IEEE Open Journal of the
Communications Society

Received 14 April 2021; revised 6 May 2021; accepted 8 May 2021. Date of publication 14 May 2021; date of current version 25 May 2021.

Digital Object Identifier 10.1109/OJCOMS.2021.3080618

RF Canceller Tuning Acceleration Using Neural Network Machine Learning for In-Band Full-Duplex Systems

KENNETH E. KOLODZIEJ (Member, IEEE), AIDAN U. COOKSON, AND BRADLEY T. PERRY (Member, IEEE)

RF Technology Group, Advanced Technology Division, MIT Lincoln Laboratory, Lexington, MA 02421, USA

CORRESPONDING AUTHOR: K. E. KOLODZIEJ (e-mail: kenneth.kolodziej@ll.mit.edu)

This work was supported by United States Air Force through Air Force under Contract FA8702-15-D-0001.

ABSTRACT The fifth-generation wireless system framework provides the option to evaluate the performance of in-band full-duplex (IBFD) operation through flexible duplexing. The resulting self-interference, however, must be mitigated within a fraction of a symbol duration for successful communication. This paper introduces the use of neural network machine learning to accelerate the tuning of multi-tap adaptive RF cancellers. Additionally, the optimal network configurations, input data structures and training dataset densities that optimize the performance of this technique are presented. The tuning results of a prototype system using a two-tap canceller were measured over 20 and 100 MHz bandwidths centered at 2.5 GHz, and demonstrated averages of 40 dB cancellation and 6 tuning iterations. These results are compared to a survey of previously-reported adaptive cancellers, and illustrate that this novel application of machine learning to RF canceller tuning provides the fastest convergence speed to date, which can enable IBFD operation in dynamic interference environments.

INDEX TERMS 5G mobile communication, in-band full-duplex, machine learning, RF cancellation, self-interference cancellation.

I. INTRODUCTION

WHILE the current fifth-generation new radio (5G NR) wireless system requirements outline the use of both frequency-division duplex (FDD) and time-division duplex (TDD) modes, flexible duplexing has also been built into the specifications. This concept initiated as a way to evolve fourth-generation (4G) Long-Term Evolution (LTE) systems by merging the best aspects of FDD and TDD to enhance network efficiency [1]. In TDD mode, this flexibility relates to dynamically assigning the number of symbols within a timeslot for uplink (transmit for handset) and downlink (receive for handset) within a given frequency, and helps support asymmetric data traffic. This, however, can result in closely-spaced frequency channels being assigned to transmit and receive at the same time, which generates system-level interference [2]. This interference has previously been managed through the use of power control algorithms [3] and device muting [4], both of which are suboptimal approaches.

When independently employing multiple frequency carriers, this flexible duplexing concept can generate scenarios where signals are transmitted and received at the same time within a channel allocation. This creates a unique subset of flexible duplex, which is similar to in-band full-duplex (IBFD), and can allow for network efficiency improvements through interference mitigation at the device level [5]. In order to support this IBFD mode, however, the resulting self-interference (SI) must be sufficiently suppressed within a fraction of the symbol timing using techniques in the propagation, analog and/or digital domains [6], [7]. In addition to studying this possible marriage for 5G NR networks, the benefits of both flexible duplex and IBFD strategies have been independently compared to traditional approaches in [8]. While the flexible duplexing concept has been researched in IBFD systems utilizing an electrical balanced duplexer (EBD) [9] and cross-polarized antennas [10] for SI mitigation, an adaptive analog-RF canceller can also be employed for this purpose, assuming it can be tuned sufficiently quickly.

Many previously-designed RF cancellers have been shown to successfully mitigate SI, but their tuning mechanisms would not be adequate to support the IBFD option within the flexible duplex operation of 5G NR networks. These multi-tap adaptation approaches include manual control [11] as well as standard gradient descent methods with both fixed learning rates [12], [13] and adaptive functions [14], [15]. Additionally, some cancellers have utilized special channel-probing waveforms [16] and received signal strength indicator information (RSSI) [17] for tuning. Others have employed dithered linear search (DLS) approaches [18] and least mean squares (LMS) algorithms [19]–[21]. While all of these RF cancellers provided sufficient cancellation performance using these various tuning techniques, many were not able to converge to their optimal settings in an efficient manner, which could be drastically accelerated through the use of machine learning and neural networks.

While machine learning has been applied to a wide array of fields, it has only recently been incorporated into IBFD systems. Multipath-rich channels have been modeled for the purposes of band-allocation prediction in cognitive radio deployments [22], and much focus has been placed on neural networks reducing computational complexity in nonlinear digital cancellation schemes [23], [24]. The latter has also been extended to a software-defined radio (SDR) platform [25] that has expanded to the use of deep neural networks [26]. Furthermore, studies have been conducted on the effects of real and complex data samples for digital cancellation [27] as well as the hardware resources required for these approaches in comparison to traditional polynomial-based nonlinear cancellation [28]. Additionally, neural networks have been investigated for both reference-based digital cancellation and beamforming operations within an IBFD phased array [29]. Overall, [23]–[29] are similar to each other in that they have only investigated the potential of applying machine learning methods to digital-based SI mitigation, and have not explored their application to techniques in the analog domain. Compared to these digital approaches, analog-RF cancellation provides the benefit of reducing SI before the receiver front-end, but presents the additional challenges of accounting for finite tuning resolution and hardware variations with time and temperature [30].

This paper moves beyond the application of machine learning to purely-digital techniques in IBFD systems, and presents the first research illustrating how analog-RF cancellers can significantly reduce their convergence time by incorporating neural networks into their tuning approaches. After training the network on practical cancellation sequences, the resulting tuning function is able to make predictions that configure the canceller near its optimal settings. This novel incorporation of machine learning to standard tuning approaches helps the canceller significantly reduce convergence time compared to other techniques, and allows it to support the IBFD mode and flexible duplex timing aspects of 5G NR. The tuning evaluation results were collected on a two-tap canceller prototype that was adapted to a multipath SI channel using only the information from the existing system without any additional hardware or probe waveforms.

This work greatly expands on our initial investigation of analog cancellation and machine learning in [31] by presenting the theoretical framework of the neural network, simulation results of various network inputs and structures to determine the optimal configuration, and the first set of statistical measurements that validate the cancellation performance and tuning acceleration. Additionally, we uniquely examine the training sequence sample space in terms of the number of complex data points mapping the SI channel to understand its effect on performance. Finally, we include an expansion of this method from 20 to 100 MHz channel bandwidths, and present a novel comparison of the evaluation results to previously-reported cancellers, indicating that this new tuning approach provides the fastest convergence speed to date. Overall, this is the first-known research to combine machine learning and analog cancellation for IBFD systems, and should help provide a practical path forward to mitigate dynamic SI on short timescales.

The remaining part of the paper is organized as follows: Section II will discuss the description of the system that includes both the flexible duplex concept and SI cancellation concerns, while Section III will focus on the various aspects of the neural network machine learning, including network formulation, training and simulation. The measurement results and the performance survey will be presented in Section IV, and conclusions will be derived in Section V.

II. SYSTEM DESCRIPTION

A. FLEXIBLE DUPLEX

Many legacy wireless networks have been designed to cover large areas with static access points and symmetric communications that utilized spectrally-inefficient FDD for channel access [5]. Recently, the trend has been towards deploying significantly smaller cells that utilize TDD to support asymmetric data traffic for user equipment, such as large file downloads and high-definition video streaming applications that are common on mobile devices.

The frequency-time diagram of Fig. 1 illustrates how flexible duplexing operates within this TDD mode of 5G NR, where each timeslot has 14 symbols that can be assigned to be either uplink (U) or downlink (D). The top part of the figure shows how carrier aggregation (CA) can be utilized to provide higher data capacity using multiple frequency bands, shown as f_3 and f_4. The top half also illustrates how traditional duplexing operates, where both carriers need to select the same U/D configuration for each individual symbol (guard periods have been removed for clarity). While there is some flexibility in U/D time ratio, network efficiency can be enhanced by utilizing the flexible duplex approach depicted in the bottom part of Fig. 1. In this case, carriers f_1 and

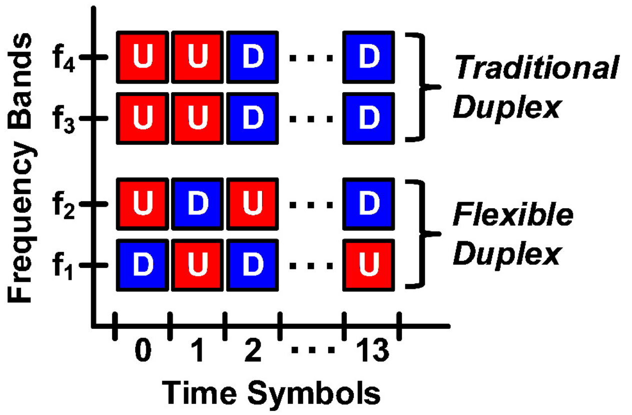


FIGURE 1. Frequency-time diagram showing four carrier frequencies (f_{1-4}) and 14 time symbols that can be assigned to uplink (U) or downlink (D) for either traditional or flexible duplex.

f_2 have been aggregated without any restrictions on their U/D configurations, which offers the possibility of IBFD operation, as mentioned and shown with one carrier transmitting and the other receiving (changing every symbol is not required, but merely illustrates the most extreme tuning case).

In order to implement this flexible IBFD mode, each transceiver can utilize an adaptive RF canceller to remove the SI that results from the adjacent carrier, which is not at the same frequency, but still within the channel bandwidth.

B. SI CANCELLATION

Fig. 2(a) demonstrates how an adaptive canceller that is comprised of two vector modulator (VM) taps can be employed to mitigate the SI coupling channel within a generic transceiver. This interference channel is comprised of both direct path SI as well as dynamic SI that is dependent on the surroundings, and has a composite impulse response that can be modeled as

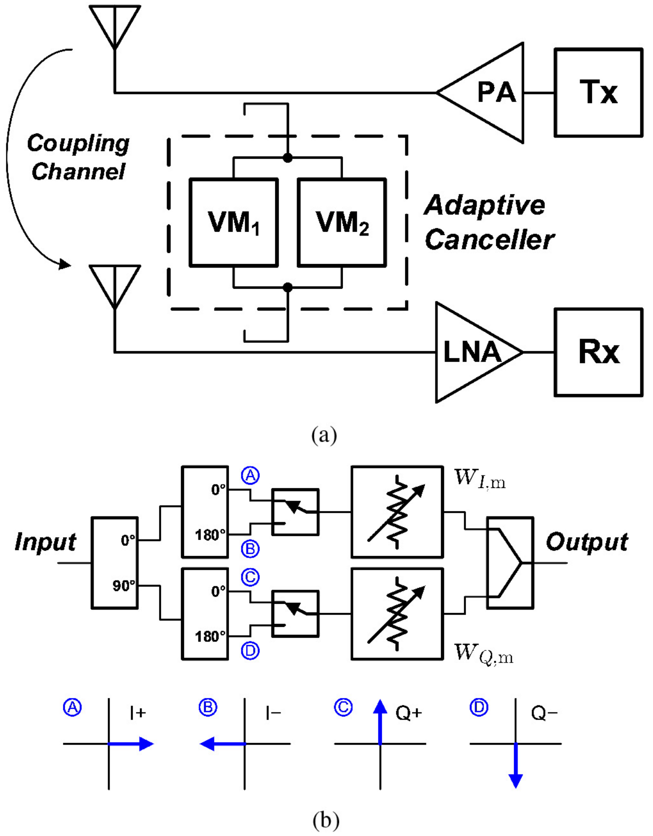
$$h_{SI}(t) = \alpha_0 \delta(t-\tau_0) * h_{\phi_0}(t) + \sum_{k=1}^{K} \alpha_k \delta(t-\tau_k) * h_{\phi_k}(t), \tag{1}$$

where the first term represents the direct path coupling and the second term captures the dynamic SI. The former is dictated by the specific antenna structure, such that α_0 captures the attenuation coefficient of the signal that is delayed by τ_0 and convolved with a phase shift response, $h_{\phi_0}(t)$. The latter multipath signals, however, experience changes based on the environment around the IBFD node that similarly alter their attenuation, phase and time delay responses, as indicated for a total of K reflections.

These key attenuation, phase shift and delay variables can be straightforwardly visualized in the frequency-domain representation of (1), such that

$$H_{SI}(j\omega) = \alpha_0 e^{-j\phi_0} e^{-j\omega\tau_0} + \sum_{k=1}^{K} \alpha_k e^{-j\phi_k} e^{-j\omega\tau_k}, \tag{2}$$

where ϕ denotes the phase shift in radians for $\omega \geq 0$.



(a)

(b)

FIGURE 2. (a) Dual-antenna IBFD system diagram, highlighting the SI coupling channel and adaptive RF canceller with two taps using vector modulators (VMs). (b) VM architecture and internal phase states controlled by tunable weights, w_I and w_Q, for a generic canceller with M taps.

Since the SI responses represented in (1) and (2) contain delays that are dependent on both the antenna and surrounding environment, an RF canceller must provide similar delays to counteract these signals. This is typically accomplished through the use of multiple canceller taps that internally delay the transmit signal to match τ_k for the various SI paths. Furthermore, these taps must provide a means to dynamically change their attenuation and phase to compensate for the unknown SI response characteristics that are location-specific. With these requirements in mind, we utilized a two-tap VM-based canceller architecture for our research, as shown in Fig. 2(a). This diagram illustrates how the canceller signal is coupled off of the transmit output after the power amplifier (PA), which allows for all of the transmit noise and nonlinear distortions to also be captured and cancelled. Additionally, it can be seen that the modified canceller output signal is coupled back into the receiver input before the low-noise amplifier (LNA), which helps minimize receiver saturation and nonlinearities.

The components utilized within the VM are represented in the upper part of Fig. 2(b), where an input signal experiences a series of fixed phase shifts to create the four vectors ($I+$, $I-$, $Q+$, $Q-$) that are indicated in the bottom part of the figure. Subsequently, the following switches select either the positive or the negative vector for both the in-phase (I) and quadrature (Q) channels before they are independently weighted using variable attenuators and then recombined at the output [32]. The combination of this vector selection and weighting allows the VM to generate arbitrary attenuation and phase changes of the input signal, which make it ideal for incorporation into an RF canceller.

For a multi-tap canceller architecture, the complex impulse response can be modeled as

$$h_c(t) = \sum_{m=1}^{M} \beta_m \big(w_{I,m} + jw_{Q,m}\big)\delta(t - \tau_m), \tag{3}$$

where β_m is the attenuation coefficient that captures the fixed insertion loss for each tap, w_I and w_Q are the m-tap tunable weights that can vary between -1 and 1, and τ_m is the corresponding tap time delay. M represents the total number of canceller taps, and should be two or greater to address the multiple components of SI, as described in (1). The frequency response of this VM canceller can be formulated using (3), such that

$$H_c(j\omega) = \sum_{m=1}^{M} \beta_m \Big(w_{I,m}e^{-j\omega\tau_m} + w_{Q,m}e^{j(\pi-\omega\tau_m)}\Big). \tag{4}$$

Equation (4) can now be compared to (2) to better understand that the purpose of the RF canceller is to adapt its response to closely match that of the phase-inverted SI channel so that when combined at the receiver, the signals sum to zero. This is accomplished by tuning the canceller weights, w_I and w_Q, for each tap to precise amounts that are dependent on the IBFD node surroundings. As M increases, the canceller gains the ability to adapt to a more diverse set of SI channels; however, the number of independent weights also increases, which makes efficient canceller tuning a significant challenge. As previously mentioned, the speed requirement on this tuning must be sufficiently high to enable flexible duplexing operation, and can be met through the use of the neural network machine learning approach discussed in the following section.

III. NEURAL NETWORK MACHINE LEARNING

A. TUNING CONSIDERATIONS

The main challenge of tuning any adaptive RF canceller to mitigate SI is that the magnitude and phase of the SI varies over time. This is a result of changes in either the assigned frequency band or the dynamic multipath environment around the wireless node. When a change is detected by the receiver (typically when the SI level rises above a given threshold), the canceller must adapt the weights for each tap to compensate and suppress the SI. This involves finding the optimal w_I and w_Q for each tap as given in (3) and (4), which can be a time-consuming process for a significant change in the interference level.

This level represents the error between the SI channel and canceller responses described in (2) and (4), respectively. When considering the errors for all the possible canceller weight configurations, a multi-dimensional error surface can be generated. The resulting cancellation-error surfaces are non-convex and present nearly-flat gradients, local minima and precise optimal locations, all of which can be demanding on adaptive tuning algorithms, potentially limiting their performance and effectively increasing their tuning time.

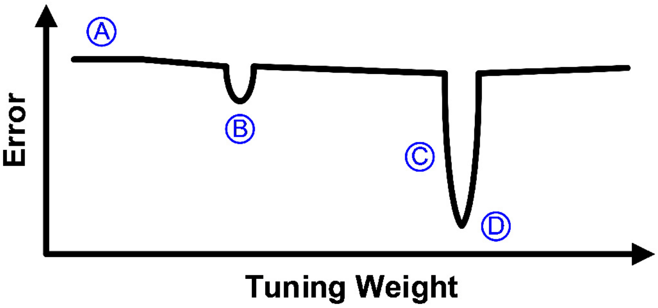


FIGURE 3. Example cancellation error surface that is a function of a single tuning weight for a realistic non-convex scenario.

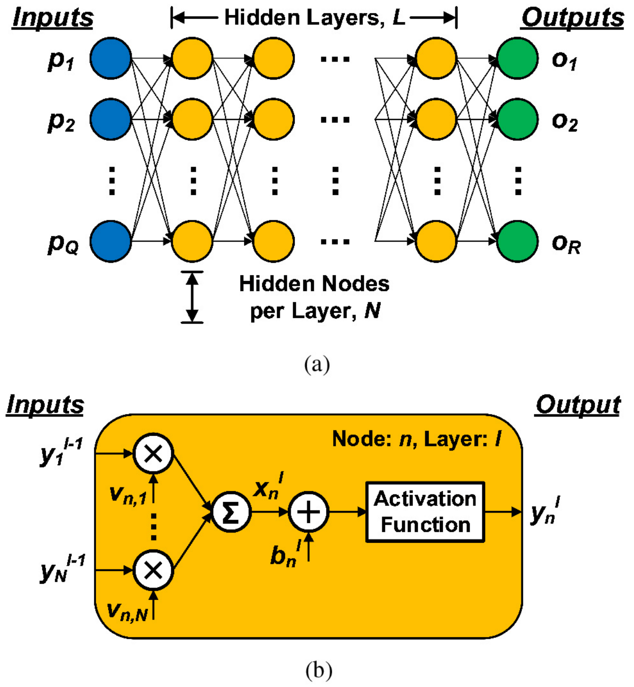
These canceller-tuning challenges are highlighted in Fig. 3, which represents a simplified one-dimensional surface for a single tuning weight. When the canceller is enabled for an unknown SI channel, it selects initial weight settings that are non-optimal (such as point A). An adaptive tuning algorithm must then accelerate its learning rate to avoid local minima (such as point B) before reducing its step size when interesting features (such as point C) are encountered before finally locating the global minimum (such as point D). This tuning can be efficiently accomplished using adaptive algorithms that learn from the error surface and provide canceller weight updates according to according to

$$w_{n+1} = w_n - \left(\frac{E(w)}{E_T} + \frac{1}{n\Big(\overline{|\nabla E(w)|}^2 + \epsilon\Big)}\right) \cdot \nabla E(w), \tag{5}$$

where w is the weight vector that contains w_I and w_Q for both VMs, $E(w)$ is the error on the surface that is a function of the tap weights, E_T is a target error value, n is the iteration number, $\nabla E(w)$ is the gradient of the error, and ϵ avoids division by zero (on the order of 1e−9). While this advanced algorithm has successfully been demonstrated to traverse non-convex error surfaces and accelerate the convergence speed, it still requires many tuning iterations that consume significant portions of a time symbol, thus rendering it insufficient for flexible duplex operation [15]. The robustness and speed enhancement provided by (5), however, can be augmented with machine learning to provide further improvements, as discussed in the next subsection.

B. NETWORK CONFIGURATION

As previously mentioned, the canceller tuning process can be accelerated using machine learning, and more specifically, a feedforward artificial neural network (NN). As seen in Fig. 4(a), a generic NN consists of inputs $\boldsymbol{p}$, and outputs $\boldsymbol{o}$, that can be of different lengths: Q and R, respectively. Between these sets of inputs and outputs exists a network



(a)

(b)

FIGURE 4. (a) Generic neural network diagram with inputs p, outputs o, hiddens layers L and hidden nodes per layer N. (b) Detailed neuron diagram indicating the input weighting v, bias constant b and activation function for node n on layer l.

of nodes, or neurons, that are "hidden" from the user and are used to create interconnects that model nonlinear system behavior for complex datasets. The accuracy of these models depends on both the number of hidden layers, L, and hidden nodes per layer, N, which can vary from layer to layer, but were assumed to be the same for all layers in this work, as is most common.

Within the network, each node is connected in a forward-propagating fashion, as indicated by the arrows. What is not shown in Fig. 4(a) is that each node's input (every arrow) is assigned some weighting value so that new inputs can propagate through the network to produce relevant outputs. This fact is illustrated in the detailed neuron diagram of Fig. 4(b) for a given node n on layer l. The inputs to this node are the outputs from all N nodes on the previous layer, denoted y_1^{l-1} through y_N^{l-1}. These inputs are multiplied by the corresponding node weights, $v_{n,1}$ through $v_{n,N}$, and summed to form

$$x_n^l = \sum_{i=1}^{N} v_{n,i}^l \, y_i^{l-1}, \tag{6}$$

where x_n^l represents the total combined input into node n on layer l [33]. The node's output can then be formulated as

$$y_n^l = f_l\left(x_n^l + b_n^l\right), \tag{7}$$

where $f(\cdot)$ represents the node's activation function, and b_n^l is the node's bias constant. Equation (6) can be inserted into (7), and rewritten in matrix form as

$$\boldsymbol{y}^l = \boldsymbol{f}_l\left(V^l \boldsymbol{y}^{l-1} + \boldsymbol{b}^l\right), \tag{8}$$

which is valid for $l = 1, 2, \ldots, L$. It should be noted that $\boldsymbol{y}^0$ is simply the network input vector, $\boldsymbol{p}$, and $\boldsymbol{y}^L$ corresponds to the network output vector, $\boldsymbol{o}$.

The neuron activation functions are inspired by their practical counterparts in biological NNs, and can be either linear or nonlinear equations that determine the node's output response based on its inputs. Linear functions, such as $f(z) = c \cdot z$, can scale the input by a constant value c, and are often used in the last network layer, L, to scale the outputs to a desired range.

Nonlinear activation functions, on the other hand, give the network the ability to provide multi-dimensional mapping with a minimal number of network nodes. While many nonlinear equations have been utilized within NNs, the logistic-sigmoid function is among the most popular, and is given by

$$f(z) = \frac{1}{1 + e^{-z}}, \tag{9}$$

which has a smooth gradient that prevents large output changes for small input variations, and provides a natural normalization since the outputs are bounded between 0 and 1.

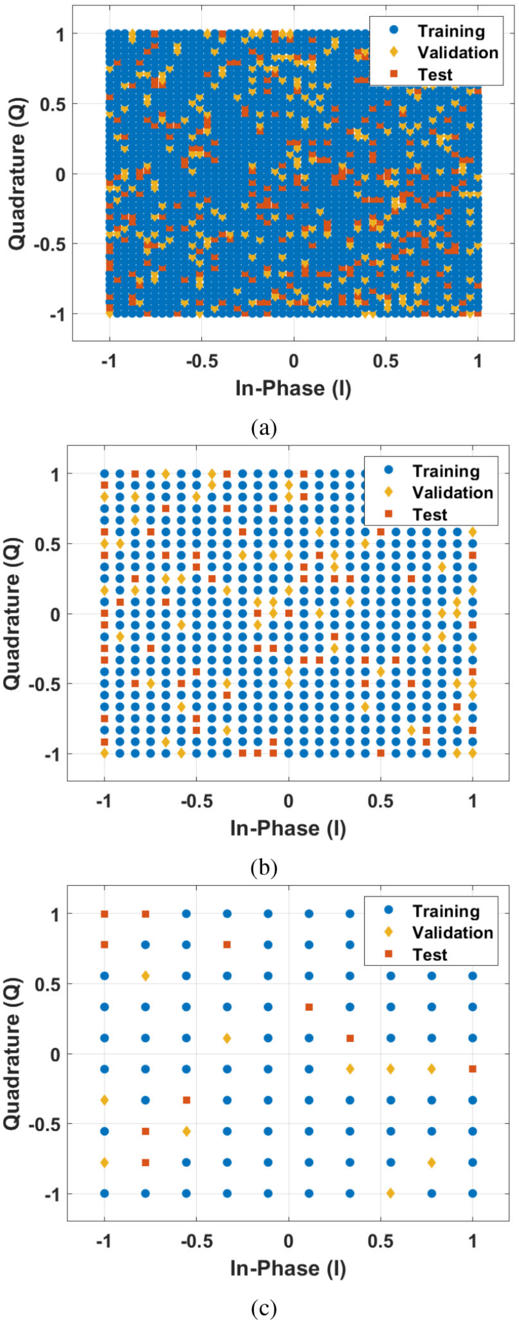
The specific NN architecture for this canceller tuning approach utilized sigmoid activation functions at every node, except for the last network layer, which contained a linear function that scaled the final outputs to the appropriate ranges for the specific canceller hardware controls and experiments described in Section IV. These final NN outputs, $\boldsymbol{o}$, contained four values ($R = 4$), and consisted of w_I and w_Q for both VMs. The NN inputs, $\boldsymbol{p}$, as well as the number of hidden layers, L, and hidden nodes per layer, N, were varied through the simulations described in Section III-D to determine the optimal network configuration.

C. NETWORK TRAINING

The network weights captured in (6) are computed one time using an input dataset that maps to a set of known-good target outputs, and is referred to as training the NN. These training sequences are comprised of sample inputs, $\boldsymbol{p}$, and desired outputs, $\boldsymbol{d}$, for a total of S pairs, such that $\{(\boldsymbol{p}_1,\boldsymbol{d}_1),(\boldsymbol{p}_2,\boldsymbol{d}_2), \ldots, (\boldsymbol{p}_S,\boldsymbol{d}_S)\}$. As shown in Fig. 4(a), the input training vector $\boldsymbol{p}_S$ is the same length, Q, as the number of network inputs, and similarly, the output training vector $\boldsymbol{d}_S$ is of length R. This allows the individual training error at output r when applying sequence s to the input to be calculated by

$$\boldsymbol{e}_{s,r} = \boldsymbol{d}_{s,r} - \boldsymbol{o}_{s,r}, \tag{10}$$

where $\boldsymbol{d}$ is the desired output vector, and $\boldsymbol{o}$ is the actual output vector [34]. These individual errors can be combined to provide a sum square error (SSE) that can be used to evaluate the network training for all sequences and outputs,



(a)

(b)

(c)

FIGURE 5. Complex sample points of composite SI channel where cancellation measurements were made to train and test the neural network for sample grids with (a) 50x50 = 2500 points, (b) 25x25 = 625 points and (c) 10x10 = 100 points.

such that

$$E(\boldsymbol{p}, \boldsymbol{v}) = \frac{1}{2}\sum_{s=1}^{S}\sum_{r=1}^{R} \boldsymbol{e}_{s,r}^{2} = \frac{1}{2}\boldsymbol{e}^{T}\boldsymbol{e}, \tag{11}$$

where $\boldsymbol{v}$ represents a vector containing all of the network weights, and $\boldsymbol{e}$ is the error vector that has the form

$$\boldsymbol{e} = \left[e_{1,1},\ e_{1,2}, \ldots, e_{1,R}, \ldots, e_{S,1},\ e_{S,2}, \ldots, e_{S,R}\right]^{T}, \tag{12}$$

where $(\cdot)^{T}$ is the transpose operator.

The training data for this IBFD application was generated using a channel emulator that deliberately covered the complete SI magnitude and phase space for a coupling channel with both direct and multipath signals present. Fig. 5(a) illustrates how a grid of 50 by 50 sample points were translated into I and Q values for the composite SI channel with a total of 2500 sequences, S. As shown, this dataset was segmented into specific points that were used for training, validation and testing. Cancellation measurements were made at each of these points using the adaptive tuning algorithm of (5), where 80 percent of the points were allocated to calculate the NN weights. Additionally, Fig. 5(b) depicts a 625-point sample space that was generated from a 25 by 25 point grid, and Fig. 5(c) illustrates the 100 samples that resulted from a sparse 10 by 10 grid. While sampling the SI channel with fewer points could increase the NN output errors, a smaller number of training sequences, S, would reduce the time required to construct the training dataset. Our investigations into the required sample space density are provided in the following subsection.

These training errors and the resulting composite vector of (12) can then be utilized within the Levenberg-Marquardt backpropagation algorithm to calculate all of the node weights according to

$$\boldsymbol{v}_{k+1} = \boldsymbol{v}_{k} - \left(\boldsymbol{J}_{k}^{T}\boldsymbol{J}_{k} + \mu\boldsymbol{I}\right)^{-1}\boldsymbol{J}_{k}^{T}\boldsymbol{e}_{k}, \tag{13}$$

where $\boldsymbol{v}$ is again a vector containing all of the network weights, μ is the scalable learning rate, $\boldsymbol{I}$ is the identity matrix, and k is the algorithm iteration number [35], [36]. This network calculation method utilizes the Jacobian matrix, $\boldsymbol{J}$, that contains the first-order partial derivatives of the network errors, as given by

$$\boldsymbol{J} = \begin{bmatrix} \frac{\partial e_{1,1}}{\partial v_{1}} & \frac{\partial e_{1,1}}{\partial v_{2}} & \cdots & \frac{\partial e_{1,1}}{\partial v_{L\cdot N}} \\ \frac{\partial e_{1,2}}{\partial v_{1}} & \frac{\partial e_{1,2}}{\partial v_{2}} & \cdots & \frac{\partial e_{1,2}}{\partial v_{L\cdot N}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_{1,R}}{\partial v_{1}} & \frac{\partial e_{1,R}}{\partial v_{2}} & \cdots & \frac{\partial e_{1,R}}{\partial v_{L\cdot N}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_{s,1}}{\partial v_{1}} & \frac{\partial e_{s,1}}{\partial v_{2}} & \cdots & \frac{\partial e_{s,1}}{\partial v_{L\cdot N}} \\ \frac{\partial e_{s,2}}{\partial v_{1}} & \frac{\partial e_{s,2}}{\partial v_{2}} & \cdots & \frac{\partial e_{s,2}}{\partial v_{L\cdot N}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_{S,R}}{\partial v_{1}} & \frac{\partial e_{S,R}}{\partial v_{2}} & \cdots & \frac{\partial e_{S,R}}{\partial v_{L\cdot N}} \end{bmatrix}, \tag{14}$$

where $L \cdot N$ represents the total number of NN weights. Overall, this Levenberg-Marquardt algorithm provides a stable approach to compute the large number of NN node weights, $\boldsymbol{v}$, in a fast and efficient manner [37].

Other than the training data shown in Fig. 5, 10 percent of the dataset was reserved for validation, which helped avoid overfitting the data by stopping the $\boldsymbol{v}$-weight iterations after generalization stopped improving. Finally, the last component of the SI dataset consisted of 10 percent of the points being reserved for testing purposes only, which provided an

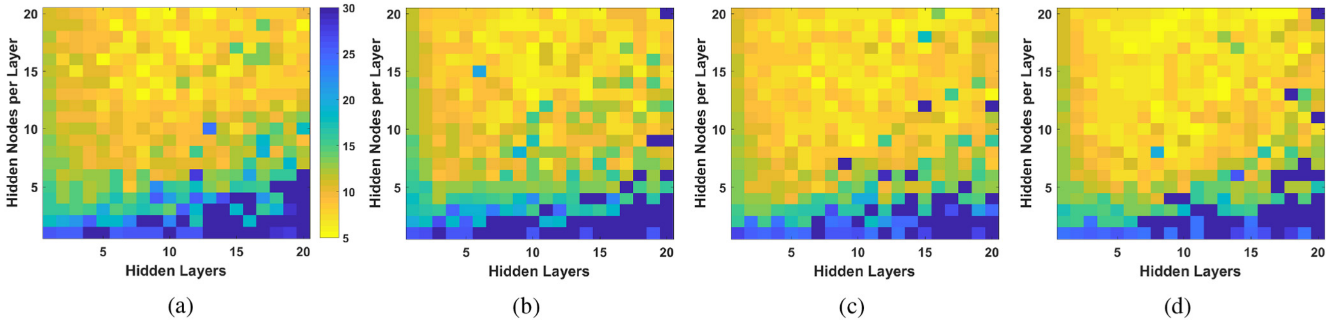


FIGURE 6. Simulated network error percentage (colorbar) for various configurations using 2500-training samples with different inputs: (a) full SI magnitude and phase responses, (b) the mean SI magnitude and phase responses, (c) the full SI real and imaginary responses, and (d) the mean SI real and imaginary responses (colorbar scale shown in (a) is the same for all plots).

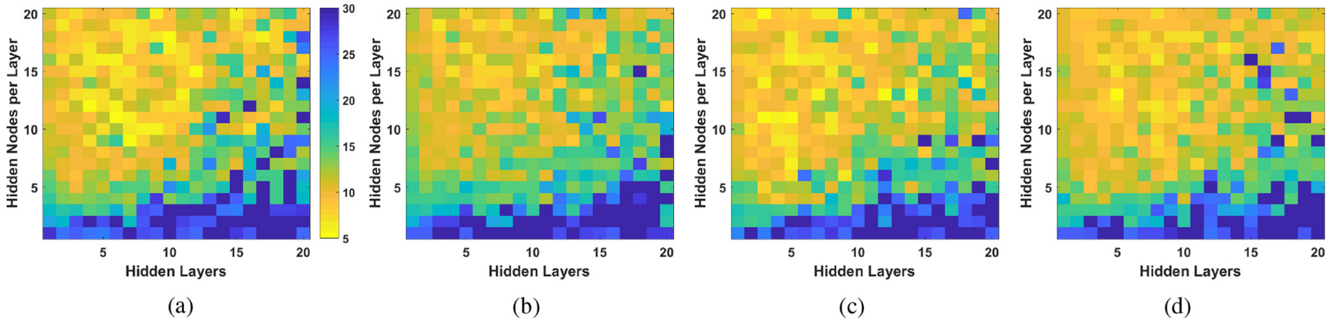


FIGURE 7. Simulated network error percentage (colorbar) for various configurations using 625-training samples with different inputs: (a) full SI magnitude and phase responses, (b) the mean SI magnitude and phase responses, (c) the full SI real and imaginary responses, and (d) the mean SI real and imaginary responses (colorbar scale shown in (a) is the same for all plots).

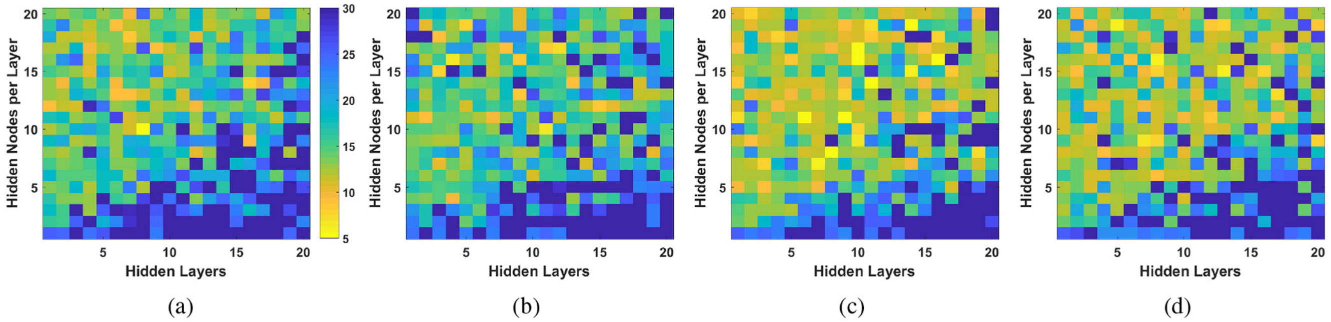


FIGURE 8. Simulated network error percentage (colorbar) for various configurations using 100-training samples with different inputs: (a) full SI magnitude and phase responses, (b) the mean SI magnitude and phase responses, (c) the full SI real and imaginary responses, and (d) the mean SI real and imaginary responses (colorbar scale shown in (a) is the same for all plots).

independent measure of network performance during formation, and were also used to evaluate the effectiveness of this concept in Section IV.

D. NETWORK SIMULATION

As mentioned in the previous subsections, we investigated the number of layers and number of nodes per layer that would minimize the total errors of (11) for the NN. This was done by solving (13) to compute all of the NN weights for these various configurations using the cancellation data measured for the three sample spaces illustrated in Fig. 5. The effectiveness of the NN simulations was evaluated by checking the residual errors for both the full training set (80 percent of the sample space) as well as the reserved test points only (10 percent of the sample space).

While all of these cases computed the same four VM outputs desired, we researched the effect of the input data structure on the network performance. The input was constructed in four different ways: the full SI magnitude and phase responses, the mean SI magnitude and phase responses, the full SI real and imaginary responses, and the mean SI real and imaginary responses. Respectively, these took the forms of

$$\boldsymbol{p} = [|H_{SI}|\ \angle H_{SI}]^T, \tag{15}$$

$$\boldsymbol{p} = \left[\overline{|H_{SI}|}\ \overline{\angle H_{SI}}\right]^T, \tag{16}$$

$$\boldsymbol{p} = [\Re H_{SI}\ \Im H_{SI}]^T, \tag{17}$$

$$\boldsymbol{p} = \left[\overline{\Re H_{SI}}\ \overline{\Im H_{SI}}\right]^T, \tag{18}$$

which consisted of 402 points for (15) and (17) (the frequency domain response sampled every 0.1 MHz for a 20 MHz bandwidth), and only 2 points for (16) and (18) (simply the mean values of (15) and (17), respectively).

Fig. 6 through 8 illustrate the residual NN error percentages on a color scale for NNs with a number of hidden layers that was varied from 1 to 20 (horizontal axis) with between 1 and 20 hidden nodes per layer (vertical axis). For example, the bottom left corners of these plots represent the most basic NNs with only one layer and one node (1 total neuron), while the top right corners represent NNs with 20 layers and 20 nodes per layer (400 total neurons). The results in Figs. 6, 7, and 8 represent NNs that were trained using the 2500, 625 and 100 sample points depicted in Fig. 5, respectively. For each of these figures, the subplots (a) through (d) correspond to the four input formats previously discussed as well as represented in (15) through (18), respectively.

The results plotted in all of these figures indicate that the NNs perform poorly when constructed with many layers but few nodes per layer (the bottom right corner of the plots). This intuitively makes sense since the networks do not have many node weights to effectively map node inputs to outputs before feeding them on to the next layer. While this collection of simulation results also illustrates that the network input type, (15)-(18), does not strongly influence the error, the number of complex sample points of the SI channel significantly impacts the network performance. As such, the lowest network tuning errors were achieved for the largest training set of 2500 samples, as depicted across Fig. 6.

As shown in Fig. 6(a), the minimum errors for the 2500 training samples occurred with the full SI magnitude and phase response inputs, and yielded 3.3 percent for a NN configuration with an L of 6 and N of 19. For the 625 length sequences, the minimum error was 4.8 percent for the full SI real and imaginary inputs, as illustrated in Fig. 7(c) for an L of 5 and an N of 11. Similarly, the lowest error for the 100 training points occurred using the full SI real and imaginary inputs, and was found to be 7.4 percent, as shown in Fig. 8(c) for an L of 10 and an N of 17.

The complete set of minimum testing errors and their associated optimal structures (in terms of hidden layers and nodes per layer) for all of the network simulation cases are summarized in Table 1. Additionally, this table depicts the average NN error that could be expected for a suboptimal selection of the structure, which provides a sense of what would occur when the configuration sweeps of Figs. 6–8 are not conducted. While the error metrics represented in this table do not show a strong correlation with the network input type for the various numbers of training samples, lower errors are generally seen for the cases that utilize the full set of input data points over frequency as opposed to their mean values. Finally, the indicative benefits of training the NN with denser sample spaces will become even more apparent in the following section, where testing was conducted on complex SI responses that were not part of the training sequences, as would be expected in practical IBFD nodes.

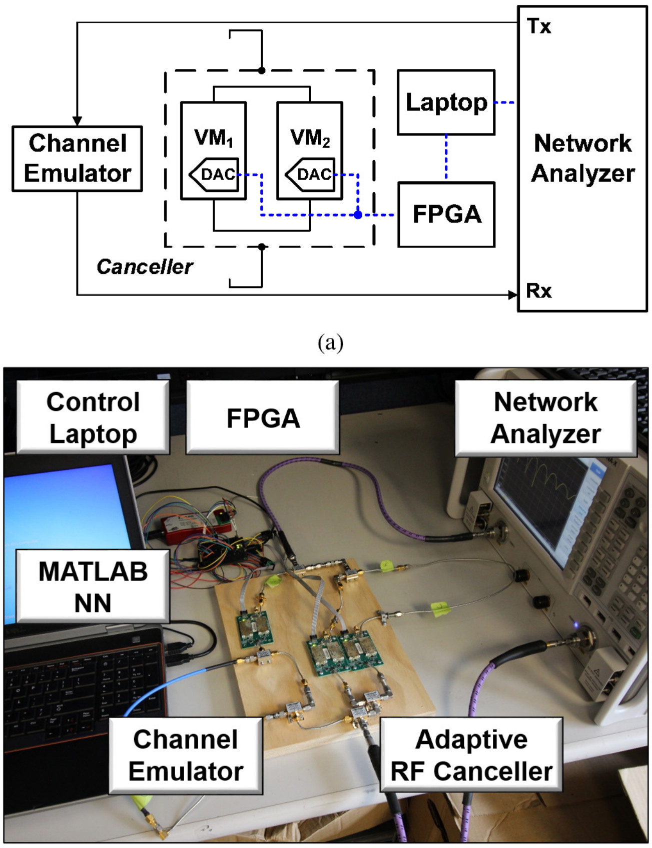
TABLE 1. Network simulation performance summary.

Network Configuration	Testing Errors (%)		Optimal Structure	
	Minimum	Average	Layers	Nodes/Layer
Samples: 2500 Input: (15)	3.3	13.4	6	19
Samples: 2500 Input: (16)	4.4	13.4	12	20
Samples: 2500 Input: (17)	5.0	12.4	13	19
Samples: 2500 Input: (18)	4.5	12.4	6	17
Samples: 625 Input: (15)	6.0	14.7	6	11
Samples: 625 Input: (16)	6.7	15.6	3	19
Samples: 625 Input: (17)	4.8	14.2	5	11
Samples: 625 Input: (18)	5.9	14.4	5	13
Samples: 100 Input: (15)	7.5	20.5	8	10
Samples: 100 Input: (16)	9.6	22.3	4	16
Samples: 100 Input: (17)	7.4	20.1	10	17
Samples: 100 Input: (18)	8.4	21.8	8	16

IV. RESULTS

Ideally, the residual errors discussed in the preceding section would be zero, but a combination of limited total neurons in these NN as well as the consistency of the training data introduce practical limitations. As captured in (10), these errors create undesired offsets in the NN outputs, the canceller weights. This effectively moves the canceller away from the global minimum (point D in Fig. 3), and necessitates additional minor weight tuning to relocate the optimal settings.

The overall quality of the NN outputs and associated additional tuning was evaluated through measurement of an IBFD node employing a two-tap VM adaptive canceller, as illustrated in Fig. 2(a). As depicted in the connection diagram of Fig. 9(a), a N5222A network analyzer from Keysight Technologies was utilized as the system's transmitter and receiver, and was responsible for feeding the receive data into a control laptop. For each test case, this computer formatted the data into one of the desired network input structures, described in (15) through (18), and then calculated the desired canceller weight outputs from the NN in MATLAB. This output generation was a straightforward process since the different NNs were already trained using (13), and the optimal configurations were stored for use with new measured inputs.



(a)

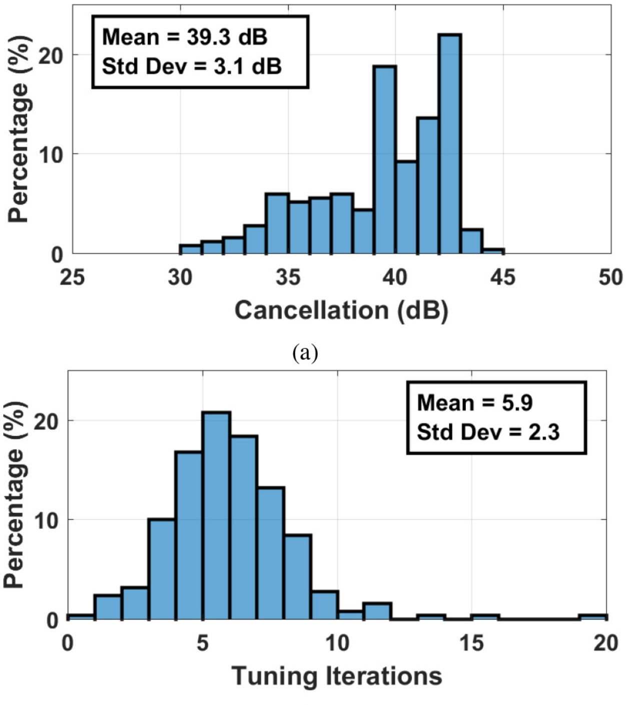
(b)

FIGURE 9. (a) Laboratory connection diagram and (b) setup photo that was used for the neural network tuning evaluation, illustrating the two-tap RF canceller, a control laptop, an FPGA, network analyzer and channel emulator.

Fig. 9(a) also illustrates how an FPGA was used to buffer the canceller weights before updating the analog attenuators on the VMs by writing to the on-board digital-to-analog converters (DACs). Additionally, the SI coupling channel was generated using a channel emulator that created both direct and multipath SI signals simultaneously, and was capable of varying the composite SI to any complex value, as depicted in Fig. 5. The initial channel setup resembled a base station node with a strong reflection approximately 0.3 m away. All the measurements in the next subsection were conducted over a 20 MHz bandwidth centered at 2.5 GHz, and a photo of this hardware setup is shown in Fig. 9(b).

A. STATISTICAL EVALUATION

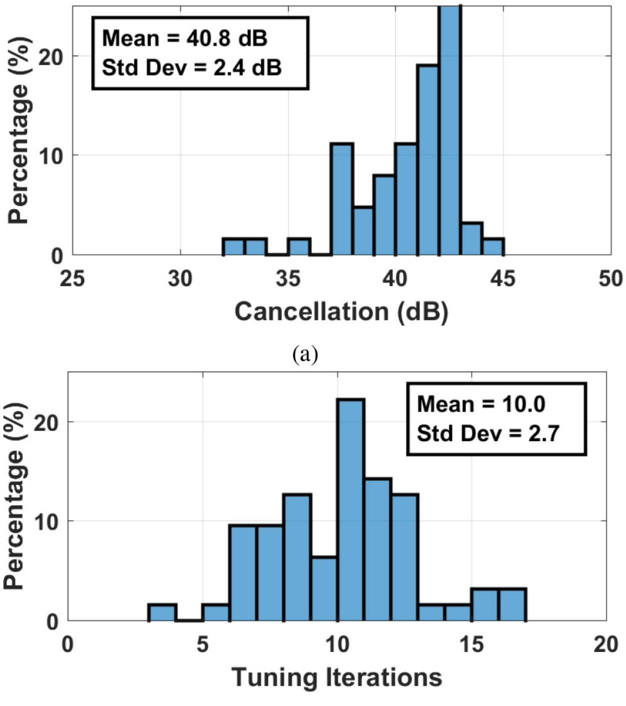
The cancellation capability and configuration speed of this novel tuning approach were evaluated using all of the different test point locations for each of the complex SI sample spaces shown in Fig. 5. It should be noted that test points – and not data points used for training – were chosen because these would present a more difficult input to the NN since these points were not used to create the network. Investigating the performance differences of the 2500-, 625- and 100-point sequences provides a better understanding of the tradeoffs associated with training dataset size.



(a)

(b)

FIGURE 10. Measured (a) cancellation performance and (b) tuning iterations for the adaptive RF canceller tuning approach utilizing a neural network trained with 2500 samples.



(a)

(b)

FIGURE 11. Measured (a) cancellation performance and (b) tuning iterations for the adaptive RF canceller tuning approach utilizing a neural network trained with 625 samples.

For each test location, the prototype node started by measuring the channel response that contained the composite SI, and then utilized the previously-trained NN configurations

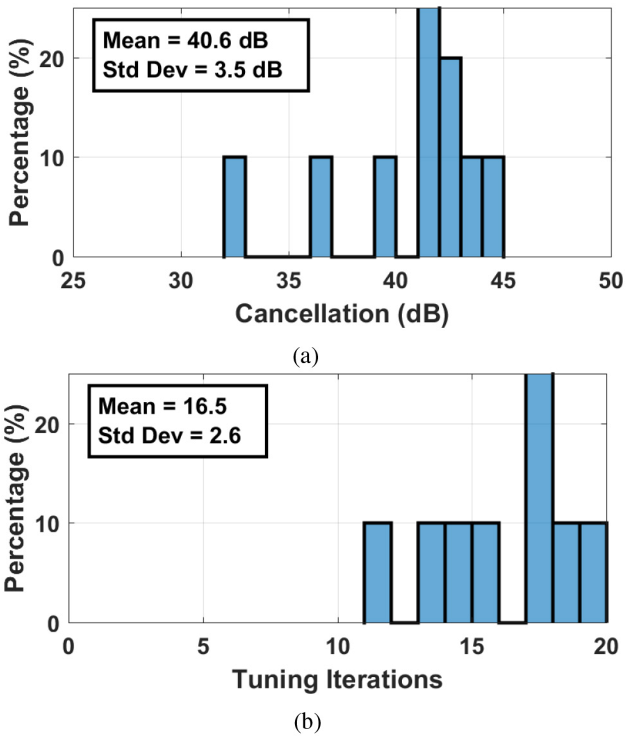


FIGURE 12. Measured (a) cancellation performance and (b) tuning iterations for the adaptive RF canceller tuning approach utilizing a neural network trained with 100 samples.

that yielded the lowest errors to estimate the initial VM weights, as discussed in the previous section (the three datasets used three different NN configurations). The resulting histograms and statistics for the cancellation performance and associated number of required tuning iterations are shown in Figs. 10 through 12 for the 2500-, 625- and 100-point training sequences, respectively. As mentioned, these results capture the NN performance at the test locations that were reserved in 10 percent of each dataset, and thus produced 250, 63 and 10 measurements, respectively.

While these plots indicate that all three training sets are capable of providing similar cancellation results (roughly 40 dB of cancellation), the number of necessary tuning iterations varies significantly as a function of the sample space density. The canceller weights converged to their optimal settings in an average of 5.9 iterations for the NN that utilized the 2500-point training sequence. The weight adjustments required 16.5 iterations on average, however, when the NN was trained with the sparser 100 sample points. Intuitively, this makes sense since training the NN with a dense sample space provides it with the ability to more accurately predict the desired outputs for unique SI responses. For IBFD applications that do not require the stringent settling times of flexible duplex operation, the NN with 625-point training can provide a tradeoff between the required dataset length and convergence time, which was measured to be 10 iterations, on average.

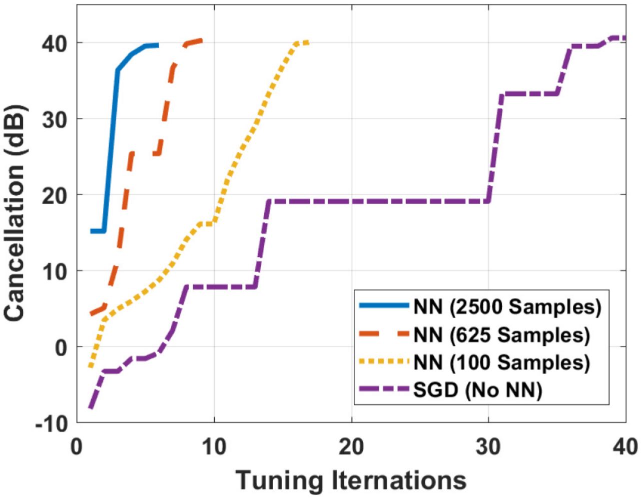


FIGURE 13. Average cancellation performance changes versus time, measured over 20 MHz centered at 2.5 GHz, for neural networks (NNs) trained with different sample densities, and compared to a standard gradient descent (SGD) method.

The influence of the training dataset density for these tuning time and cancellation results are plotted in Fig. 13 for representative performances at a single test location. These measurements are also compared to a standard gradient descent (SGD) tuning approach to illustrate the time acceleration provided by the use of the NN technique proposed. While all of the curves approach a final cancellation around 40 dB, both their initial performance and required iterations vary drastically. For both of the denser training sample spaces (2500 and 625 points), the cancellation started above 0 dB, which indicates that the NN outputs accurately predicted weights that were near the optimal settings for the canceller (such as point C in Fig. 3). For both the 100-sample NN and the SGD method, however, the initial cancellation was less than 0 dB, which required many additional iterations of the adaptive algorithm before locating the optimal canceller weights. This plot also highlights that NNs trained with denser sample spaces converge in fewer tuning iterations, as previous discussed. Overall, these results indicate that using a NN to tune adaptive RF cancellers can significantly accelerate their speed compared to traditional approaches.

The tuning speedup provided by this NN approach is critical when the SI changes rapidly and the node has to mitigate different SI before the start of every new symbol, such as in the flexible duplex example of Fig. 1. Fig. 14 illustrates the tuning results of this realistic scenario using a constant transmit output and a SI channel power that changes between symbols within a flexible duplex timeslot for the 2500-sample NN implementation. These channel powers were measured over a 20 MHz bandwidth centered at 2.5 GHz, and included the analyzer noise floor for reference. All 14 symbols of a timeslot for a subcarrier spacing of 60 kHz are shown to illustrate the concept, which highlights how the NN-tuning approach is necessary to provide tuning times that are less than 10 μs and consume only a small fraction of the symbol duration. This timing estimate assumed

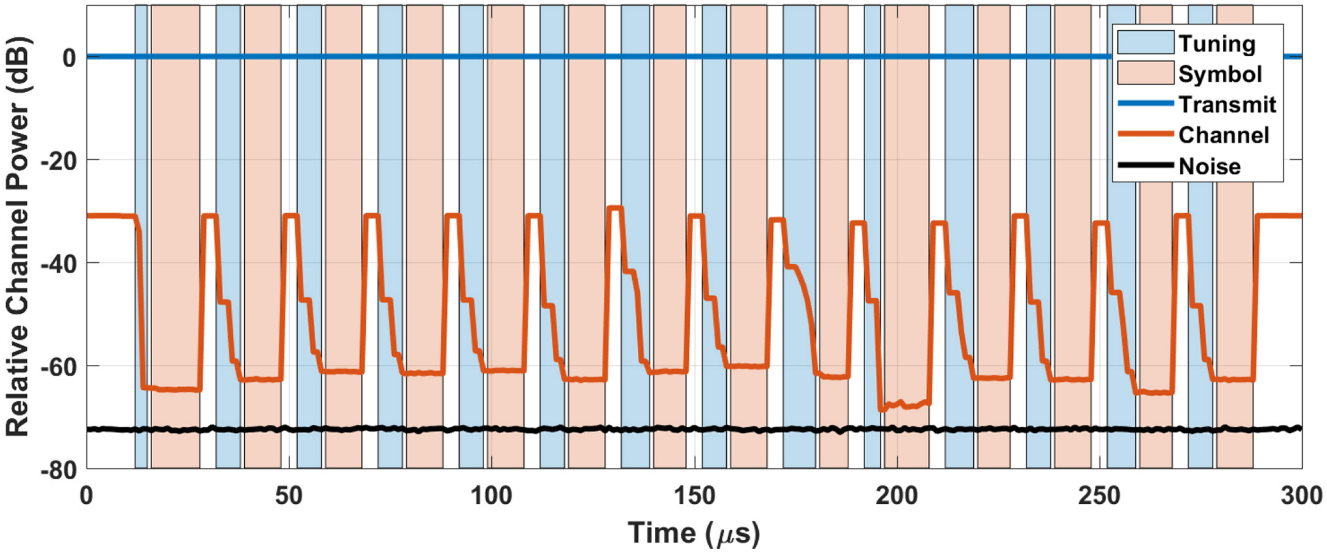


FIGURE 14. Example flexible duplex tuning scenario with average channel and noise power plotted relative to the transmit output, as well as highlighting the neural network-based canceller tuning times during the 14 symbol durations of a 5G NR timeslot.

a modest 200 MSPS analog-to-digital converter (ADC) and 200 data samples for each tuning measurement.

B. EXPANDED APPLICATIONS

While the previous subsection demonstrated the effectiveness of the NN-based tuning approach using a two-tap canceller and a 20 MHz bandwidth, this concept can straightforwardly be expanded to cancellers with more taps and wider instantaneous bandwidths. In some cases, NN configurations trained using sample data collected over 20 MHz bandwidths can be utilized to accurately predict the canceller weights for wider bandwidth targets. Fig. 15 plots the measured cancellation performance in the frequency domain for one such scenario. A 20 MHz bandwidth is highlighted to illustrate the desired cancellation region for the initial NN output, which produced an average cancellation of 41.3 dB after 5 tuning iterations. Without retraining or reconfiguring the network, the same starting canceller weights were used to tune to a 100 MHz bandwidth as indicated in the plot with a lower residual over the 2.45 to 2.55 GHz band. For this case, a cancellation average of 39.8 dB was achieved over the 100 MHz band in 8 iterations, and approached the instrument noise floor near the center of the band. While this represents only one example of the NN-tuning technique supporting an alternate application, similar performance can be expected by incorporating additional canceller taps and training datasets targeting wider bandwidths into the initial NN design.

C. PERFORMANCE SURVEY

Based on the results discussed in the previous subsections, this novel NN-based tuning approach can provide substantial cancellation performance in as few as 5.9 μs, on average. This offers a significant improvement over other adaptive RF canceller approaches that have reported tuning times of 10000 μs [12], 1000 μs [16], [17], 500 μs [18], approximately 250 μs [19], [20], 50 μs [13], 28 μs [15], and 20 μs [21]. While these prototypes were designed with various numbers of canceller taps, their common goal was to deliver cancellation over sufficiently wide channel bandwidths. With that said, a simple figure of merit (FOM) can be created, such that

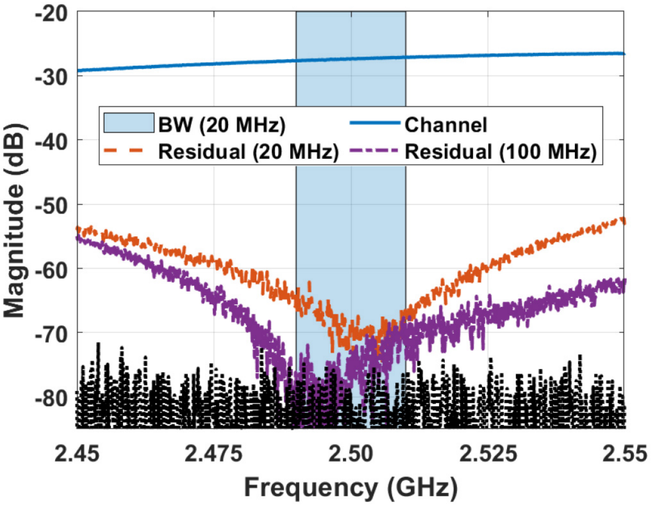


FIGURE 15. Magnitude of channel response, canceller residual signals for 20 and 100 MHz target bandwidths, and instrument noise for both bandwidths centered at 2.5 GHz.

$$\mathrm{FOM} = \mathrm{CANC} \cdot \mathrm{IBW}, \tag{19}$$

where CANC represents the cancellation performance achieved and IBW is the instantaneous bandwidth in MHz over which the cancellation was measured.

This FOM is plotted for the results discussed here and other previously-reported analog-RF cancellers in Fig. 16

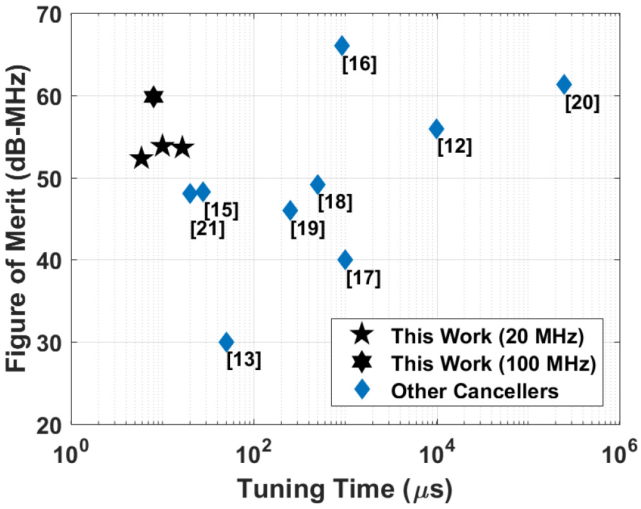


FIGURE 16. Measured FOM (cancellation-bandwidth product) versus tuning time on a logarithmic scale, comparing this work and the other approaches referenced.

(reference numbers included). This graph captures the cancellation-bandwidth performance and tuning time landscape for state-of-the-art designs, where the ideal combination resides in the top-left corner of the figure (i.e., a high cancellation-bandwidth product with minimal tuning time). The average cancellation and tuning time performance for the 2500-, 625- and 100-point NN training results are represented by the five-pointed star symbols plotted from left to right, respectively (all measured over a 20 MHz bandwidth). Additionally, the application of this NN-approach to the 100 MHz bandwidth discussed in the previous subsection is illustrated with a six-pointed star. Not only do these results represent the fastest tuning speeds reported, they also indicate that NNs can drastically accelerate canceller tuning and provide the speed required to enable IBFD operation within the flexible duplex aspect of 5G NR wireless networks.

V. CONCLUSION

The flexible duplex option within the 5G NR specification can enable IBFD operation when combined with carrier aggregation in TDD mode. In order to mitigate the resulting SI, a node can utilize an adaptive RF canceller, but it must tune within a fraction of a symbol duration. To address this need, we uniquely applied machine learning – specifically, a feedforward neural network – to accurately estimate the canceller weights and drastically accelerate the tuning speed. This novel concept was investigated for various network configurations in terms of both layer and node counts as well as the optimal input data structure and training sample density. The resulting cancellation performance was measured to be approximately 40 dB over a 20 MHz bandwidth centered at 2.5 GHz, and converged in an average of 6 iterations (or 6 μs), which represents the fastest reported approach. Our future efforts will include expanding the NN flexibility and prediction capability as well as incorporating this tuning approach into a complete IBFD node for system-level evaluation.

DISTRIBUTION STATEMENT A

Approved for public release. Distribution is unlimited.

ACKNOWLEDGMENT

The authors would like to express their gratitude to M. Fosberry for RF measurement support and R. Yandow for help with the PCB design; and E. Allen and N. Shinopulos for their help procuring all of the components. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Air Force.

REFERENCES

[1] L. Wan, M. Zhou, and R. Wen, "Evolving LTE with flexible duplex," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, 2013, pp. 49–54.

[2] H. Liu, Y. Jiao, Y. Gao, L. Sang, and D. Yang, "Performance evaluation of flexible duplex implement based on radio frame selection in LTE heterogeneous network," in *Proc. 22nd Int. Conf. Telecommun. (ICT)*, 2015, pp. 308–312.

[3] K. Balachandran, J. H. Kang, K. Karakayali, and K. M. Rege, "Flexible duplex in FDD spectrum," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, 2017, pp. 296–301.

[4] Q. Liao, "Dynamic uplink/downlink resource management in flexible duplex-enabled wireless networks," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, 2017, pp. 625–631.

[5] P. Pirinen, "Challenges and possibilities for flexible duplexing in 5G networks," in *Proc. IEEE 20th Int. Workshop Comput. Aided Model. Design Commun. Links Netw. (CAMAD)*, 2015, pp. 6–10.

[6] K. E. Kolodziej, *In-Band Full-Duplex Wireless Systems Handbook*. Norwood, MA, USA: Artech House, 2021.

[7] K. E. Kolodziej, B. T. Perry, and J. S. Herd, "In-band full-duplex technology: Techniques and systems survey," *IEEE Trans. Microw. Theory Techn.*, vol. 67, no. 7, pp. 3025–3041, Jul. 2019.

[8] J. Bai, S.-P. Yeh, F. Xue, Y.-S. Choi, P. Wang, and S. Talwar, "Full-duplex in 5G small cell access: System design and performance aspects," 2019. [Online]. Available: arXiv:1903.09893.

[9] M. Beach, L. Laughlin, C. Zhang, K. Morris, J. Haine, and M. McCullagh, "Flexible duplex transceivers for 5G and beyond wireless access," in *Proc. 1st URSI Atlantic Radio Sci. Conf. (URSI AT-RASC)*, 2015, p. 1.

[10] S.-M. Kim, H. Cha, S.-L. Kim, and C.-B. Chae, "Demo: A reinforcement learning-based flexible duplex systems for B5G with sub-6 Ghz," in *Proc. IEEE Wireless Commun. Netw. Conf. Workshops (WCNCW)*, 2020, pp. 1–2.

[11] R. Palaniappan, V. Gurumurthy, and S. Aniruddhan, "A spectral shaper based two-tap RF self-interference canceller for full-duplex radios," in *Proc. IEEE MTT-S Int. Microw. Symp. (IMS)*, 2019, pp. 614–617.

[12] Z. Zhang, Y. Shen, S. Shao, W. Pan, and Y. Tang, "Full duplex 2x2 MIMO radios," in *Proc. 6th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, 2014, pp. 1–6.

[13] B. Keogh and A. Zhu, "Wideband self-interference cancellation for 5G full-duplex radio using a near-field sensor array," in *Proc. IEEE MTT-S Int. Microw. Workshop Series 5G Hardw. Syst. Technol. (IMWS-5G)*, 2018, pp. 1–3.

[14] K. E. Kolodziej, S. Yegnanarayanan, and B. T. Perry, "Photonic-enabled RF canceller for wideband in-band full-duplex wireless systems," *IEEE Trans. Microw. Theory Techn.*, vol. 67, no. 5, pp. 2076–2086, May 2019.

[15] K. E. Kolodziej, A. U. Cookson, and B. T. Perry, "Adaptive learning rate tuning algorithm for RF self-interference cancellation," *IEEE Trans. Microw. Theory Techn.*, vol. 69, no. 3, pp. 1740–1751, Mar. 2021.

[16] D. Bharadia, E. McMilin, and S. Katti, "Full duplex radios," in *Proc. ACM Conf. (SIGCOMM)*, 2013, pp. 375–386.

[17] M. Jain *et al.*, "Practical, real-time, full duplex wireless," in *Proc. Annu. Int. Conf. Mobile Comput. Netw.*, 2011, pp. 301–312.

[18] K. E. Kolodziej, J. G. McMichael, and B. T. Perry, "Multitap RF canceller for in-band full-duplex wireless communications," *IEEE Trans. Wireless Commun.*, vol. 15, no. 6, pp. 4321–4334, Jun. 2016.

[19] T. Huusari, Y.-S. Choi, P. Liikkanen, D. Korpi, S. Talwar, and M. Valkama, "Wideband self-adaptive RF cancellation circuit for full-duplex radio: Operating principle and measurements," in *Proc. IEEE 81st Veh. Technol. Conf. (VTC Spring)*, 2015, pp. 1–7.

[20] J. Tamminen *et al.*, "Digitally-controlled RF self-interference canceller for full-duplex radios," in *Proc. 24th Eur. Signal Process. Conf. (EUSIPCO)*, 2016, pp. 783–787.

[21] Y. Cao, X. Cao, H. Seo, and J. Zhou, "An integrated full-duplex/FDD duplexer and receiver achieving 100Mhz bandwidth 58dB/48dB self-interference suppression using hybrid-analog-digital autonomous adaptation loops," in *Proc. IEEE/MTT-S Int. Microw. Symp. (IMS)*, 2020, pp. 1203–1206.

[22] A. Bakshi, Y. Mao, K. Srinivasan, and S. Parthasarathy, "Fast and efficient cross band channel prediction using machine learning," in *Proc. 25th Annu. Int. Conf. Mobile Comput. Netw.*, 2019, pp. 1–16.

[23] A. Balatsoukas-Stimming, "Non-linear digital self-interference cancellation for in-band full-duplex radios using neural networks," in *Proc. IEEE 19th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, 2018, pp. 1–5.

[24] Y. Kurzo, A. Burg, and A. Balatsoukas-Stimming, "Design and implementation of a neural network aided self-interference cancellation scheme for full-duplex radios," in *Proc. 52nd Asilomar Conf. Signals Syst. Comput.*, 2018, pp. 589–593.

[25] H. Guo, J. Xu, S. Zhu, and S. Wu, "Realtime software defined self-interference cancellation based on machine learning for in-band full duplex wireless communications," in *Proc. Int. Conf. Comput. Netw. Commun. (ICNC)*, 2018, pp. 779–783.

[26] H. Guo, S. Wu, H. Wang, and M. Daneshmand, "DSIC: Deep learning based self-interference cancellation for in-band full duplex wireless," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2019, pp. 1–6.

[27] Q. Wang, F. He, and J. Meng, "Performance comparison of real and complex valued neural networks for digital self-interference cancellation," in *Proc. IEEE 19th Int. Conf. Commun. Technol. (ICCT)*, 2019, pp. 1193–1199.

[28] Y. Kurzo, A. T. Kristensen, A. Burg, and A. Balatsoukas-Stimming, "Hardware implementation of neural self-interference cancellation," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 10, no. 2, pp. 204–216, Jun. 2020.

[29] I. T. Cummings, T. J. Schulz, T. C. Havens, and J. P. Doane, "Neural networks for real-time adaptive beamforming in simultaneous transmit and receive digital phased arrays: Student submission," in *Proc. IEEE Int. Symp. Phased Array Syst. Technol. (PAST)*, 2019, pp. 1–8.

[30] A. T. Le, L. C. Tran, X. Huang, and Y. J. Guo, "Beam-based analog self-interference cancellation in full-duplex MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 19, no. 4, pp. 2460–2471, Apr. 2020.

[31] K. E. Kolodziej, A. U. Cookson, and B. T. Perry, "Machine learning for accelerated IBFD tuning in 5G flexible duplex networks," in *Proc. IEEE/MTT-S Int. Microw. Symp. (IMS)*, 2020, pp. 691–694.

[32] K. E. Kolodziej and B. T. Perry, "Wideband vector modulator for RF cancellers in star systems," in *Proc. IEEE Radio Wireless Symp. (RWS)*, 2018, pp. 64–67.

[33] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the Marquardt algorithm," *IEEE Trans. Neural Netw.*, vol. 5, no. 6, pp. 989–993, Nov. 1994.

[34] H. Liu, "On the Levenberg-Marquardt training method for feed-forward neural networks," in *Proc. 6th Int. Conf. Nat. Comput.*, vol. 1, 2010, pp. 456–460.

[35] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *Quart. Appl. Math.*, vol. 2, no. 2, pp. 164–168, 1944.

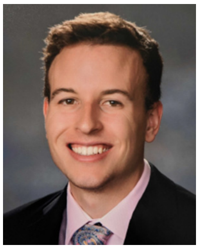
[36] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *J. Soc. Ind. Appl. Math.*, vol. 11, no. 2, pp. 431–441, 1963.

[37] J. S. Smith, B. Wu, and B. M. Wilamowski, "Neural network training with Levenberg-Marquardt and adaptable weight compression," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 2, pp. 580–587, Feb. 2019.

KENNETH E. KOLODZIEJ (Member, IEEE) received the B.E. and M.E. degrees in electrical engineering from the Stevens Institute of Technology, Hoboken, NJ, USA, in 2007.

In 2007, he joined BAE Systems in Wayne, NJ, USA, to design RF electronics for handheld communication devices. Since 2010, he has been working with MIT Lincoln Laboratory, Lexington, MA, USA, and he has conducted research on RF, microwave and photonic circuits, including antenna, radar and communications systems. He is currently working in the Advanced Technology Division's RF Technology Group, designing compact transceivers and cancelation techniques for in-band full-duplex applications. He also teaches an electromagnetics course to undergraduate students with Massachusetts Institute of Technology (MIT), and several "Build-a-Radar" courses on MIT campus.

Mr. Kolodziej is an Editor of the *In-Band Full-Duplex Wireless Systems Handbook* (Artech House). He has been a member of the technical program committees of several conferences, including the IEEE International Microwave Symposium since 2019; the IEEE Radio & Wireless Symposium since 2022; the IEEE International Symposium on Antennas and Propagation since 2017; the IEEE International Symposium on Phased Array Systems and Technology since 2019; and the IEEE International Conference on Communications since 2020.



AIDAN U. COOKSON received the B.S. degree in electrical and computer engineering from Worcester Polytechnic Institute, Worcester, MA, USA, in 2019. He is currently pursuing the M.S. degree in electrical and computer engineering with the University of California at Los Angeles, Los Angeles. He interned with MIT Lincoln Laboratory, Lexington, MA, USA, in 2019, where he conducted research in IBFD self-interference cancellation techniques within the RF Technology Group.

BRADLEY T. PERRY (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering from Michigan State University in 2001, 2002, and 2005, respectively, during which he focused on antenna design and electromagnetic propagation through layered media.

Since 2005, he has been with MIT Lincoln Laboratory, where he is currently a part of the RF Technology Group focusing on the development of next-generation electronic systems and technology aspects related to isolation improvement in electronic systems. These developments include work in the areas of microwave circuit and antenna design, compact receiver and transmitter designs, and RF cancellation techniques for simultaneous transmit and receive (STAR) applications.

Dr. Perry served as the Chairman of the Boston Section of the IEEE Antennas and Propagation Society from 2006 to 2008 and continued in the role of past chair in 2009. He also served as the student programs coordinator for 2013 and 2016 IEEE International Symposium on Phased Array Systems and Technology. He has presented work at many IEEE and Antenna Measurement Techniques Association symposiums and published articles in a number of refereed journals. He is a member of Commission B of the International Union of Radio Science and the IEEE Antennas and Propagation and Microwave Theory and Techniques Societies.