Received 17 May 2022; accepted 2 June 2022. Date of publication 9 June 2022; date of current version 20 June 2022.

Digital Object Identifier 10.1109/OJCOMS.2022.3181685

SICNet—Low Complexity Sample Adaptive Neural Network-Based Self-Interference Cancellation in LTE-A/5G Mobile Transceivers

OLIVER PLODER [1] (Graduate Student Member, IEEE), CHRISTINA AUER [1] (Graduate Student Member, IEEE), CHRISTIAN MOTZ [1], THOMAS PAIREDER [1], OLIVER LANG [2] (Member, IEEE), AND MARIO HUEMER [2] (Senior Member, IEEE)

[1]Christian Doppler Laboratory for Digitally Assisted RF Transceivers for Future Mobile Communications, Institute of Signal Processing, Johannes Kepler University Linz, 4040 Linz, Austria

[2]Institute of Signal Processing, Johannes Kepler University Linz, 4040 Linz, Austria

CORRESPONDING AUTHOR: O. PLODER (e-mail: oliver.ploder@jku.at)

This work was supported in part by the Austrian Federal Ministry for Digital and Economic Affairs; in part by the National Foundation for Research, Technology and Development; and in part by the Christian Doppler Research Association.

ABSTRACT The limited transmitter-to-receiver stop-band isolation of the duplexers in long term evolution (LTE) and 5G/NR frequency division duplex transceivers induces leakage signals from the transmitter(s) (Tx) into the receiver(s) (Rx). These leakage signals are the root cause of a multitude of self-interference (SI) problems in the receiver path(s) diminishing a receiver's sensitivity. Traditionally, these effects are counteracted by the use of various different SI cancellation (SIC) architectures which typically solely target one specific problem. In this paper, we propose two novel neural networks based architectures that can handle a variety of different SI effects without the need for a different architecture for each effect. We additionally show the suitability of the proposed architecture on SI effects occurring in in-band full duplex transceivers. Further, we introduce two novel low-cost training algorithms to enable online adaptation (as opposed to offline training currently proposed in literature). The combination of these two concepts is shown to not only beat existing algorithms in their cancellation performance, but also to provide sufficiently low computational complexity allowing on-chip implementations.

INDEX TERMS 4G, machine learning, online learning, self-interference cancellation.

I. INTRODUCTION

MACHINE LEARNING (ML) has experienced immense growth over the last decade, thanks to the tremendous boost in computational power available to the everyday researcher and user. It is commonly thought of *the* solution to hard-to-crack problems like automated image annotation [1] and recognition [2] in computer vision (CV), natural language processing (NLP) [3] or medical applications [4]. Neural networks (NNs) have also been shown to yield excellent performance in the field of system identification [5]–[7]. Typically, these NNs are smaller (in terms of layers and neurons) than their counterparts used in CV or NLP, where complexity is not as much an issue as training can be done offline in a datacenter. However, this is not necessarily true for all NN applications, especially if they shall learn to adapt to certain changes in a system (online learning) and are meant to be implemented in small (battery-powered) consumer hardware, e.g., cell phone chips or Internet of Things (IoT) devices.

In order to still be able to implement NN-based methods onto small devices, researchers came up with specialized architectures with reduced complexity for on-chip implementations [8]–[13]. However, most of these architectures are meant to be only used during inference (i.e., execution),

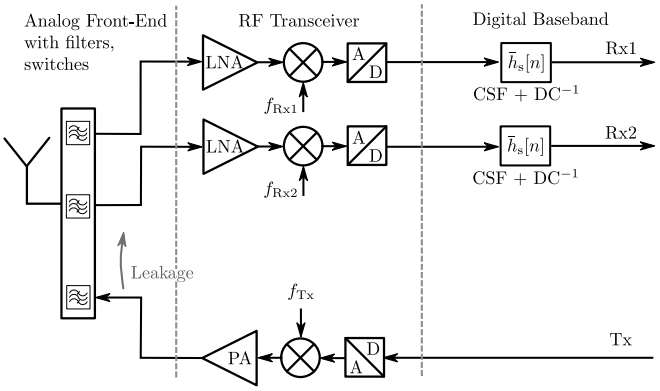

FIGURE 1. Block diagram of an LTE-A CA transceiver.

for an already trained model. Hence, training an NN on the device itself is still a problem to be solved. Obviously, this severely limits the applications of such solutions to areas were the underlying system does not change over time.

An application where NNs have been proven to yield exceptional performance, but which also requires fast adaptation as the underlying system changes frequently, is self-interference cancellation (SIC) in communication transceivers [5]–[7]. Specifically, modern direct-conversion long term evolution (LTE) frequency division duplex (FDD) transceivers experience a multitude of non-ideal effects which can significantly diminish the receiver's sensitivity [14]. This is partly due to the limited duplexer stop-band attenuation of typically around 50 dB [15], allowing part of the transmit (Tx) signal to leak into the receive (Rx) path. If this leakage signal is occurring in combination with other non-idealities (e.g., nonlinearities) in either the transmitter or receiver, the resulting self-interference (SI) signal may fall into the Rx band or baseband (BB) and therefore interfere with the actual Rx signal.

