electronics

MDPI

Article

Support Vector Regression for the Modeling and Synthesis of Near-Field Focused Antenna Arrays

Rafael González Ayestarán

Department of Electrical Engineering, Universidad de Oviedo, Campus Universitario, Edif. Polivalente, 33203 Gijón, Asturias, Spain; rayestaran@uniovi.es; Tel.: +34-985182212

Received: 29 September 2019; Accepted: 13 November 2019; Published: 15 November 2019

Abstract: The powerful support vector regression framework is proposed in a novel method for near-field focusing using antenna arrays. By using this machine-learning method, the set of weights required in the elements of an array can be calculated to achieve an assigned near-field distribution focused on one or more positions. The computational cost is concentrated in an initial training process so that the trained system is fast enough for applications where moving devices are involved. The increased learning capabilities of support vector machines allow using a reduced number of training samples. Thus, these training samples may be generated with a prototype or a convenient electromagnetic analysis tool, and hence realistic effects, such as coupling or the individual radiation patterns of the elements of the arrays, are accounted for. Illustrative examples are presented.

Keywords: near field; antennas; focusing; multifocusing; support vector regression

1. Introduction

Near-field focusing (NFF) [1–5] is one of the state-of-the-art topics related to antenna design that is gaining more attention in the recent years. It has become one of the most interesting approaches to emerging applications such as RFID [2,4], medical hyperthermia [6], wireless energy transfer (WET) [7], wireless energy and information transfer (WEIT) [8,9], imaging, Internet of things (IoT) or 5G mobile telephony, which are based on wireless links between devices usually located at short distances. NFF allows antenna arrays to concentrate the energy on an assigned position or spot in the near-field (NF) region of the antenna, reducing the waste of energy directed to positions of space where it is not necessary. The conjugate-phase (CP) method proposed in [2,10,11] is an excellent technique for calculating the phase-shift that must be applied to each array element so that all the signal contributions from the individual elements arrive in-phase to the position of interest, creating a constructive interference that increases the field level at such position. This is a robust and simple approach whose performance has been undoubtedly proven. However, the CP method has been lacking for certain cases where it is limited, for example in emerging applications with more demanding specifications that require concentrating the field on more than one focal point or arbitrary volumes. In such cases, near-field multi-focusing (NFMF) has arisen as an excellent alternative [12,13].

NFMF is based on the minimization of a proper cost function designed to account for all the requirements. Different optimization techniques are used to calculate the set of weights that must be applied to the array elements so that the resulting field distribution is focused on the assigned spots. Although the resulting method is flexible and powerful with increased focusing performance, it is time consuming due to the iterative nature of the optimization techniques involved. In the case of applications requiring fast adaptation, for example because moving devices are involved, this could represent an important limitation as the time required to calculate the new weights to focus on the new positions could be unacceptable.

Recently, in [14] neural networks (NN) [15] have been proposed as an alternative to optimization for NFMF applications, adapting previous successful approaches used for far field (FF) synthesis problems [16,17]. The resulting focusing performance is similar to that achieved using optimization techniques, but its time–cost is strongly reduced once the NN has been trained; it is able to calculate the weights required to achieve a given field distribution without relevant delay. All the computing time is concentrated in the previous training step, but providing solutions almost in real time once trained. The training or learning of any machine learning (ML) system consists of presenting known pairs of inputs and outputs so that the system is able to extract the relationship between them and provide outputs to new inputs. In [14], the training pairs presented to the NN are samples of the NF distributions and the weights applied to the array to achieve them. Although NNs are a very popular tool in the framework of ML methods, other alternatives have been shown to present improved learning capabilities, avoiding the problem of *overfitting* [15] if the NN is trained excessively, and requiring much less training data. In [14], the number of pairs required to train the NN is in the order of 5000. If the training patterns are obtained through simulation, the time requested to obtain such a number of patterns might be excessive; obtaining them through measurement of a real antenna is impossible in practice. In this paper we propose the use of support vector regression (SVR), a powerful ML method based on support vector machines (SVM) and the associated structural risk minimization principle (SRM) [18] to obtain the weights to be applied to the array elements so that an antenna array is able to multifocus or to generate a near-field (NF) distribution complying with the specifications. It requires a strongly reduced set of training patterns with respect to NNs, so that obtaining them becomes easier, and the system still operates fast enough for real time applications. Although SVR has been used in [19] for NFF applications, it was only used as an auxiliary tool used to calculate a coupling matrix required to model interaction between elements in the results of a conventional optimization method. In this paper we propose using SVR directly to synthesize the weights required for focusing on some assigned points.

This paper is organized as follows: Section 2 presents the use of SVR to create a model of the array able to relate field samples and weights. In Section 3 this model is used to perform the synthesis. Some illustrative results are presented in Section 4. These results and some conclusions are discussed in Section 5.

