IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 25 January 2023, accepted 28 February 2023, date of publication 6 March 2023, date of current version 13 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3253639

RESEARCH ARTICLE

Inertially Controlled Two-Dimensional Phased Arrays by Exploiting Artificial Neural Networks and Ultra-Low-Power AI-Based Microcontrollers

RICCARDO COLELLA, (Senior Member, IEEE), LUIGI SPEDICATO, LAURA LAQINTANA, AND LUCA CATARINUCCI, (Senior Member, IEEE)
Innovation Engineering Department, University of Salento, 73100 Lecce, Italy

Corresponding author: Riccardo Colella (Riccardo.colella@unisalento.it)

ABSTRACT The use of Artificial Intelligence (AI) in electronics and electromagnetics is opening many attractive research opportunities related to the smart control of phased arrays. This is particularly challenging especially in some high-mobility contexts, such as drones, 5G, automotive, where the response time is crucial. In this paper a novel method combining AI with mathematical models and firmware for orientation estimation is proposed. The goal is to control two-dimensional phased arrays using an Inertial Measurement Unit (IMU) by exploiting a feed-forward neural network. The neural network takes the IMU-based beam direction as input and returns the related phase shift matrix. To make the method computationally efficient, the network structure is carefully chosen. Specific and discretized cross-section regions of the array factor (AF) main lobe are considered to compute the phase shift matrices, used in turn to train the neural network. This approach achieves a balance between the number of phase-shifting processes and spatial resolution. Without loss of generality, the proposed method has been tested and verified on 4 × 4 and 6 × 6 arrays of 2.4 GHz antennas. The obtained results demonstrate that reconfigurability time, easiness of use, and scalability are suitable for a wide range of high-mobility applications.

INDEX TERMS Two-dimensional phased array, planar array, AI-based microcontroller, neural network, radio front-end.

I. INTRODUCTION

The design of smart electronic and electromagnetic systems for controlling phased arrays and radio antenna front-ends using artificial intelligence (AI) is gaining increasing attention from the scientific community. Indeed, more and more users can interact with each other through countless mobile devices that are increasingly connected. In this scenario, there is a strong and massive need for efficient algorithms and edge-computing methods to enhance the communication systems by controlling phased arrays radiation patterns, particularly in some high-mobility applications, such as those related to unmanned aerial vehicles (UAVs) —where a communication link with ground control stable over time and regardless of orientation is mandatory— rather than to 5G and automotive —where the reconfigurability time is a crucial parameter. Some works focus on specific material and chemical properties to obtain smart electronic devices, such as the use of graphene to have frequency selective surfaces [1]. Furthermore, other works relate to electronically controlled antennas with the aim of optimizing the radiation pattern without moving the radiators. For instance, compact-size, low-cost antennas have been recently proposed to realize electronically switchable radiation patterns [2] and reconfigurable polarizations [3], [4]. A two-dimensional scanning smart antenna is presented in [5] to realize a cognitive radio. To concentrate signals at desired directions, reconfigurable antenna arrays can be used, such as shape changeable devices [6], or rather a set of array elements can be selected, thereby considering subarrays and reducing complexity [7].

Recently, neural networks have been introduced to control phased array, reducing the computational complexity

The associate editor coordinating the review of this manuscript and approving it for publication was Fulvio Schettino.

required by traditional driving methods. For example, neural networks have been largely used to accurately estimate the direction of arrival (DOA) [8], [9] in place of classical techniques based on multiple signal classification (MUSIC) [10]. In the last decade, promising algorithms of machine learning (ML) have been suggested to make smart antennas [11], [12] and some of them combine traditional approaches, such as MUSIC, with ML [13]. As a branch of ML, deep learning (DL) has been also applied owing to its capability to extract features layer-by-layer and to combine them at different levels [14], [15], [16], [17], [18], [19]. For instance, some DL models are applied to smart antennas with the aim of reducing the energy consumption and increasing efficiency [20]. DL is robust with respect to data variations and flexible to be adapted to new problems when dealing with unknown configurations, because of its automatic feature detection. Nevertheless, it requires very large amount of data with respect to other techniques and it is computationally onerous to train due to the model complexity. As an alternative, supervised learning techniques of ML can be implemented when a-priori knowledge is available, for example on the number of radiation elements and on their position in antenna arrays. At this regards, innovative methods based on support vector machine (SVM) [21], [22], [23], [24], [25] and radial basis function (RBF) [26], [27], [28], [29] have been proposed. Novel approaches have been suggested both to perform detection and DOA estimation by means of a neural multiple source tracking (N-MUST) algorithm that consists in a family of radial basis function neural networks [30].

In parallel with the rise of innovative techniques of AI, the objective of focusing the radiated power towards a specific direction (or estimating the DOA) can be achieved by introducing inertial systems, such as Inertial Measurement Units (IMUs), to get the absolute antenna orientation. For example, in [31] an IMU is used to correct the orientation according to the signal level; in [32] an IMU is integrated into a GPS based-on system to achieve high position accuracy; in [33] the DOA is estimated using virtual antenna arrays where the array coordinates are estimated from raw IMU measurements. It is worth highlighting that the IMU based-on algorithms are generally used either to mechanically drive an antenna or to support localization systems, but none of them is employed to electronically control the antenna beam by the phase shift matrix. To solve this problem, the combined use of both supervised learning methods of ML and inertially-based models could be a winning approach.

In this paper, a novel method is purposely proposed to control two-dimensional phased arrays by exploiting both an IMU-based algorithm and a feed-forward neural network. The IMU acts as an inertial sensor, integrating an accelerometer, a gyroscope and a magnetometer into a unique board. The purpose of changing the radiation pattern is achieved through phase shifters, able to alter the phase electronically, thus quickly steering the beam to the desired direction. For this aim, the feed forward neural network receives the processed IMU data as input and returns the phase shift for each radiator as output. The network is trained by processing a set of samples, each of which contains a known pointing direction and the related phase shifts. To optimize the network structure, in terms of the number of nodes in the hidden layer and to speed up the training phase (assignment of weights) the number of training samples is suitably chosen. Moreover, the main lobe of the array factor has been cross-sectioned at selectable level curves (at −3dB in this paper without loss of generality). This allows to achieve the best compromise between number of phase-shifting processes over the time and spatial resolution. Since the network structure is a-priori determined and appropriate weights of the neural connections are found after training once for all, the proposed approach is much faster than exploring maps simply obtained by finding the maximum value of the array factor for different phase shift configurations.

II. FEED-FORWARD NEURAL NETWORK TO CONTROL RADIATION PATTERN OF TWO-DIMENSIONAL ARRAYS

In this section the proposed method to control the radiation pattern of a generic array based on processed inertial data is presented by exploiting a feed-forward neural network. At this regards the problem is to link 3D orientation data, generally available in form of rotation matrices or quaternions, with a corresponding phase shift matrix that is used to drive the array. The process of determining the phase shift matrix corresponding to a desired pointing direction is not trivial under both mathematical and computational point of view, especially when a rapid and stable radiation pattern reconfiguration is expected to follow the array movements.

Formally, the inverse problem of extracting the phase shifts related to the point at which the radiated power is maximum should be solved. Unfortunately, it is not possible to find a solution in closed form to this kind of problems. To overcome this limit, the proposed approach consists in solving the direct problem of finding the target direction starting from the phase shift matrix, and using the obtained outcomes to train a feed-forward neural network. After training, the inverse problem assumes the form of a solvable classification problem, whose complexity is reduced, in terms of required time and memory resources.

To rigorously explain the whole approach, some theoretical considerations are recalled in the following.

A N-by-M planar array is illustrated in Fig. 1 with respect to the Cartesian reference system having coordinates x, y and z and origin O.

The generic element at position *(n,m)* is excited by means of a signal with amplitude $a_{n,m}$ and phase $\varphi_{n,m}$. The related well-known formula of the Array Factor AF is given in (1), where ψ_1 and ψ_2 are expressed in (2) and (3) respectively, with respect to the distance d between two generic elements, to the wave number $\beta = 2\pi/\lambda$, the zenith angle θ varying in the interval $[0, \pi]$ and the azimuthal angle φ varying in the

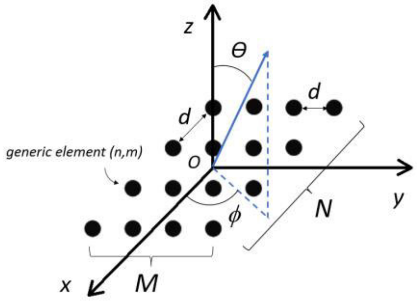


FIGURE 1. The two-dimensional array and the considered reference system.