One example of such an SI effect are so called second-order intermodulation distortions (IMD2). Those are even order nonlinearities and fall directly into the Rx BB. The IMD2 may result from a coupling of the down-conversion mixer's radio frequency (RF)- and local oscillator (LO) ports in combination with the leaked Tx signal. In the worst case, this interference can be up to around 20 dB stronger than the desired Rx signal [16].

Further, in the case of LTE-Advanced (LTE-A) FDD transceivers operating in downlink carrier-aggregation (CA) with at least two active receive paths (cf. Fig. 1), so-called Tx harmonics may cause another type of SI under certain conditions. The power amplifier (PA) of the transmit chain typically exhibits nonlinear behavior and generates signals at multiples of the Tx frequency f_{Tx}. If one of these components overlaps with a receive band f_{Rx}, it will directly be down-converted to the Rx BB, and will interfere with the BB Rx signal [16].

In the same scenario (LTE-A with downlink CA), modulated spurs may be caused by a leaked Tx signal in combination with multiple clock sources and dividers – needed for all possible CA scenarios – which create various harmonics in the analog front-end of the transceiver. This can lead to a downconversion of the distorted Tx leakage signal into the baseband of one or more of the receive paths. This leaked and frequency shifted Tx signal in the RX paths represents the SI signal.

Lastly, we investigate the case of in-band full duplex SI, where a misaligned Tx mixer causes I/Q-imbalance and a non-ideal PA introduces further nonlinearities. The resulting Tx signal then travels via the leakage channel toward the Rx path where it can significantly diminish the Rx signal's quality and therefore limit throughput.

Contribution: In this work we propose two novel architectures for deploying NNs in an online learning manner for system identification in mobile transceivers for the purpose of SIC. Further, we introduce two adaptations to well-known training algorithms to make them either more robust/stable or less computationally intensive in order to ease the implementation burden. We additionally show that one and the same architecture (including hyperparameters) can be used to combat a multitude of SI effects in both, FDD and in-band full duplex transceivers, which – to the best of our knowledge – has not been shown in prior works. This eliminates the need to implement various SIC algorithms to mitigate different SI effects.

This paper is organized as follows, in Section II the actual self-interference problems are introduced before Section III reviews the current state-of-the-art in terms of SIC architectures and NN hardware implementations. Afterwards, Section IV discusses the proposed SIC architectures before the actual online learning algorithms are derived in Section V. Section VI takes a look at the complexity of all considered architectures, and simulation results are presented in Section VII. Finally, Section VIII concludes the paper.

II. PROBLEM STATEMENT

In the following, the three considered SI problems occurring in modern FDD transceivers as well as an SI type occurring in in-band full duplex transceivers are explained in greater detail.

A. MODULATED SPURS

The leaked Tx signal can be described by

$$y_{\mathrm{RF}}^{\mathrm{Tx,L}}(t) = A_{\mathrm{PA}}\,\Re\left\{\left(x_{\mathrm{BB}}^{\mathrm{Tx}}(t) * h_{\mathrm{BB}}^{\mathrm{Tx,L}}(t)\right) e^{j2\pi f_{\mathrm{Tx}} t}\right\}, \tag{1}$$

where A_{PA} represents the amplification factor of the PA, neglecting nonlinear effects, $x_{\mathrm{BB}}^{\mathrm{Tx}}$ is the complex valued BB Tx signal, $h_{\mathrm{BB}}^{\mathrm{Tx,L}}(t)$ denotes the impulse response of the Tx-Rx leakage channel, and f_{Tx} is the Tx carrier frequency [17].

Therefore, the total received signal, after the low noise amplifier (LNA) is given by

$$y_{\mathrm{RF}}^{\mathrm{Tot}}(t) = A_{\mathrm{LNA}}\left[y_{\mathrm{RF}}^{\mathrm{Tx,L}}(t) + y_{\mathrm{RF}}^{\mathrm{Rx}}(t) + \eta_{\mathrm{RF}}(t)\right], \tag{2}$$

and consists of three components, the leaked Tx signal centered at f_{Tx}, the wanted Rx signal $y_{\mathrm{RF}}^{\mathrm{Rx}}(t)$ centered at f_{Rx}, and the thermal noise $\eta_{\mathrm{RF}}(t)$. We assume an ideal LNA model with amplification A_{LNA}.

The mixer downconverts the wanted Rx signal to the baseband. If a spur frequency f_{sp} lies close to f_{Tx}, also the leaked Tx signal will be downconverted to the Rx baseband. After the mixing stage but in front of the analog-to-digital converter (ADC), the baseband Rx signal can therefore be formulated as [17]

$$y_{\mathrm{RF}}^{\mathrm{Tot}}(t) = A_{\mathrm{LNA}} A_{\mathrm{sp}}\, y_{\mathrm{RF}}^{\mathrm{Tx,L}}(t)\, e^{-j2\pi f_{\mathrm{sp}} t} + A_{\mathrm{LNA}}\Big[\big(y_{\mathrm{RF}}^{\mathrm{Rx}}(t) + \eta_{\mathrm{RF}}(t)\big)\, e^{-j2\pi f_{\mathrm{Rx}} t}\Big], \quad (3)$$

where A_{sp} denotes the gain of the spur. Combining (1) and (3) and defining $\Delta f = f_{\mathrm{Tx}} - f_{\mathrm{sp}}$ allows to represent the baseband Rx signal after the ADC and channel select filter (CSF), while neglecting the frequency content attenuated by the anti-aliasing filter, as [17]

$$y_{\mathrm{BB}}^{\mathrm{Tot}}[n] = \left(A_{\mathrm{Tot}}\left(x_{\mathrm{BB}}^{\mathrm{Tx}}[n] * h_{\mathrm{BB}}^{\mathrm{Tx,L}}[n]\right) e^{\frac{j2\pi\Delta f n}{f_{\mathrm{s}}}}\right) * \bar{h}_{\mathrm{s}}[n] + A_{\mathrm{LNA}}\big(y_{\mathrm{BB}}^{\mathrm{Rx}}[n] + \eta_{\mathrm{BB}}[n]\big) * \bar{h}_{\mathrm{s}}[n]. \quad (4)$$

