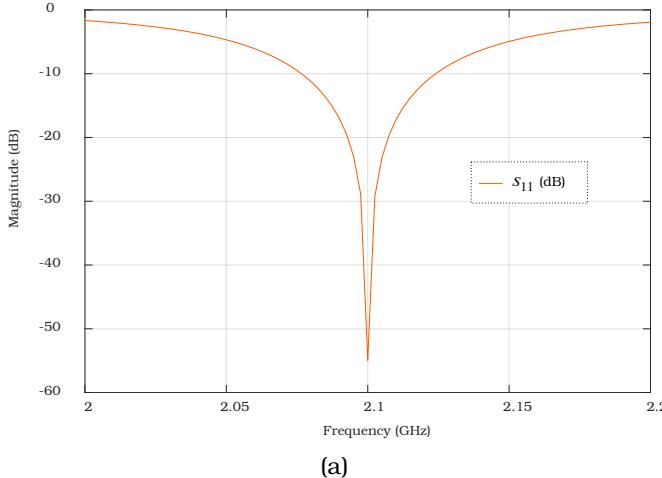
(b)

FIGURE 4: Reflection coefficient (a) and impedance (b) plots depending on $f \in [2.0, 2.1 \text{ GHz}]$ 

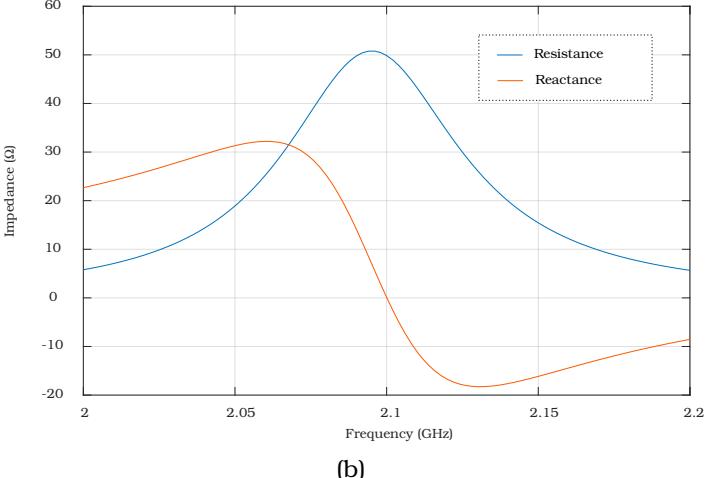
(a)



(b)

FIGURE 5: First (a) and second (b) contour plots depending on the patch size (L_{patch} and W_{patch} variations)

(a)



(b)

FIGURE 6: Final Γ (a) and impedance matching (b) plots after further refinement including w_{feed} change

Overall array performance

In this last part of the project, the array factor and the element antenna (PIFA or PCB stack) designs will be combined so that their

total effect will be examined. As it's been already mentioned in the previous paragraph, some of the information describing the overall array performance can be asked from both the SAA (where the array of PIFAs was designed) and the PCB/AAD (where the single patch

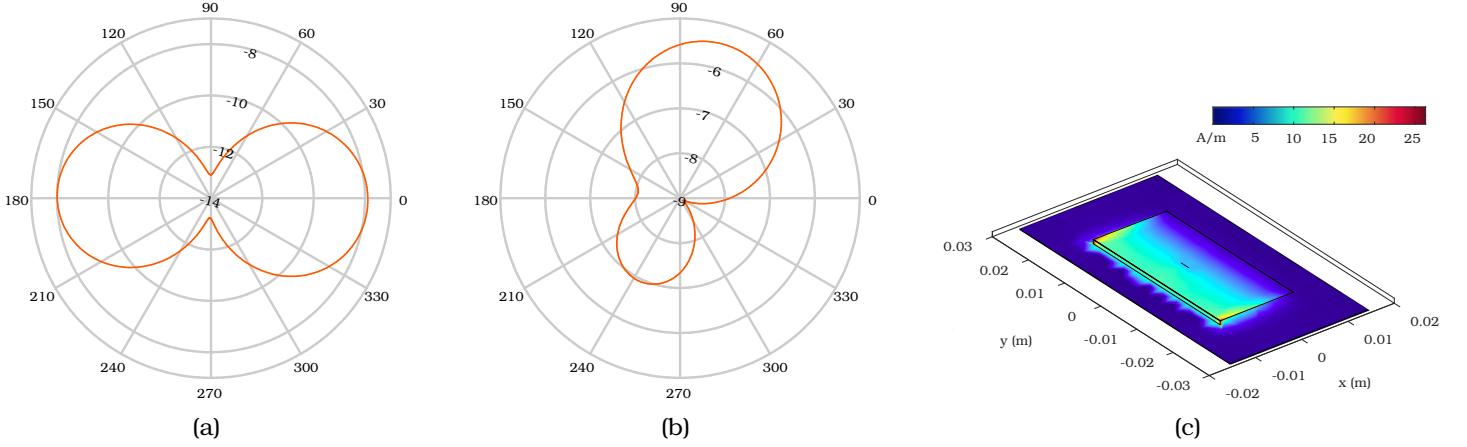


FIGURE 7: Gain patterns (a) in the nulle elevation plane, (b) in the null azimuth plane and (c) 3D current plot on the patch antenna

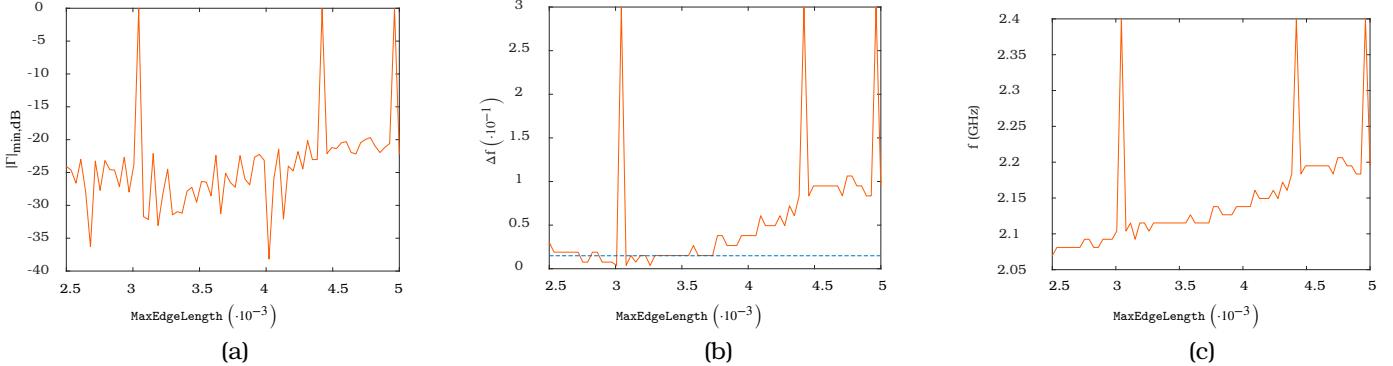


FIGURE 8: Comparison of the meshing parameter e_{\max} with minimum of reflection coefficient (a), variation of the resonant frequency with respect to the design one (the dotted line represents the value with respect to automatic meshing) (b), resonant frequency identified by the minimum of Γ (c)

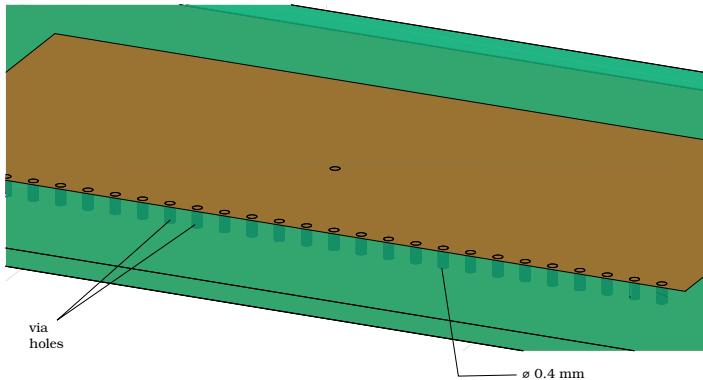


FIGURE 9: Cropped zoom over PCB via holes. For an overall picture see fig. 16

antenna made starting from the PCB stack and also the whole array can be constructed). The performance of the overall array will be evaluated in two cases: in the broadside case (90°) but also at 45° off the boresight direction. First of all, it is very simple to identify the phase shift coefficients in the broadside case because they all equal 0° (there's no actual phase shift between antennas). In the second case, some manual calculation can be made in order to in-

sert the phase shifts between the single antennas (this is the case of the PCB stack array), and it can be also automatically computed by using the array of PIFAs. The generic procedure is shown in eq. (12). Just before moving to the last part of the analysis, where only the characteristics of the array of PCBs will be considered, a last comparison between the array of PCBs and the array of PIFAs will be made in terms of 2D directivity patterns (both in broadside and in 45° off the boreside direction, see fig. 19). An error analysis based on the difference between the PCBs array main lobe and that of the PIFAs array, but also on the difference between the PCBs array main side lobe and that of the PIFAs array will be shown (see table 5). The patterns and the numerical results according to the numerical error values illustrate how the array of PCBs represent a good enough approximation of the array of PIFAs (see fig. 10 and fig. 19).

Total array gain

In order to obtain and describe the last plots of the project, it is not possible to recur to the SAA because some of the commands, such as `EHfields()` and `patternMultiply()`, are not available in this tool. Thus, in this last part, only the array of PCBs will be used, because it's compatible with these commands. The

Direction	$D_{ML,az}$ error	$D_{SL,az}$	$D_{ML,el}$ error	$D_{SL,el}$
Broadside (90°)	$\Delta D_{ML} = 2.39 \text{ dB}$ $\Delta D_{ML}^r = 3.3 \%$	$\Delta D_{SL} = 0.96 \text{ dB}$ $\Delta D_{SL}^r = 2.7 \%$	$\Delta D_{ML} = 0.81 \text{ dB}$ $\Delta D_{ML}^r = 1.1 \%$	$\Delta D_{SL} = 0.35 \text{ dB}$ $\Delta D_{SL}^r = 1.6 \%$
Off Boresight (45°)	$\Delta D_{ML} = 0.05 \text{ dB}$ $\Delta D_{ML}^r = 0.07 \%$	$\Delta D_{SL} = 1.37 \text{ dB}$ $\Delta D_{SL}^r = 3.4 \%$	$\Delta D_{ML} = 0.03 \text{ dB}$ $\Delta D_{ML}^r = 0.04 \%$	$\Delta D_{SL} = 4.1 \text{ dB}$

TABLE 5: Comparison between the array of PIFAs and the array of PCBs in terms of the directivity evaluated in their main lobes and first side lobes, both in the broadside and 45° off the boresight direction, both in the azimuth cut (az) and elevation cut (el) planes. The comparison has been made in terms of the directivity difference ($\Delta D_\# = D_{PIFAs} - D_{PCBs}$ in dB) in the corresponding positions and also as a relative percentage error ($\Delta D_\#^r \%$, the ratio between $\Delta D_\#$ and the directivity in the array of PIFAs case (D_{PIFAs}))

$f_{fre} \rightsquigarrow$ Fresnel limit $f_{fraun} \rightsquigarrow$ Fraunhofer limit	Position	E -field (90°)	H -field (90°)	E -field (45°)	45 -field (90°)
<i>PCB fed by C_{-2}</i>	(0, $-2d_{opt}$, $z_{fre}/2$)	0.1027	0.0139	0.2071	0.0269
	(0, $-2d_{opt}$, $z_{fraun}/2$)	0.0848	0.0114	0.2071	0.0269
	(0, $-2d_{opt}$, $4z_{fraun}$)	0.0161	0.0020	$0.165 \cdot 10^{-3}$	$0.1360 \cdot 10^{-3}$
<i>PCB fed by C_{-1}</i>	(0, $-d_{opt}$)	0.2923	0.0377	0.1889	0.0100
	(0, $-d_{opt}$, $z_{fraun}/2$)	0.1144	0.0143	0.1889	0.0100
	(0, $-d_{opt}$, $4z_{fraun}$)	0.163	0.0020	$0.1493 \cdot 10^{-3}$	$0.1365 \cdot 10^{-3}$
<i>Main PCB, fed by C_0</i>	(0, 0, $z_{fre}/2$)	0.4189	0.0522	0.2422	0.0174
	(0, 0, $z_{fraun}/2$)	0.1258	0.0155	0.2492	0.0174
	(0, 0, $4z_{fraun}$)	0.163	0.0020	$0.1285 \cdot 10^{-3}$	$0.1364 \cdot 10^{-3}$
<i>PCB fed by C_1</i>	(0, d_{opt} , $z_{fre}/2$)	0.3306	0.0367	0.2619	0.0215
	(0, d_{opt} , $z_{fraun}/2$)	0.1191	0.0145	0.2619	0.0215
	(0, d_{opt} , $4z_{fraun}$)	0.162	0.0020	$0.1053 \cdot 10^{-3}$	$0.1357 \cdot 10^{-3}$
<i>PCB fed by C_2</i>	(0, $2d_{opt}$, $z_{fre}/2$)	0.1350	0.0143	0.2922	0.0270
	(0, $2d_{opt}$, $z_{fraun}/2$)	0.0971	0.0116	0.2922	0.0270
	(0, $2d_{opt}$, $4z_{fraun}$)	0.162	0.0020	$0.0744 \cdot 10^{-3}$	$0.1343 \cdot 10^{-3}$

TABLE 6: Particular electric (E) and magnetic (H) field values measured in both broadside (90°) and in the 45° off the boresight directions. The E -field is measured in $\left[\frac{V}{m}\right]$ while the H -field in $\left[\frac{A}{m}\right]$. The fields have been measured with respect to the centers of each element antenna vertically off the array plane at different height levels of z_{fre} (inductive region), z_{fraun} (radiative region) and $4z_{fraun}$ (particular coordinate in the far field region).

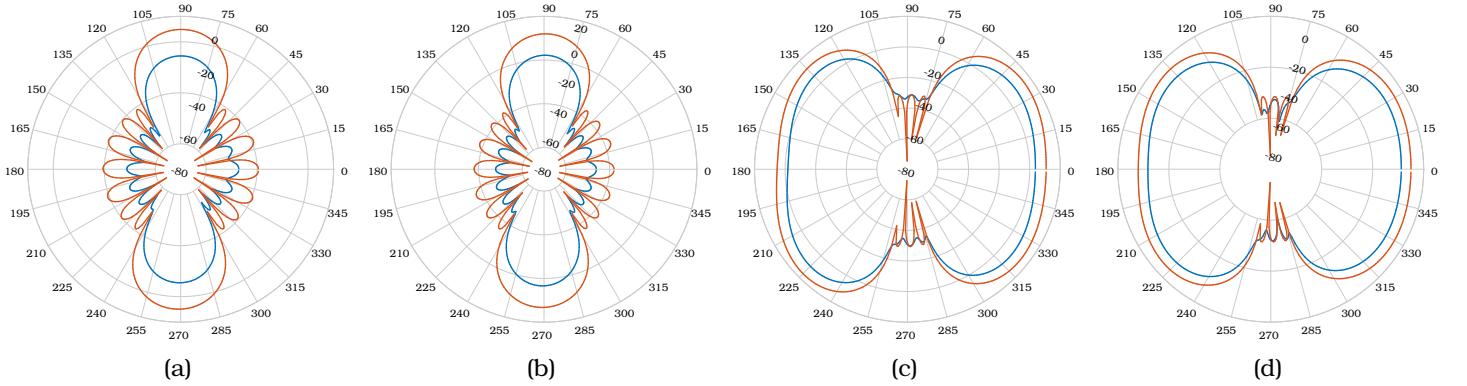


FIGURE 10: 2D gain patterns (ideal ■ and real ■) in both the broadside case ((a) azimuth cut and (b) elevation cut) and 45° off the boresight direction ((c) azimuth cut and (d) elevation cut).

total array gain will be computed and displayed in two different cases: by means of a fullwave model (with no approximation) and by using the pattern multiplication principle. This second case is idealistic because it makes an approximation of the real patterns by not considering the mutual induced interactions between the single antennas of the array; however, a "mathematical" interaction is considered in this model because it comes out of a formulation

that mixes the single element effect (which gives an element gain factor G_0) on the gain pattern with the angular filtering influence of the array factor, weighted by a combination of the current coefficients (which gives an array gain factor G_F). The overall idealistic case is based on the pattern multiplication principle, which (in a case such as the Tchebyshev array synthesis model, thus with non uniform amplitude feed), will take into account also the effect of

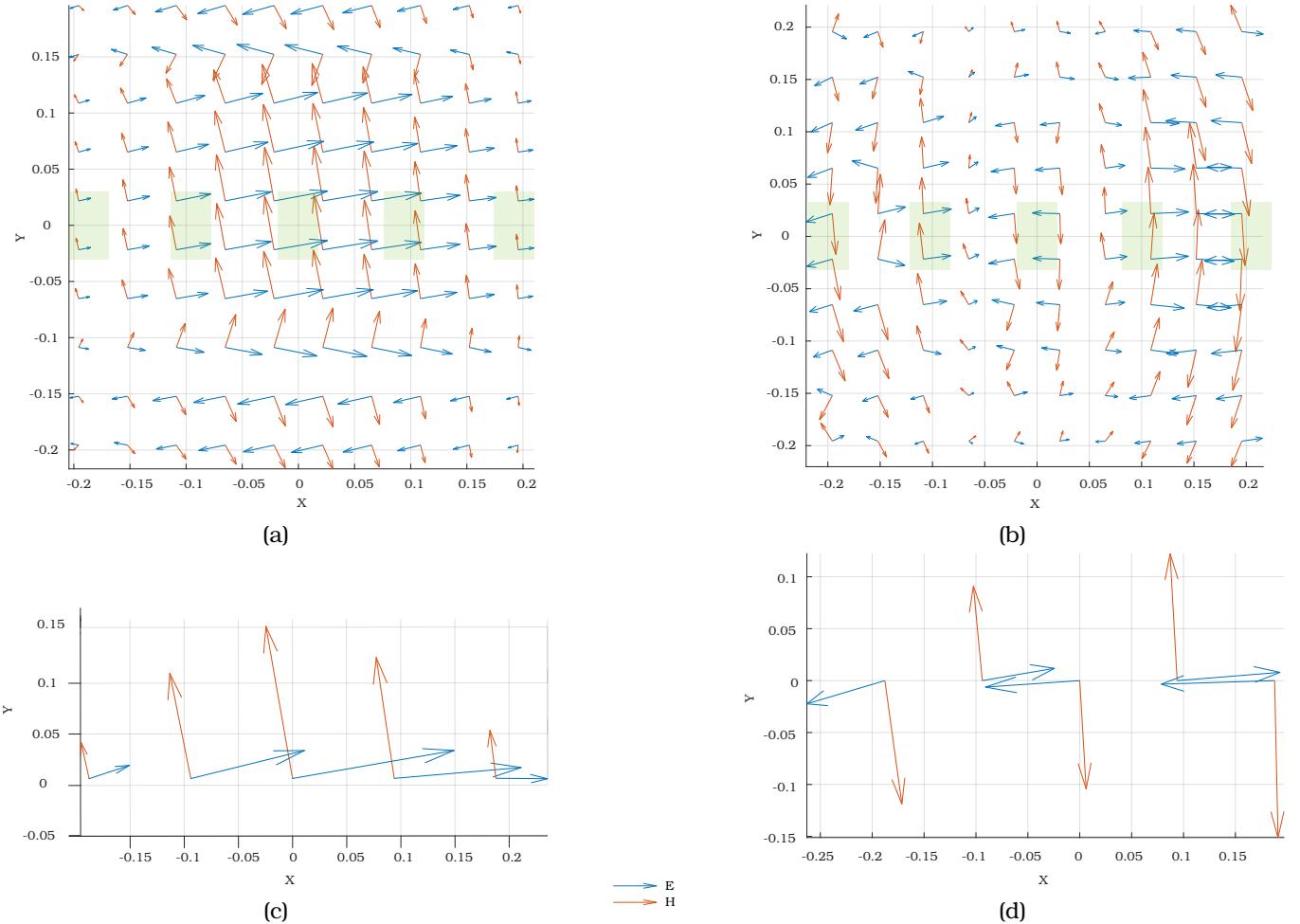


FIGURE 11: Electric and magnetic fields in Fresnel region in both broadside (a) and 45° off the boresight direction (b). Same fields calculated in the center of the position of the antennas in both broadside (c) and 45° off the boresight direction (d). The contribution of the various coefficients is evident.

the tapering efficiency (η_T) in its formulation (see eq. (13)). Thus, the pattern multiplication principle can be applied angle by angle and this will provide a set of values that can be computed and inserted in matrices by using MatLab. Starting from the matrices, the 2D patterns can be represented. On the other hand, an easier way is to use the `patternMultiply()` command and get the ideal patterns.

Electric and magnetic near fields

The near field characteristics (distinguishing between its inductive and radiative regions) have been analyzed in both the broadside and by beamsteering to 45° off the boresight conditions. Two situations have been analyzed: the general near field behaviour (in its electric and magnetic field components) but also the specific orientation of the fields in the center position of each of the five elements composing the antenna (so, considering the x and y coordinates of each patch corresponding to the length and width directions of its planar development and the z coordinate representing the vertical component of the distance between the observation point and the array). The vertical distance z has been considered in three different cases. First, the Fresnel region (inductive near field region) has been analyzed at $z_{fre} = 0.62\sqrt{L_{max}^3/\lambda}$ (where L_{max} is the maximum size of the array, thus $L_{max} = 4d_{opt} + L_{patch}$),

while the radiative near field region was measured, starting from the Fraunhofer limit expression ($z_{fraun} = 2D^2/\lambda$), at $z_{fraun}/2$. Moreover, it's been considered an observation point at a generic distance from the array in the far field region ($4z_{fraun}$). The three cases can be compared both in the broadside and in the 45° off the boresight direction examples. In the broadside case, the inductive and radiative near electric and magnetic fields can be ordered by following the increasing of their intensities as a criterion. It can be noted that the order of these intensities (calling them E_i , H_j , with $i, j = -2, 2$) is the same of that of the feed currents (the relationship is shown for the E field intensities - see eq. (11) - but it works analogously for the H field). In the far field, a totally different phenomenon is observable (at least in the broadside case and with good approximation in the 45° direction too): the E field components are all equal, as those relative to the magnetic field H . This can be interpreted as follows: in the near field, the elements of the array still don't behave completely as one radiating object so that their individual radiative effect can be observed, prevailing on the mutual interactions that will make them work as a whole at a further distance (i.e. in the far field). This fact can be explicitly analyzed by looking at the electric and magnetic fields registered at the specific distances: everything is shown in table 6. The facts

described above can be also observed in **fig. 11**.

$$\begin{array}{ll} \text{If} & C_0 > C_1 > C_2 \\ \text{and} & C_0 > C_{-1} > C_{-2} \\ \text{then} & E_0 > E_1 > E_2 \\ \text{then} & E_0 > E_{-1} > E_{-2} \end{array} \quad (11)$$

Methods

In this conclusive paragraph, a list of the MatLab commands that are not available for all the tools that have been used will be pointed out. Next, the computational costs related to the different parts of the project will be discussed. In the end, a comparison between the array of PIFAs realized in Matlab and the CST Studio Suite (Student edition) will be shown.

The theoretical formulations extracted from **[Balanis 1]** and adopted for this project regarding the pre-optimized PIFA model

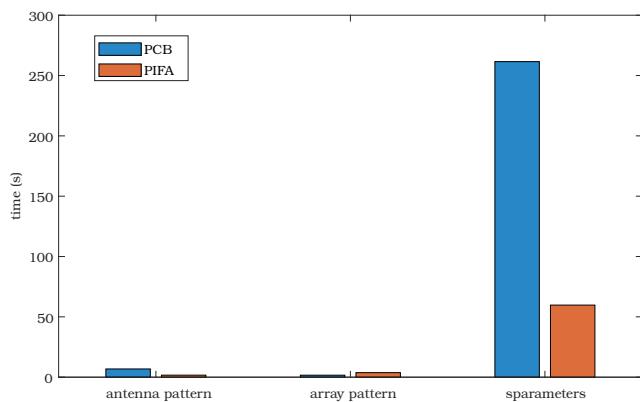


FIGURE 12: Analysis of the computational cost of PIFA vs PCB in the calculation of the radiation pattern (single antenna and array) and of the sparameters

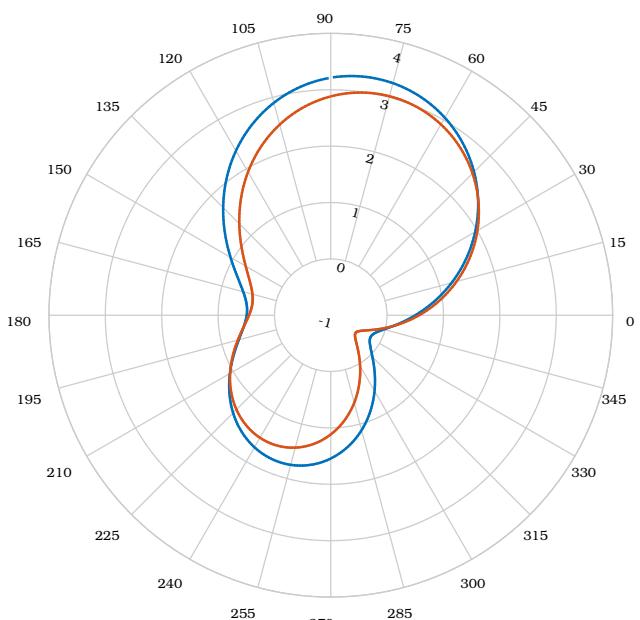


FIGURE 13: 2D gain patterns in the azimuth cut: comparison between the PIFA created in Matlab (■) and that designed in CST Studio Suite (■).

and the Tchebyshev array factor design have been also implemented in Wolfram Mathematica, which enables the user to write and compute formulas in a very fast and clear fashion (for this purpose the software acts as a Computer Algebra System).

Explaining what are the limitations of every tool and our previous suggestions about how to combine them (this made it possible to get conclusive results) may help the reader who wants to engage himself into doing some similar studies about array of antennas. Four different MatLab tools belonging to two different toolboxes have been used for this project. By using the Antenna Designer tool, the PIFA structure has been generated and analyzed, while the PCB stack has been created and studied through the PCB Antenna Designer and we acquired the array of PCBs from the AAD. All these tools belong to the Antenna Toolbox (**[AntennaToolbox]**), while the SAA (by which the study of the array of PIFAs is possible) is part of the Phased SAA (**[PhasedArraySystemToolbox]**). The differences in terms of the informations than can be directly extracted by using commands from the two main toolboxes used for this projects are shown in **table 7**.

We dedicate also a short paragraph to discuss about the computational costs. In terms of *random access memory* (RAM) occupied by the different elements during simulations, the single PIFA weights 960 MB, while the single PCB 1.2 GB; the array of PIFAs occupies 4 GB while the array of PCBs engages 5.6 GB of memory. The RAM occupation is not enough in order to describe the computational costs, but it represents just a starting point (it gives information about the "stone" that needs to be carried, but not about how long). Thus, the informations about the the time spent by the workstation adopted for the project to complete the calculations of different simulations are collected in **table 8**. The running time for one the radiation pattern simulation (in both single PIFA and PCB antennas and arrays) and the **sparameters()** function (which calculates a list of parameters which Γ belongs to) for both the single PIFA and the PCB is shown in **fig. 12**.

We realized a PIFA model also in CST Studio Suite and plotted its 2D radiation pattern diagrams. For the seek of a further comparison, the data related to the gain pattern in the azimuth cut have been exported as a .txt file which have been converted into an array in Matlab so that the gain pattern in this cut could be overlapped to that of the PIFA realized with the Antenna Designer (see).

$$u_0 = \alpha d_{opt} = 2\pi \frac{d_{opt}}{\lambda} \cos(\theta_0) \quad (12)$$

$$\alpha_n = n \alpha d_{opt} = n u_0 \quad (n = \overline{1, 5})$$

$$G_{total} = \eta_T G_0 G_T \quad (13)$$

Conclusions

The different analysis conducted in this paper have fulfilled the project requirements (none of them has been omitted). The toolbox I toolbox which have been used made possible to reach the final results (related to electric and magnetic fields) using the PCB structure as the antenna element of the array since the array of PIFAs isn't compatible for this type of study. Using the array of PCBs introduced an admissible set of errors. The use of the array

Commands	SAA	AAD
<code>pattern(...)</code> (it makes 2D or 3D patterns - i.e. gain, directivity, power - starting from different inputs; specifying the object and the frequency are two absolutely necessary inputs)	✓ *	✓
<code>patternAzimuth(...)/patternElevation(...)</code> (they create 2D patterns - i.e. directivity, power, ecc. - in the azimuth and elevation cut planes, respectively. Syntax specs to the previous command)	✓	✗
<code>patternMultiply</code> (the several patterns coming out of this command - i.e. gain, directivity, power - are computed by means of the pattern multiplication principle, a mathematical approximation of the radiation diagrams which doesn't take into account mutual coupling between the elements of the array)	✗	✓
<code>arrayFactor(...)</code> (it shows different patterns - i.e. gain, directivity, power - by disregarding completely their influence from the specific element used in the array)	✗	✓
<code>phased.SteeringVector</code> (it allows running beamsteering analysis by specifying the two spatial angles; it's the input of another command which gives all the weights related to possible phase shifts between the elements of the array)	✓	✗
<code>conformalArray(...)</code> (any element from the antenna or array library can be used to generate an antenna array, except of the PIFA and some other cases)	✓	✗
<code>polarpattern(...)</code> (different patterns from all the tools can be extracted as 2D matrices data objects. This gives the opportunity to overlap and compare patterns coming from different tools, by introducing the matrices as inputs of this command.)	✓	✓

* only 3D

TABLE 7: This table is listing some of the commands which have been used for this project. Some of them are partly or fully compatible with both the tools used for the array analysis and others have been implemented by Matlab to work only in one of them. The main issue of the simulation part was the array of PIFAs could be realized with SAA but using that tool didn't allow to study the electromagnetic field. Instead, the AAD permitted the realization of an array of PCBs, which electromagnetic field could be represented and discussed.

Function	Time* (s)
<code>refinement_time(...)</code>	$6.649 \cdot 10^3$
<code>refinement_time2(...)</code>	$3.816 \cdot 10^3$
<code>refinement_time3(...)</code>	$6.994 \cdot 10^3$
<code>contour_plot(...)</code>	$7.149 \cdot 10^3$
<code>impedance_matching(...)</code>	$1.752 \cdot 10^2$

* Intel i7-10850H @2.70 GHz - 32 GB RAM - Dell Precision 3551

TABLE 8: The optimization process went through different steps, as described in the previous sections. Here, some of the computational duration required to completely execute some of the simulations (related to one of the workstations which have been used) are listed so the reader may have an idea of the computational costs for this project

of PCBs led to a notable increase of the computational costs and this may represent an obstacle to users that would like to replicate the results but having machines who don't have enough memory resources.

There are also available more plot and information about antenna and array properties.

There are all the raw data for plot and calculation.

Data availability

All the Matlab code is available in the online repository (file `final_wet_project.m`) at the following link: <https://github.com/mastroalex/antenna-design>.

In the appendix there are different patterns (for PCB and PIFA comparison), 3D gain patterns, Smith diagram, and different antenna / array diagrams. There is also a project workflow and some code snippets.

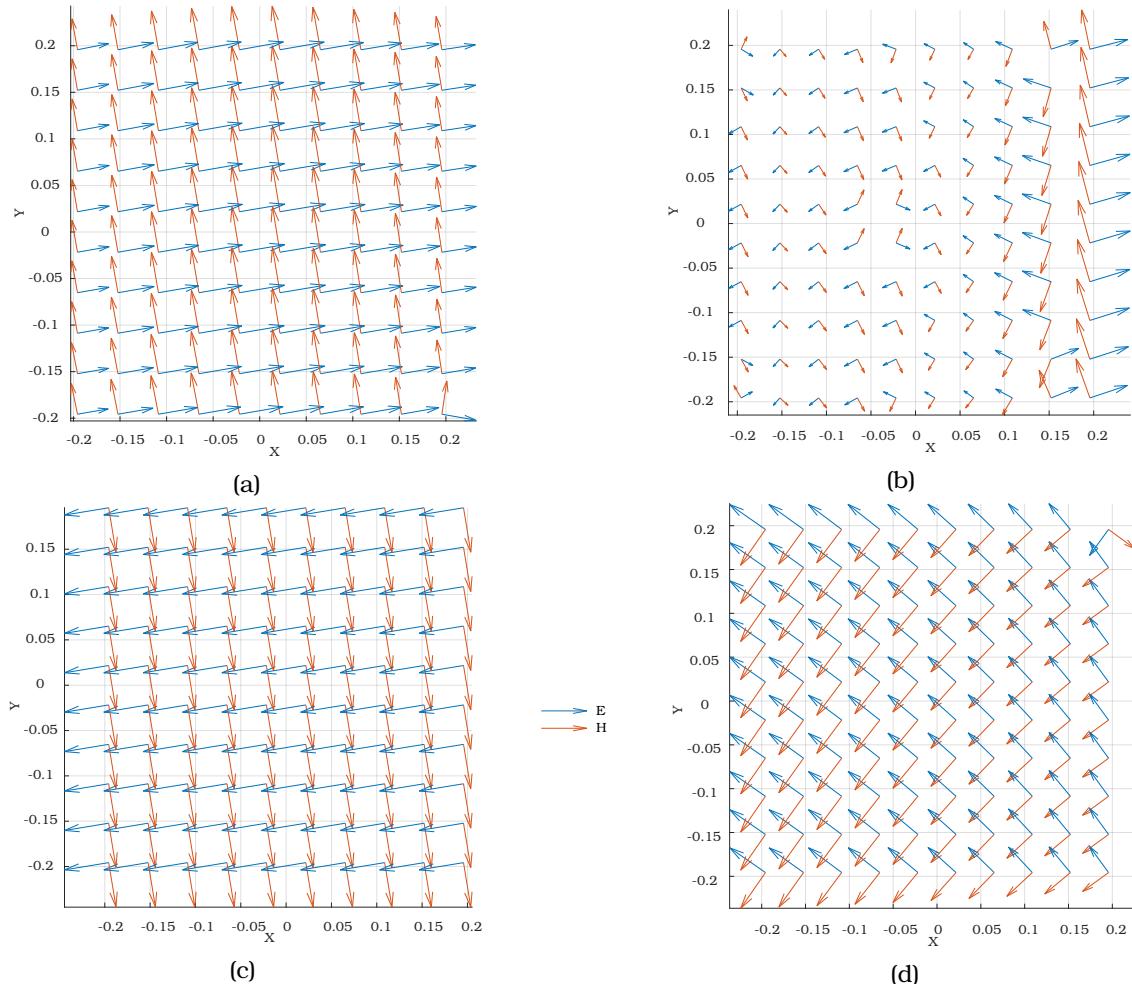


FIGURE 14: EH Fields in radiative near field region for broadside (a) and 45° off boresight direction (b) and in far field region for broadside (c) and 45° off boresight direction (d)

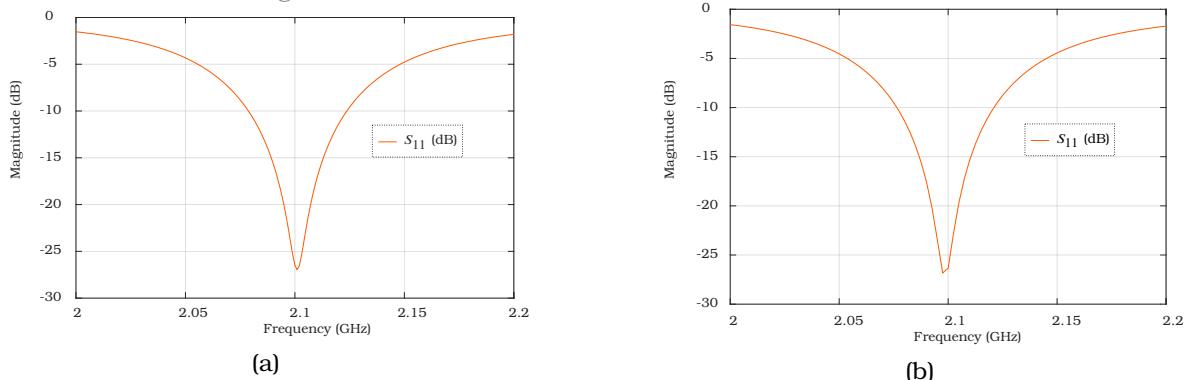


FIGURE 15: The two best Γ plots depending on the specific patch size expressed coupled variations ($L_{\text{patch}}, W_{\text{patch}}$) and represented in a frequency range of [2.0, 2.1] GHz

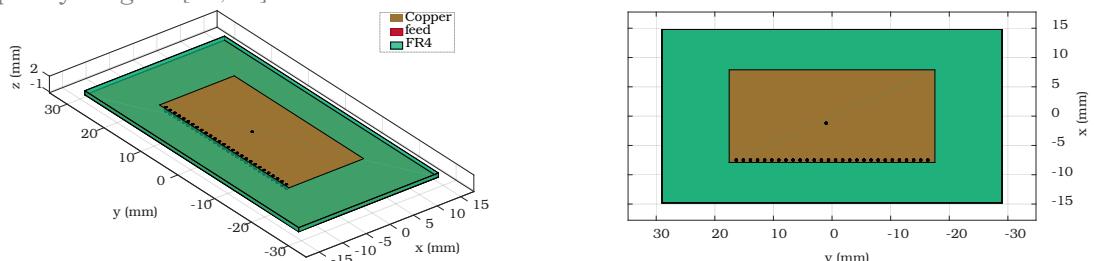


FIGURE 16: 3D view and top view of the PIFA approximation realized with a PCB stack and cylindrical shorting pins

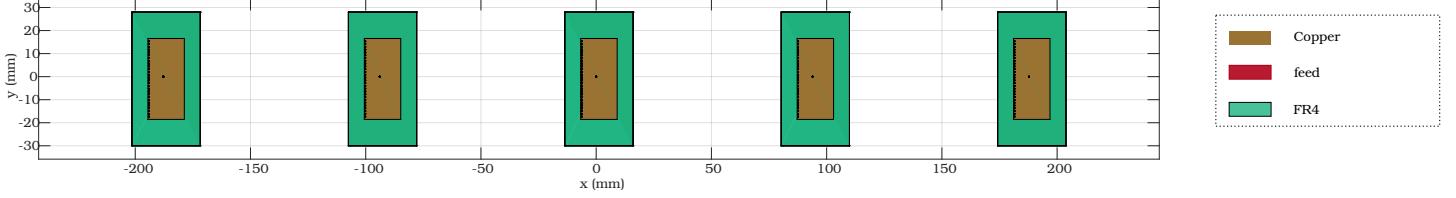


FIGURE 17: PCB array

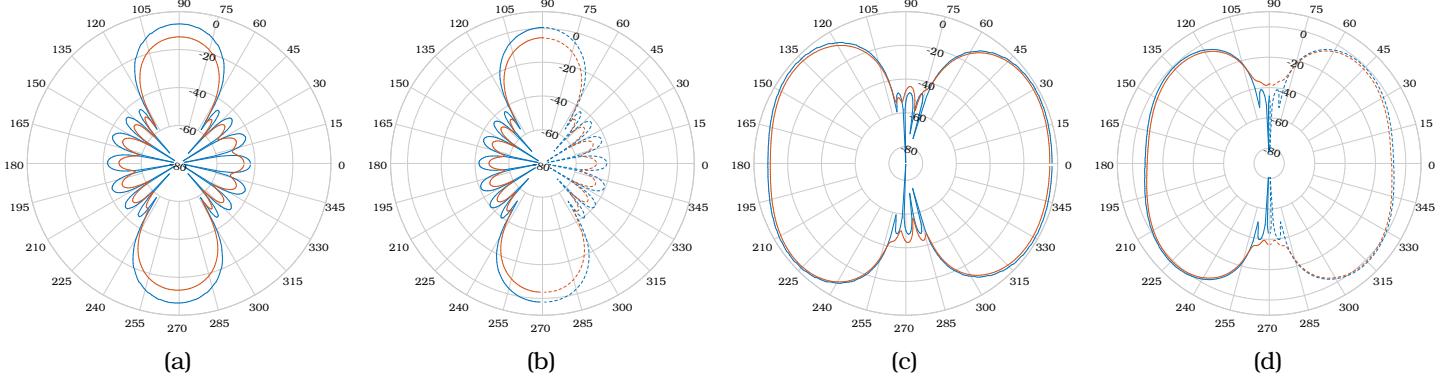


FIGURE 18: 2D gain patterns (PIFA (■) and PCB (□)) in both the broadside case ((a) azimuth cut and (b) elevation cut) and 45° off the boresight direction ((c) azimuth cut and (d) elevation cut). The curve with the dashed line has not been calculated regularly but mirrored, due to the limitation in the maximum elevation angle range which can be asked for in the SAA patterns.

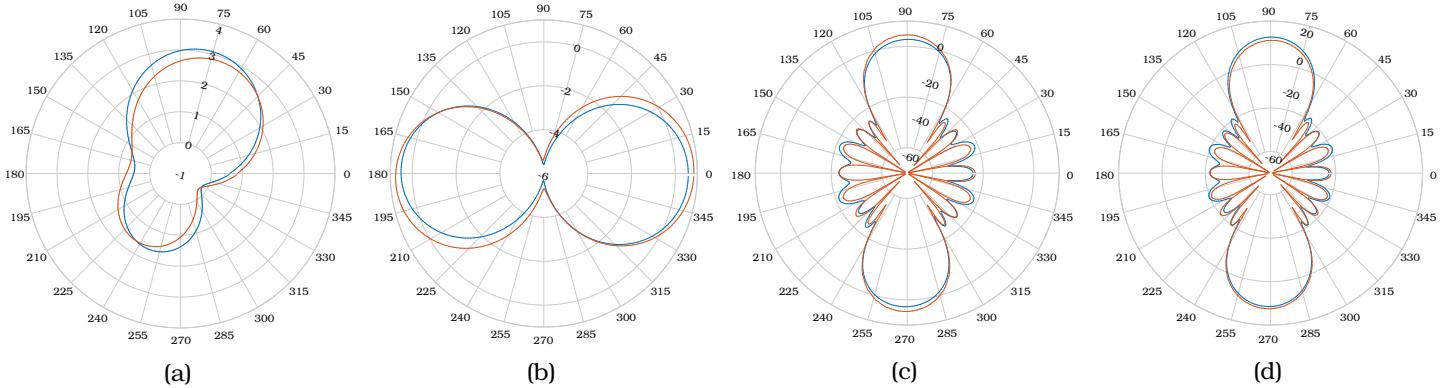


FIGURE 19: PIFA (■) and PCB (□) single antenna directivity patterns (dB) in the Azimuth cut ($\theta_{el} = 0^\circ$, (a)) and in the Elevation cut ($\phi_{az} = 0^\circ$, (b)). PIFA and PCB arrays directivity patterns (dB) in the Azimuth cut ($\theta_{el} = 0^\circ$, (c)) and in the Elevation cut ($\phi_{az} = 0^\circ$, (d))

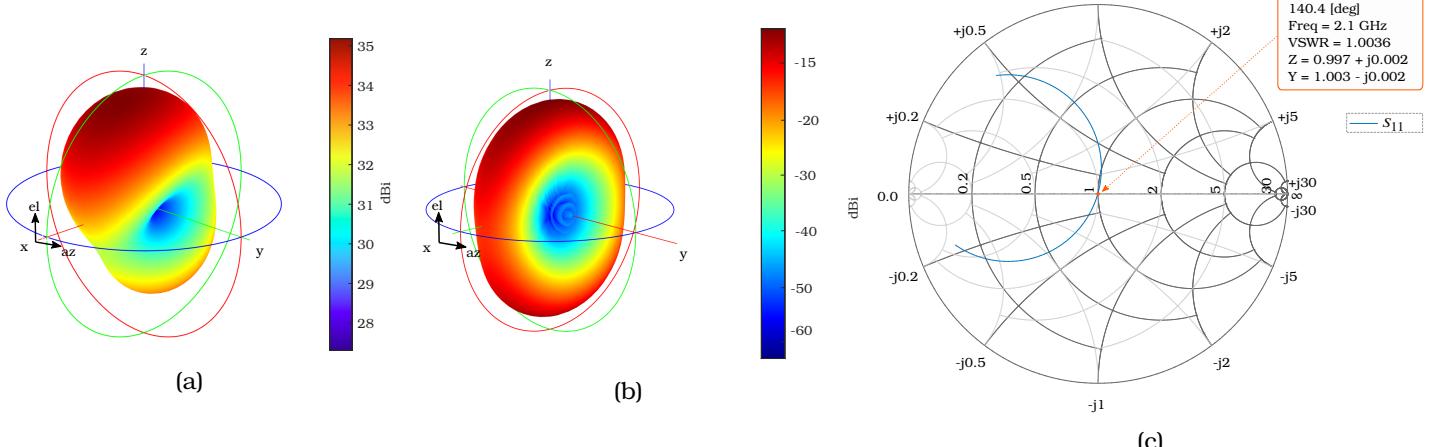


FIGURE 20: Gain pattern for PIFA single antenna (a) and PCB array (b). Smith chart (c)

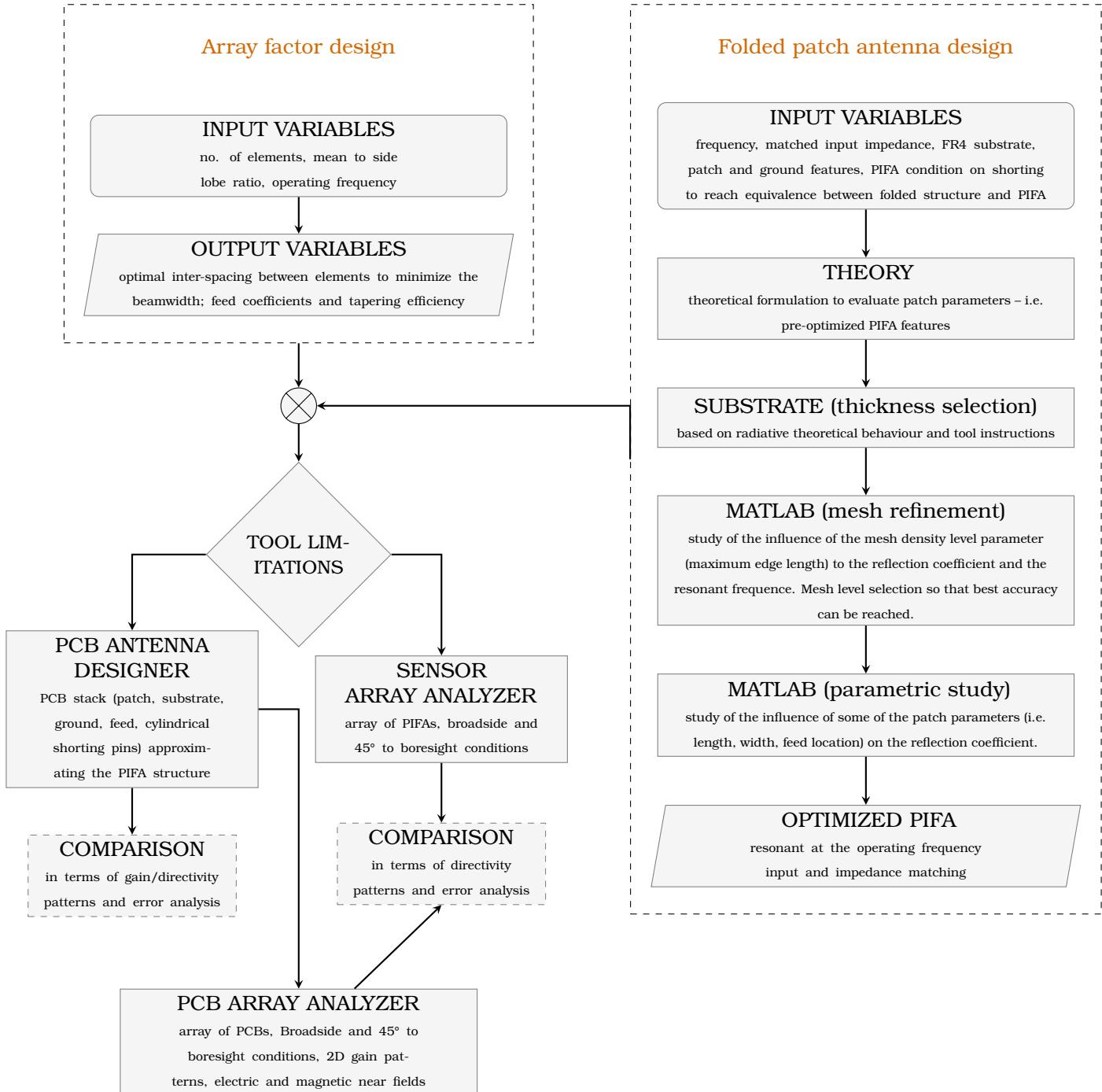


FIGURE 21: Flowchart

```

1 pat_dotchy_az=pattern(dotchy,2.1e9
2   ,-180:180,0,'Type','directivity');
3 pat_kfa_az=pattern(kfa,2.1e9,-180:180,0,
4   'Type','directivity');
5 Df_az=10.^((pat_kfa_az-pat_dotchy_az)
6   /10);
7 F=sqrt(Df_az*(dot(chebc,chebc)/0.79));
8 Fn=sqrt(Df_az*(dot(chebn,chebn)/0.79));
9 FdB=20*log10(Fn);
10 subfigure(3,4,3);
11 polarpattern(FdB);

```

Listing (1) Calculation of array factor to avoid Toolbox incompatibility

```

1 u0 = -k0*dopt*cos(pi/4);
2 an = zeros(5,1);
3 for i=0:4
4   an(i+1,1) = i*u0;
5 end
6 kfa_ster=kfa;
7 kfa_ster.PhaseShift = an*(180/pi);

```

Listing (2) Manual phase shift of PCB array for beamsteering