2. Support Vector Regression-Based Inverse Model of the Array

The field radiated at a certain position $\vec{r}$ by an antenna array with N elements is given by the superposition of the field radiated by its individual elements. A general and simple formulation representing this sum of contributions is

$$E(\vec{r}) = \sum_{n=1}^{N} \omega_n \cdot g_n(\vec{r}) = \mathbf{g}^T(\vec{r}) \cdot \mathbf{w} \tag{1}$$

where $E(\vec{r})$ is the value of the considered component of the field at $\vec{r}$, ω_n is the feeding weight applied to the n-th array element, whose contribution to the field at $\vec{r}$ is $g_n(\vec{r})$. These values are also represented in vector form as $\mathbf{w} = [\omega_1, \omega_2, \ldots, \omega_N]^T$ and $\mathbf{g}(\vec{r}) = [g_1(\vec{r}), g_2(\vec{r}), \ldots, g_N(\vec{r})]^T$, with $(.)^T$ representing the transpose operator.

If M positions of the near-field region of the antenna are evaluated, $\vec{r}_m, m = 1 \ldots M$, Equation (1) may be rewritten in matrix form as

$$\mathbf{e} = \mathbf{G} \cdot \mathbf{w} \tag{2}$$

with $\mathbf{e} = [E(\vec{r}_1), E(\vec{r}_2), \ldots, E(\vec{r}_M)]^T$ and $\mathbf{G} = [\mathbf{g}(\vec{r}_1), \mathbf{g}(\vec{r}_2), \ldots, \mathbf{g}(\vec{r}_M)]^T$. A proper modeling of the array involves determining or calculating $\mathbf{G}$. However, for synthesis purposes an inverse model is more adequate.

In the usual case of $M > N$ (i.e., the number of positions where the field is evaluated is greater than the number of elements in the array) Equation (2) represents an overdetermined system of linear equations. If a least-squares criterion is chosen to find a solution, the problem becomes

$$\underset{\mathbf{w}}{\text{minimize}}\ ||\mathbf{e} - \mathbf{G} \cdot \mathbf{w}||^2 \tag{3}$$

where $||.||$ stands for the Euclidean norm.

The solution to Equation (3) is well known [20] and given by

$$\mathbf{w} = (\mathbf{G}^H \cdot \mathbf{G})^{-1} \cdot \mathbf{G}^H \cdot \mathbf{e} = \mathbf{G}^+ \cdot \mathbf{e} \tag{4}$$

where $\mathbf{G}^+$ is the pseudoinverse matrix for the overdetermined problem, and it represents an inverse model of the antenna array. Equation (4) shows that a linear model $\mathbf{A} = \mathbf{G}^+$ may be obtained to calculate the weights that must be applied to an antenna array so that it radiates according to a given field distribution specified by its samples, $\mathbf{w} = \mathbf{A} \cdot \mathbf{e}$.

The SVR framework is proposed to calculate the mentioned model. Let us consider that a set of P pairs $\{\mathbf{w}^{(p)}, \mathbf{e}^{(p)}\}$, $p = 1 \ldots P$, are available for training purposes. Each training pair (or training pattern) consists of a vector of weights applied to the elements of the array and the corresponding field distribution represented by its samples. In this training dataset, the inputs correspond to the field distributions represented by their samples, so that the number of samples for each pattern is defined by the number M of locations in the NF region of the antenna that are considered. The outputs are represented by the complex weights applied to the elements of the array, i.e., a total of N complex values for each sample. It is straightforward that the total number of values involved is $M \cdot P$ in the inputs and $N \cdot P$ in the outputs. These training patterns may have been obtained by measurements, EM simulations or by any means. If they are obtained using a method that accounts for the realistic properties of the array (including coupling effects between elements, non-idealities, etc.), the training patterns also take these properties into account, hence increasing its accuracy.

If we focus on the n-th element of the array in order to obtain the feeding weight that must be applied to it, the SRM principle establishes that the n-th row of $\mathbf{A}$, $\mathbf{a}_n$, can be obtained through minimizing a cost function given by [18]:

$$J(\mathbf{a}_n) = \frac{1}{2}||\mathbf{a}_n||^2 + C \sum_{p=1}^{P} |\omega_n^{(p)} - \mathbf{a}_n \cdot \mathbf{e}^{(p)}|_\varepsilon \tag{5}$$

where

$$|\omega_n^{(p)} - \mathbf{a}_n \cdot \mathbf{e}^{(p)}|_\varepsilon = \max\left(0, |\omega_n^{(p)} - \mathbf{a}_n \cdot \mathbf{e}^{(p)}| - \varepsilon\right) \tag{6}$$

is Vapnik's ε-insensitive loss function, and $C > 0$ is a penalty value used to balance the model complexity (controlled by the first term of Equation (5)) and the cost of deviations larger than ε. This parameter is related to the precision of the field sampling, the noise or any other uncertainty factor that might affect the accuracy of the training patterns. This cost function implies the choice of a linear kernel for the SVR [18]. This is a straightforward choice due to the linear relation between inputs and outputs made explicit in Equation (4).

In order to solve this minimization problem, a set of positive slack variables ξ_p and $\tilde{\xi}_p$ are introduced so that Equation (5) may be rewritten as a constrained minimization problem given by

$$\underset{\mathbf{a}_n}{\text{minimize}}\ \frac{1}{2}||\mathbf{a}_n||^2 + C \sum_{p=1}^{P} (\xi_p + \tilde{\xi}_p) \tag{7}$$

subject to

$$\begin{aligned} \mathbf{a}_n \cdot \mathbf{e}^{(p)} - \omega_n^{(p)} &\leq \varepsilon + \xi_p, \qquad \forall p = 1 \ldots P \\ \omega_n^{(p)} - \mathbf{a}_n \cdot \mathbf{e}^{(p)} &\leq \varepsilon + \tilde{\xi}_p, \qquad \forall p = 1 \ldots P \\ \xi_p, \tilde{\xi}_p &\geq 0, \qquad \forall p = 1 \ldots P \end{aligned}$$

This problem is usually solved making use of the Lagrange multiplier technique [18], which leads to an optimal regressor (i.e., the optimal values of the n-th row of model $\mathbf{A}$) given by the following expression:

$$\hat{\mathbf{a}}_n = \sum_{p=1}^{P} (\tilde{\alpha}_p - \alpha_p) \mathbf{e}^{(p)} \tag{8}$$

which is a function of the field distribution training patterns, linearly combined using $\tilde{\alpha}_p$ and α_p, positive Lagrange multipliers resulting from the maximization of the quadratic dual problem expressed as

$$W(\tilde{\alpha}, \alpha) = -\sum_{p=1}^{P} \varepsilon(\tilde{\alpha}_p + \alpha_p) + \sum_{p=1}^{P} \omega_n^{(p)} (\tilde{\alpha}_p + \alpha_p) - \frac{1}{2} \sum_{p,q=1}^{P} (\tilde{\alpha}_p + \alpha_p)(\tilde{\alpha}_q + \alpha_q) \langle \mathbf{e}^{(p)}, \mathbf{e}^{(q)} \rangle \tag{9}$$

subject to $0 \leq \tilde{\alpha}, \alpha \leq C$. The expression $\langle \mathbf{e}^{(p)}, \mathbf{e}^{(q)} \rangle$ stands for the inner product, and it is applied to pairs of near-field distributions taken from the training patterns. The solution of this dual quadratic programming (QP) problem can be efficiently found taking advantage of its convexity (for example, using [21,22]). The support vector theory states that only some training patterns (typically less than P) contribute to Equation (8) while the other terms in the summation are canceled because their corresponding Lagrange multipliers are both zero-valued. The training patterns with non-null multipliers are referred to as *support vectors*.

The solution to Equation (8) allows calculating the weight ω_n that must be applied to the n-th element of the array. This regression must be repeated for the total N elements, obtaining the rows of the estimated model $\hat{\mathbf{A}}$. Finally, the total set of weights to be applied to the array elements to produce a given field distribution can be estimated as

$$\hat{\mathbf{w}} = \hat{\mathbf{A}} \cdot \mathbf{e}. \tag{10}$$

The number of training patterns P used to perform the regression is a critical choice for the accuracy of the model in ML methods. Unfortunately, none of the methods proposed in the literature for the choice of P has been accepted as valid in general situations. A strategy based on the cross-evaluation of an independent test or *validation* set (usually a subset of the available patterns not used for training) is typically followed. In the examples presented along this paper, the validation set consists of new patterns obtained using the same procedure as with the training set. In all cases, the number of validation patterns has been chosen to be 10% of the training set, i.e., $0.1P$. Although other machine learning methods may be affected by the problem of *overfitting* (for example neural networks [15]), the SRM principle applied in SVR guarantees robustness against overfitting if P is too high, and improved learning capabilities (i.e., a smaller P is necessary for similar performance).

2.1. Handling Complex Values in Real-Valued Schemes

The optimization problems proposed in Equations (7) and (9) are complex-valued, as both the field samples and the weights applied to the array are complex, and also the model relating them is composed of complex values. Nowadays most optimization schemes and toolboxes are able to handle complex values, but some ML methods are designed to work with real numbers. In such cases, the regression can be reformulated to be handled by the real-valued algorithms.

Let us consider a vector of real values $\tilde{\mathbf{w}} = [\Re\{\mathbf{w}^T\}\ \Im\{\mathbf{w}^T\}]^T$ obtained by stacking the real and imaginary parts of the weight vector $\mathbf{w}$. The real and imaginary parts of the samples of the field distribution may also be stacked to create an extended real-valued vector $\tilde{\mathbf{e}} = [\Re\{\mathbf{e}^T\}\ \Im\{\mathbf{e}^T\}]^T$ so that both extended vectors are related according to

$$\tilde{\mathbf{w}} = \tilde{\mathbf{A}} \cdot \tilde{\mathbf{e}} \tag{11}$$

with an extended model relating both vectors given by

$$\tilde{\mathbf{A}} = \begin{bmatrix} \Re\{\mathbf{A}\} & -\Im\{\mathbf{A}\} \\ \Im\{\mathbf{A}\} & \Re\{\mathbf{A}\} \end{bmatrix}. \tag{12}$$

The optimization problem Equation (7) can be adapted to deal with this extended model by simply considering this real-valued formulation. The regression must be applied $2N$ times to calculate the rows of the extended model $\tilde{\mathbf{A}}$:

$$\underset{\mathbf{a}_{n'}}{\text{minimize}}\ \frac{1}{2}||\mathbf{a}_{n'}||^2 + C\sum_{p=1}^{P}(\xi_p + \tilde{\xi}_p), \qquad \text{for } n' = 1\ldots 2N \tag{13}$$

subject to

$$\begin{aligned} \mathbf{a}_{n'} \cdot \mathbf{e}^{(p)} - \omega_{n'}^{(p)} &\leq \varepsilon + \xi_p, && \forall p = 1\ldots P \\ \omega_{n'}^{(p)} - \mathbf{a}_{n'} \cdot \mathbf{e}^{(p)} &\leq \varepsilon + \tilde{\xi}_p, && \forall p = 1\ldots P \\ \xi_p, \tilde{\xi}_p &\geq 0, && \forall p = 1\ldots P. \end{aligned}$$

From the extended model it is straightforward to calculate the complex-valued model.

It is interesting to notice that the training dataset has to be modified to use this real-valued approach. In this case, instead of using the weight vector and the complex samples of the corresponding field distribution, the real-valued version must also be used by stacking the real and imaginary parts of both vectors, $\{\tilde{\mathbf{w}}^{(p)}, \tilde{\mathbf{e}}^{(p)}\}, p = 1\ldots P$, with $\tilde{\mathbf{w}}^{(p)} = [\Re\{\mathbf{w}^{(p)T}\}\ \Im\{\mathbf{w}^{(p)T}\}]^T$ and $\tilde{\mathbf{e}}^{(p)} = [\Re\{\mathbf{e}^{(p)T}\}\ \Im\{\mathbf{e}^{(p)T}\}]^T$. In this training dataset, the inputs correspond again to the field distributions represented by their samples, but the number of values for each pattern is $2M$ due to the separation of real and imaginary parts. In the same way, the outputs are represented by $2N$ real values for each sample. The total number of values involved is now $2M \cdot P$ in the inputs and $2N \cdot P$ in the outputs.

3. Near-Field Focusing Using the SVR-Based Inverse Model

Some previous methods proposed for near-field focusing aplications are based on the use of models relating a target NF distribution and the weights applied to the array elements. For example, the method presented in [13] makes use of a typical formulation based on a near-field array factor represented in matrix form. In order to specify the NF requirements it builds a target field distribution defined as a unitary value for the samples corresponding to the focal points and null values at any other position. The same procedure for the specification of the NF was used in the NN method proposed in [14]. It is a very flexible specification procedure as far as it allows assigning multiple spots or arbitrary focusing volumes just by setting unitary values at different sets of samples. It is obvious that such artificial target distribution with only 1 and 0 values is not physically feasible, but it has been shown to be quite effective for NFF purposes, so it will be used as basis for the method proposed here.

In the method proposed in this paper, the SVR-based inverse model relates a target NF distribution designed according to the NFF requirements, and the weights that must be applied to the array. The design of the target NF distribution allows non-unitary values to be specified for the focal points, so that different power densities can be assigned to them, or more complicated shaped field distributions can be requested. The resulting target distribution is built according to the following formulation:

$$E(\vec{r}_m) = \begin{cases} C_m, & \text{if } \vec{r}_m \text{ is a focal point} \\ 0, & \text{if } \vec{r}_m \text{ is not a focal point} \end{cases} \tag{14}$$

where C_m is the field value required at the position $\vec{r}_m$, provided that it is a point where non-vanishing field is requested.

Once the target field distribution has been determined, the weights to be applied to the array can be calculated using the SVR model according to Equation (10).

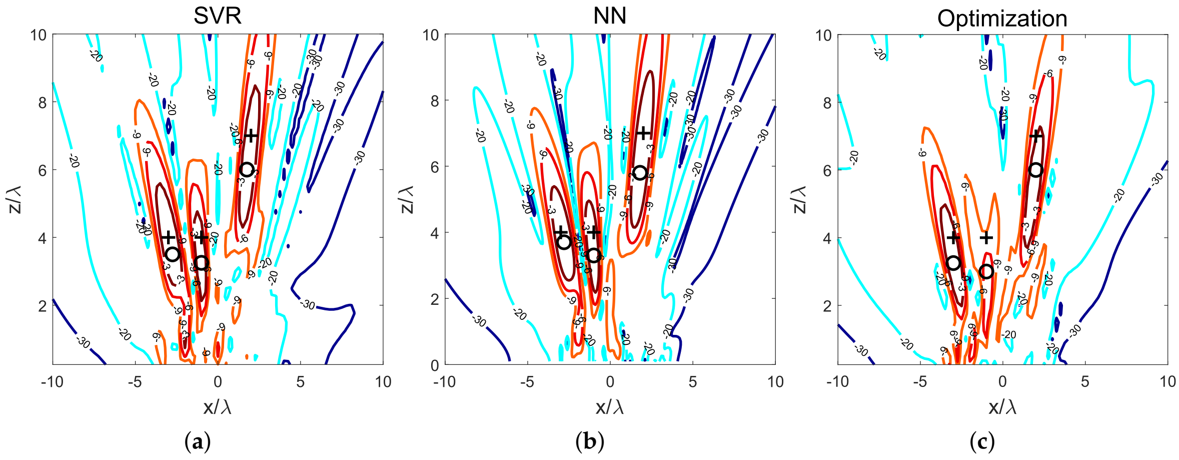
4. Results

Some experiments have been carried out in order to evaluate the performance of the proposed method. A regular planar antenna array with 12 × 12 elements ($N = 144$) is considered in first place. The aperture is placed centered in the plane $z = 0$, so that the broadside direction coincides with the z-axis. The interelement distance is 0.6λ and λ is the operation wavelength. The chosen individual radiating elements are hemispherical dielectric resonator antennas (DRA) [23], already used in [14,19,24] for testing due to their low losses but relevant mutual coupling effects when included in an array. The radius of each hemisphere is 12.5 mm and the relative permitivity is $\epsilon_r = 9.8$. A metallic pin with radius 0.63 mm, height 6.5 mm and offset 6.5 mm is used to feed the DRA. The working frequency is 3.6 GHz, hence exciting the TE_{11} mode. The range of variation of S_{11} for the elements is between -10.17 and -13.72 dB, being the maximum of the rest of S parameters -12.35 dB, so that the coupling effects are relevant in the resulting field distribution. In the results presented in this paper, the method of moments (MoM) [25] has been chosen to obtain the training patterns, as it is able to analyze quasi-realistically the real properties of the radiating structure. An infinite ground plane has been considered for the MoM analysis. A set of feeding voltages has been applied to each element, then calculating the resulting NF distribution to create one of the P training patterns. The NF distribution has been sampled around the antenna in a region limited to $x \in [-5\lambda, 5\lambda]$, $y \in [-5\lambda, 5\lambda]$ and $z \in [0.5\lambda, 10\lambda]$, with a sampling period $\lambda/2$ resulting in 8820 samples of the field distribution. A set of 120 training patterns has been generated, each of them formed by a vector of 144 complex weights and the corresponding samples of the NF distribution calculated using MoM. The SVR is used to obtain an inverse model of the array able to relate field samples and weights applied to the array, setting the trade-off parameter $C = 1$ and with $\varepsilon = 0.001$ (a low value as the results of simulations are not affected by noise). The mean square error after the training is 0.038 for the training patterns and 0.04 for the validation set.

Once the inverse model of the antenna array is obtained, it is used to calculate the weights that must be applied to the array in order to obtain a simultaneous focus on three focal points at $\{x, y, z\} = \{2\lambda, 0, 7\lambda\}$, $\{-\lambda, 0, 4\lambda\}$ and $\{-3\lambda, 0, 4\lambda\}$, to reproduce the experiment presented in [14]. A target field distribution is built using Equation (14), using $C_m = 1$ for the assigned focal points and 0 for any other position, and the resulting vector $\mathbf{e}$ is applied to Equation (10). The resulting weights have been used in the MoM model of the antenna to evaluate the resulting radiated field. Figure 1 shows the NF power density distribution in the plane $y = 0$ where all the focal points have been specified. The SVR-based model has been able to generate -3 dB focal spots containing the three focal points, and spending 0.02 s. The NN presented in [14] and the optimization method proposed in [13] have been used for comparison. The NN-based method has spent 0.19 s for the same problem, while the optimization technique has spent 82.7 s (as stated in [14], this method presents a lower focusing performance as one of the focal points lays out of a -3 dB spot with respect to the maximum field power in the NF region). The calculations have been carried out in a conventional PC with an Intel Core i5-7500 CPU, 3.4 GHz and 8 GB RAM, using MatLab R2019a as programming tool, and averaging 20 simulations for all the three methods. Both the SVR- and the NN-based methods spent times suitable for applications with moving devices, but the NN has required 5000 training pairs and the SVR has used 120 pairs to achieve similar performance and faster operation.

The assigned focal points are shown in Figure 1 along with the resulting synthesized spots and the points where the radiated field power is actually maximum. For comparison purposes, the results obtained in [13,14] have been reproduced. The proposed SVR model generates a solution with an NF distribution with the three assigned focal points into −3dB spots, and maximum values of radiated field power density at the positions $\{1.8\lambda, 0, 6\lambda\}$, $\{-\lambda, 0, 3.3\lambda\}$ and $\{-2.8\lambda, 0, 3.65\lambda\}$. These points are located at an average distance of 0.7λ from the assigned focal points. The NN method outputs a set of weights corresponding to an NF distribution with three maxima located in the positions $\{1.8\lambda, 0, 5.8\lambda\}$, $\{-1\lambda, 0, 3.3\lambda\}$ and $\{-2.8\lambda, 0, 3.7\lambda\}$, with an average distance 0.75λ between the assigned focal points and the actual maximum points. The optimization method, not accounting for coupling effects, provides a set of weights that correspond to a radiated NF distribution with −3dB spots including only two of the three assigned focal points; its maximum points are located at the positions $\{2\lambda, 0, 6\lambda\}$, $\{-\lambda, 0, 3\lambda\}$ and $\{-3\lambda, 0, 3.25\lambda\}$ with an average distance to the assigned focal points 0.91λ, considering also the focal point out of the −3 dB spot. The size of the resulting −3 dB spots is summarized in Table 1.

From these results it may be observed that the use the use of training patterns obtained including the realistic properties of the array leads to improved focusing accuracy without requiring complicated formulations. This is actually one of the main advantages of using machine learning tools for the modeling of the array.



(**a**) (**b**) (**c**)

Figure 1. Example #1. Normalized near-field power density at $y = 0$ for the support vector regression (SVR) method (**a**), the neural network in [14] (**b**) and the optimization in [13] using a non-coupling model (**c**). The symbols + and ∘ represent the focal and synthesized maximum points respectively.

Table 1. Example #1. Axis length (major, minor) using SVR, neural networks (NN) and optimization (non-coupling).

Focal Point	$\{2\lambda, 0, 7\lambda\}$	$\{-\lambda, 0, 4\lambda\}$	$\{-3\lambda, 0, 4\lambda\}$
Axis length SVR	$3.95\lambda, 0.99\lambda$	$2.54\lambda, 1.09\lambda$	$3.10\lambda, 1.02\lambda$
Axis length NN	$4.85\lambda, 1.22\lambda$	$2.74\lambda, 0.84\lambda$	$2.97\lambda, 0.94\lambda$
Axis length Opt	$3.53\lambda, 1.09\lambda$	no	$3.21\lambda, 0.92\lambda$

In order to verify the ability of the proposed method to deal with different sizes of arrays, the number of elements of the array has been increased to 32 × 32 elements, and the interelement distance has been set to 0.7λ so that the resulting aperture is reasonably larger than the first structure, and hence its focusing capabilities are increased. All the other properties of the array and its elements remain the same, as well as the specifications. The higher number of degrees of freedom should lead, if properly handled by the method, to better focusing performance. The number of training patterns used to obtain the model is 150, and MoM has been used to calculate them. The hyperparameters of the regression remain $C = 1$ and $\varepsilon = 0.001$. The training process stops with a square mean error of 0.036 and 0.037 for the training and validation sets respectively.

Figure 2 shows the resulting NF distribution at the plane $y = 0$ where all the focal points are assigned. Both the focal points and the resulting maxima are also plotted. It may be observed that the field is much more concentrated around the assigned points, and that the accuracy in the location of the maximum power density points is much higher than using the smaller structure. The positions where the maximum power density is found are $\{2\lambda, 0, 7\lambda\}$, $\{-\lambda, 0, 4\lambda\}$ and $\{-3\lambda, 0, 4\lambda\}$, all of them exactly at the positions where the focal points are assigned, resulting in an average distance of 0. The axis lengths for the -3 dB spots are summarized in Table 2.

Table 2. Example #1. Axis length (major, minor) using SVR and a 32 × 32 element array.

Focal Point	$\{\mathbf{2\lambda, 0, 7\lambda}\}$	$\{\mathbf{-\lambda, 0, 4\lambda}\}$	$\{\mathbf{-3\lambda, 0, 4\lambda}\}$
Axis length SVR 32 × 32	$1.4\lambda, 0.55\lambda$	$1.25\lambda, 0.5\lambda$	$1.25\lambda, 0.5\lambda$

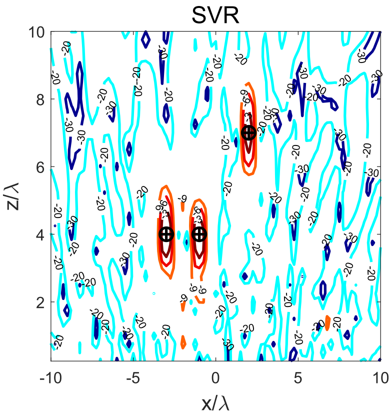
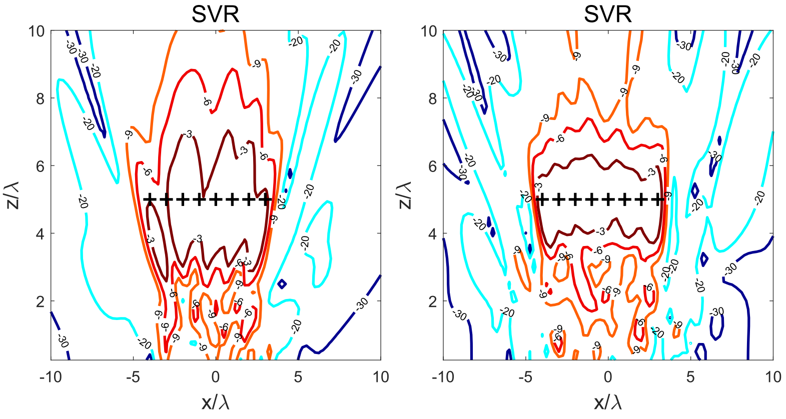


Figure 2. Example #1. Normalized near-field power density at $y = 0$ for the SVR method using a 32 × 32 element array. The symbols + and ○ represent the focal and synthesized maximum points respectively.

In example #2, a wider coverage area is requested in the NF region of the antenna. To specify it, a set of focal points representing *samples* of the coverage area has been selected. The assigned focal points are placed at $y = 0$, $z = 5\lambda$ and $x = [3\lambda, 2\lambda, \lambda, 0, -\lambda, -2\lambda, -3\lambda, -4\lambda]$, resulting in eight focal points. In all the cases the corresponding value C_m has been set to 1. The same regular array with 12 × 12 DRA elements separated 0.6λ has been considered, operating again at 3.6 GHz. A set of 140 training patterns was obtained using MoM as analysis tool. The hyperparameters of the regression are again $C = 1$ and $\varepsilon = 0.001$. The training process leads to a square mean error of 0.037 and 0.039 for the training and validation sets respectively. The obtained inverse model is used to calculate the set of weights corresponding to the field distribution shown in Figure 3a. It can be observed how a wider region is obtained over -3 dB, with all the assigned focal points but one into it. The focal point out of the -3 dB spot receives a power density -3.2 dB with respect to the maximum power in the NF region. In order to verify that larger arrays have increased focusing capabilities, the number of elements of the array has been increased to 16 × 16, and the interelement distance has been set to 0.7λ, again to create a larger aperture. The resulting field distribution is plotted in Figure 3b, where the coverage area can be observed clearly defined in a -3 dB spot containing all the assigned focal points.



(**a**) (**b**)

Figure 3. Example #2. Normalized near-field power density at $y = 0$ for the SVR method considering a coverage area defined by the assigned focal points represented by the symbol +. The number of elements of the array is 12 × 12 separated 0.6λ in (**a**) and 16 × 16 separated 0.7λ in (**b**).

5. Discussion

A machine learning approach to near-field focusing based on the powerful and elegant support vector machines framework is presented. Support vector regression is performed to develop an accurate NF inverse model of a given antenna array. It is calculated from a set of training patterns consisting on known pairs of weights applied to the array elements and the corresponding NF distribution. These patterns may be obtained through measurements or simulation. The necessary number of patterns is strongly reduced with respect to other ML alternatives such as neural networks, due the increased learning capabilities of SVM techniques.

Once the model has been obtained, it can be used for synthesis of focused distributions by following a simple strategy, and without relevant computational cost or time, as synthesis becomes a simple matrix-vector product. The resulting method is suitable for applications requiring fast synthesis to operate in scenarios where real-time calculations are required, for example where moving devices in a near environment are involved (e.g., 5G femtocells, Internet of things, etc.). Although the most popular approach to NFF, conjugate-phase, is very fast, simple and accurate, it cannot deal with multiple specifications; an optimization method, able to account for multifocus requirements, is not fast enough to be used in real-time applications due to its iterative nature; the NN approach is able to be accurate, fast and deal with multifocus, but requires thousands of training pairs, that might overflow many applications and make impossible the use of measured training data. The proposed SVR method overcomes all these difficulties due to the reduced number of required training patterns, its fast operation once trained, and its ability to deal with NF distributions specified in different ways. Additionally, it is able to account for the real properties of the array (realistic effects, coupling effects, individual radiation patterns, non-uniformities, etc.) provided that the method for generating the training patterns also accounts for all of them.

As a future work to be addressed in future developments, more sophisticated specifications such as a mask or template, phase-only distributions or additional constraints to be considered, may be included in the synthesis scheme; by doing so, the resulting method will become much more flexible and powerful. Meanwhile, SVR represents an interesting step for applications where different devices are involved and located in the near-field region of the antenna, even when some of them are moving.

Funding: This work was partially supported by the Spanish *Ministerio de Ciencia, Innovacion y Universidades* under the project TEC2017-86619-R (ARTEINE), and by the Gobierno del Principado de Asturias under project GRUPIN-IDI-2018-000191.

Conflicts of Interest: The author declares no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

NF	Near-Field
FF	Far-Field
NFF	Near-Field Focusing
CP	Conjugate-Phase
NFMF	Near-Field Multi-Focusing
WET	Wireless Power Transfer
WEIT	Wireless Power and Information Transfer
IoT	Internet of Things
SVM	Support Vector Machine
SVR	Support Vector Regression
SRM	Structural Risk Minimization
NN	Neural Network
ML	Machine Learning
EM	Electromagnetic
RFID	Radio Frequency Identification

References

1. Hansen, R. Focal region characteristics of focused array antennas. *IEEE Trans. Antennas Propag.* **1985**, *33*, 1328–1337. [CrossRef]
2. Buffi, A.; Serra, A.A.; Nepa, P.; Chou, H.T.; Manara, G. A Focused Planar Microstrip Array for 2.4GHz RFID Readers. *IEEE Trans. Antennas Propag.* **2010**, *58*, 1536–1544. [CrossRef]
3. Buffi, A.; Nepa, P.; Manara, G. Design Criteria for Near-Field-Focused Planar Arrays. *IEEE Antennas Propag. Mag.* **2012**, *54*, 40–50. [CrossRef]
4. Chou, H.T.; Hung, T.M.; Wang, N.N.; Chou, H.H.; Tung, C.; Nepa, P. Design of a Near-Field Focused Reflectarray Antenna for 2.4 GHz RFID Reader Applications. *IEEE Trans. Antennas Propag.* **2011**, *59*, 1013–1018. [CrossRef]
5. Chou, H.; Hung, K.; Chou, H. Design of Periodic Antenna Arrays With the Excitation Phases Synthesized for Optimum Near-Field Patterns via Steepest Descent Method. *IEEE Trans. Antennas Propag.* **2011**, *59*, 4342–4345. [CrossRef]
6. Gee, W.; Lee, S.W.; Bong, N.K.; Cain, C.A.; Mittra, R.; Magin, R.L. Focused Array Hyperthermia Applicator: Theory and Experiment. *IEEE Trans. Biomed. Eng.* **1984**, *BME-31*, 38–46. [CrossRef] [PubMed]
7. González Ayestarán, R.; León, G.; Pino, M.R.; Nepa, P. Wireless Power Transfer Through Simultaneous Near-Field Focusing and Far-Field Synthesis. *IEEE Trans. Antennas Propag.* **2019**, *67*, 5623–5633. [CrossRef]
8. Monti, G.; Costanzo, A.; Mastri, F.; Mongiardo, M.; Tarricone, L. Rigorous design of matched wireless power transfer links based on inductive coupling. *Radio Sci.* **2016**, *51*, 858–867. [CrossRef]
9. Costanzo, A.; Masotti, D. Smart Solutions in Smart Spaces: Getting the Most from Far-Field Wireless Power Transfer. *IEEE Microw. Mag.* **2016**, *17*, 30–45. [CrossRef]
10. Nepa, P.; Buffi, A. Near-Field-Focused Microwave Antennas: Near-field shaping and implementation. *IEEE Antennas Propag. Mag.* **2017**, *59*, 42–53. [CrossRef]
11. Nepa, P.; Buffi, A.; Michel, A.; Manara, G. Technologies for Near-Field Focused Microwave Antennas. *Int. J. Antennas Propag.* **2017**, *2017*, 7694281. [CrossRef]
12. Alvarez, J.; Ayestaran, R.G.; Las-Heras, F. Design of antenna arrays for near-field focusing requirements using optimisation. *Electron. Lett.* **2012**, *48*, 1323–1325. [CrossRef]
13. Alvarez, J.; Ayestaran, R.G.; Leon, G.; Herran, L.F.; Arboleya, A.; Lopez-Fernandez, J.A.; Las-Heras, F. Near field multifocusing on antenna arrays via non-convex optimisation. *IET Microw. Antennas Propag.* **2014**, *8*, 754–764. [CrossRef]

14. Ayestarán, R.G. Fast Near-Field Multifocusing of Antenna Arrays Including Element Coupling Using Neural Networks. *IEEE Antennas Wirel. Propag. Lett.* **2018**, *17*, 1233–1237. [CrossRef]
15. Haykin, S. *Neural Networks: A Comprehensive Foundation*; Mcmillan College Publishing Company Inc.: New York, NY, USA, 1994.
16. Ayestaran, R.G.; Heras, F.L. Neural networks and equivalent source reconstruction for real antenna array synthesis. *Electron. Lett.* **2003**, *39*, 956–958. [CrossRef]
17. Ayestaran, R.G.; Las-Heras, F.; Herran, L.F. Neural Modeling of Mutual Coupling for Antenna Array Synthesis. *IEEE Trans. Antennas Propag.* **2007**, *55*, 832–840. [CrossRef]
18. Vapnik, V.N. *The Nature of Statistical Learning Theory*; Springer: New York, NY, USA, 1995.
19. Alvarez, J.; Ayestaran, R.G.; Laviada, J.; Las-Heras, F. Support vector regression for near-field multifocused antenna arrays considering mutual coupling. *Int. J. Numer. Model. Electron. Netw. Devices Fields* **2016**, *29*, 146–156. [CrossRef]
20. Hayes, M.H. *Statistical Digital Signal Processing and Modeling*, 1st ed.; John Wiley & Sons, Inc.: New York, NY, USA, 1996.
21. Grant, M.; Boyd, S. *CVX: Matlab Software for Disciplined Convex Programming*, Version 2.1; 2014. Available online: http://cvxr.com/cvx (accessed on 14 November 2019).
22. Grant, M.; Boyd, S. Graph implementations for nonsmooth convex programs. In *Recent Advances in Learning and Control*; Blondel, V.; Boyd, S.; Kimura, H., Eds.; Lecture Notes in Control and Information Sciences; Springer Verlag London: London, UK, 2008; pp. 95–110.
23. McAllister, M.W.; Long, S.A. Resonant hemispherical dielectric antenna. *Electron. Lett.* **1984**, *20*, 657–659. [CrossRef]
24. Ayestaran, R.G.; Laviada, J.; Las-Heras, F. Realistic Antenna Array Synthesis in Complex Environments Using a MOM-SVR Approach. *J. Electromagn. Waves Appl.* **2009**, *23*, 97–108. [CrossRef]
25. Harrington, R.F. *Field Computation by Moment Methods*; Wiley-IEEE Press: Piscataway, NJ, USA, 1993.