Here f_{s} is the sampling frequency of the system, $A_{\mathrm{Tot}} = A_{\mathrm{PA}} A_{\mathrm{LNA}} A_{\mathrm{sp}}$ and $\bar{h}_{\mathrm{s}}[n] = h_{\mathrm{CSF}}[n] * h_{\mathrm{DC}}[n]$ denotes the combined channel-select and DC filter. The first term in (4) represents the unwanted Tx modulated spur $y_{\mathrm{BB}}^{\mathrm{MS}}[n]$ in the Rx BB [17].

B. TX HARMONICS

The up-converted Tx signal can be described by

$$x_{\mathrm{RF}}^{\mathrm{Tx}}(t) = \Re\left\{x_{\mathrm{BB}}^{\mathrm{Tx}}(t)\, e^{j2\pi f_{\mathrm{Tx}} t}\right\}, \quad (5)$$

where $x_{\mathrm{BB}}^{\mathrm{Tx}}$ is the complex valued BB Tx signal, f_{Tx} is the Tx carrier frequency and $\Re\{\cdot\}$ denotes the real operator. This signal is amplified by the PA, which is assumed to experience non-ideal behavior, e.g., due to the saturation of the PA output at high input amplitudes. In the case of second order Tx harmonics, this can be described as

$$x_{\mathrm{RF}}^{\mathrm{Tx,H2}}(t) = A_{\mathrm{PA}}\, \Re\left\{x_{\mathrm{BB}}^{\mathrm{Tx}}(t)\, e^{j2\pi f_{\mathrm{Tx}} t}\right\} + \beta_2 A_{\mathrm{PA}}^2\, \Re\left\{x_{\mathrm{BB}}^{\mathrm{Tx}}(t)\, e^{j2\pi f_{\mathrm{Tx}} t}\right\}^2, \quad (6)$$

where A_{PA} represents the amplification factor of the PA and β_2 is a PA specific amplification factor for the second harmonic. If the transceiver is working in downlink CA, the second part of (6) can directly fall into one of the receive bands and therefore cause the leakage signal

$$y_{\mathrm{RF}}^{\mathrm{Tx,L}}(t) = \beta_2 A_{\mathrm{PA}}^2\, \Re\left\{\left(x_{\mathrm{BB}}^{\mathrm{Tx}}(t)\, e^{j2\pi f_{\mathrm{Tx}} t}\right)^2 * h_{\mathrm{RF}}^{\mathrm{Tx,L}}(t)\right\}, \quad (7)$$

to interfere with the actual Rx signal , with $h_{\mathrm{RF}}^{\mathrm{Tx,L}} = h_{\mathrm{RF}}^{\mathrm{BB,L}} \cdot \exp(j2\pi f_{\mathrm{Tx}} t)$. Therefore, the total received signal, after the LNA is given by

$$y_{\mathrm{RF,LNA}}^{\mathrm{Tot}}(t) = A_{\mathrm{LNA}}\Big[y_{\mathrm{RF}}^{\mathrm{Tx,L}}(t) + y_{\mathrm{RF}}^{\mathrm{Rx}}(t) + \eta_{\mathrm{RF}}(t)\Big], \quad (8)$$

and consists of three components, the leaked Tx signal component centered at $2f_{\mathrm{Tx}}$, the wanted Rx signal $y_{\mathrm{RF}}^{\mathrm{Rx}}(t)$ centered at $f_{\mathrm{Rx,1}}$, and the thermal noise $\eta_{\mathrm{RF}}(t)$. We assume an ideal LNA model with amplification A_{LNA}.

The mixer down-converts the wanted Rx signal to the BB. If, for example, $f_{\mathrm{Rx,1}} = 2f_{\mathrm{Tx}}$, also the leaked Tx signal component will be down-converted to the Rx BB. After the mixing stage but in front of the ADC, the BB Rx signal can therefore be formulated as

$$y_{\mathrm{RF,LNA}}^{\mathrm{Tot}}(t) = A_{\mathrm{LNA}}\, y_{\mathrm{RF}}^{\mathrm{Tx,L}}(t)\, e^{-j2\pi f_{\mathrm{Rx,1}} t} + A_{\mathrm{LNA}}\Big[\big(y_{\mathrm{RF}}^{\mathrm{Rx}}(t) + \eta_{\mathrm{RF}}(t)\big)\, e^{-j2\pi f_{\mathrm{Rx,1}} t}\Big]. \quad (9)$$

Combining (7) and (9) allows to represent the BB Rx signal after the ADC and CSF, while neglecting the frequency content attenuated by the anti-aliasing filter, as

$$y_{\mathrm{BB}}^{\mathrm{Tot}}[n] = A_{\mathrm{Tot}}\left(x_{\mathrm{BB}}^{\mathrm{Tx}}[n]\right)^2 * h_{\mathrm{BB}}^{\mathrm{Tx,L}}[n] * \bar{h}_{\mathrm{s}}[n] + A_{\mathrm{LNA}}\big(y_{\mathrm{BB}}^{\mathrm{Rx}}[n] + \eta_{\mathrm{BB}}[n]\big) * \bar{h}_{\mathrm{s}}[n]. \quad (10)$$

Here, $A_{\mathrm{Tot}} = A_{\mathrm{PA}} A_{\mathrm{LNA}}$ and $\bar{h}_{\mathrm{s}}[n] = h_{\mathrm{CSF}}[n] * h_{\mathrm{DC}}[n]$ denotes the combined channel-select and DC filter. The first term in (10) represents the unwanted second order Tx harmonics $y_{\mathrm{BB}}^{\mathrm{H2}}[n]$ in the Rx BB.

C. IMD2

The derivation of the IMD2 interference follows the same pattern as the previous model. However, an ideal PA model is assumed. In addition to the regular down-conversion, the mixer may generate IMD2 which can be represented by squaring the input signal (assuming a polynomial nonlinearity of order 2). With the mixer's RF-to-LO coupling coefficient $\alpha_2 = \alpha_2^{\mathrm{I}} + j\alpha_2^{\mathrm{Q}}$, the down-converted signal including IMD2 can be expressed as [16]

$$y_{\mathrm{RF,Mix}}^{\mathrm{Tot}}(t) = \alpha_1 y_{\mathrm{RF,LNA}}^{\mathrm{Tot}}(t)\, e^{-j2\pi f_{\mathrm{Rx}} t} + y_{\mathrm{RF,LNA}}^{\mathrm{Tot}}(t)\big(\alpha_2\, y_{\mathrm{RF,LNA}}^{\mathrm{Tot}}(t)\big). \quad (11)$$

Using $\Re\{\nu e^{j\kappa}\} = \frac{1}{2}(\nu e^{j\kappa} + \nu^* e^{-j\kappa})$ and neglecting the frequency content attenuated by the anti-aliasing filter of the ADC, the total received BB signal after the ADC becomes

$$y_{\mathrm{BB,ADC}}^{\mathrm{Tot}}[n] = \alpha_1 \frac{A_{\mathrm{LNA}}}{2}\big(y_{\mathrm{BB}}^{\mathrm{Rx}}[n] + \eta_{\mathrm{BB}}[n]\big) + \frac{\alpha_2 A_{\mathrm{LNA}}^2}{2}\Big(\Big|A_{\mathrm{PA}} x_{\mathrm{BB}}^{\mathrm{Tx}}[n] * h_{\mathrm{BB}}^{\mathrm{Tx,L}}[n]\Big|^2 + \big|y_{\mathrm{BB}}^{\mathrm{Rx}}[n]\big|^2 + 2\Re\big\{y_{\mathrm{BB}}^{\mathrm{Rx}}[n]\eta_{\mathrm{BB}}^*[n]\big\} + |\eta_{\mathrm{BB}}[n]|^2\Big). \quad (12)$$

Assuming $|\alpha_2| \ll 1$ and a weak received signal, the fourth, fifth and sixth terms can be neglected compared to the down-converted noise signal [16], [18]. After channel-select- and DC-filtering, the received BB signal including the IMD2 interference is given by

$$y_{\mathrm{BB}}^{\mathrm{Tot}}[n] = \alpha_1 \frac{A_{\mathrm{LNA}}}{2}\big(y_{\mathrm{BB}}^{\mathrm{Rx}}[n] + \eta_{\mathrm{BB}}[n]\big) * \bar{h}_{\mathrm{s}}[n] + \frac{\alpha_2}{2}\Big|A_{\mathrm{LNA}} A_{\mathrm{PA}} x_{\mathrm{BB}}^{\mathrm{Tx}}[n] * h_{\mathrm{BB}}^{\mathrm{Tx,L}}[n]\Big|^2 * \bar{h}_{\mathrm{s}}[n]. \quad (13)$$

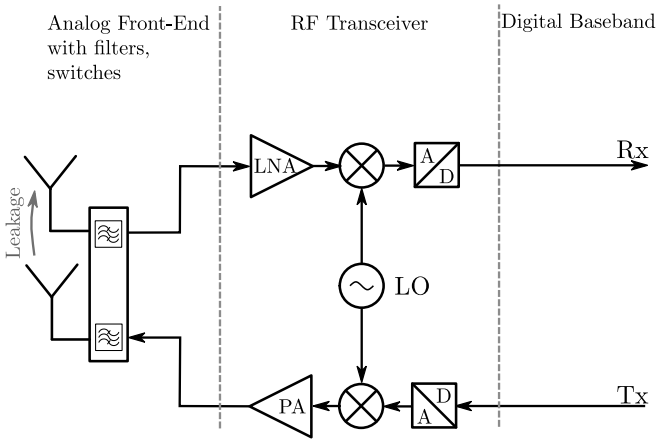

FIGURE 2. Simplified block diagram of an in-band full duplex transceiver [7].

The IMD2 interference, which is represented by the second term in (13), is mainly characterized by the squared envelope of the leaked Tx signal. It can also be expressed as [16]

$$y_{\mathrm{BB}}^{\mathrm{IMD2}}[n] = y_{\mathrm{BB}}^{\mathrm{IMD2,I}}[n] + j\, y_{\mathrm{BB}}^{\mathrm{IMD2,Q}}[n] = \xi \left| x_{\mathrm{BB}}^{\mathrm{Tx}}[n] * h[n] \right|^2 * \bar{h}_{\mathrm{s}}[n]\,, \tag{14}$$

with $h[n] = \sqrt{\alpha_2^{\mathrm{I}}/2}\, A_{\mathrm{LNA}} A_{\mathrm{PA}} h_{\mathrm{BB}}^{\mathrm{Tx,L}}[n]$ and $\xi = 1 + j\,\alpha_2^{\mathrm{Q}}/\alpha_2^{\mathrm{I}}$.

D. IN-BAND FULL DUPLEX SI

As a relevant SI problem for an in-band full duplex transceiver, we chose the case of I/Q imbalance coupled with a non-ideal PA, as has been presented in [7], [19], [20], and is repeated here for clarity. A simplified block diagram of a full duplex transceiver with shared LO is shown in Fig. 2.

As with the previous cases, $x_{\mathrm{BB}}^{\mathrm{Tx}}$ represents the complex-valued Tx samples in the BB, in front of the digital-to-analog converter (DAC). After digital-to-analog conversion, the Tx signal is up-converted by the IQ-mixer, the digital BB equivalent signal, after I/Q imbalance has been introduced by the mixer, can be represented as [19]

$$x_{\mathrm{IQ}}^{\mathrm{Tx}}(t) = K_1 x_{\mathrm{BB}}^{\mathrm{Tx}}(t) + K_2 \left(x_{\mathrm{BB}}^{\mathrm{Tx}}(t)\right)^*, \tag{15}$$

with $K_1, K_2 \in \mathbb{C}$ and, usually, $K_1 \gg K_2$. This signal now serves as an input to the non-ideal PA, which introduces non-linearities that can be modelled using a parallel Hammerstein model [7], [19]

$$x_{\mathrm{PA}}^{\mathrm{Tx}}(t) = \sum_{\substack{p=1 \\ p\,\mathrm{odd}}}^{P} h_{\mathrm{PA},p}(t) * \left(x_{\mathrm{IQ}}^{\mathrm{Tx}}(t) \left| x_{\mathrm{IQ}}^{\mathrm{Tx}}(t) \right|^{p-1} \right), \tag{16}$$

where the memory length of the PA is given by M, and $h_{\mathrm{PA},p}$ denotes the p^{th} order nonlinearities impulse response. Further, assuming an ideal LNA, down-conversion mixer and ADC, the leaked Tx signal in the Rx digital BB can be modelled as [7], [19]

$$y_{\mathrm{BB}}^{\mathrm{Tx,L}}[n] = \sum_{l=0}^{L-1} h_{\mathrm{BB}}^{\mathrm{Tx,L}}[l]\, x_{\mathrm{PA}}^{\mathrm{Tx}}[n-l]\,, \tag{17}$$

with $h_{\mathrm{BB}}^{\mathrm{Tx,L}}$ denoting the SI channel impulse response with a memory length of L. Therefore, the overall signal model can be expressed as [7], [19], [20]

$$y_{\mathrm{BB}}^{\mathrm{Tx,L}}[n] = \sum_{\substack{p=1 \\ p\,\mathrm{odd}}}^{P} \sum_{q=0}^{p} \sum_{m=0}^{M+L-1} h_{p,q}[m] \left(x_{\mathrm{BB}}^{\mathrm{Tx}}[n-m]\right)^q \left(\left(x_{\mathrm{BB}}^{\mathrm{Tx}}[n-m]\right)^{p-q} \right)^*, \tag{18}$$

where $h_{p,q}$ models the channel combining $K_1, K_2, h_{\mathrm{PA},p}$ and $h_{\mathrm{BB}}^{\mathrm{Tx,L}}$.

III. STATE-OF-THE-ART

In this section we firstly review current state-of-the-art (SOTA) SIC architectures before discussing the advantages and drawbacks of ML powered SIC.

A. SELF-INTERFERENCE CANCELLATION

With regard to the considered problem of SIC, current solutions include traditional as well as machine learning based architectures. The former usually rely on some sort of (most often adaptive) filter and/or additional analog hardware in order to provide a proper estimate of the SI signal. The latter usually makes use of NNs, kernel based methods or, highly specialized architectures acquired via deep-unfolding. This section serves as a review of all these architectures and highlights advantages as well as drawbacks of the respective solutions.

1) TRADITIONAL SIC

As mentioned in the introduction, traditional SIC – in the context of this paper – refers to methods that make use of modeling the interference signal appropriately and then try to build an (adaptive) filter that is able to estimate this model with sufficient accuracy. Typically, these architectures consist either of only digital processing in the BB (fully-digital) or make use of an additional auxiliary (aux) analog receive path (mixed-signal/hybrid).

Fully-digital solutions include adaptive filter based architectures [21]–[27] as well as least-squares based system models [18], [28], [29]. Generally speaking, adaptive filter based solutions tend to be less computationally complex (with the exception of recursive least squares (RLS) filters) than least-squares based ones. Nevertheless, all architectures provide sufficient cancellation performance and good to adequate convergence behavior (in case of adaptive filters being used).

Mixed-signal and hybrid architectures [17], [30]–[35] yield excellent convergence performance and good to excellent cancellation performance. The downside of this type of solutions is, of course, the need for additional analog hardware, adding to the chip space, power and tuning needed for proper operation.

While all of these methods work very well for SIC, their main drawback is that different solutions are needed for different SI effects (e.g., IMD2, Tx harmonics, modulated spurs,

etc.). As the number of SI effects increases with the race to more throughput and more complex transceiver architectures, one can imagine that quite a bit of chip area is needed to deploy solutions for all possible SI effects.

2) ML-BASED SIC

In order to ease the burden to have different SIC architectures for different SI effects, researchers tried to find the one-for-all solution in machine learning, especially NNs. These have been shown to yield better performance or even to surpass traditional estimators for various SI effects [5]–[9], [36], [37]. More complex ML models like kernel powered estimators have been proven to result in proper SIC for multiple different SI signals [38]–[42], at the cost of adding tremendous complexity to the solution.

With that in mind, it seems that NNs seem to be promising candidates for SIC (and system identification in general) with minimal complexity compared to other ML methods. The big, so far unsolved, problem for these kind of estimators is, however, how to adapt the weights and biases in a way that does not require a huge amount of computational power on the device itself. Previous work [5]–[7], [36] usually makes use of Adam [43] or AdaDelta [44] for training the NN on a given system. The problem with these algorithms is, though, that they require significantly more operations (additions, divisions, multiplications) per processed sample than simpler algorithms like vanilla stochastic gradient descent (SGD), hence they are unsuitable candidates for a real-life implementation. Of course standard SGD could be used, but – as will be seen in the simulation section of this work – the performance heavily fluctuates with the underlying system parameters and generally results in worse performance than other algorithms. All of this combined makes current NN based solutions rather poor choices for practical implementations as the estimator needs to be able to track changes in the system.

One way to enable the NN-based estimator to track changes in the system, is to decouple the (usually less time-invariant) nonlinear and linear parts of the underlying model and only use a least mean squares (LMS) filter for the linear part, while the nonlinear one is still modeled by an NN [5], [45]. This generally minimizes the size of the NN and therefore the update complexity, significantly undercutting the update complexity of using one NN for the whole system [5]. However, this approach – obviously – requires more in-depth knowledge of the system and is therefore not as flexible.

B. NN HARDWARE

Researchers came up with a variety of hardware (HW) implementations of neural networks, even though they are mostly focused on inference and not on adapting the weights and biases of the NN itself. In [8], [9], [37] the authors show that their NN model can surpass traditional estimators in both, cancellation performance and throughput, however, the presented architecture might experience performance issues if the NN is chosen bigger than just one hidden- and output layer. Other proposals include quantizing the weights of the NN to powers of two [10] (and therefore avoiding multipliers in hardware), compression [11] or other methods and/or specialized hardware [11]–[13]. However, while all of these works describe inference of the NN, none of them solves the issue of having to train the NN on the field programmable gate array (FPGA) or the application-specific integrated circuit (ASIC) itself. This therefore remains an open research topic. As already mentioned, this is largely due to the increased computational complexity required for training with state-of-the-art algorithms compared to inference, which is relatively cheap to compute in comparison.

IV. PROPOSED ARCHITECTURES

Similar to past publications [5]–[7], we propose the usage of a real-valued NN as the main estimator of the SI signal. However, the architecture is extended by one (or two, in the case of the second architecture) building block(s) which are updating the parameters of the NN in order to be able to adapt to the ever-changing system in terms of both, parameters and SI effects.

A. ARCHITECTURE I

The first architecture proposal can be seen in Fig. 3, where the Tx samples are collected in a tapped delay line (TDL) to obtain the vector

$$\mathbf{x}_{\mathrm{TDL}}[n] = \begin{bmatrix} x[n] \\ x[n-1] \\ \vdots \\ x[n-P] \end{bmatrix}, \tag{19}$$

with P being the length of the TDL. The output is then converted from complex to real valued signals via stacking the real and imaginary parts, i.e.,

$$\mathbf{x}_{\mathrm{NN,\,in}}[n] = \begin{bmatrix} \Re\{\mathbf{x}_{\mathrm{TDL}}[n]\} \\ \Im\{\mathbf{x}_{\mathrm{TDL}}[n]\} \end{bmatrix}, \tag{20}$$

where $\Im\{\cdot\}$ denotes the imaginary operator. This signal then serves as input to the NN which estimates the interference effect. Afterwards, the output of the NN is scaled and again put together to a complex scalar. The whole process can be summarized as

$$\hat{y}_{\mathrm{BB}}^{\mathrm{AC}}[n] = \gamma_{\mathrm{norm}}\big[1, j\big]\mathbf{f}_{\mathrm{NN}}\big(\mathbf{x}_{\mathrm{NN,\,in}}[n]\big) \tag{21}$$

$$= \gamma_{\mathrm{norm}}\big[1, j\big]\hat{\mathbf{y}}[n], \tag{22}$$

where γ_{norm} is the de-normalization constant and $\mathbf{f}_{\mathrm{NN}}(\cdot) : \mathbb{R}^M \rightarrow \mathbb{R}^2$ denotes the operations of the NN.

The resulting estimate $\hat{y}_{\mathrm{BB}}^{\mathrm{AC}}[n]$ is then subtracted from the total Rx signal which serves as input to the adaptive algorithm. The NN parameters are tuned into the direction of the negative gradient of the cost function, i.e.,

$$\boldsymbol{\theta}_i[n+1] = \boldsymbol{\theta}_i[n] - \mu \frac{\partial J\big(\mathbf{y}[n], \hat{\mathbf{y}}[n]\big)}{\partial \boldsymbol{\theta}_i[n]}, \tag{23}$$

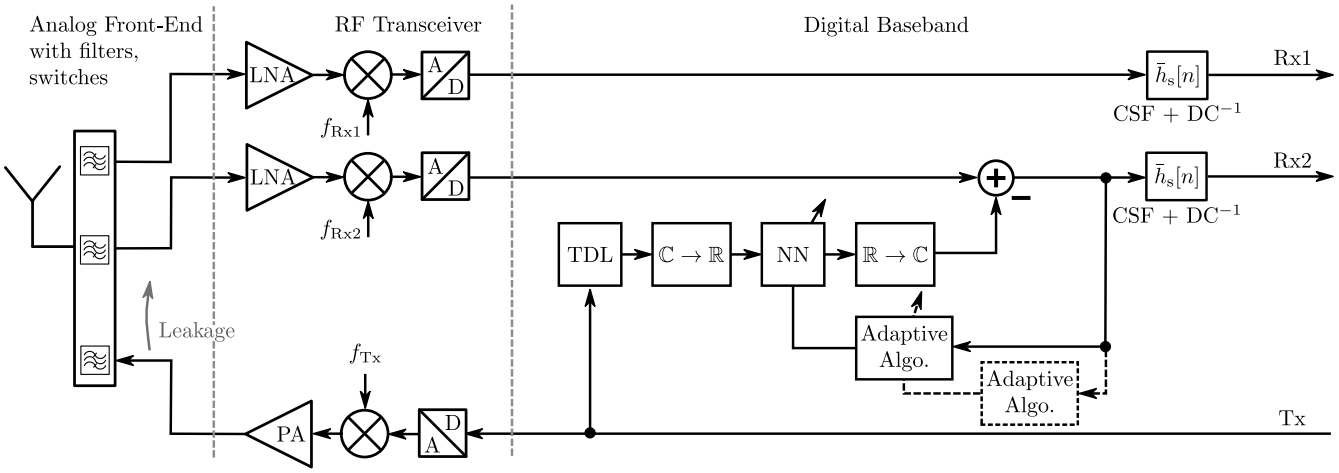

FIGURE 3. Block diagram of an LTE-A CA transceiver with NN based interference mitigation. The dashed lines indicate additional components for architecture II.

where $\boldsymbol{\theta}_i$ denotes the parameters of the NN,[1] the vector $\mathbf{y}$ contains the (complexified) Rx signal, $\hat{\mathbf{y}}$ represents the estimated SI signal, μ is the learning rate (LR), and $J(\cdot)$ is the cost function which is usually the mean squared error (MSE) in the context of this paper (but can of course be replaced with whatever function works best for a given problem). The MSE is defined as

$$J\big(\mathbf{y}[n], \hat{\mathbf{y}}[n]\big) = \frac{1}{N}\sum_{k=0}^{N-1}\big(y_k[n] - \hat{y}_k[n]\big)^2, \tag{24}$$

with the subscript k denoting the k^{th} element of the corresponding vector and N being the number of outputs of the NN (which equals to two in the context of this paper).

B. ARCHITECTURE II

As will be seen later, simply using an NN within architecture I works perfectly fine, but the performance directly depends on the chosen LR. Further, different problems and situations require different LR values to reach their full potential. The second architecture (cf. Fig. 3 including the dashed lines) operates similar to the first one, with the exception of an additional adaptive filter that controls the parameters (mainly the step-size) in (23). Therefore, this addition ensures that the resulting LR of the algorithm adapting the NN is as close as possible to the ideal step-size. Of course this additional block requires an LR of its own, however, it has been shown [46] that this type of architecture is less sensitive to choosing the perfect LR compared to the first architecture. Further, it is possible to stack multiples of these adaptation blocks to end up with more and more tuned LRs. This further eases the need to find the one LR for which the architecture yields the best result possible.

Additionally, this method allows to easily adapt multiple parameters of the first update algorithm on a per parameter basis. That is, every parameter $\boldsymbol{\theta}_i$ of the NN has its own, unique LR μ_i, i.e.,

$$\boldsymbol{\theta}_i[n+1] = \boldsymbol{\theta}_i[n] - \mu_i[n]\frac{\partial J\big(\mathbf{y}[n], \hat{\mathbf{y}}[n]\big)}{\partial \boldsymbol{\theta}_i[n]}. \tag{25}$$

This is achieved by adapting $\mu_i[n]$ according to [46]

$$\mu_i[n+1] = \mu_i[n] - \kappa_i\frac{\partial J\big(\mathbf{y}[n], \hat{\mathbf{y}}[n]\big)}{\partial \mu_i[n]}. \tag{26}$$

As will be seen, this further improves performance as the influence of individual parameters on the overall performance is considered in the tuning of the LRs.

V. PROPOSED ALGORITHMS

To achieve low-cost and robust online adaptation of the proposed architectures, current state-of-the-art algorithms need to either be extended (in the case of SGD) or simplified (in the case of RMSProp) to allow reasonable complexity while still delivering the required performance. We therefore introduce two novel extensions/modifications to these algorithms in this section.

A. NORMALIZED SGD

The first obvious choice for anyone attempting to implement a low-complexity online learning NN is SGD, as it is the most simple of all available gradient-based training algorithms. However, it is well known, that its performance heavily depends on the chosen step-size and initialization of the network (as will be shown in the simulation section). To counteract such behavior, traditional methods like the LMS adaptive filter make use of a normalization of the update equation. However, applying the same to NNs can be tricky, if possible at all. Therefore, we propose three ways to normalize the SGD (NSGD) algorithm at minimal additional cost.

1. E.g., for a 2-layered fully connected NN with no activation functions (calculating $\hat{\mathbf{y}} = \mathbf{W}_2(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2$), the optimization (23) would be called four times with $i \in \{1, \ldots, 4\}$, $\boldsymbol{\theta}_1 = \mathbf{W}_1$, $\boldsymbol{\theta}_2 = \mathbf{b}_1$, $\boldsymbol{\theta}_3 = \mathbf{W}_2$ and $\boldsymbol{\theta}_4 = \mathbf{b}_2$.

1) ALGORITHM OVERVIEW

A standard SGD update for the parameter $\boldsymbol{\theta}_i$ is given by

$$\boldsymbol{\theta}_i[n+1] = \boldsymbol{\theta}_i[n] - \mu \frac{\partial J\big(\mathbf{y}[n], \hat{\mathbf{y}}[n]\big)}{\partial \boldsymbol{\theta}_i[n]} \tag{27}$$

$$= \boldsymbol{\theta}_i[n] - \mu \nabla_{\boldsymbol{\theta}_i} J\big(\mathbf{y}[n], \hat{\mathbf{y}}[n]\big). \tag{28}$$

That is, the parameter $\boldsymbol{\theta}_i$ is updated towards the negative direction of the gradient of the cost function w.r.t. the parameter. This formulation coincides with the standard update equation of the LMS adaptive filter, where it is commonly known that its normalized variant yields better results and more stable behavior [47].

Therefore, we attempt to follow the same principle as for the LMS in order to normalize the update equations of any given NN. One way is to approximate the first order Taylor series of the error [48]–[50]

$$\mathbf{e}[n+1] \approx \mathbf{e}[n] + \left.\frac{\partial \mathbf{e}}{\partial \boldsymbol{\theta}_i}\right|_{\boldsymbol{\theta}_i = \boldsymbol{\theta}_i[n-1]} (\boldsymbol{\theta}_i[n] - \boldsymbol{\theta}_i[n-1]) \,, \tag{29}$$

with the error $\mathbf{e}[n] = \mathbf{y}[n] - \hat{\mathbf{y}}[n]$. Doing so commonly results in

$$\mathbf{e}[n+1] \approx \mathbf{e}[n] - 2\mu \mathbf{e}[n] \left\| \nabla_{\boldsymbol{\theta}_i} \mathbf{e}[n] \right\|_2^2. \tag{30}$$

Isolating the error, enforcing that $\mathbf{e}[n+1] < \mathbf{e}[n]$ and solving for the LR results in the update equation

$$\boldsymbol{\theta}_i[n+1] = \boldsymbol{\theta}_i[n] - \frac{\bar{\mu} \nabla_{\boldsymbol{\theta}_i} J\big(\mathbf{y}[n], \hat{\mathbf{y}}[n]\big)}{\epsilon + \left\| \nabla_{\boldsymbol{\theta}_i} \mathbf{e}[n] \right\|_2^2} \tag{31}$$

$$0 \leq \bar{\mu} < 1 \,, \tag{32}$$

where the parameter ϵ is a small positive constant in case the normalization of the error is close to (or is) zero. This concept works nicely if the parameter $\boldsymbol{\theta}_i$ is a scalar or vector, as all derivatives are defined. In this case, a similar result to [48]–[50] is achieved for the normalized LR.[2] If $\boldsymbol{\theta}_i$ is a matrix, however, the derivative of the error vector w.r.t. the matrix $\boldsymbol{\theta}_i$ is not defined, and this approach cannot be used like described above. Similarly, the approaches presented in [48], [49] also break down if the parameter $\boldsymbol{\theta}_i$ has more than one dimension. In this case, other methods need to be applied in order to get the desired result. One way to do this (if the NN structure permits it) is to vectorize the matrix $\boldsymbol{\theta}_i$ (and change the equations accordingly). E.g., for a fully connected layer (without activation function), an estimate can be defined by

$$\hat{\mathbf{y}} = \mathbf{W}\mathbf{x} + \mathbf{b} \tag{33}$$

$$= \tilde{\mathbf{X}}\tilde{\mathbf{w}} + \mathbf{b} \,, \tag{34}$$

with the parameters $\mathbf{W}$ and $\mathbf{b}$ being the parameters of the NN we would like to update. Obviously the formulation in (33) would not allow to compute (29) as $\mathbf{W}$ is a matrix. