Received 11 July 2025; accepted 10 August 2025. Date of publication 13 August 2025; date of current version 28 August 2025.

Digital Object Identifier 10.1109/OJCOMS.2025.3598546

AI-Driven Optimization of Wave-Controlled Reconfigurable Intelligent Surfaces

GAL BEN-ITZHAK (Graduate Student Member, IEEE),
MIGUEL SAAVEDRA-MELO (Graduate Student Member, IEEE), ENDER AYANOGLU (Fellow, IEEE),
FILIPPO CAPOLINO (Fellow, IEEE), AND A. LEE SWINDLEHURST (Fellow, IEEE)

Department of Electrical Engineering and Computer Science, University of California at Irvine, Irvine, CA 92697, USA

CORRESPONDING AUTHOR: E. AYANOGLU (e-mail: ayanoglu@uci.edu)

This work was supported in part by the National Science Foundation (NSF) under Grant ECCS-2030029.

ABSTRACT A promising type of Reconfigurable Intelligent Surface (RIS) employs tunable control of its varactors using biasing transmission lines below the RIS reflecting elements. Biasing standing waves (BSWs) are excited by a time-periodic signal and sampled at each RIS element to create a desired biasing voltage and control the reflection coefficients of the elements. A simple rectifier can be used to sample the voltages and capture the peaks of the BSWs over time. Like other types of RIS, attempting to model and accurately configure a wave-controlled RIS is extremely challenging due to factors such as device non-linearities, frequency dependence, element coupling, etc., and thus significant differences will arise between the actual and assumed performance. An alternative approach to solving this problem is data-driven: Using training data obtained by sampling the reflected radiation pattern of the RIS for a set of BSWs, a neural network (NN) is designed to create an input-output map between the BSW amplitudes and the resulting sampled radiation pattern. This is the approach discussed in this paper. In the proposed approach, the NN is optimized using a Genetic Algorithm (GA) to minimize the error between the estimated and measured radiation patterns. The BSW amplitudes are then designed via Simulated Annealing (SA) to optimize a signal-to-leakage-plus-noise ratio measure by iteratively forward-propagating the BSW amplitudes through the NN and using its output as feedback to determine convergence. The resulting optimal solutions are stored in a lookup table to be used both as settings to instantly configure the RIS and as a basis for determining more complex radiation patterns.

INDEX TERMS Neural network (NN), simulated annealing (SA), genetic algorithm (GA), reconfigurable intelligent surface (RIS), machine learning (ML).

I. INTRODUCTION

A RECONFIGURABLE Intelligent Surface (RIS) is a promising building block for next-generation wireless networks, potentially solving issues related to limited coverage, sensitivity to blockages, high path loss, etc., by providing alternative propagation paths and enabling beamforming for desired directions [1]. An RIS is a type of metasurface [2], [3] consisting of periodic arrangements of subwavelength-sized passive reflecting elements whose reflection coefficients can be individually controlled.

The RIS architecture studied in this paper is illustrated in Fig. 1, and is based on the general wave-controlled approach described in [4]. In this approach, biasing standing waves (BSWs) are excited on a transmission line (TL) behind the RIS structure and sampled at discrete locations to provide the DC bias voltages for varactor diodes that modulate the reflection coefficients. In a previous paper [5], various optimization algorithms were developed that demonstrate the capability of the wave-controlled RIS to create beams and nulls in the radiation pattern for multiple desired directions. The performance of the optimization algorithms was demonstrated for two different types of voltage sampling circuits: a *rectifier* that is easier to implement but more difficult to use for optimization, and a *sample-and-hold* circuit, which is more costly to implement but enables analytical solutions to the beam pattern optimization problem.

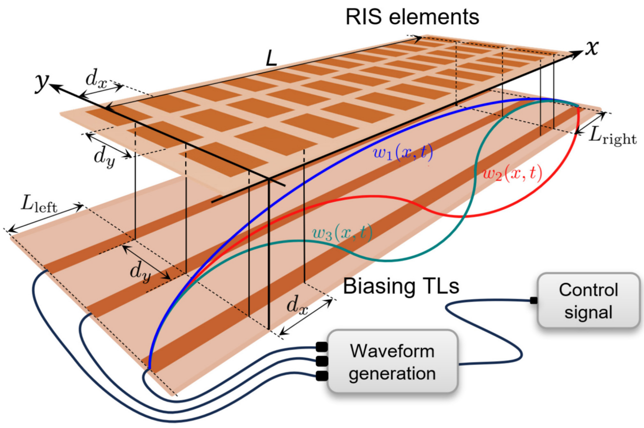


FIGURE 1. Wave-controlled RIS composed of two physical layers [5]. Top layer: M elements in each row along the x direction, where each element is connected to a varactor diode. Bottom layer: N BSWs are excited on each TL to create the biasing voltages sampled at each RIS element. Each row is controlled via the connection on the left where a signal with N adjustable frequency components is injected by a waveform generator. Adjacent metallic patches and varactors on the top layer, and adjacent DC voltage outputs on the bottom layer are uniformly separated by distance d_x in the x direction and d_y in the y direction.

In this work, we consider the rectifier implementation, where the beam pattern optimization is nonlinear and constrained, and we assume the RIS is deployed in an unknown environment without knowledge of channel state information (CSI). Thus, there is no exact mathematical model that can be used to predict the response created by the RIS to an excitation by the BSWs. Consequently, we will turn to the use of machine learning (ML) tools to solve the RIS configuration problem.

Optimizing the wireless propagation environment using RIS has been prevalently explored through various optimization algorithms, including model-based, ML-based, and heuristic approaches, as well as hybrid methods combining multiple techniques [6]. While model-based optimization offers theoretical rigor, its practical application is hindered by discrepancies between simulations and real-world environments and circuitry. Conventional mathematical models fail to account for frequency-dependent behavior and device nonlinearities, and direct control of individual RIS reflection coefficients is constrained by element coupling.

The ML-based approaches discussed in [6] show applications in CSI prediction, RIS phase control and optimization, and have demonstrated effectiveness in improving data rates and secrecy rates. However, many rely on assumptions regarding CSI availability, for example, considering using partial CSI to predict full CSI or RIS phase shifts to maximize the data rate. This is often impractical, or at least difficult to implement, due to the complexity of CSI acquisition with passive RIS [7], [8]. Additionally, neural network (NN) architecture design presents its own challenges, including susceptibility to overfitting. Common mitigation techniques include adding dropout layers [9], expanding datasets, and reducing the number of hidden layers [10]. Yet, hyperparameter tuning remains case-specific and labor-intensive. The specific types of nonlinearities exploited by the NN must be carefully chosen, such that it may require many iterations to train an ML model, which may not guarantee optimality. A systematic framework for model development and evaluation is thus desirable. Other work has shown the feasibility of using unsupervised learning for Signal-to-Noise Ratio (SNR) maximization in a single-user scenario, but it also prompts scalability concerns as the size of the dataset grows with the number of RIS elements [11].

Recent work has applied deep-learning methods to capture the relationships between RIS configurations and achievable rates [12]. It proposes using a Deep Neural Network (DNN) to model the RIS-assisted environment and optimize the achievable rate at a given user direction by sweeping the entire codebook of RIS configurations as inputs to the DNN for inference. However, inference based on exhaustive codebook search introduces latency, which grows significantly with the number of RIS elements and allowable configurations. Other codebook approaches used for channel estimation and passive beamforming RIS design are explored in [13], [14], [15]. While codebook-based approaches provide structured solutions, they rely on predefined phase shifts, limiting flexibility in complex environments. More training iterations would be required as new users are introduced, or existing users move around.

In the approach presented in this paper, we address these shortcomings. In particular, we focus on the use of wave-controlled RIS, which as described above is a novel architecture that accommodates the physical limitations of an RIS and simplifies the hardware required to operate it. The wave-controlled RIS achieves reduced-dimension control of the M RIS elements by configuring $N \ll M$ BSW amplitudes on a TL. This approach also accounts for the coupling between RIS elements by maintaining a relatively smooth variation in the voltage profiles across adjacent elements. We assume the BSW amplitudes can each be configured continuously and are not confined to a discrete codebook, thus enhancing the versatility of the system. We address challenges in system modeling and optimization by employing a data-driven method that does not require explicit CSI or real-time feedback. Our methodology involves:

1) *Training Data Collection:* Randomly exciting the RIS and sampling the received power at specific directions.
2) *Neural Network Design and Optimization:* Designing an NN to estimate the input-output relationship between BSW amplitudes and received powers, with architecture refinement automated via a Genetic Algorithm (GA).
3) *Passive Beamforming via Simulated Annealing (SA):* Using a heuristic approach for offline optimization of beam patterns based on a signal-to-leakage-plus-noise ratio (SLNR) metric, using feedback from the NN during iterations of the algorithm.
4) *Efficient Lookup Table Deployment:* Storing optimized BSW configurations for rapid RIS beam steering and

future complex radiation pattern adjustments, using the existing configurations as initialization for SA.

By departing from conventional codebook-based methods and eliminating the need for direct CSI acquisition, our framework provides a more adaptable and scalable solution for RIS optimization. Sampling powers at directions of interest allows us to represent the entire RIS-aided wireless environment as a radiation pattern which can be customized based on desired quality of service metrics. We note that previous success in using heuristic algorithms for RIS optimization, in particular using SA and GA, has been explored in literature, see for example [5], [16], [17]. However, in this work we introduce the novel approach of using a GA as an automation tool to optimize the NN architecture and hyperparameters, while using SA to generate optimal sets of BSWs to configure the RIS phase shifts for a desired beam pattern, employing feedback from the NN during inference.

We would like to note that a main novelty in this paper is the fact that its set of algorithms are specifically designed for the wave-controlled RIS. However, we would also like to note that the approaches used in this paper do not hold only for the wave-controlled RIS, but also for other RIS optimization scenarios where parameters other than the BSW amplitudes can be optimized, such as varactor bias voltages or direct reflection coefficients.

The paper is organized as follows. Section II presents the electromagnetic and circuit models for simulating the wave-controlled RIS. Section III outlines the ML-based RIS control, dataset creation, and GA-driven optimization. Section IV integrates SA and the ML model for beamforming and nullforming, and shows simulation results. Section V evaluates the performance gain achieved by employing the lookup table, also backed by simulations.

II. MODEL, ASSUMPTIONS, AND NOTATION

A. RIS MODEL

In Fig. 1, we show an RIS with uniform linear arrays (ULAs) of M metallic patches in the x direction with period d_x. The figure shows three parallel linear arrays as an example. Each RIS element is connected to a varactor diode which is used to vary the reflective properties of the element, as illustrated in Fig. 2. On the backside of the structure, there is a biasing TL of length L along x on which a set of N BSWs are excited over the entire length of the TL. We assume that the fundamental standing wave is resonant with the total TL length L_{tot}, leading to the fundamental angular frequency $\omega_{\text{b}} = 2\pi f_{\text{b}} = \pi v_{\text{ph}}/L_{\text{tot}}$ [18]. The other BSWs oscillate at the harmonic frequencies nf_{b}, with $n = 1, 2, \ldots, N$. The overall BSW signal, assuming a short circuit termination at the end of the biasing TL, is given by

$$w(x,t) = W_0 + \sum_{n=1}^{N} W_n \sin\left(\frac{n\pi\,(x + L_{\text{left}})}{L_{\text{tot}}}\right)\sin(n\omega_{\text{b}}t), \quad (1)$$

where the individual BSWs are controlled by tuning their amplitudes, which are stored in the vector $\boldsymbol{W} = [W_1, W_2, \ldots, W_N]$. The W_0 term is a DC component used to center the biasing of the varactors on their best working range for capacitance control.

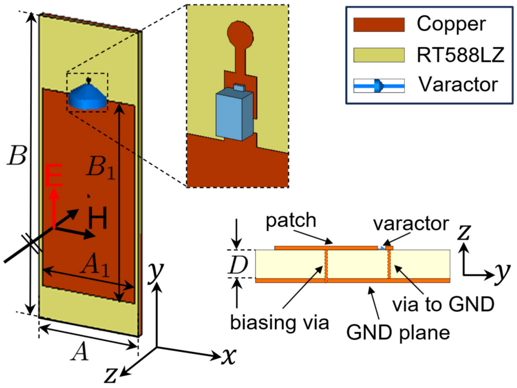


FIGURE 2. RIS unit cell geometry. Each rectangular metallic conductor is biased by the sampled voltage through a via and connected to a grounded varactor. The reverse-biased varactors act as tunable capacitors to polarize the incident electric field along the y direction.

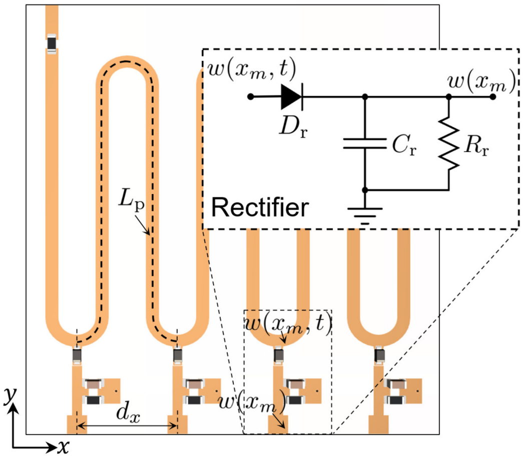


FIGURE 3. Geometry of the biasing TL. The length L_{p} of the TL path for one unit cell and the distance d_x between two adjacent rectifier circuits are detailed. These circuits, which rectify the BSWs [5], are located at the bottom, with one assigned to each RIS element. The voltage $w(x_m, t)$ is extracted from the biasing TL at the location m, while the rectified voltage $w(x_m)$ provides the bias to the varactor of the mth element. $w(x_m, t)$ is rectified using the diode D_r and by following its envelope or peaks through the RC circuit shown, with a carefully chosen time constant to minimize voltage drops in $w(x_m)$ due to capacitor discharge.

To minimize the control signaling overhead and reduce the variation of the bias voltage from element to element, it is desired that $N \ll M$. Connected to each RIS element with a vertical via is a rectifier circuit, shown in Fig. 3, which rectifies the BSW signal at each RIS element location $x_m = md_x$, with $m = 0, 1, 2, ..M-1$. We define the reference point $x = 0$ at the location where the first rectifier is located and L_{left} as the length of the biasing TL in the x direction before (to the left of) this detector. We also define L_{right} as the length of the biasing TL between the last rectifier and the short circuit at the end of the TL on the right. The total length of the waveguide along x is given by

$$L_{\text{tot}} = L_{\text{left}} + L + L_{\text{right}}. \quad (2)$$

The voltage variation along x for each BSW is based on the phase velocity v_{ph}, which we assume due to low frequency dispersion to be the same for all the waves. The phase velocity is given by $v_{\mathrm{ph}} = c/n_{\mathrm{slow}}$, where c is the speed of light and n_{slow} is the slowness factor that depends on the geometry and material properties of the biasing TL. As shown in Fig. 3, L_{p} represents the physical path length between two adjacent DC voltage outputs along the biasing TL, while d_x is the corresponding separation along the x direction. The effective refractive index, n_{eff}, of the biasing TL is determined by its effective permittivity ϵ_{eff} as $n_{\mathrm{eff}} = \sqrt{\epsilon_{\mathrm{eff}}}$. Consequently, the slowness factor in this case is given by $n_{\mathrm{slow}} = (L_{\mathrm{p}}/d_x)n_{\mathrm{eff}}$.

The voltage is uniformly sampled at discrete locations $x_m = md_x$, where $m = 0, 1, \ldots, M-1$, and $0 \le x_m \le L$.

We rewrite (1) in terms of the RIS element indices m as

$$w(md_x, t) = W_0 + \sum_{n=1}^{N} W_n \sin\left(\frac{n\pi\,(md_x + L_{\mathrm{left}})}{L_{\mathrm{tot}}}\right) \sin(n\omega_{\mathrm{b}}t). \quad (3)$$

The varactor diode connected to each RIS element is biased in reverse using the rectifier circuit as in the inset of Fig. 3.

We assume that the rectified voltage of the BSWs at each discrete location x_m is equivalent to the peak voltage (maximum voltage at each location over time), yielding

$$w(md_x) = W_0 + \max_t \left(\sum_{n=1}^{N} W_n \sin\left(\frac{n\pi\,(md_x + L_{\mathrm{left}})}{L_{\mathrm{tot}}}\right) \sin(n\omega_{\mathrm{b}}t) \right), \quad (4)$$

with W_0 taken out of the $\max_t(\cdot)$ term as it is a constant independent of time. This positive voltage is then applied at the cathode of the varactor, thus reverse-biasing it, determining its capacitive properties, and consequently tuning the reflection coefficient of the RIS element to provide the desired response.

Each RIS unit cell shown in Fig. 2 is designed with (all dimensions in mm) $A = 20$, $A_1 = 18.5$, $B = 72.7$, $B_1 = 36.7$, $D = 1.27$. The dielectric spacer is the substrate Rogers RT5880LZ, with relative permittivity $\epsilon_r = 2$ and dielectric loss $\tan(\delta) = 0.0021$. To simulate the tunable impedance of each RIS element, the equivalent circuit model shown in Fig. 4 is used and optimized with full-wave simulations [3], [19]. The unit cell elements have resistance $R_{\mathrm{d}} = 0.1671\ \Omega$, capacitance $C_{\mathrm{d}} = 0.97821$ pF, and inductance $L_{\mathrm{d}} = 1.9177$ nH. There is an additional inductive term $L_{\mathrm{s}} = 1.5959$ nH in parallel to the unit cell element and varactor that accounts for the grounded substrate, creating the "magnetic resonance" effect discussed in [2], [20], [21].

The varactor SMV1231-040LF provided by Skyworks Solutions, Inc. is chosen due to its desirable characteristics, including resistance below 0.6 Ω and low series inductance (0.45 nH) for our frequencies of interest. The varactor is modeled by an equivalent series RLC circuit with inductance L_{v}, capacitance $C_{\mathrm{v}}(\mathrm{V})$, and resistance $R_{\mathrm{v}}(\mathrm{V})$. The inductance $L_{\mathrm{v}} = 2.34$ nH is constant and models the package and parasitic inductance when the varactor is connected across the gap. The values for $C_{\mathrm{v}}(\mathrm{V})$ and $R_{\mathrm{v}}(\mathrm{V})$ shown in Fig. 5 are obtained from a parametric sweep simulation using Advanced Design System (ADS) software, as functions of the reverse voltage bias across the varactor model provided by the vendor, limited to the range [4 V, 15 V].

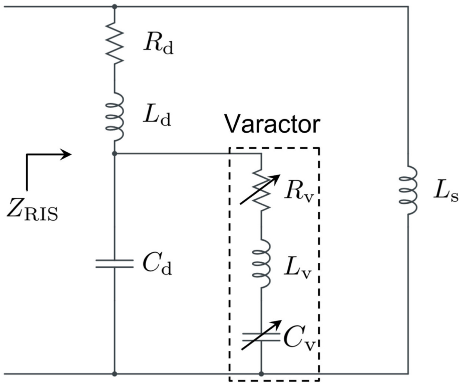


FIGURE 4. Analytical model of the RIS unit cell including an RLC model of the varactor to calculate the reflection coefficient as a function of frequency or varactor biasing voltage. The equivalent impedance Z_{RIS} is used to find the reflection coefficient Γ.

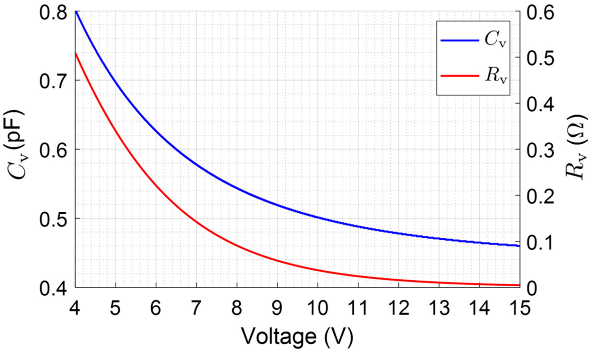


FIGURE 5. Equivalent resistance and capacitance for the SMV1231-040LF varactor as functions of the biasing voltage across the varactor model shown in Fig. 4.

The total equivalent RIS impedance is expressed by

$$Z_{\mathrm{RIS}} = \left(R_{\mathrm{d}} + j\omega L_{\mathrm{d}} + \left(R_{\mathrm{v}} + j\omega L_{\mathrm{v}} + \frac{1}{j\omega C_{\mathrm{v}}} \right) \| \frac{1}{j\omega C_{\mathrm{d}}} \right) \| j\omega L_{\mathrm{s}}, \quad (5)$$

and is used to determine the reflection coefficient of the RIS elements by

$$\Gamma = \frac{Z_{\mathrm{RIS}} - Z_0}{Z_{\mathrm{RIS}} + Z_0}, \quad (6)$$

where Z_0 is the free-space impedance. This model predicts the magnitude and phase of the local reflection coefficient for variable frequency and varactor biasing voltage V. An example is given in Fig. 6, where the magnitude and phase of the reflection coefficients as functions of the varactor reverse bias voltage are plotted for the frequency $f_c = 2.45$ GHz.

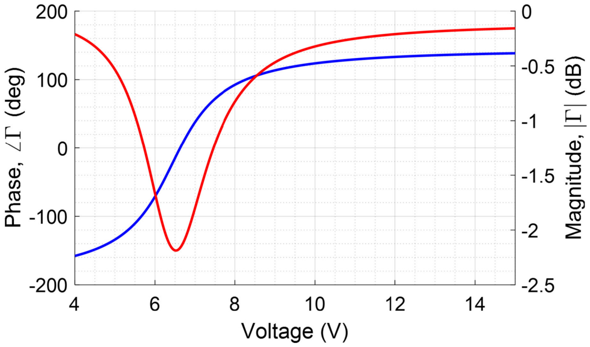


FIGURE 6. Reflection magnitude and phase of the RIS unit cell described in Fig. 2 as functions of the reverse bias voltage across the varactor at the carrier frequency $f_c = 2.45$ GHz.

TABLE 1. Mathematical notations.

Variable	Definition
N	Number of BSW frequencies
M	Number of RIS elements
W_0	Transmission line DC voltage bias
$\boldsymbol{W}$	BSW amplitudes vector
ω_b	Fundamental BSW angular frequency
d_x	Distance between adjacent varactors
$w(x,t)$	Transmission line voltage at location x and time t
$\boldsymbol{\Gamma}$	Diagonal matrix with RIS reflection coefficients
K	Number of desired user-end Rx
L	Number of eavesdropper Rx
$\boldsymbol{h}_k$	Channel vector from the RIS to the k-th Rx
$\boldsymbol{g}$	Channel vector from the Tx to the RIS
ρ_s	Average transmitted symbol power
σ_s^2	Noise power of AWGN channel
SLNR	Signal-to-leakage-plus-noise ratio
n_{angles}	Number of sampled Rx directions in training dataset
P_θ	Received power at direction θ
$\boldsymbol{P}$	Received power array at all n_{angles} directions

The main notations used in this work are summarized in Table 1.

B. SIGNAL MODELS

We adopt the assumptions of [5], where all channels are narrowband line-of-sight (LoS), flat-fading, and in the far-field. There is a single-antenna transmitter (Tx) and K single-antenna receivers (Rxs), with no direct path between the Tx and each one of the Rxs. The signal y_k at the k-th Rx is given by

$$y_k = \boldsymbol{h}_k^T \boldsymbol{\Gamma} \boldsymbol{g} s + n_k, \tag{7}$$

where s is the transmitted signal, n_k is Additive White Gaussian Noise (AWGN), and $\boldsymbol{h}_k$ and $\boldsymbol{g}$ are the $M \times 1$ channels from the RIS to the k-th Rx and from the Tx to the RIS [22]. The RIS response is represented by a diagonal matrix containing the RIS reflection coefficients:

$$\boldsymbol{\Gamma} = \operatorname{diag}[\Gamma(0), \Gamma(1), \ldots, \Gamma(M-1)]. \tag{8}$$

The element responses are passive ($|\Gamma(m)| \leq 1$ for all $m = 0, 1, \ldots, M-1$) and are determined by the biasing voltage applied to the m-th RIS element. The Tx is assumed to be located at the broadside of the RIS, at the far-field, such that there is normal incidence of the signal on all RIS elements, so the Tx-RIS channel is given by

$$\boldsymbol{g} = [1, 1, \ldots, 1], \tag{9}$$

and the channels from the RIS to each Rx are given by the steering vectors of the form

$$\boldsymbol{h}(\theta_k) = \left[1, e^{-j\kappa(\theta_k)}, e^{-j2\kappa(\theta_k)}, \ldots, e^{-j(M-1)\kappa(\theta_k)}\right], \tag{10}$$

where θ_k is the azimuth angle of the k-th Rx from the RIS with spatial frequencies $\kappa(\theta) = 2\pi(d_x f_c/c)\sin(\theta)$. We assume the far-field channels remain static during the operation of the RIS, to ensure the integrity of the ML model designed by the initial training data.

The SNR at each Rx k is defined as the ratio of the received signal power divided by the noise power σ_s^2:

$$\text{SNR}_k = \frac{|E[y_k]|^2}{\sigma_s^2} = \frac{|E[\boldsymbol{h}_k^T \boldsymbol{\Gamma} \boldsymbol{g} x]|^2}{\sigma_s^2} = \frac{\rho_s |\boldsymbol{h}_k^T \boldsymbol{\Gamma} \boldsymbol{g}|}{\sigma_s^2}, \tag{11}$$

where ρ_s is the average transmitted symbol power. To evaluate the feasibility of shaping the radiation patterns generated by the wave-controlled RIS, we define the following SLNR (Signal-to-Leakage-plus-Noise Ratio) measure for $K \geq 1$ desired Rx and $L \geq 0$ undesired Rx ("eavesdroppers"):

$$\max_{\boldsymbol{W}} \text{SLNR} = \max_{\boldsymbol{W}} \frac{\min_{i \in \{1,2,\ldots,K\}} \rho_s |\boldsymbol{h}_{d,i}^T \boldsymbol{\Gamma} \boldsymbol{g}|^2}{\max_{j \in \{1,2,\ldots,L\}} \rho_s |\boldsymbol{h}_{e,j}^T \boldsymbol{\Gamma} \boldsymbol{g}|^2 + \sigma_s^2}, \tag{12}$$

where each $\boldsymbol{h}_{d,i}$ denotes the channel from the RIS to the i-th desired Rx, and $\boldsymbol{h}_{e,j}$ denotes the channel from the RIS to the j-th eavesdropper Rx. The independent variable used for optimization is $\boldsymbol{W}$, the vector containing all the BSW amplitudes, which determines the reflection coefficients in $\boldsymbol{\Gamma}$. We choose the SLNR metric since it is well suited for quantifying performance in terms of beam patterns, and is general enough to apply in cases whether or not one wishes to steer a null in a particular direction.

III. MACHINE LEARNING MODEL DESIGN AND CALIBRATION

A. BACKGROUND

Numerous factors differentiate the ideal RIS model often used in simulations from the actual behavior of an RIS in a physical environment. Starting with the design of the biasing TL, there may be impedance mismatches, losses, and reflections along the path, creating signal losses and distortion [23]. Since the wave-controlled RIS operates with a wide range of BSWs, there may be frequency-dependent losses along the biasing TL and other nonlinearities caused by skin effects, noise, crosstalk, as well as jitter and phase delays from the waveform generator. There is also a non-negligible voltage drop across the rectifier diode. Depending on the peak detector architecture, its output

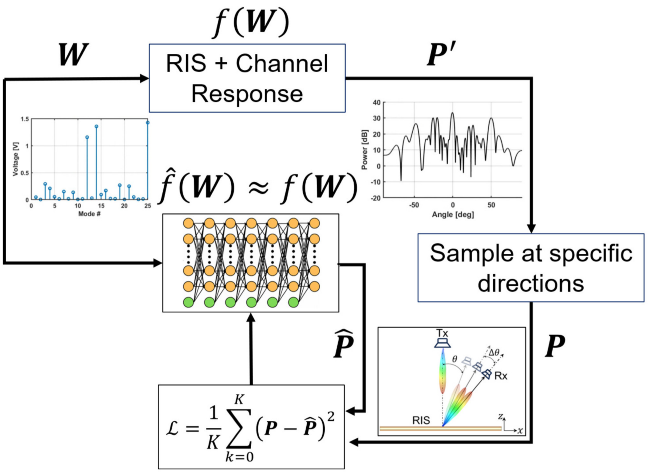


FIGURE 7. The RIS and channel response are represented by the function $f(W)$ which takes an input set of BSW amplitudes W and outputs a radiation pattern. The radiation pattern P' is sampled at specific locations and stored as P. An NN is then trained to estimate the sampled power values $\hat{P}$ from the same data, where the MSE between the actual and estimated power values is used as the loss function $\mathcal{L}$.

voltage could be affected by the charging and discharging times of the capacitor. These and other non-ideal effects are difficult to model, and can only be characterized using full-wave simulations or extensive experimental measurements. Furthermore, the reflection coefficients created by the varactor biasing may not be perfectly modeled, as varactor characteristics may vary due to environmental (thermal), fabrication, and electrical conditions [24], as well as mutual coupling between elements [25]. The metasurface response also varies according to the angle of arrival of the incoming signal [26]. Lastly, there is the issue of channel estimation for the Tx-RIS and RIS-Rx links required to simulate the performance of the entire system and optimize the resulting beam patterns [27], [28], [29], [30].

Rather than relying on mathematical models that attempt to mimic the RIS and physical environment, an alternative is to use data from the environment in which the RIS is deployed to characterize the behavior of the entire system – from the input BSW amplitudes to the waveguide and to the resulting radiation patterns. When an RIS is initially deployed, CSI is unavailable and the waveform sampling required to excite the metasurface is unknown. To learn how the system behaves, we can excite the RIS with a combination of BSW amplitudes and sample the power obtained at receivers in different directions of interest. This procedure can be repeated many times to obtain improved knowledge of the input-output relationship of the system, which can in turn be used to design an ML model that models this relationship. The corresponding functional representation of the system and the ML model training procedure can be visualized as in Fig. 7. If the ML model is well calibrated, it will provide a sufficiently accurate representation of the nonlinear and complex mapping between the BSW amplitudes and the sampled powers, and can be used to perform offline optimization to obtain the desired beam patterns. Once found, the optimal BSW amplitudes are used to excite the physical RIS and create the actual radiation patterns. Thus, this ML-based approach does not require any knowledge of CSI or channel estimation for the beam pattern optimization.

The main problems that must be solved when using this approach include

1) What combinations of BSW amplitudes should be used to excite the RIS?
2) Which ML model provides a sufficiently generalized representation of the system?

The answer to the first question requires some knowledge of the system, such as how large should the maximum voltage swing on the biasing TL be. For this question, for example, we can simulate the ideal scenario where there is no attenuation on the biasing TL, and use these results to verify that the peak voltage generated by each set of BSW amplitudes will not exceed the maximum voltage swing constraint. The answer to the second question requires a deeper analysis of available network architectures and their optimization. Creating a map between the N input BSW amplitudes and the n_{angles} spatially distributed power values at the receivers is fundamentally a regression problem. Since the map is high-dimensional, nonlinear, and complex, most kernel methods and surrogate function estimators that involve linear transformations and interpolations between training samples will fail. Such approaches also require large matrix operations as more training samples are added [31], [32].

An NN, or more specifically, a fully-connected (dense) Multilayer Perceptron (MLP) [33] is a better approach for modeling the system, as they achieve universal function approximation by leveraging multiple layers of nonlinear functions with many hidden variables [34]. The MLP approach takes an input set of N BSW amplitudes, propagates them through multiple hidden layers with nonlinear activation functions, and provides as output a set of n_{angles} power values from the corresponding radiation pattern, sampled at discrete locations. This is a supervised learning problem, where the network tries to minimize the mean squared error (MSE) between the actual and estimated outputs, while also generalizing the complex relationship between the BSW amplitudes and their corresponding radiation patterns. To come up with a good MLP architecture for the given problem, one may manually design an optimal MLP through trial-and-error, but this process is highly time-consuming. Moreover, the architecture of the RIS, the numbers of inputs and outputs, the frequencies of the BSWs, the locations of the Rx, the channel models, and many other factors affect the regression map, often in a case-by-case manner. As a result, no single MLP architecture can be generalized for every RIS modeling problem across all environments. To address this challenge, rather than manually tuning the MLP architecture for each case, an optimization algorithm can automate the process by creating an MLP that accurately models the input-output relationship while satisfying architectural constraints. We propose using a GA that creates a diverse pool of MLPs trained via back-propagation, refined through evolution, and optimized to minimize validation loss – thereby enhancing model performance [35], [36].

The following subsections will address the data processing steps required to use the GA to generate an MLP that will estimate the desired mapping with high accuracy. It is important to note that the procedures discussed in this paper are not only relevant for the specific wave-controlled RIS, rectifier-based multi-user beamforming problem considered here. The proposed procedure and algorithms can be generalized to other RIS modeling and optimization problems, including calibration of a wave-controlled RIS with the sample-and-hold circuit in [5].

B. DATA GENERATION AND PROCESSING

Simulation of a wave-controlled RIS and corresponding wireless channels was executed using MATLAB with the models defined in Section II. The RIS is modeled as a single-row surface with $M = 100$ elements, biased by a superposition of $N = 25$ BSWs. The RIS elements are separated by $d_x = 20$ mm, while behind the surface the length of the TL path between adjacent rectifiers is $L_\mathrm{p} = 131.42$ mm. The effective permittivity of the TL is $\epsilon_\mathrm{eff} = 8.66$, and therefore the slowness factor is $n_\mathrm{slow} = 19.34$. The transmitted signal has carrier frequency $f_\mathrm{c} = 2.45$ GHz, and the DC voltage offset is $W_0 = 4$ V, as this is the minimal voltage required to excite the varactor, and as seen in (3), the BSWs will always create a higher voltage than W_0 at any RIS element location. This allows generation of voltages across the entire allowable range of the varactor. The distance between the beginning of the TL and the first RIS element along the x direction is $L_\mathrm{left} = 0.5d_x$. The distance is the same between the last RIS element and the end of the TL. For better control of the voltage over the length $L = 99d_x$ used to bias the RIS elements, an additional TL of the same length and geometry as the first biasing TL is connected at its end through a 50 mm connector. Thus, the overall length from the last varactor to the end of the extended TL, normalized by d_x/L_p, is

$$L_\mathrm{right} = 0.5d_x + 50\mathrm{mm} \cdot \frac{d_x}{131.42\mathrm{mm}} + 100d_x \approx 100.88d_x \tag{13}$$

This extended TL results in a fundamental BSW frequency of $f_\mathrm{b} = 1.93$ MHz.

To enable good coverage of the 25-dimensional input space, a dataset was generated with $N_\mathrm{max} = 100{,}000$ pairs of data arrays, each consisting of a random $\boldsymbol{W}$ array of voltages as the input, and samples from the resulting radiation pattern vector $\boldsymbol{P}$ in dB as outputs when the RIS waveguide is excited by the corresponding BSWs. The azimuth angles corresponding to the sampled points on the radiation pattern are in the interval $[-60°, 60°]$. The simulated 3 dB beamwidth for the given $M = 100$ element RIS is approximately 1.7°, so we used discrete steps of 1.5° spaced uniformly between $[-60°, 60°]$, resulting in power samples at $n_\mathrm{angles} = 81$ different angles.

Algorithm 1 Dataset Generation

```
 1: $i \leftarrow 0$
 2: while $i < N_\mathrm{max}$ do
 3:     $\boldsymbol{W} \leftarrow [0, 0, \ldots, 0]^T$
 4:     $\boldsymbol{W}(n) \leftarrow$ sample from $|\mathcal{N}(0, \sigma_1^2)|$ for $n = 1, \ldots, N$.
 5:     $k \leftarrow$ integer sample from $\mathcal{U}(1, N)$.
 6:     $\boldsymbol{S} \leftarrow$ unique set of $k$ uniformly chosen integers from 1 to $N$.
 7:     $\boldsymbol{W}(n) \leftarrow$ sample from $|\mathcal{N}(0, \sigma_2^2)|$ $\forall n \in \boldsymbol{S}$.
 8:     Calculate maximum absolute value $v_\mathrm{max}$ of the resulting BSW superposition over time from $W_0$ and $\boldsymbol{W}$.
 9:     if $v_\mathrm{max}$ is out of varactor biasing bounds then
10:         Go back to line 3.
11:     end if
12:     Sample the power at all $n_\mathrm{angles}$ desired directions, store in $\boldsymbol{P}$.
13:     Store $\boldsymbol{W}$ and $\boldsymbol{P}$ as $\boldsymbol{W}_i$ and $\boldsymbol{P}_i$.
14:     $i \leftarrow i + 1$
15: end while
```

To reduce the likelihood of producing chaotic radiation patterns caused by fully randomized inputs which will negatively affect the learning by the MLP [37], the following methodology is proposed. Since the MLP will later be used to estimate the model for optimization by SA, which randomly perturbs all the BSW amplitudes in the input vector to converge towards an optimal SLNR measure, a small random Gaussian voltage with a standard deviation $\sigma_1 = 0.008$ V whose absolute value is taken, is added to all elements of an initially empty $\boldsymbol{W}$. A randomly selected subset of the BSW amplitudes is then excited by Gaussian-distributed voltages with a much larger standard deviation $\sigma_2 = 0.8$ V, also converted to absolute values, to put more emphasis on those BSWs and create more unique radiation patterns. Note that only positive voltage amplitudes are used to reduce the dataset complexity by truncating the Gaussian distributions by taking the absolute values of their samples. This ensures that only the amplitudes of the BSWs are modulated, rather than both the amplitudes and the phases. The entire procedure to create the training dataset is described in Algorithm 1. Since we use back-propagation and gradient descent, it is beneficial to normalize the data to improve convergence [38]. In particular, we scale each of the 100,000 $\boldsymbol{W}$ vectors to be within the range $[0, 1]$, and we scale the resulting power values in $\boldsymbol{P}$ to be in the range $[-1, 1]$.

C. NEURAL NETWORK OPTIMIZATION USING A GENETIC ALGORITHM

Optimizing MLP hyperparameters is a challenging task due to the high-dimensional, non-convex nature of the search space. These hyperparameters–including the number of hidden layers, node counts, activation functions, and training parameters–are not independent; rather, they interact in a complex, non-linear fashion [39]. Traditional optimization techniques such as grid search or gradient-based methods often struggle with these challenges, either requiring excessive computational resources or failing to escape local optima in the search space.

To efficiently navigate this optimization problem, we leverage a GA, a global search technique inspired by natural evolution [40]. GA excels in exploring large, non-convex solution spaces by maintaining a diverse population of candidate architectures and evolving them iteratively. Unlike gradient-based methods, GA does not require differentiability, making it well-suited for optimizing discrete and mixed hyperparameters. Additionally, crossover and mutation operators enable GA to generate novel MLP architectures, ensuring efficient convergence toward an optimal design.

While GA has been successfully applied to classification tasks, as demonstrated in [41], its flexibility allows us to adapt the approach for regression problems. By customizing crossover and mutation mechanisms to handle variable hyperparameter counts, GA becomes a powerful tool for automating MLP architecture optimization, enhancing model performance while eliminating manual tuning efforts. The following GA adaptation ensures effective exploration of the hyperparameter space, leading to robust MLP architectures optimized for regression tasks:

- *Initial population* – We start with a collection of MLP realizations or "individuals" with randomly chosen numbers of epochs, batch sizes, numbers of hidden layers, numbers of neurons in each layer, and types of activation functions.
- *Objective* – The fitness of each MLP is defined as the validation loss from training it on the same dataset containing the RIS BSW amplitudes and corresponding radiation patterns, by evaluating the MSE between the predicted radiation patterns and the actual radiation patterns for a given BSW amplitudes array.
- *Parent selection* – Parents are randomly selected individuals from the population, and their children are reproductions of themselves with probabilities for crossover and mutation.
- *Crossover* – The number of epochs, batch size, and number of hidden layers of the child are each chosen to be equal to the corresponding value of one of the parents with equal probability between the two parents. For each hidden layer, the child will inherit randomly the activation function and the number of nodes from the corresponding layer of either parent. If the number of hidden layers of the parents are different, then there may be a case where the child inherits the larger number of layers, which is best explained by an example. Suppose that Parent A has 5 hidden layers, Parent B has 3 hidden layers, and the child has 5 hidden layers. Hidden layers 1-3 of the child will have randomly chosen parameters between layers 1-3 of each parent. For layers 4-5, the parameters will be randomly chosen between layers 4-5 of Parent A, or layer 3 of Parent B.
- *Mutation* – Only occurs when a child has a larger number of hidden layers than one of the parents. This involves random shuffling of the number of nodes and activation functions between all hidden layers of the child.
- *Evolution* – Only the fittest k individuals from the current population (the original MLPs and the children) move to the next generation, where individuals reproduce again and the cycle repeats. The number k decreases during each evolution step.

The algorithm converges after the populations evolve all the way to a single MLP with the lowest validation loss.

In our simulations, the GA was implemented with randomly chosen parameters for each MLP from the following sets:

- Number of epochs: $\{80, 81, \ldots, 200\}$.
- Batch sizes: $\{32, 64, 128, 256, 512, 1024\}$.
- Number of hidden layers: $\{1, 2, \ldots, 6\}$.
- Number of nodes per hidden layer: $\{64, 128, 256, 512, 1024, 2048\}$.
- Activation functions [42]: {ReLU, PReLU, Sigmoid, Tanh}.

The initial population has R randomly generated individuals. Then, $\frac{R}{2}$ randomly selected pairs of parents are chosen with replacement to have 2 children each. From both the original population and the new children, a population of $k = \frac{R}{2}$ individuals with the lowest validation loss is created. This process is repeated until there is only a single individual left. In our simulations, we chose R to be 64 as a good tradeoff between the number of solutions explored versus the time to convergence. A total of 190 MLPs were trained, and the population evolved over 7 generations until the final MLP was determined.

The MLPs were generated and trained using TensorFlow Keras in Python [43]. All MLPs were trained with an L2 regularization penalty of 5×10^{-7} to improve generalization [44], using the Adam Optimizer [45] and the MSE loss between the actual and predicted scaled power values. The learning rate is reduced by half if the validation loss stops improving after 20 consecutive epochs, and the training is terminated if the validation loss does not improve after 30 consecutive epochs. The training set consisted of the first 90,000 samples from the dataset and the validation set consisted of the last 10,000 samples. The fitness of each MLP was determined by its validation loss.

The final MLP architecture that resulted from the GA optimization has 5 hidden layers with nodes and activation functions as illustrated in Fig. 8. The MLP has a batch size of 128 and goes through 147 training epochs. The final MSE training loss was 7.4851×10^{-4} and the final MSE validation loss was 8.4779×10^{-4}. We note that the converged MLP architecture may be sufficient for this

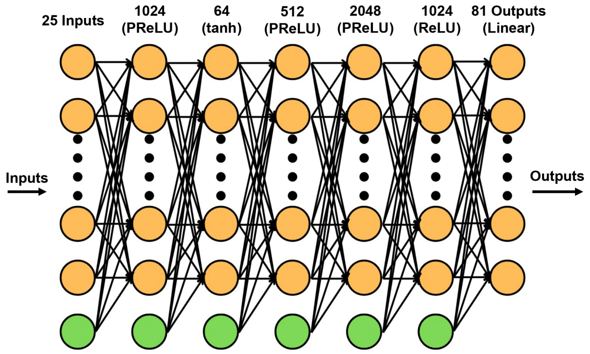


FIGURE 8. Illustration of the proposed NN, including bias nodes (green) in the input layer and each hidden layer. The number and type of activation functions in each layer are given above each layer. The black arrows correspond to the weights between neurons. Inputs flow from the left, through the neurons in the hidden layers of the network, to the outputs on the right.

implementation of the wave-controlled RIS with rectifiers, $N = 25$ BSWs, $M = 100$ elements, and the channel models described in Section II. For other channel models or RIS architectures, an MLP that effectively generalizes the mapping between BSW amplitudes and received powers will likely be different. This highlights the importance of automating MLP parameter tuning using the prescribed GA.

IV. OFFLINE OPTIMIZATION USING SIMULATED ANNEALING

A. MACHINE LEARNING MODEL PERFORMANCE

The NN models the system as the mapping $\mathbb{R}^N \to \mathbb{R}^{n_{\text{angles}}}$ denoted by the nonlinear function $\hat{f}(\boldsymbol{W}) = \hat{\boldsymbol{P}}$, where it is desired that $\hat{\boldsymbol{P}} \approx \boldsymbol{P}$, as illustrated in Fig. 7. We are interested in offline optimization of the BSW amplitudes to configure the RIS to produce a desired radiation pattern. For this optimization, SA [46] is a reasonable choice, given our prior success using it to create SLNR-optimized radiation patterns for the wave-controlled RIS implemented with the rectifiers [5]. In this case, the NN can be used to estimate the powers sampled at specific directions after each update of $\boldsymbol{W}$ as the SA converges towards improved SLNR values. Once the algorithm converges to a particular set of optimal BSW amplitudes, these amplitudes can be used to reconfigure the physical RIS and generate the optimal radiation pattern.

We used a similar SA algorithm as in [5] to determine the BSW amplitudes that optimize the SLNR, as outlined in Algorithm 2. This is a stochastic optimization that randomly perturbs the set of BSW amplitudes as it searches for improved SLNR. The algorithm starts at a "high energy" state corresponding to a maximum temperature parameter T, in which it is more likely to explore different BSW amplitudes even if they do not necessarily produce better SLNR values. This ultimately helps in escaping local minima. As the algorithm converges over multiple iterations, it moves to a "lower energy" state with a low value for T, where it is less likely to explore solutions with lower SLNR. The likelihood of moving from the current set of BSW amplitudes $\boldsymbol{W}$ to the next one $\boldsymbol{W}_{\text{new}}$ is given according to a probability measure

Algorithm 2 Simulated Annealing With Neural Network

```
 1: $\boldsymbol{W} \leftarrow [0, 0, \ldots, 0]^T$
 2: $\boldsymbol{W}_{\text{best}} \leftarrow \boldsymbol{W}$
 3: $i_{\text{best}} \leftarrow 0$
 4: $\text{SLNR}_{\text{current}} \leftarrow$ SLNR calculated from (12) by forward-propagating $\boldsymbol{W}$ through the NN.
 5: $\text{SLNR}_{\text{best}} \leftarrow \text{SLNR}_{\text{current}}$
 6: while $i < i_{\max}$ do
 7:     if $i - i_{\text{best}} \geq i_{\text{rst}}$ then
 8:         $\boldsymbol{W} \leftarrow \boldsymbol{W}_{\text{best}}$
 9:         $i_{\text{best}} \leftarrow i$
10:         $\text{SLNR}_{\text{current}} \leftarrow \text{SLNR}_{\text{best}}$
11:     end if
12:     $T \leftarrow T_{\text{scale}}\left(1 - \frac{i}{i_{\max}}\right)$
13:     $\boldsymbol{W}_{\text{new}}(n) \leftarrow |\boldsymbol{W}(n) + \lambda\epsilon|, \quad n = 1, 2, \ldots, N$
14:     Calculate varactor DC voltages using (4).
15:     if DC biases exceed varactor limits then
16:         Increment $i$, go to line 6.
17:     end if
18:     $\text{SLNR}_{\text{new}} \leftarrow$ SLNR calculated from (12) by forward-propagating $\boldsymbol{W}_{\text{new}}$ through the NN.
19:     if $\text{SLNR}_{\text{new}} > \text{SLNR}_{\text{best}}$ then
20:         $\text{SLNR}_{\text{current}} \leftarrow \text{SLNR}_{\text{new}}$
21:         $\text{SLNR}_{\text{best}} \leftarrow \text{SLNR}_{\text{new}}$
22:         $\boldsymbol{W} \leftarrow \boldsymbol{W}_{\text{new}}$
23:         $\boldsymbol{W}_{\text{best}} \leftarrow \boldsymbol{W}_{\text{new}}$
24:         $i_{\text{best}} \leftarrow i$
25:     else
26:         Calculate $p$ using (14)
27:         if $p \geq \text{rand}(1)$ then
28:             $\boldsymbol{W} \leftarrow \boldsymbol{W}_{\text{new}}$
29:             $\text{SLNR}_{\text{current}} \leftarrow \text{SLNR}_{\text{new}}$
30:         end if
31:     end if
32: end while
```

$$p = \begin{cases} 1 & \text{SLNR}_{\text{new}} > \text{SLNR}_{\text{current}} \\ e^{\left(-\frac{\text{SLNR}-\text{SLNR}_{\text{new}}}{k_c T}\right)} & \text{SLNR}_{\text{new}} \leq \text{SLNR}_{\text{current}}, \end{cases} \tag{14}$$

which depends on the SLNR corresponding to $\boldsymbol{W}$, the new SLNR corresponding to $\boldsymbol{W}_{\text{new}}$, the temperature T, and a cooling factor constant k_c. The advantage of the SA algorithm is that it is not computationally intensive – the random perturbations are simple addition and subtraction operations, and the SLNR values are calculated from the forward pass of the BSW amplitudes through the neural network during inference. The algorithm is initialized with $\boldsymbol{W} = [0, 0, \ldots, 0]^T$.

Note that the SLNR optimization in this system is constrained by the allowed range of DC biasing voltages to the varactors (in this case, between [4 V and 15 V]) given by the peak-detection-over-time expression in (4). This

TABLE 2. SLNR [dB] after convergence of simulated annealing for the NN system and the simulated system.

					NN System				Simulated System			
Beam 1	Beam 2	Null 1	Null 2		Avg	Min	Max	Std. dev	Avg	Min	Max	Std. dev
-40.5°	-	-	-	Real	27.8578	27.2220	28.6893	0.5420	31.7187	30.5748	33.6562	0.8947
				Estimated	28.8694	27.8159	30.1901	0.8664	-	-	-	-
25.5°	-	-	-	Real	32.7547	32.3752	33.9274	0.5156	33.6853	32.5867	34.3495	0.6721
				Estimated	36.1271	35.2647	36.5626	0.3988	-	-	-	-
-30°	-	-49.5°	-	Real	27.7722	24.0745	31.4893	3.3380	29.9627	25.6659	33.0271	2.1174
				Estimated	29.0995	25.0857	32.5343	3.2351	-	-	-	-
-19.5°	-49.5°	-	-	Real	26.5683	22.9833	28.4797	2.3155	27.6214	25.8481	28.6254	0.9598
				Estimated	27.5753	24.1129	29.6426	2.2093	-	-	-	-
-19.5°	49.5°	10.5°	-40.5°	Real	19.3474	14.6896	24.4148	3.4994	21.4651	17.0254	24.7522	2.2578
				Estimated	20.7587	15.0296	24.4111	3.9277	-	-	-	-
30°	-	0°	-	Real	4.3228	2.1095	5.6858	1.0251	6.4071	2.3809	9.7270	2.5564
				Estimated	6.0771	5.1976	7.5983	0.6638	-	-	-	-

constraint can be estimated by applying each update of $\boldsymbol{W}$ during each iteration of SA into the same expression before evaluating the new SLNR. Since this expression does not account for nonlinearities or frequency-dependent losses along the transmission line, it can be used as a worst-case metric to determine whether the varactor biasing limits may be exceeded.

We compare the performance of SA executed offline by inference using the MLP for $\boldsymbol{W}$ during each iteration estimate the power values, against the performance of SA executed in the deterministic simulation, where $\boldsymbol{W}$ excites the RIS and the exact simulated power values at the Rx are calculated at each iteration. The first system will be referred to as the "NN System" and the second system will be referred to as the "Simulated System." The SA algorithms in both cases have the same hyperparameters:

- Cooling factor: $k_c = 0.002$.
- Maximum number of iterations: $i_{\max} = 2000$.
- Random perturbations ϵ for each BSW amplitude chosen from $\epsilon \sim \mathcal{N}(0, 1)$.
- Scaling factor for random steps: $\lambda = 0.015$.
- Maximum number of iterations between current SLNR and previous best SLNR: $i_{\text{rst}} = 200$.
- Temperature scaling factor: $T_{\text{scale}} = 100$.

The major difference between the two is that for the NN system, only positive BSW amplitudes are used because the network was trained only on positive amplitudes, whereas for the simulated system, both positive and negative amplitudes are used, to allow another degree of freedom that the NN does not have. This provides a fair comparison between the beam patterns achievable by each system, without limiting the simulated system to the same constraint that the NN system has. We studied multiple scenarios with single beams, multiple beams, and combinations of beams and nulls, and evaluated the performance of each system in terms of SLNR after convergence, assuming a noise variance of $\sigma_s^2 = 1$ in (12). Table 2 compares the SA performance using feedback from the NN versus using direct Rx feedback from the simulated setup where the RIS response and channel models are exact. The final results are divided into two categories: "Estimated," which reports the final SLNR value that SA in the NN system converged to, and "Real," which is the SLNR found by the deterministic simulation using the optimal $\boldsymbol{W}$ arrays that SA in each system converged to.

We note that the simulated average SLNR in all cases is quite close between the offline optimization using the NN and the online optimization from the simulation, with the worst matching having around 4 dB of difference between the average SLNRs of the two. This indicates that the MLP was able to generalize the true mapping between BSW amplitudes and radiation patterns to high accuracy. Even for cases where the objective function includes forming deep nulls, which are sensitive to small changes in the reflection coefficients, the MLP nearly matches the performance of the simulated scenario.

Note that the cases illustrated in Table 2 only involve angles on which the MLP was trained. It would also be useful to generalize the algorithm to create beams and nulls in any arbitrary direction within the sampled range.

B. ANGLE INTERPOLATION

When uniformly sampling the radiation pattern to create the initial dataset, we used the 3 dB beamwidth as the criterion for maximum angular separation between any two sampled Rx directions to reconstruct the radiation pattern. As stated previously, this helps with beamforming and nullforming at Rx directions between any two neighboring sampled directions, because attempting to change the power directed towards one direction will significantly affect the power directed toward its nearby directions within a beamwidth. We can estimate the power values at Rx directions between the discrete samples through linear interpolation, as follows.

We define the floating-point index of each angle of interest θ^* as

$$i = \frac{\theta^* - \theta_{\min}}{\theta_{\max} - \theta_{\min}} \times \left(n_{\text{angles}} - 1\right), \tag{15}$$

TABLE 3. SLNR [dB] after convergence of simulated annealing for the NN system and the simulated system for arbitrary directions.

					NN System				Simulated System			
Beam 1	Beam 2	Null 1	Null 2		Avg	Min	Max	Std. dev	Avg	Min	Max	Std. dev
-35°	-	-50°	-	Real	25.4248	11.0920	30.4218	5.8103	29.2951	26.4624	31.4586	1.6349
				Estimated	25.7507	13.4766	30.3057	5.0270	-	-	-	-
-20°	50°	10°	-40°	Real	5.7627	2.1418	12.4138	3.4949	24.4674	20.9093	27.9945	2.2030
				Estimated	14.0304	11.2299	18.1563	2.3760	-	-	-	-

where $\theta_{\min}$ is the smallest (most negative) angle sampled in the training dataset (in this case, $-60°$) and $\theta_{\max}$ is the largest (most positive) angle in the training dataset (60°). Since the powers are sampled at discrete angles represented by integer indices in the interval $[0, n_{\text{angles}} - 1]$, we calculate the fractional part of i as

$$\Delta_i = i - \lfloor i \rfloor. \tag{16}$$

Finally, we define P_{θ^-} and P_{θ^+} as the powers sampled at the angles corresponding to $\lfloor i \rfloor$ and $\lceil i \rceil$, respectively. When evaluating the SLNR measure, we calculate the power at interpolated directions using the convex combination

$$P_{\theta^*} = (1 - \Delta_i)P_{\theta^-} + \Delta_i P_{\theta^+}. \tag{17}$$

We simulated two cases involving beam- and nullforming using the same SA algorithm as before, except that the power calculated in each direction by the MLP in the NN system is defined as in (17). The simulated system is allowed to sample the power in any given direction rather than being confined to a discrete grid. The results are summarized in Table 3 averaged over 10 simulations. We note that the MLP performance is degraded in this case, although it is able to create both beams and nulls. This degradation can be seen both by the low mean SLNRs and the high standard deviations. The primary reason for this degradation is that the SLNR objective function becomes too complicated for SA to maximize when we start with an empty $\boldsymbol{W}$ vector, since it must account for more directions in both beamforming and nullforming. In the next section we consider the use of a lookup table to improve algorithm convergence, by storing optimal $\boldsymbol{W}$ vectors in memory and using them as initialization for SA for more complex cases.

V. ADAPTIVE OPTIMIZATION USING A LOOKUP TABLE

A lookup table can be used to store BSW amplitude arrays that correspond to radiation patterns that were already optimized for different cases. This serves two main purposes: First, the optimal arrays can be used to configure the RIS instantly to create desired radiation patterns that have already been generated in the past, thus requiring fewer executions of the SA algorithm. Second, the optimal arrays can be used to optimize related, but more complex radiation patterns, by being the initial arrays used in the SA algorithm. For example, if there was a requirement in the past to generate a beam at direction θ_1, the same BSW amplitudes that were designed to create this beam can be used as the baseline to produce a radiation pattern with a beam at θ_1 and another beam at θ_2. This approach greatly helps with convergence as multiple beams and/or multiple nulls are desired, as demonstrated here.

Algorithm 3 Adaptive Optimization and Lookup Table Construction

```
1:  $P_{\max} \leftarrow -\infty$
2:  $\boldsymbol{W} \leftarrow [0, 0, \ldots, 0]^T$
3:  if lookup table does not contain any of $\boldsymbol{\theta}_{\text{beam}}$ then
4:      for each $\theta_b$ in $\boldsymbol{\theta}_{\text{beam}}$ do
5:          $P_{\theta_b} \leftarrow$ max power (17) of $\theta_b$ in training dataset.
6:          if $P_{\theta_b} > P_{\max}$ then
7:              $P_{\max} \leftarrow P_{\theta_b}$
8:              $\boldsymbol{W} \leftarrow \boldsymbol{W}_b$ (set of BSW amplitudes corresponding to $P_{\theta_b}$ from dataset).
9:          end if
10:     end for
11:     Define an empty objective function $\mathcal{C}$ based on the SLNR measure from (12), initially not taking into account any beam or null directions.
12:     for each $\theta_b$ in $\boldsymbol{\theta}_{\text{beam}}$ do
13:         Update $\mathcal{C}$ to include the SNR corresponding to beam direction $\theta_b$ in the numerator of (12).
14:         Optimize $\boldsymbol{W}$ using SA to maximize $\mathcal{C}$.
15:         Store $\boldsymbol{W}$, beam directions from $\mathcal{C}$, and SLNR in the lookup table.
16:     end for
17:     Update $\mathcal{C}$ to include the SNRs corresponding to all null directions $\boldsymbol{\theta}_{\text{null}}$, if any, in the denominator of (12).
18:     Optimize $\boldsymbol{W}$ using SA to maximize $\mathcal{C}$.
19:     Store $\boldsymbol{W}$, beam and null directions from $\mathcal{C}$, and SLNR in the lookup table.
20: else
21:     Find the $\boldsymbol{W}$ that creates peaks in the directions from the largest subset $\boldsymbol{\theta}_{\text{s},b}$ of $\boldsymbol{\theta}_{\text{beam}}$ from the lookup table. If two or more subsets with the same cardinality exist, choose the one with the highest SLNR and initialize $\mathcal{C}$ to the corresponding objective function with its beam and null directions.
22:     Perform the steps in lines 12-19, starting with $\mathcal{C}$ from line 21, iterating through all $\theta_b \in \{\boldsymbol{\theta}_{\text{beam}} \setminus \boldsymbol{\theta}_{\text{s},b}\}$ and then all $\boldsymbol{\theta}_{\text{null}}$, if any.
23: end if
```

We propose the following approach to construct the lookup table, as outlined in Algorithm 3, where $\boldsymbol{\theta}_{\text{beam}}$ and $\boldsymbol{\theta}_{\text{null}}$ are arrays respectively containing the K and L directions

TABLE 4. SLNR [dB] after convergence of simulated annealing for the NN system using the lookup table.

					NN System			
Beam 1	Beam 2	Null 1	Null 2		Avg	Min	Max	Std. dev
-40.5°	-	-	-	Simulated	33.6676	33.6292	33.7279	0.0361
				Estimated	35.9271	35.8175	36.0131	0.0503
25.5°	-	-	-	Simulated	33.9228	33.8352	33.9872	0.0549
				Estimated	36.9640	36.8946	37.0644	0.0615
-30°	-	-49.5°	-	Simulated	27.6829	26.5275	28.4869	0.6647
				Estimated	34.2191	33.9985	34.4114	0.1206
-19.5°	-49.5°	-	-	Simulated	28.3460	27.2197	28.9696	0.7212
				Estimated	28.6516	27.6357	29.2950	0.6165
-19.5°	49.5°	10.5°	-40.5°	Simulated	23.0232	20.5317	25.1426	1.6340
				Estimated	27.9592	27.6960	28.3134	0.2082
30°	-	0°	-	Simulated	5.3311	4.7714	5.9562	0.3877
				Estimated	7.5584	7.4377	7.6830	0.0720
-35°	-	-50°	-	Simulated	31.4067	29.3688	32.9923	1.2463
				Estimated	32.4866	32.3519	32.6637	0.1030
-20°	50°	10°	-40°	Simulated	25.4036	21.8868	27.9905	1.7480
				Estimated	26.1828	25.2871	26.7366	0.4128

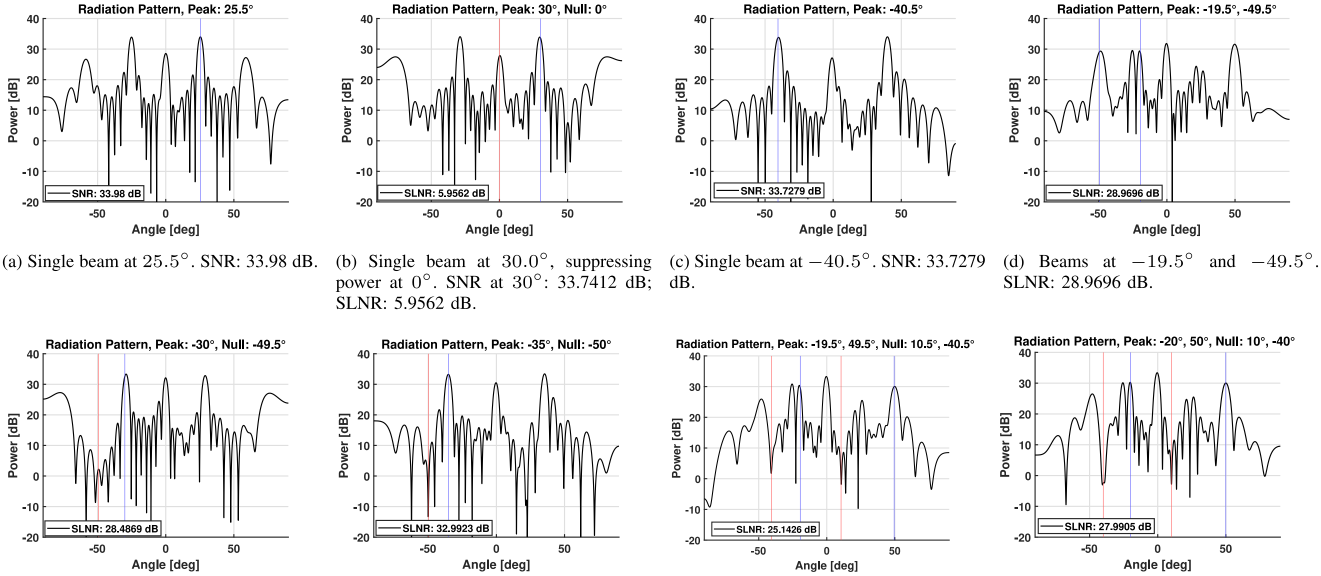
in which beams and nulls are to be created. Assume for example that we wish to create a single beam in a specific direction. Given the large dataset used for training, which is the collection of random $\boldsymbol{W}$ vectors and their corresponding sampled power values, we find the BSW amplitude input $\boldsymbol{W}$ that corresponds to the highest power ($P_{\max}$) in the same direction as the desired beam, and we use $\boldsymbol{W}$ as the initialization for SA. After SA converges, we store the resulting $\boldsymbol{W}$ in the lookup table with its corresponding beam direction. To create beams in two directions, we repeat the same procedure – find the strongest beam between the two in the lookup table and use this result as the basis to design the second beam using SA. The same procedure can be generalized for larger numbers of beams as well. When forming both beams and nulls, we repeat the same procedure as above to create all the beams first, and then create all the nulls in the last optimization steps. The table may contain an entry of the form:

Beams: $\{\theta_{\text{b1}}, \theta_{\text{b2}}, \ldots\}$; **Nulls**: $\{\theta_{\text{n1}}, \theta_{\text{n2}}, \ldots\}$; **SLNR**: y; $\boldsymbol{W}$.

We analyze the performance of SA assisted by the lookup table in Table 4. It is evident that the starting points created in Algorithm 3 significantly improve the SLNR in nearly all cases, allowing the offline optimization using the NN to achieve superior performance over the online optimization in the simulated system. Furthermore, the standard deviations of the SLNR values are significantly smaller than before, indicating that the lookup table-based initializations reduce the risk that SA will not converge to a good solution. To further illustrate this result, all test cases are plotted in Fig. 9.

It is observed from the plots that the offline and adaptive data-based optimizations successfully create radiation patterns that satisfy the criteria defined by the objective functions, creating peaks and nulls at the desired directions. However, peaks are also created in unintended directions, especially in the broadside (0°) direction and at angles symmetric with the desired beams, which may lead to undesirable behavior in a given environment [47]. This is at least partially due to the use of the SLNR measure, which does not penalize radiation patterns with spurious beams. But a careful choice of the parameters used when creating the training dataset for the machine learning model can reduce such artifacts. For example, taking the case of Fig. 9(b) with a single desired beam at 30°, it was found that setting the dominant BSW amplitude in an optimal $\boldsymbol{W}$ array to 3.5 V yields a higher peak at 30° with more attenuation at 0°, as seen in Fig. 10. The ML model was likely unable to explore this scenario since this value is too close to the maximum value the model was trained on. Thus, there is a low probability of such a high value appearing frequently enough in the dataset for its effects to be accurately modeled. Moreover, by increasing the maximum number of BSW frequencies from $N = 25$ to $N = 50$, more degrees of freedom are available for beam pattern optimization at higher frequencies. This also helps reduce the reflected power of the symmetric beam at $-30°$, as shown in Fig. 11. These observations lead to the following conclusions:

- *The voltage range of the randomly generated BSW amplitudes used for training should be carefully chosen depending on the desired performance of the RIS.* If the requirement from the ML model is to explore a wide range of radiation patterns, one should use relatively low random voltage amplitudes such as those used in this paper to avoid saturating the varactor voltage limits while allowing more BSWs to be activated at once. On the contrary, one can generate input arrays where fewer BSWs are excited, but with higher amplitudes, that will dominate the RIS biasing profile. This approach may also reduce the amount of data required for model training.



(a) Single beam at 25.5°. SNR: 33.98 dB.

(b) Single beam at 30.0°, suppressing power at 0°. SNR at 30°: 33.7412 dB; SLNR: 5.9562 dB.

(c) Single beam at -40.5°. SNR: 33.7279 dB.

(d) Beams at -19.5° and -49.5°. SLNR: 28.9696 dB.

(e) Beam at -30°, null at -49.5°. SLNR: 28.4869 dB.

(f) Beam at -35°, null at -50°. SLNR: 32.9923 dB.

(g) Beams at -19.5° and 49.5°, nulls at 10.5° and -40.5°. SLNR: 25.1426 dB.

(h) Beams at -20° and 50°, nulls at 10° and -40°. SLNR: 27.9905 dB.

FIGURE 9. Radiation patterns generated by the optimized NN, using SA and the lookup table.

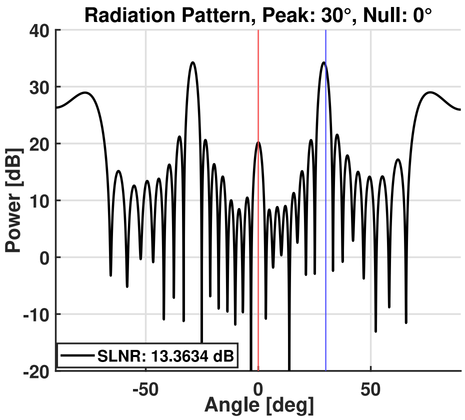


FIGURE 10. Generating a peak at 30° using SA and a dominant BSW at 3.5V. SNR at 30° is 33.5964 dB, peak at 0° is 13.3634 dB lower than the desired peak.

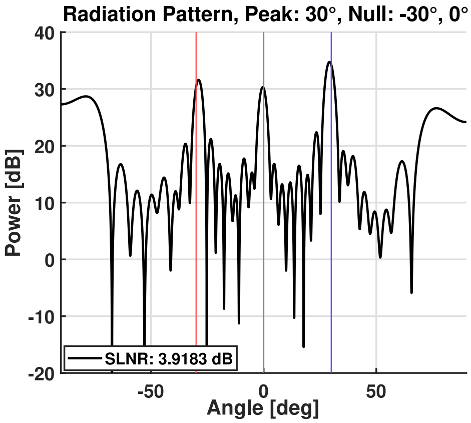


FIGURE 11. Generating a peak at 30° using SA and $N = 50$ BSWs instead of $N = 25$. SNR at 30° is 34.2803 dB, symmetric peak at −30° is attenuated by 3.9183 dB.

- *The number of BSWs and their frequencies should be chosen depending on the hardware constraints and the complexity of the system model.* Using more BSWs creates a tradeoff between the increased control over the radiation patterns generated by the RIS, and the higher-dimensional optimization problems, potentially increased amount of data collection required when generating an ML model, hardware complexity of the RIS, and control signal overhead.

An inherent weakness of using a lookup table to store optimal RIS configurations is that its size increases exponentially with the number of beam pattern constraints. Possible solutions to this problem include interpolation between configurations in the lookup table, using for example manifold methods [48], empirical interpolation [49], or additional ML models to reduce the memory requirements of the lookup table and the required number of SA executions for beam pattern optimization.

VI. CONCLUSION AND FURTHER RESEARCH

We have presented a novel data-driven approach for optimizing wave-controlled RIS without requiring extensive modeling of the RIS behavior or channel state information. The RIS is simulated in an environment where the channel models are unknown. It is excited with random combinations of BSW amplitudes, and received powers are sampled at various directions of interest for each set of input BSWs. An NN that generalizes the relationship between the BSW amplitudes and the corresponding radiation patterns is designed and optimized using a GA, achieving low training and validation losses on the collected data. SA is then used for offline beam pattern optimization using the NN-generated model. The optimal BSW weights are then used to excite the simulated RIS to create the desired radiation patterns. In conjunction, a lookup table is used to store optimal sets of BSW amplitudes to rapidly determine the biasing voltages for standard beam patterns and to create baselines for more complicated radiation patterns. The combination of the lookup table and SA was demonstrated to be capable of achieving superior performance in terms of SLNR in both sampled and interpolated directions via offline optimization using the NN, compared to simulated scenarios where live feedback from the receivers is used to determine convergence for SA.

The GA that was used in this paper to optimize the NN architecture for RIS modeling is applicable to many other optimization and modeling problems. The optimal NN architecture created for our application resembles an autoencoder that learns the general features from the BSW amplitude combinations in the first few hidden layers and maps them to the resulting radiation pattern as the information propagates through subsequent layers. Designing a good NN architecture to model high-dimensional nonlinear mappings is often not an intuitive or a straightforward task, requiring significant tuning and verification. This is especially crucial for regression problems where the model must generalize well for both the training and validation data. The GA was proven to be effective in automating the tuning process and determining system parameters that allow the NN to accurately represent the given model. Overall, the combination of backpropagation, GA, and SA was shown to lead to desirable RIS performance in the absence of an accurate physical model for the RIS or the environment, and provides a promising avenue for other RIS-related applications.

We would like to reiterate that a main novelty in this paper is the fact its algorithms are specifically designed for a wave-controlled RIS. On the other hand, the algorithms presented in this paper that specifically employ synthesis of BSW amplitudes, namely Algorithm 1 (Dataset Generation) and Algorithm 2 (Simulated Annealing), do not have to strictly use BSW amplitudes as the optimization parameters. The dataset generation could be different depending on choice of parameters, such as varactor voltage biases or reflection coefficients instead of BSW amplitudes and modified depending on the use-case (e.g., using a codebook or randomly generating the feature vectors). SA can apply to other objective functions instead of SLNR and optimize for other parameters as well.

Future work will also discuss the use of the data-driven approach presented in this work with the sample-and-hold implementation of the wave-controlled RIS described in [5]. An extension to the conventional RIS structure we discussed is stackable RIS systems, sometimes called Stacked Intelligent Metasurfaces (SIMs) [50], [51]. They are an advanced form of RIS that consist of multiple layers of programmable metasurfaces. SIMs are capable of performing more complex and powerful manipulation of electromagnetic waves compared to single-layer RIS which results in advanced signal processing, beamforming and beam steering, enhanced coverage and throughput, increased energy efficiency, etc. Employing our proposed wave control on stacked metasurfaces can be very advantageous because of the simplified hardware requirement; the control wiring of each element on each metasurface can be substituted by only one biasing TL for each metasurface. Biasing TLs could be placed in multiple layers below the stacked metasurfaces or next to each other in the same layer with DC-supporting vias connected to the elements of each metasurface. The presence of ground planes under the stacked metasurfaces (to separate the biasing TLs) would require using the multilayer RIS in reflection mode. When multiple biasing TLs are used, particular attention should be paid to be sure that the biasing signals do not interfere with each other, by either using ground plane separations or by using electromagnetic barriers made of shielding vias (like metallic pins) between the biasing TLs on the same layer to improve isolation. Further extensions to the basic RIS system are under consideration, such as Simultaneously Transmitting and Reflecting RISs (STAR-RISs) and other novel ways to extend the bandwidth of RIS structures. The use of wave control for these applications can be a new area of research.

REFERENCES

[1] C. Pan et al., “Reconfigurable intelligent surfaces for 6G systems: Principles, applications, and research directions,” *IEEE Commun. Mag.*, vol. 59, no. 6, pp. 14–20, Jun. 2021.

[2] F. Capolino, A. Vallecchi, and M. Albani, “Equivalent transmission line model with a lumped X-circuit for a metalayer made of pairs of planar conductors,” *IEEE Trans. Antennas Propag.*, vol. 61, no. 2, pp. 852–861, Feb. 2013.

[3] D. Hanna, M. S. Melo, F. Shan, and F. Capolino, “A versatile polynomial model for reflection by a reflective intelligent surface with varactors,” in *Proc. IEEE Int. Symp. Antennas Propagat. USNC-URSI Radio Sci. Meeting (AP-S/URSI)*, 2022, pp. 679–680.

[4] E. Ayanoglu, F. Capolino, and A. L. Swindlehurst, “Wave-controlled metasurface-based reconfigurable intelligent surfaces,” *IEEE Wireless Commun.*, vol. 29, no. 4, pp. 86–92, Aug. 2022.

[5] G. Ben-Itzhak, M. Saavedra-Melo, B. Bradshaw, E. Ayanoglu, F. Capolino, and A. Lee Swindlehurst, “Design and operation principles of a wave-controlled reconfigurable intelligent surface,” *IEEE Open J. Commun. Soc.*, vol. 5, pp. 7730–7751, 2024.

[6] H. Zhou, M. Erol-Kantarci, Y. Liu, and H. V. Poor, “A survey on model-based, heuristic, and machine learning optimization approaches in RIS-aided wireless networks,” *IEEE Commun. Surveys Tuts.*, vol. 26, no. 2, pp. 781–823, 2nd Quart., 2024.

[7] G. Zhou, C. Pan, H. Ren, P. Popovski, and A. L. Swindlehurst, “Channel estimation for RIS-aided multiuser Millimeter-wave systems,” *IEEE Trans. Signal Process.*, vol. 70, pp. 1478–1492, Mar. 2022.

[8] C. Pan et al., “An overview of signal processing techniques for RIS/IRS-aided wireless systems,” *IEEE J. Sel. Topics Signal Process.*, vol. 16, no. 5, pp. 883–917, Aug. 2022.

[9] X. Hu, C. Masouros, and K.-K. Wong, “Reconfigurable intelligent surface aided mobile edge computing: From optimization-based to location-only learning-based solutions,” *IEEE Trans. Commun.*, vol. 69, no. 6, pp. 3709–3725, Jun. 2021.

[10] Y. Song, M. R. A. Khandaker, F. Tariq, K.-K. Wong, and A. Toding, “Truly intelligent reflecting surface-aided secure communication using deep learning,” in *Proc. IEEE 93rd Veh. Technol. Conf. (VTC)*, 2021, pp. 1–6.

[11] J. Gao, C. Zhong, X. Chen, H. Lin, and Z. Zhang, “Unsupervised learning for passive beamforming,” *IEEE Commun. Lett.*, vol. 24, no. 5, pp. 1052–1056, May 2020.

[12] B. Sheen, J. Yang, X. Feng, and M. M. U. Chowdhury, "A deep learning based modeling of reconfigurable intelligent surface assisted wireless communications for phase shift configuration," *IEEE Open J. Commun. Soc.*, vol. 2, pp. 262–272, 2021.

[13] J. An et al., "Codebook-based solutions for reconfigurable intelligent surfaces and their open challenges," *IEEE Wireless Commun.*, vol. 31, no. 2, pp. 134–141, Apr. 2024.

[14] Z. Yu, J. An, L. Gan, H. Li, and S. Chatzinotas, "Weighted codebook scheme for RIS-assisted point-to-point MIMO communications," *IEEE Wireless Commun. Lett.*, vol. 14, no. 5, pp. 1571–1575, May 2025.

[15] J. An, C. Xu, L. Gan, and L. Hanzo, "Low-complexity channel estimation and passive beamforming for RIS-assisted MIMO systems relying on discrete phase shifts," *IEEE Trans. Commun.*, vol. 70, no. 2, pp. 1245–1260, Feb. 2022.

[16] Z. Yahong, L. Cuiran, and L. Yongjie, "Simulated annealing genetic algorithm based RIS phase optimization in the high-speed communication of the railways," in *Proc. 8th Int. Conf. Commun., Image Signal Process. (CCISP)*, 2023, pp. 1–5.

[17] Z. Peng, T. Li, C. Pan, H. Ren, W. Xu, and M. D. Renzo, "Analysis and optimization for RIS-aided multi-pair communications relying on statistical CSI," *IEEE Trans. Veh. Technol.*, vol. 70, no. 4, pp. 3897–3901, Apr. 2021.

[18] M. Saavedra-Melo, K. Rouhi, and F. Capolino, "Wave-controlled RIS: A novel method for reconfigurable elements biasing," in *Proc. IEEE Int. Symp. Antennas Propagat. USNC-URSI Radio Sci. Meeting (USNC-URSI)*, 2023, pp. 979–980.

[19] F. Costa and M. Borgese, "Electromagnetic model of reflective intelligent surfaces," *IEEE Open J. Commun. Soc.*, vol. 2, pp. 1577–1589, 2021.

[20] D. Sievenpiper, L. Zhang, R. F. J. Broas, N. G. Alexopolous, and E. Yablonovitch, "High-impedance electromagnetic surfaces with a forbidden frequency band," *IEEE Trans. Microw. Theory Techn.*, vol. 47, no. 11, pp. 2059–2074, Nov. 1999.

[21] S. R. Best and D. L. Hanna, "Design of a broadband dipole in close proximity to an EBG ground plane," *IEEE Antennas Propag. Mag.*, vol. 50, no. 6, pp. 52–64, Dec. 2008.

[22] E. Basar, M. Di Renzo, J. De Rosny, M. Debbah, M.-S. Alouini, and R. Zhang, "Wireless communications through reconfigurable intelligent surfaces," *IEEE Access*, vol. 7, pp. 116753–116773, 2019.

[23] R. Achar and M. S. Nakhla, "Simulation of high-speed interconnects," *Proc. IEEE*, vol. 89, no. 5, pp. 693–728, May 2001.

[24] A. Sedra, K. Smith, T. Carusone, and V. Gaudet, *Microelectronic Circuits* (Oxford Series in Electrical and Computer Engineering). Oxford, U.K.: Oxford Univ. Press, 2020.

[25] G. Gradoni and M. Di Renzo, "End-to-end mutual coupling aware communication model for reconfigurable intelligent surfaces: An electromagnetic-compliant approach based on mutual impedances," *IEEE Wireless Commun. Lett.*, vol. 10, no. 5, pp. 938–942, May 2021.

[26] M. Yazdi and M. Albooyeh, "Analysis of metasurfaces at oblique incidence," *IEEE Trans. Antennas Propag.*, vol. 65, no. 5, pp. 2397–2404, May 2017.

[27] S. Basharat, S. A. Hassan, H. Pervaiz, A. Mahmood, Z. Ding, and M. Gidlund, "Reconfigurable intelligent surfaces: Potentials, applications, and challenges for 6G wireless networks," *IEEE Wireless Commun.*, vol. 28, no. 6, pp. 184–191, Dec. 2021.

[28] X. Wei, D. Shen, and L. Dai, "Channel estimation for RIS assisted wireless communications—Part I: Fundamentals, solutions, and future opportunities," *IEEE Commun. Lett.*, vol. 25, no. 5, pp. 1398–1402, May 2021.

[29] D. Mishra and H. Johansson, "Channel estimation and low-complexity beamforming design for passive intelligent surface assisted MISO wireless energy transfer," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2019, pp. 4659–4663.

[30] A. L. Swindlehurst, G. Zhou, R. Liu, C. Pan, and M. Li, "Channel estimation with reconfigurable intelligent surfaces—A general framework," *Proc. IEEE*, vol. 110, no. 9, pp. 1312–1338, Sep. 2022.

[31] V. Franc and V. Hlaváč, "Greedy algorithm for a training set reduction in the kernel methods," in *Proc. 10th Int. Conf. Comput. Anal. Images Patterns*, 2003, pp. 426–433.

[32] Y. Sun, P. Babu, and D. P. Palomar, "Majorization-minimization algorithms in signal processing, communications, and machine learning," *IEEE Trans. Signal Process.*, vol. 65, no. 3, pp. 794–816, Feb. 2017.

[33] M.-C. Popescu, V. Balas, L. Perescu-Popescu, and N. Mastorakis, "Multilayer perceptron and neural networks," *WSEAS Trans. Circuits Syst.*, vol. 8, pp. 579–588, Jul. 2009.

[34] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, no. 5, pp. 359–366, 1989.

[35] M. A. J. Idrissi, H. Ramchoun, Y. Ghanou, and M. Ettaouil, "Genetic algorithm for neural network architecture optimization," in *Proc. 3rd Int. Conf. Logist. Opera. Manage. (GOL)*, 2016, pp. 1–4.

[36] H. Ramchoun, Y. Ghanou, M. Ettaouil, and M. A. J. Idrissi, "Multilayer perceptron: Architecture optimization and training," *Int. J. Interact. Multimedia Artif. Intell.*, vol. 4, no. 1, pp. 26–30, Jun. 2016.

[37] H. Taghvaee et al., "Radiation pattern prediction for metasurfaces: A neural network-based approach," *Sensors*, vol. 21, no. 8, p. 2765, 2021.

[38] J. Sola and J. Sevilla, "Importance of input data normalization for the application of neural networks to complex industrial problems," *IEEE Trans. Nucl. Sci.*, vol. 44, no. 3, pp. 1464–1468, Jun. 1997.

[39] L. N. Smith, "A disciplined approach to neural network hyper-parameters: Part 1—Learning rate, batch size, momentum, and weight decay," 2018, *arXiv:1803.09820*.

[40] S. Forrest, "Genetic algorithms," *ACM Comput. Surv.*, vol. 28, no. 1, pp. 77–80, 1996.

[41] J. V. Domashova, S. S. Emtseva, V. S. Fail, and A. S. Gridin, "Selecting an optimal architecture of neural network using genetic algorithm," *Procedia Comput. Sci.*, vol. 190, pp. 263–273, Jan. 2021.

[42] B. Ding, H. Qian, and J. Zhou, "Activation functions and their characteristics in deep neural networks," in *Proc. Chin. Control Decis. Conf. (CCDC)*, 2018, pp. 1836–1841.

[43] M. Abadi et al. "TensorFlow: Large-scale machine learning on heterogeneous systems." 2015. [Online]. Available: https://www.tensorflow.org/

[44] G. Shi, J. Zhang, H. Li, and C. Wang, "Enhance the performance of deep neural networks via L2 regularization on the input of activations," *Neural Process. Lett.*, vol. 50, no. 1, pp. 57–75, Aug. 2019.

[45] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017, *arXiv:1412.6980*.

[46] V. Cerny, "Thermodynamical approach to the Traveling salesman problem: An efficient simulation algorithm," *J. Optim. Theory Appl.*, vol. 45, pp. 41–51, Jan. 1985.

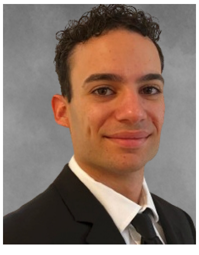
[47] M. Merluzzi and A. Clemente, "Anomalous and specular reflections of reconfigurable intelligent surfaces: Configuration strategies and system performance," *IEEE Wireless Commun. Lett.*, vol. 13, no. 10, pp. 2707–2711, Oct. 2024.

[48] R. Zimmermann, "Manifold interpolation," *Model Order Reduct.*, vol. 1, pp. 229–274, Nov. 2021.

[49] S. Chaturantabut and D. C. Sorensen, "Nonlinear model reduction via discrete empirical interpolation," *SIAM J. Sci. Comput.*, vol. 32, no. 5, pp. 2737–2764, 2010.

[50] J. An et al., "Stacked intelligent metasurface-aided MIMO transceiver design," *IEEE Wireless Commun.*, vol. 31, no. 4, pp. 123–131, Aug. 2024.

[51] J. An et al., "Stacked intelligent metasurfaces for efficient holographic MIMO communications in 6G," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 8, pp. 2380–2396, Aug. 2023.



GAL BEN-ITZHAK (Graduate Student Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from the University of California at Irvine, Irvine, CA, USA, in 2023 and 2024, respectively, where he is currently pursuing the Ph.D. degree. His current research and professional interests include high-speed communications, optimization, digital signal processing, high-speed circuit design, and signal integrity.

MIGUEL SAAVEDRA-MELO (Graduate Student Member, IEEE) received the B.S. and M.S. degrees in electronic engineering from the Universidad Nacional de Colombia, Bogotá, Colombia, in 2014 and 2018, respectively. He is currently pursuing the Ph.D. degree in electrical engineering with the University of California at Irvine, Irvine, CA, USA. His current research interests include electron beam devices, reflective intelligent surfaces, and high-power microwaves.

ENDER AYANOGLU (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, USA, in 1986. He was with the Bell Laboratories Communications Systems Research Laboratory, Holmdel, NJ, USA. From 1999 to 2002, he was a Systems Architect with Cisco Systems Inc., San Jose, CA, USA. Since 2002, he has been a Professor with the Department of Electrical Engineering and Computer Science, University of California at Irvine, Irvine, CA, USA, where he was the Director of the Center for Pervasive Communications and Computing and the Conexant-Broadcom Endowed Chair, from 2002 to 2010. He was a recipient of the IEEE Communications Society Stephen O. Rice Prize Paper Award in 1995, the IEEE Communications Society Best Tutorial Paper Award in 1997, and the IEEE Communications Society Communication Theory Technical Committee Outstanding Service Award in 2014. In 2023, he received the IEEE Communications Society Joseph L. LoCicero Award for outstanding contributions to IEEE Communications Society journals as an Editor, the Editor-in-Chief (EiC), and the Founding EiC. From 2000 to 2001, he was the Founding Chair of the IEEE-ISTO Broadband Wireless Internet Forum, an industry standards organization. He served on the Executive Committee for the IEEE Communications Society Communication Theory Committee from 1990 to 2002, and its Chair from 1999 to 2002. From 1993 to 2014, he was an Editor of IEEE TRANSACTIONS ON COMMUNICATIONS. He was the Editor-in-Chief of IEEE TRANSACTIONS ON COMMUNICATIONS from 2004 to 2008, and the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS-Series on Green Communications and Networking from 2014 to 2016. He was the Founding Editor-in-Chief of IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING from 2016 to 2020. He served as an IEEE Communications Society Distinguished Lecturer from 2022 to 2023 and will serve in the same capacity in 2024-2025.

FILIPPO CAPOLINO (Fellow, IEEE) received the Laurea (cum laude) and Ph.D. degrees in electrical engineering from the University of Florence, Italy, in 1993 and 1997, respectively. From 1997 to 1999, he was a Fulbright and then a Postdoctoral Fellow with the Department of Aerospace and Mechanical Engineering, Boston University, Boston, MA, USA. From 2000 to 2001, part of 2005 and in 2006, he was a Research Assistant Visiting Professor with the Department of Electrical and Computer Engineering, University of Houston, Houston, TX, USA. From 2002 to 2008, he was an Assistant Professor with the Department of Information Engineering, University of Siena, Italy. He has been a Visiting Professor with the Fresnel Institute, Marseille, France, in 2003, and with the Centre de Recherche Paul Pascal, Bordeaux, France, 2010. He is currently a Professor with the Department of Electrical Engineering and Computer Science, University of California at Irvine, Irvine, CA, USA. In 2022, he held the title of Cathedra of Excellence with the University of Carlos III, Madrid, Spain. He is an Editor of the *Metamaterials Handbook* (Boca Raton, FL, USA: CRC Press, 2009). His current research interests include metamaterials and their applications, antennas and wireless systems, sensors in both microwave and optical ranges, millimeter wave technology, electron beam devices, plasmonics, microscopy, optical devices, and applied electromagnetics in general. He received the R. W. P. King Prize Paper Award from the IEEE Antennas and Propagation Society for the Best Paper of the Year 2000, by an author under 36. He was the Founder and an EU Coordinator of the EU Doctoral Programs on Metamaterials from 2004 to 2009. From 2002 to 2008, he served as an Associate Editor for the IEEE TRANSACTIONS ON ANTENNAS AND PROPAGATION. He is a Fellow of OPTICA.

A. LEE SWINDLEHURST (Fellow, IEEE) received the B.S. and M.S. degrees in electrical engineering from Brigham Young University (BYU) in 1985 and 1986, respectively, and the Ph.D. degree in electrical engineering from Stanford University in 1991. He was with the Department of Electrical and Computer Engineering, BYU from 1990 to 2007, where he served as the Department Chair from 2003 to 2006. From 1996 to 1997, he held a joint appointment as a Visiting Scholar with Uppsala University and the Royal Institute of Technology, Sweden. From 2006 to 2007, he was on leave working as the Vice President of Research with ArrayComm LLC, San Jose, CA, USA. Since 2007, he has been with the Electrical Engineering and Computer Science Department, University of California Irvine, where he is a Distinguished Professor and currently serving as the Department Chair. His research focuses on array signal processing for radar, wireless communications, and biomedical applications. He received the 2000 IEEE W. R. G. Baker Prize Paper Award, the 2006 IEEE Communications Society Stephen O. Rice Prize in the Field of Communication Theory, the 2006, 2010, and 2021 IEEE Signal Processing Society's Best Paper Awards, the 2017 IEEE Signal Processing Society Donald G. Fink Overview Paper Award, the Best Paper award at the 2020 and 2024 IEEE International Conferences on Communications, the 2022 Claude Shannon-Harry Nyquist Technical Achievement Award from the IEEE Signal Processing Society, and the 2024 Fred W. Ellersick Prize from the IEEE Communications Society. From 2014 to 2017, he was also a Hans Fischer Senior Fellow in the Institute for Advanced Studies at the Technical University of Munich, and in 2016, he was elected as a Foreign Member of the Royal Swedish Academy of Engineering Sciences (IVA).