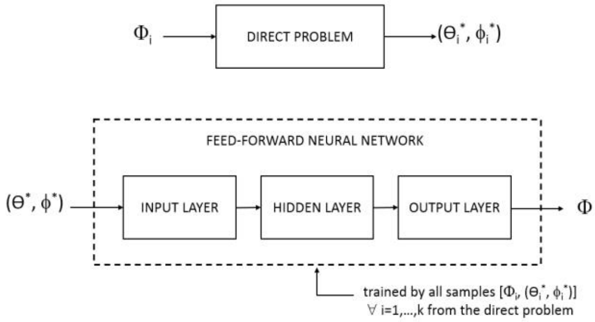

Φi
DIRECT PROBLEM
(Θi*, φi*)
FEED-FORWARD NEURAL NETWORK
(Θ*, φ*)
INPUT LAYER
HIDDEN LAYER
OUTPUT LAYER
Φ
trained by all samples [Φi, (Θi*, φi*)]
∀ i=1,...,k from the direct problem

FIGURE 2. The direct problem and the feed-forward neural network.

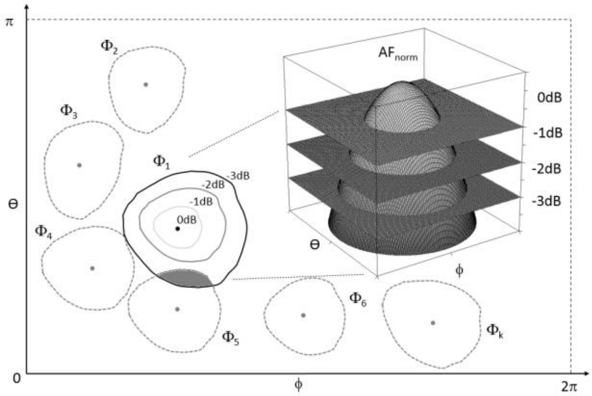


FIGURE 3. The normalized array factor and cross sections.

interval [0,2π].

$$AF = \sum_{n=1}^{N}\sum_{m=1}^{M} a_{n,m} e^{j\psi_1(n-1)} e^{j\psi_2(m-1)} e^{j\varphi_{n,m}} \tag{1}$$

$$\psi_1 = \beta d \sin\theta \cos\phi \tag{2}$$

$$\psi_2 = \beta d \sin\theta \sin\phi \tag{3}$$

The signal phase $\varphi_{n,m}$ is the generic element of the phase shift matrix Φ as indicated in (4).

$$\Phi = \begin{pmatrix} \varphi_{11} & & \varphi_{1M} \\ & \ddots & \\ \varphi_{N1} & & \varphi_{NM} \end{pmatrix} \tag{4}$$

A. DIRECT PROBLEM AND TRAINING OF THE NEURAL NETWORK

The direct problem, mainly consisting in the evaluation of the maximum value of $|AF|$ for a given Φ, must be preliminary solved to train the neural network. If the phase shift matrix Φ is known, the module of the array factor is a function of two variables $|AF(\theta, \phi)|$, that are the angles θ and ϕ. In order to evaluate the maximum value of the two-variable function, the first step is to find the critical points (θ^*, ϕ^*) where the gradient is the null vector $\nabla(|AF(\theta^*, \phi^*)|) = \mathbf{0}$. Then, the second partial derivative test can be carried out and the determinant of the Hessian matrix can be calculated to evaluate the surface concavity. Alternatively, the gradient ascent method (or equivalently the gradient descent method for the dual minimum problem) can be implemented to find the local maximum by starting from an initial point (θ_0, ϕ_0) and evaluating the gradient at each step to converge toward (θ^*, ϕ^*).

Based on these theoretical aspects, the neural network can be trained by assigning a number k of samples, in terms of critical points (θ_1^*, ϕ_1^*), …,(θ_k^*, ϕ_k^*) and corresponding shift phase matrices Φ_1,…, Φ_k, as illustrated in Fig. 2, where the network structure is also shown in terms of input, hidden and output layers.

On the one hand, the higher is k the better is the performance reached in terms of minimum classification error. On the other hand, higher values of k implicate a great number of neuronal units in the hidden layer of the neural network with higher computational load, longer response time, and possible lack of convergence. In this circumstance, the system could be stressed out by continuous updates of the phase shift matrix thus getting orientation-sensitive until it becomes unstable, also for small perturbations or sensing noise. In other words, the radiation pattern would be constantly reconfigured owing to extremely closed points with consequent important delays, local oscillations, and instability.

To avoid these drawbacks, the training process explained in Fig. 2 is closely enhanced by taking into consideration the power beamwidth extracted from level curves of the array factor. Specifically, the array factor normalized with respect to the maximum $AF_{norm} = |AF(\theta, \phi)|/|AF(\theta^*, \phi^*)|$ is considered to define common cutting planes useful to facilitate the process of arrangement of samples for training the network.

As reported in the inset of Fig. 3, the graph of AF_{norm} is a surface that can be sectioned by planes parallel to the coordinate plane $\theta\phi$. The level curves as cross sections of the main lobe taken at different quotes and related to the specified dB value are also illustrated in Fig. 3. Once set the cutting level, a sufficient number of regions enclosed by the level curves should be considered to fill the whole rectangular region of the plane $\theta\phi$ in Fig. 3 for all possible values of θ and ϕ.

In this way, the specific phase shift matrix related to the maximum point (θ^*, ϕ^*) can be associated with the region that contains such a maximum point. Consequently, the number of samples k can be drastically reduced, and it can be suitably tuned, by acting on the cutting level, in order to achieve the desired compromise among the number of available phase shift matrices, the spatial resolution, and the system response time.

Based on the previously explained approach, a more efficient way to design and train the feed-forward neural network can be implemented. At this regard, the first step is to generate a set of random points (θ, ϕ) uniformly distributed in the rectangular region of the plane $\theta\phi$ of Fig. 3. Each random point is labeled according to the region subtended by the level curve at the chosen quote to which it belongs, as shown in Fig. 4 for the case of -3dB. In this particular case, which is considered without loss of generality, all the boundary points lying on the curve at -3dB concern with the half-power beamwidth (HPBW).

Once all the random points are labeled and assigned to each membership region, associated to the related phase shift matrix, and represented by a specific class label, the dataset to train the neural network is complete. For instance, in Fig. 4 a focus on two class labels of Fig. 3 is given to better describe the proposed approach. As observed, the black points are assigned to a class label (i.e., class label 1) and the cyan points are assigned to another one (i.e., class label 5). Of course, the procedure is iterated to consider all the class labels as depicted by the green points in Fig. 4. It is worth highlighting that some intersections among level curves can occur and some random points can lie on the intersection region as shown by the gray region obtained from the level curves at -3dB in Fig. 4. In order to classify these points, the normalized array factors AF_{norm} associated to each superimposed region are evaluated for them. Each of such points in the intersection is then assigned to the label class for which the greater calculated value of AF_{norm} is obtained. After generating the set of random points and labeling all of them, the collected samples can be used to train the neural network. For the sake of clearness, all the described steps of the method are summarized in Fig. 5.

The training process consists in determining the weights of neural connections, that are the network parameters useful to obtain the required link between the input point (θ, ϕ) and the output class label, thus regulating the network.

In detail, the weights are adjusted by the Levenberg-Marquardt algorithm [34], that is an iterative procedure to minimize the error between obtained and desired output values. The number of input nodes of the input layer is predetermined by the number of features in the input data, that are the two angles (θ, ϕ) defining the target direction. As well, the number of output nodes of the output layer is predetermined by the number of classes in the outcome, that correspond to the specific phase shift matrices used to drive the two-dimensional antenna array. As regards the neurons in the hidden layer, there is no analytical approach to determine their number and it has to be chosen according to the amount of training data. This choice is critical because the hidden layer plays a crucial role to take away the risk of a system that is computationally expensive and slow to train and test. If this parameter is carefully fixed, the neural network appears to be the best solution for the inverse problem of determining the phase shift matrix needed to control the radiation pattern by fixing the target direction.

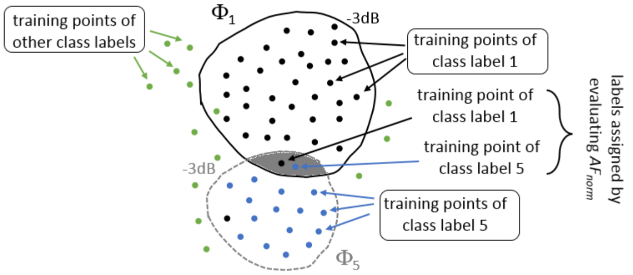

training points of other class labels
Φ_1
-3dB
training points of class label 1
training point of class label 1
training point of class label 5
labels assigned by evaluating AF_{norm}
-3dB
training points of class label 5
Φ_5

FIGURE 4. The labeling process from the level curves of normalized array factors.

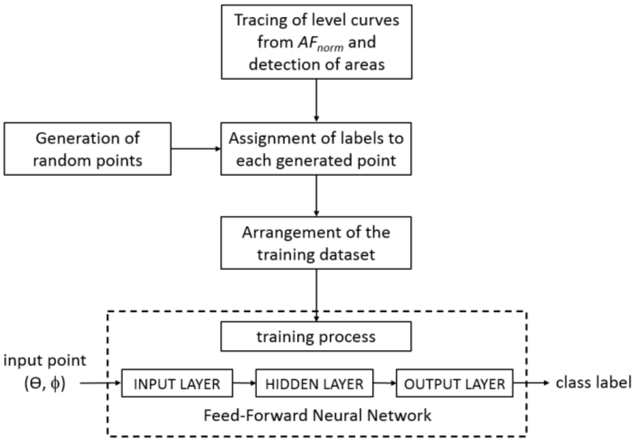

Tracing of level curves from AF_{norm} and detection of areas
Generation of random points
Assignment of labels to each generated point
Arrangement of the training dataset
training process
input point (Θ, ϕ)
INPUT LAYER
HIDDEN LAYER
OUTPUT LAYER
class label
Feed-Forward Neural Network

FIGURE 5. The steps to arrange the training dataset useful to control the radiation pattern of the two-dimensional array of antennas.

B. PROCESSING OF IMU DATA TO CONTROL THE RADIATION PATTERN

The aim of controlling the radiation pattern for the two-dimensional array of antennas is achieved by exciting the array elements with appropriate shift phases. In more detail, the array can be electrically driven to concentrate the radiated power at a specific target direction according to the IMU data. The inertial sensor data are processed by implementing a complementary filter and an analytic geometry model according to the scheme in Fig. 6.

The considered IMU has nine degrees of freedom and it includes a triaxial accelerometer, gyroscope and magnetometer rispectively. It is used to measure the proper acceleration (rate of change of velocity in g), the angular velocity (in *rad/s*) and the relative change of magnetic field (in T) of the

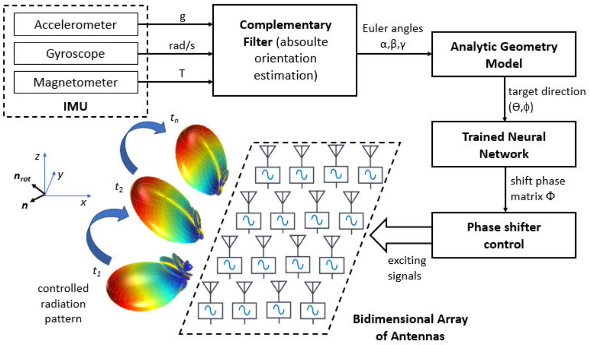


FIGURE 6. The whole procedure to control the radiation pattern by processing the IMU data through complementary filter and analytical geometry model.

two-dimensional array or rather of the mobile device where it is placed on. These quantities are combined by a properly-designed filter to keep track of the spatial orientation. When using only an accelerometer, all forces acting are detected, small forces can disturb the process of orientation estimation and driving forces are also perceived in actuated systems.

Moreover, the exclusive use of a gyroscope brings to measurements that have the tendency to drift because of the integration error over time. The best solution to these weaknesses consists in the combination of both accelerometer and gyroscope data by means of a filter. Most of research works focus on the use of the Kalman filter [35], that is a recoursive filter keeping track of the estimated state of a system and of the variance or uncertainty of the estimate, as well. However, the Kalman filter requires significant hardware resources to be used and this is a strong limitation for the proposed application in which the system has to be responsive and the minimal computational effort is needed. In order to obtain the orientation, in terms of Euler angles, a complementary filter [36], [37], [38] has been implemented since it is computationally more efficient (also for real time applications) with respect to the Kalman filter [39], and it is not susceptible to small disturbing forces, to external forces, and it does not drift.

Formally, the time derivative of the Euler attitude $\dot{\omega}$ can be expressed in terms of the gyroscope measures (p, q, r) by (5), where the angles of pitch, roll and yaw are denoted as α, β, and γ respectively, and the subscript g refers to the fact that the angular variations are estimated from the gyroscope measures.

$$\dot{\omega} = \begin{pmatrix} \dot{\alpha}_g \\ \dot{\beta}_g \\ \dot{\gamma}_g \end{pmatrix} = \begin{pmatrix} 1 & \sin\alpha\tan\beta & \cos\alpha\tan\beta \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha/\cos\beta & \cos\alpha/\cos\beta \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix} \tag{5}$$

Moreover, the roll α and the pitch β can be obtained from the accelerometer measures (a_x, a_y, a_z) by (6) and (7), where the subscript a indicates that the accelerometer is used.

$$\alpha_a = \tan^{-1}(a_y/a_z) \tag{6}$$

$$\beta_a = \tan^{-1}(a_x/\sqrt{a_y^2 + a_z^2}) \tag{7}$$

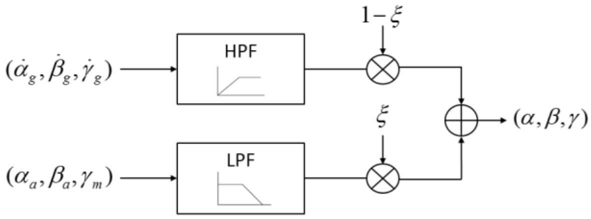

$(\dot{\alpha}_g, \dot{\beta}_g, \dot{\gamma}_g)$
$(\alpha_a, \beta_a, \gamma_m)$
HPF
LPF
$1-\xi$
ξ
(α, β, γ)

FIGURE 7. Block diagram of the complementary filter.

Finally, the yaw γ can be evaluated from the magnetometer measures (m_x, m_y, m_z) by (8), in which the subscript m refers to the the use of the magnetometer.

$$\gamma_m = \tan^{-1}\left(\frac{m_z\sin\alpha - m_y\cos\alpha}{m_x\cos\beta + m_y\sin\alpha\sin\beta + m_z\sin\beta\cos\alpha}\right) \tag{8}$$

The basic structure of the complementary filter is given by the block diagram in Fig. 7 that accepts the time derivative of the Euler attitude and the preliminar estimations of roll, pith and yaw as inputs to return the Euler angles α, β and γ. As shown in Fig. 7, the inputs are filtered by high pass filter (HPF) and low pass filter (LPF) correspondingly, and the outputs are weighted through a weighting factor $\xi \epsilon$ [0,1].

These filters have specific transfer function $\mathrm{G_{HPF}}(s)$ and $\mathrm{G_{LPF}}(s)$ in Laplace domain, such that $\mathrm{G_{HPF}}(s) \cdot \mathrm{G_{LPF}}(s) = 1$ [39]; for instance, $\mathrm{G_{HPF}}(s) = \tau s/(1+\tau s)$ and $\mathrm{G_{LPF}}(s) = 1/(1+\tau s)$ can be chosen for a fixed cut-off frequency $1/\tau$.

The weighted sum of the filter outputs gives the expected triplet of angles α, β and γ. The Euler angles describe the absolute orientation with respect to a fixed coordinate system of the object where the IMU is placed on. In order to get the target direction (θ, ϕ), the Euler angles have to be given as inputs to an analytic geometry model. With respect to the fixed reference system depicted in Fig. 6, the roll α represents the rotation around the x axis, the pitch β represents the rotation around the y axis and, finally, the yaw γ represents the rotation around the z axis. Any orientation can be achieved by composing three elemental rotations around the axes z, y and x (in that order), starting from a known standard orientation. These rotations bring to a composed rotation that can be described by the rotation matrix R in (9), where for a generic angle u, $s_u = \sin(u)$ and $c_u = \cos(u)$ are used for compacteness.

$$R = \begin{pmatrix} \mathrm{c}_\beta \mathrm{c}_\gamma & -\mathrm{c}_\beta \mathrm{s}_\gamma & \mathrm{s}_\beta \\ \mathrm{c}_\alpha \mathrm{s}_\gamma + \mathrm{c}_\gamma \mathrm{s}_\alpha \mathrm{s}_\beta & \mathrm{c}_\alpha \mathrm{c}_\gamma - \mathrm{s}_\alpha \mathrm{s}_\beta \mathrm{s}_\gamma & -\mathrm{c}_\beta \mathrm{s}_\alpha \\ \mathrm{s}_\alpha \mathrm{s}_\gamma - \mathrm{c}_\alpha \mathrm{c}_\gamma \mathrm{s}_\beta & \mathrm{c}_\gamma \mathrm{s}_\alpha + \mathrm{c}_\alpha \mathrm{s}_\beta \mathrm{s}_\gamma & \mathrm{c}_\alpha \mathrm{c}_\beta \end{pmatrix} \tag{9}$$

In this way, a generic direction defined by the vector $\boldsymbol{n}$ in the fixed reference system can be mapped to a new rotated direction $\boldsymbol{n_{rot}} = R \cdot \boldsymbol{n}$ through the procedure illustrated in Fig. 6. Lastly, the zenith angle θ and the azimuthal angle ϕ can be calculated according to the longitude and colatitude formulas in (10) and (11), where $n_{rot,x}$, $n_{rot,y}$, $n_{rot,z}$ are the components of $\boldsymbol{n}_{\mathbf{rot}}$ along the axes x, y and z respectively.

$$\theta = a\cos(n_{rot,z}) \tag{10}$$

$$\phi = \tan^{-1}(n_{rot,y}/n_{rot,x}) \tag{11}$$

After dermining the target direction (θ, ϕ), the previously trained neural network can be used to extract the concerning phase shift matrix Φ. A topic of particular interest consists in the possibility of using the proposed neural network as a fitting function rather than simple classifier. Indeed, once the implemented neural network has learnt from the training data, it forms a generalization of the input-output relationship and can be used to generate outputs for inputs that it was not trained on. In other words, the input point (θ, ϕ) in the block scheme of Fig. 5, and also reported in Fig. 6, could not belong to the training dataset arranged to train the network. Anyway, the neural network is able to assign to the not-previously observed point the concerning class label, and ultimately the related phase shift matrix, with the minimum fitting error.

III. RESULTS

In this Section the method to inertially control the radiation pattern of two-dimentional antenna arrays by using a feed-forward neural network has been implemented to test and validate the performance in two specific reference use cases.

As a first use case, a 2.4 GHz 4 × 4 antenna array has been designed by using Matlab Antenna Toolbox. The parameter d of Fig. 1, namely the distance between two contiguous radiating elementes, has been set to *λ/2,* i.e. $d = 62.5$ *mm.* Each radiating element is fed by a signal with unitary amplitude and variable phase, according to the corresponding element of the computed phase shift matrix. The feeding signal can be written in matrix form as reported in (12):

$$\bar{S} = \bar{A} * e^{i\bar{\Phi}_c}, \tag{12}$$

where $\bar{S}$ is the 4 × 4 feeding matrix, $\bar{A}$ is the 4 × 4 amplitude matrix (having the value "1" in all positions in this case), and $\bar{\Phi}_c$ is the computed 4 × 4 phase shift matrix.

To control the antenna array, the feed-forward neural network has been implemented in Matlab, and the procedure of Fig. 5 has been properly executed. Level curves at -3 dB have been purposely selected in the specific test case. Therefore, the AF_{norm} has been computed and 81 regions (i.e. classes) at -3 dB have been determined for the specific case, as shown in Fig. 8, where the regions are represented on a $\theta\phi$ plane. According to the procedure, a specific dataset of random points (θ_i, ϕ_i) has been then generated to perform the training of the neural network. Based on the proposed theory, each point has been classified, associated to a region subtended by a specific level curve, and properly labeled to address the specific phase shift matrix corresponding to the desired AF_{norm} orientation.

With this general setup, different tests have been implemented to evaluate the performance of the designed system in terms of percentage of detection of the correct phase shift matrix, given a certain input point. Tests have been performed by varying:

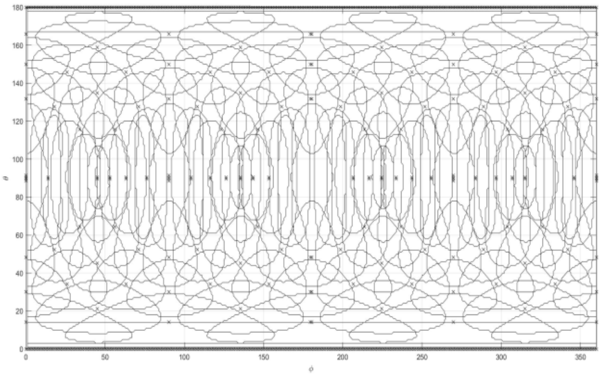


FIGURE 8. Regions subtended by -3dB level curves computed for the radiation pattern of a 4 × 4 antenna array on the θ-ϕ plane.

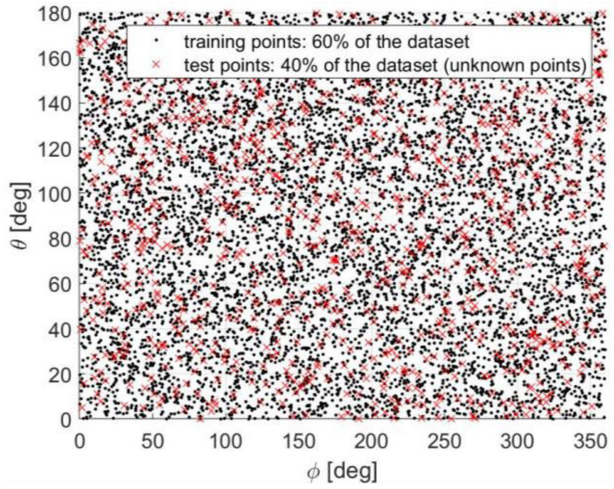


FIGURE 9. Example of a 10,000-point dataset with 60% of training points, and 40% of (unknown) test points referred to a neural network with 60 hidden neurons controlling a 4 × 4 phased array.

1. number of neurons of the hidden layer,
2. maximum number of training points within the interval [1,000-10,000],
3. number and typology of points used to perform the test of the neural network. More specifically, the neural network has been tested by exploiting a certain percentage of "known" or "unknown" points. Known points are points of the dataset that the neural network already observed during the training phase. Conversely, unknown points are points of the dataset never observed by the neural network.

The example of Fig. 9 shows a graphical representation on the $\theta\phi$ plane of a 10,000-point dataset with 60% of training points, and 40% of unknown test points referred to the neural network with 60 neurons in the hidden layer.

As for the obtained neural network performance evaluation, Fig. 10 shows three different graphs related to 60 a), 80 b), and 100 c) hidden neurons, respectively. Graphs show the percentage of detection of the correct phase shift matrix when varying cardinality of the dataset in the given interval. For each dataset, continuous lines are referred to unknown test points, and dashed lines are referred to known points.

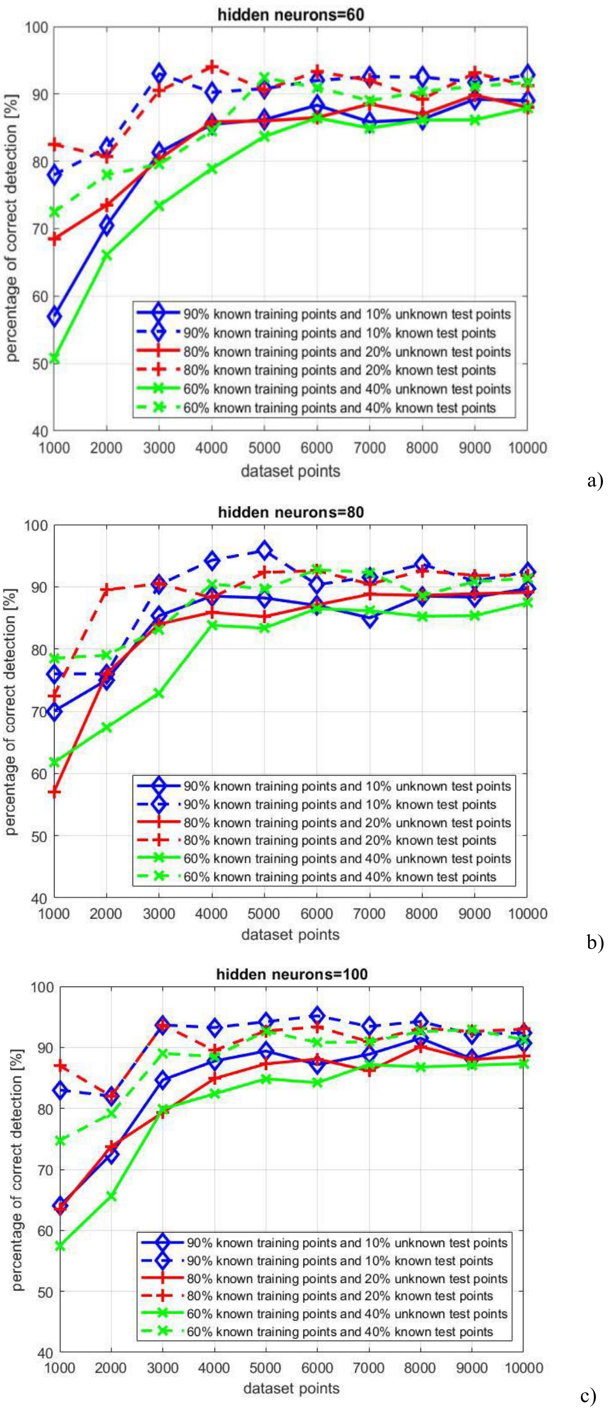


FIGURE 10. Percentage of correct detection referred to the neural network a controlling a 4 × 4 phased array: 60 hidden neurons a); 80 hidden neurons b); 100 hidden neurons c).

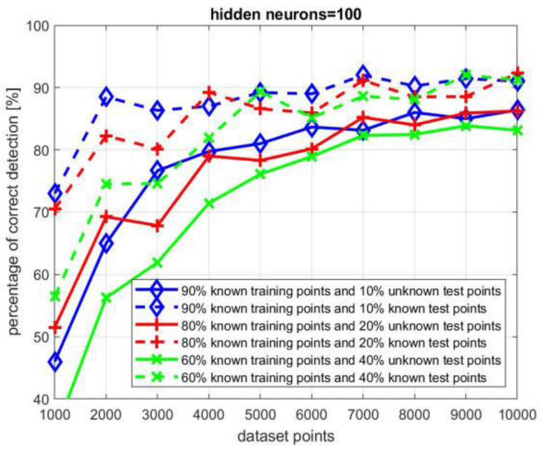


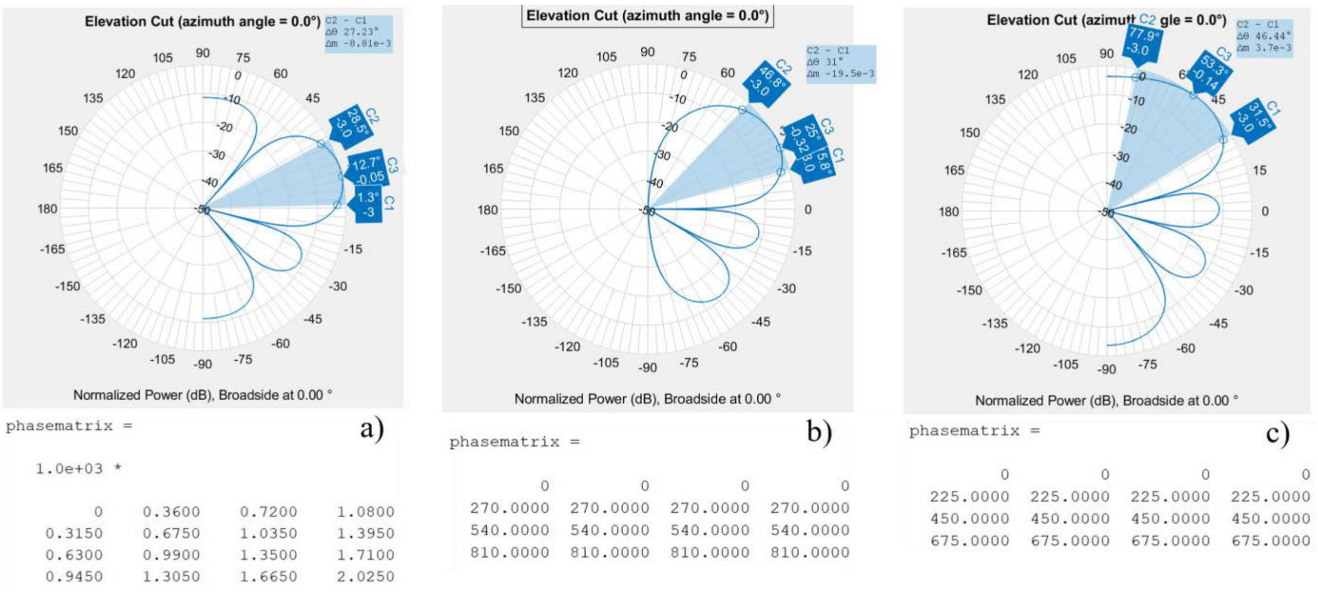
FIGURE 11. Percentage of correct detection referred to the neural network with 100 hidden neurons a controlling a 6 × 6 phased array.

Let us consider the graph of Fig. 10a, which is referred to a number of hidden neurons (80) comparable with the number of classes (81), according to various criteria adopted to avoid both overfitting or underfitting problems [40]. First, as expected, curves referred to a certain percentage of known test points concern with higher accuracy than the companion versions referred to the same percentage of unknown test points. Nevertheless, it is important to observe that, while increasing the dataset points, such a difference becomes particularly reduced, thus demonstrating that regardless the status of the points (known or unknown) an appreciable percentage of correct detection, between 87% and 93%, is achieved. Another important deduction is that while varying the dataset points all the graphs show an initial growing trend which converges toward an asymptotic value. In the specific case, the smallest cardinality of the dataset guaranteeing a stable percentage is roughly 6,000 points. This value sets the smaller number of dataset points to train the neural network, since a higher number of points would only impact on the computational time. Finally, the comparison among curves obtained when increasing the percentage of training points with respect to the test points, is rather relevant. Indeed, what it is necessary is that the neural network does not depend on the number of test points. Hence, even if apparently the curve obtained with 90% of training points becomes stable at around 4,000 points instead of 6,000 (when 60% of training points is used), the significant gap among all the curves suggests that the highest value should be considered.

For the sake of completeness, the same test has been carried out also with a number of hidden neurons equal to 60 (smaller that the number of classes) and 100 (higher than the number of classes), and results reported in Fig. 10b and Fig 10c, respectively.

The sole slightly appreciable difference is referred to the rapidity of convergence of the curve with 60% of training points. No evident effects are observed neither on the percentage of correct detection in the steady state, nor in the number of points individuating such a steady state.

Consequently, the choose of the number of hidden neurons can be made by considering other parameters such as, for instance, the computational time, which has been properly evaluated in the three addressed cases. More specifically, Matlab tests have been performed on an off-the-shelf workstation powered by an Intel Core I7 10th generation processor, and mounting 16 GB of RAM. The average time to



a)

```
phasematrix =

   1.0e+03 *

        0    0.3600    0.7200    1.0800
   0.3150    0.6750    1.0350    1.3950
   0.6300    0.9900    1.3500    1.7100
   0.9450    1.3050    1.6650    2.0250
```

b)

```
phasematrix =

          0           0           0           0
   270.0000    270.0000    270.0000    270.0000
   540.0000    540.0000    540.0000    540.0000
   810.0000    810.0000    810.0000    810.0000
```

c)

```
phasematrix =

          0           0           0           0
   225.0000    225.0000    225.0000    225.0000
   450.0000    450.0000    450.0000    450.0000
   675.0000    675.0000    675.0000    675.0000
```

FIGURE 12. Computed normalized radiation pattern of a 4 × 4 phased array for three input angles and corresponding computed phase matrices.

compute a phase shift matrix (i.e. the time necessary to switch from one radiation pattern to another) resulted in $t_{60,4\times4} = 5.6$ ms, $t_{80,4\times4} = 5.9$ ms, $t_{100,4\times4} = 7.8$ ms, for 60, 80, and 100 hidden neurons, respectively. This suggests that the use of 60 hidden neurons is the preferrable solution to control the 4 × 4 phased array.

The same kind of result has been obtained for a more complex array composed of 6 × 6 elements. In this case the number of classes is 120. The test has been carried out for 100, 120, and 150 hidden neurons, respectively. Since the results are similar, for the sake of conciseness only the case of 100 is reported in Fig. 11. It emerges that the steady state is reached with a number of dataset points slightly higher than the 4 × 4 case (around 7,000). The computational time resulted in $t_{100,6\times6} = 9.1$ ms.

This demonstrates how, thanks to a neural network, there is a relevant reduction of the computational time required to get the result. Indeed, in the proposed approach, neural networks are used because of their suitability to build models without resorting to complicated mathematical formulas. This allows not only to solve complex problems by simulating the basic functions of biological neurons, but also (and above all) to return the problem solution in an reasinable time. Several factors impact on the net complexity, such as the network structure and the number of hidden neurons [40]. The used feed-forward scheme appears to be the best way to keep the computation time low, avoiding cycles or loops when the information moves from the input to the output layer.

Once reasonable numbers of hidden neurons and dataset points have been individuated, the neural network for the 4 × 4 case (without loss of generality) has been used to steer the beam of the radiation pattern toward pre-defined angles. In particular, the goal is to demonstrate that the neural network correctly selects the proper phase shift matrix when an arbitrary couple (θ, ϕ) is given in input. For the sake of conciseness and graphical representability, only three cases — and all referred to $\phi = 0$ — have been reported in Fig.12, along with the classified matrices. Fig. 12a, for instance, is refereed to $\theta = 12.7°$. It can be observed that, with reference to the selected phase shift matrix, the chosen angle falls in the highlighted -3dB beamwidth, as desired.

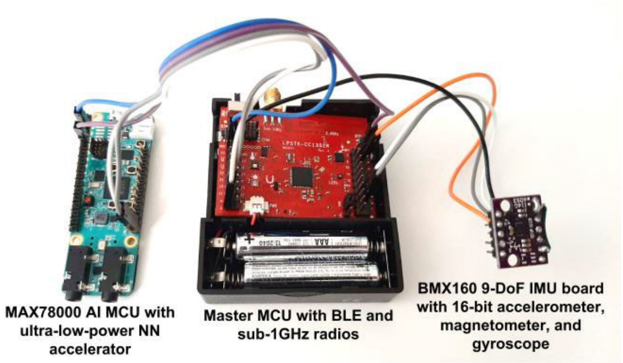

MAX78000 AI MCU with ultra-low-power NN accelerator
Master MCU with BLE and sub-1GHz radios
BMX160 9-DoF IMU board with 16-bit accelerometer, magnetometer, and gyroscope

FIGURE 13. Setup of the MAX78000 AI-based microcontroller interfaced with the CC1352R BLE microcontroller and the BMX160 IMU board to drive an external 4 × 4 phased array.

In Figg. 12b and 12c the results referred to two other randomly chosen angles, i.e. $\theta = 25.0°$ and $\theta = 53.3°$, respectively, are reported. It can be clearly observed that the trained neural network performs beam steering according to the desired direction, while guaranteeing the already discussed computational performance.

In the final test, the capability of the designed system of correctly steering the radiation pattern according to real time data coming from an IMU board has been verified using the

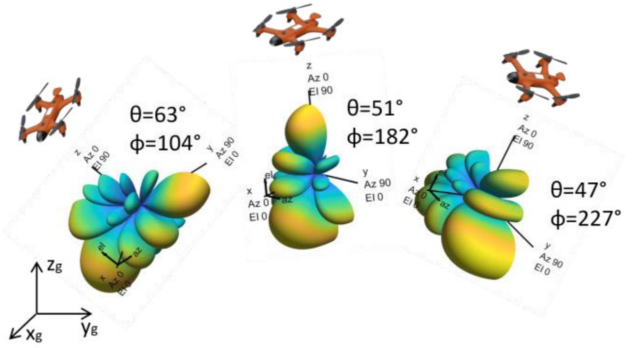


FIGURE 14. Calculated radiation patterns according to drone orientation.

setup of Fig. 13. The systems is based on the MAX78000 a new kind of Artificial Intelligence microcontrollers embedding and ultra-low-power neural network accelerator [41]. The MAX78000 has been programmed to run an artificial neural network trained to drive a 4 × 4 two-dimensional phased array according to the proposed methodology. In order to perform the transmission of the computed angular data (toward an host-pc for data visualization) the MAX78000 has been interfaced, via i2c interface with a CC1352R-LPSTK Bluetooth low Energy (BLE) wireless microcontroller board acting as a primary device [42]. This last has been then connected via i2c to the 16-bit BMX160 IMU unit [43] integrating accelerometer, magnetometer, and gyroscope. The BLE master microcontroller performs sensing and data fusion and generates triplets of Euler angles α, β and γ that send to the MAX78000 for neural network processing. The MAX78000 sends back the result that is finally transmitted toward the host-pc, for array feed and data visualization under MATLAB environment. Specifically, in the studied test case, the IMU board is considered integral with the array (which is implemented in a virtual environment in this test, without loss of generality) and with an object capable of moving, such as a flying drone.

In the proposed test-case the ground has been chosen as the target direction of the antenna beam. Then, the system has been configured to maximize the radiating power toward the ground and to recompute the pointing direction accordingly, regardless the actual drone orientation during its activity.

In Fig 14, the results obtained for some of the many investigated angles of the drone —and hence IMU and array— are reported. All the rotation angles are computed with respect to the reference system (x_g,y_g,z_g), where the plane (x_g,y_g) corresponds to the ground. The reference system for the radiation pattern is, as usual, those of Fig. 1, where the broadside direction of the 4 × 4 array is aligned with the front direction of the drone.

The reported studied cases correspond to triplet of Euler angles (representative of drone orientations) which, once processed, generate the couples ($\theta = 63°$,$\phi = 104°$), ($\theta = 51°$,$\phi = 182°$), and (θ=$47°$,$\phi = 227°$), respectively. As can be observed, regardless the angles, the algorithm selects the correct phase shift matrix which guarantees the desired orientation of the beam of the radiation pattern. More specifically, this use case demonstrates the capability of the whole designed system of opportunely compensating —within the imposed −3dB tolerance and in real time (switching time of 5.6 ms)— the arbitrary angles of the drone.''

IV. CONCLUSION

The use of artificial intelligence (AI) algorithms in the electromagnetic field promises interesting advances in various application contexts, such as the design of intelligent electromagnetic environments rather than the development of optimizers as part of electromagnetic simulators. The design of an AI-based system for the rapid and smart orientation of the beam of a phased array according to a given event, such as for instance the real-time data coming from an IMU, is another promising area of interest which is proposed in this work. The main issue from the analytical point of view consists of inverting the Array Factor (AF) function to evaluate the phase shift matrix associated to a desired orientation of the beam. Indeed, a closed-form solution for such a problem is not possible. To overcome this limit, a feed-forward neural network acting on specific and discretized cross-section regions of the AF main lobe has been implemented and opportunely trained to individuate the proper phase shift matrix guaranteeing the desired radiation pattern of two-dimensional phased arrays.

To drive the neural network, inertial sensor data are fused and processed through both a complementary filter and an analytic geometrical model so to provide proper angles according to a certain user-defined control logic.

The whole system has been tested on 4 × 4 and 6 × 6 arrays, in terms of computational performance, pointing resolution, and global functioning in a realistic use case.

The obtained real-time responses of a few milliseconds, together with the capability of correctly steering the beam, are the main features of the proposed method. Several practical applications could benefit from the possibility of real-time controlling a radiation pattern by exploiting the quick response of a feed-forward neural network.

REFERENCES

[1] R. M. S. D. Oliveira, R. R. Paiva, N. R. N. M. Rodrigues, and M. R. E. Filho, ''Smart graphene-based frequency selective surface designed to act as single or dual band device with reconfigurable operation bands,'' *J. Microw., Optoelectron. Electromagn. Appl.*, vol. 18, no. 4, pp. 452–481, Dec. 2019.

[2] C. Cui, W. T. Li, X. T. Ye, P. Rocca, Y. Q. Hei, and X. W. Shi, ''An effective artificial neural network-based method for linear array beampattern synthesis,'' *IEEE Trans. Antennas Propag.*, vol. 69, no. 10, pp. 6431–6443, Oct. 2021, doi: 10.1109/TAP.2021.3069467.

[3] C. Gu, S. Gao, H. Liu, Q. Luo, T.-H. Loh, M. Sobhy, and J. Li, ''Compact smart antenna with electronic beam-switching and reconfigurable polarizations,'' *IEEE Trans. Antennas Propag.*, vol. 63, no. 12, pp. 5325–5333, Dec. 2015, doi: 10.1109/TAP.2015.2490239.

[4] K. S. Kim, J. S. Yoo, J. W. Kim, S. Kim, J. W. Yu, and H. L. Lee, ''All-around beam switched antenna with dual polarization for drone communications,'' *IEEE Trans. Antennas Propag.*, vol. 68, no. 6, pp. 4930–4934, Jun. 2020, doi: 10.1109/TAP.2019.2952006.

[5] A. B. Guntupalli and K. Wu, ''60 GHz circularly-polarized smart antenna system for high throughput two-dimensional scan cognitive radio,'' in *IEEE MTT-S Int. Microw. Symp. Dig.*, Jun. 2013, pp. 1–3, doi: 10.1109/MWSYM.2013.6697662.

[6] S. Kalra, R. Datta, B. S. Munjal, and B. Bhattacharya, ''Smart reconfigurable parabolic space antenna for variable electromagnetic patterns,'' *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 311, no. 1, pp. 1–8, 2018.

[7] X. Wang, E. Aboutanios, M. Trinkle, and M. G. Amin, ''Reconfigurable adaptive array beamforming by antenna selection,'' *IEEE Trans. Signal Process.*, vol. 62, no. 9, pp. 2385–2396, May 2014, doi: 10.1109/TSP.2014.2312332.

[8] A. Rawat, R. N. Yadav, and S. C. Shrivastava, ''Neural network applications in smart antenna arrays: A review,'' *AEU-Int. J. Electron. Commun.*, vol. 66, no. 11, pp. 903–912, 2012.

[9] K. Terabayashi, R. Natsuaki, and A. Hirose, ''Ultrawideband direction-of-arrival estimation using complex-valued spatiotemporal neural networks,'' *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 9, pp. 1727–1732, Sep. 2014.

[10] S. Liu, Z. Zhang, and Y. Guo, ''2-D DOA estimation with imperfect L-shaped array using active calibration,'' *IEEE Commun. Lett.*, vol. 25, no. 4, pp. 1178–1182, Apr. 2021, doi: 10.1109/LCOMM.2020.3047222.

[11] C. G. Christodoulou, J. A. Rohwer, and C. T. Abdallah, ''The use of machine learning in smart antennas,'' in *Proc. IEEE Antennas Propag. Soc. Symp.*, vol. 1, Jun. 2004, pp. 321–324, doi: 10.1109/APS.2004.1329637.

[12] C. Maeurer, P. Futter, and G. Gampala, ''Antenna design exploration and optimization using machine learning,'' in *Proc. 14th Eur. Conf. Antennas Propag. (EuCAP)*, Mar. 2020, pp. 1–5, doi: 10.23919/EuCAP48036.2020.9135530.

[13] A. Randazzo, M. A. Abou-Khousa, M. Pastorino, and R. Zoughi, ''Direction of arrival estimation based on support vector regression: Experimental validation and comparison with MUSIC,'' *IEEE Antennas Wireless Propag. Lett.*, vol. 6, pp. 379–382, 2007.

[14] M. Chen, Y. Gong, and X. Mao, ''Deep neural network for estimation of direction of arrival with antenna array,'' *IEEE Access*, vol. 8, pp. 140688–140698, 2020, doi: 10.1109/ACCESS.2020.3012582.

[15] Z.-M. Liu, C. Zhang, and P. S. Yu, ''Direction-of-arrival estimation based on deep neural networks with robustness to array imperfections,'' *IEEE Trans. Antennas Propag.*, vol. 66, no. 12, pp. 7315–7327, Dec. 2018, doi: 10.1109/TAP.2018.2874430.

[16] A. Gupta, C. G. Christodoulou, M. Martinez-Ramon, and J. L. Rojo-Alvarez, ''Deep neural nets for DOA estimation with random arrays,'' in *Proc. IEEE Int. Symp. Antennas Propag. North Amer. Radio Sci. Meeting*, Jul. 2020, pp. 2043–2044, doi: 10.1109/IEEECONF35879.2020.9330446.

[17] O. J. Famoriji, O. Y. Ogundepo, and X. Qi, ''An intelligent deep learning-based direction-of-arrival estimation scheme using spherical antenna array with unknown mutual coupling,'' *IEEE Access*, vol. 8, pp. 179259–179271, 2020, doi: 10.1109/ACCESS.2020.3027623.

[18] W. Zhu, M. Zhang, P. Li, and C. Wu, ''Two-dimensional DOA estimation via deep ensemble learning,'' *IEEE Access*, vol. 8, pp. 124544–124552, 2020, doi: 10.1109/ACCESS.2020.3005221.

[19] S. Komeylian, ''Deep neural network modeling of different antenna arrays; analysis, evaluation, and application,'' *IEEE Can. J. Electr. Comput. Eng.*, vol. 44, no. 3, pp. 261–274, Summer 2021, doi: 10.1109/ICJECE.2020.3043756.

[20] T. R. V. Anandharajan, C. Murugalakshmi, B. Adhitya, and K. Swetha, ''Deep learning based energy efficient scheme for massive MIMO,'' *Int. J. Eng. Adv. Technol.*, vol. 8, pp. 1776–1780, Sep. 2019.

[21] M. Pastorino and A. Randazzo, ''A smart antenna system for direction of arrival estimation based on a support vector regression,'' *IEEE Trans. Antennas Propag.*, vol. 53, no. 7, pp. 2161–2168, Jul. 2005.

[22] Y. Gao, D. Hu, Y. Chen, and Y. Ma, ''Gridless 1-b DOA estimation exploiting SVM approach,'' *IEEE Commun. Lett.*, vol. 21, no. 10, pp. 2210–2213, Oct. 2017.

[23] M. Tarkowski and L. Kulas, ''RSS-based DoA estimation for ESPAR antennas using support vector machine,'' *IEEE Antennas Wireless Propag. Lett.*, vol. 18, no. 4, pp. 561–565, Apr. 2019, doi: 10.1109/LAWP.2019.2891021.

[24] D. R. Prado, J. A. López-Fernández, M. Arrebola, and G. Goussetis, ''Support vector regression to accelerate design and crosspolar optimization of shaped-beam reflectarray antennas for space applications,'' *IEEE Trans. Antennas Propag.*, vol. 67, no. 3, pp. 1659–1668, Mar. 2019, doi: 10.1109/TAP.2018.2889029.

[25] J. Gao, Y. Tian, and X. Chen, ''Antenna optimization based on co-training algorithm of Gaussian process and support vector machine,'' *IEEE Access*, vol. 8, pp. 211380–211390, 2020, doi: 10.1109/ACCESS.2020.3039269.

[26] L.-Y. Xiao, W. Shao, F.-L. Jin, B.-Z. Wang, and Q. H. Liu, ''Radial basis function neural network with hidden node interconnection scheme for thinned array modeling,'' *IEEE Antennas Wireless Propag. Lett.*, vol. 19, no. 12, pp. 2418–2422, Dec. 2020, doi: 10.1109/LAWP.2020.3034481.

[27] K.-C. Lee, S.-C. Cheng, Y.-H. Chen, C.-H. Wang, and J.-Y. Jhang, ''Application of radial basis function based neural networks to arrays of nonlinear antennas,'' in *Proc. IEEE Antennas Propag. Soc. Int. Symp.*, Jul. 2006, pp. 955–958, doi: 10.1109/APS.2006.1710689.

[28] S. Mishra, R. N. Yadav, and R. P. Singh, ''Directivity estimations for short dipole antenna arrays using radial basis function neural networks,'' *IEEE Antennas Wireless Propag. Lett.*, vol. 14, pp. 1219–1222, 2015, doi: 10.1109/LAWP.2015.2399453.

[29] M. Agatonovic, Z. Stankovic, B. Milovanovic, and N. Doncov, ''DOA estimation using radial basis function neural networks as uniform circular antenna array signal processor,'' in *Proc. 10th Int. Conf. Telecommun. Mod. Satell. Cable Broadcast. Services (TELSIKS)*, Oct. 2011, pp. 544–547, doi: 10.1109/TELSKS.2011.6143173.

[30] A. H. E. Zooghby, C. G. Christodoulou, and M. Georgiopoulos, ''A neural network-based smart antenna for multiple source tracking,'' *IEEE Trans. Antennas Propag.*, vol. 48, no. 5, pp. 768–776, May 2000, doi: 10.1109/8.855496.

[31] A. A. Galkin, V. V. Puzikov, A. V. Mikheev, A. V. Tulush, and A. S. Timoshenkov, ''Mobile satellite antenna control system based on MEMS-IMU,'' in *Proc. IEEE Conf. Russian Young Researchers Electr. Electron. Eng. (ElConRus)*, Jan. 2021, pp. 2655–2659, doi: 10.1109/ElConRus51938.2021.9396118.

[32] X. He, Y. Le, and W. Xiao, ''MEMS IMU and two-antenna GPS integration navigation system using interval adaptive Kalman filter,'' *IEEE Aerosp. Electron. Syst. Mag.*, vol. 28, no. 10, pp. 22–28, Oct. 2013, doi: 10.1109/MAES.2013.6642828.

[33] M. A. Yaqoob, F. Tufvesson, A. Mannesson, and B. Bernhardsson, ''Direction of arrival estimation with arbitrary virtual antenna arrays using low cost inertial measurement units,'' in *Proc. IEEE Int. Conf. Commun. Workshops (ICC)*, Jun. 2013, pp. 79–83, doi: 10.1109/ICCW.2013.6649205.

[34] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. Cham, Switzerland: Springer, 2006.

[35] R. V. Vitali, R. S. McGinnis, and N. C. Perkins, ''Robust error-state Kalman filter for estimating IMU orientation,'' *IEEE Sensors J.*, vol. 21, no. 3, pp. 3561–3569, Feb. 2021, doi: 10.1109/JSEN.2020.3026895.

[36] Z. M. Naing, S. Anatolii, H. S. Paing, and L. V. Thang, ''Evaluation of microelectromechanical system gyroscope and accelerometer in the object orientation system using complementary filter,'' in *Proc. IEEE Conf. Russian Young Researchers Electr. Electron. Eng. (ElConRus)*, Jan. 2021, pp. 2777–2781, doi: 10.1109/ElConRus51938.2021.9396198.

[37] S. O. H. Madgwick, S. Wilson, R. Turk, J. Burridge, C. Kapatos, and R. Vaidyanathan, ''An extended complementary filter for full-body MARG orientation estimation,'' *IEEE/ASME Trans. Mechatronics*, vol. 25, no. 4, pp. 2054–2064, Aug. 2020, doi: 10.1109/TMECH.2020.2992296.

[38] R. B. Widodo, H. Edayoshi, and C. Wada, ''Complementary filter for orientation estimation: Adaptive gain based on dynamic acceleration and its change,'' in *Proc. Joint 7th Int. Conf. Soft Comput. Intell. Syst. (SCIS) 15th Int. Symp. Adv. Intell. Syst. (ISIS)*, Dec. 2014, pp. 906–909, doi: 10.1109/SCIS-ISIS.2014.7044755.

[39] P. Narkhede, S. Poddar, R. Walambe, G. Ghinea, and K. Kotecha, ''Cascaded complementary filter architecture for sensor fusion in attitude estimation,'' *Sensors*, vol. 21, pp. 1–18, Mar. 2021, doi: 10.3390/s21061937.

[40] K. G. Sheela and S. N. Deepa, ''Review on methods to fix number of hidden neurons in neural networks,'' *Math. Problems Eng.*, vol. 2013, pp. 1–11, Jan. 2013, doi: 10.1155/2013/425740.

[41] MAX78000. *AI Microcontroller With Neural Network Accelerotor*. Accessed: Nov. 26, 2022. [Online]. Available: https://www.analog.com/en/products/max78000.html#product-overview

[42] LPSTK-CC1352R. *SimpleLink? Multi Band CC1352R Wireless MCU Launchpad? SensorTag Kit*. Accessed: Nov. 26, 2022. [Online]. Available: https://www.ti.com/tool/LPSTK-CC1352R

[43] BMX160. *IMU Board*. Accessed: Nov. 26, 2022. [Online]. Available: https://www.bosch-sensortec.com/products/motion-sensors/absolute-orientation-sensors/bmx160/

RICCARDO COLELLA (Senior Member, IEEE) received the M.Sc. degree (Hons.) in telecommunication engineering and the Ph.D. degree from the University of Salento, Italy, in 2010 and 2015, respectively. He is currently an Assistant Professor with the Department of Innovation Engineering, University of Salento. He has authored about 150 papers appeared on international journals and national and international conferences and two book chapters with international diffusion. He is the co-inventor of one patent. His research interests include the design of innovative RF/RFID electronic devices, antennas, wirelessly powered IoT sensing systems, wireless devices for healthcare applications, and 3-D-printable RF electronic systems and antennas.

LUIGI SPEDICATO received the M.S. degree in automation engineering and the Ph.D. degree in mechanical and industrial engineering from the University of Salento, Lecce, Italy, in 2009 and 2013, respectively. Since 2013, he has been with the Department of Innovation Engineering, University of Salento. In 2021, he began collaborating with the EMTech Laboratory with the University of Salento, as a Research Fellow, focusing primarily on AI algorithms applied to electromagnetic issues. His research interests include mechatronic systems and control of mechanical systems, as well as sensing and signal processing. His research interests include the areas of machines and intelligent machines, such as mobile robots and robotic arms, including aerial navigation and space exploration.

LAURA LAQINTANA received the M.S. degree in telecommunications engineering from the University of Salento, Lecce, Italy, in 2022. Her M.S. thesis was titled, "the Application of Artificial Intelligence to Electromagnetics," showcasing her strong interest and expertise in AI and its potential applications in telecommunications. She is currently working on the framework of cyber-security and also applied it to radio and wireless systems.

LUCA CATARINUCCI (Senior Member, IEEE) is currently an Associate Professor in electromagnetic with the Department of Innovation Engineering, University of Salento, Italy. He is the chair of "Microwaves" and "Electromagnetic Solutions for Hi-Tech" and leads the Research Group focused on RFID and Emerging Technologies for IoT. He has authored more than 200 papers published in international journals and conferences and four chapters books with international diffusion. He is the co-inventor of two patents. His research interests include the implementation of high-performance electromagnetic simulation tools, the electromagnetic characterization of heterogeneous materials, and the use of time-domain reflectometry for the qualitative and quantitative characterization of fluids. He is currently involved in RFID-related activities, ranging from antenna and system design, integration between sensors and RFID tags, RFID-based robot navigation, and new techniques for tag characterization, optimization, and design. Moreover, he collaborates with a multitude of research groups on topics referring to antenna arrays, antenna manufacturing, 3-D Printing in electromagnetics, and others. He is the Vice President of Technical Activities at the IEEE Council on RFID (CRFID) for the two-year term 2022–2023 and the Chair of the IEEE Technical Committee on Additively Manufactured Electronic Systems.

Open Access funding provided by 'Università del Salento' within the CRUI CARE Agreement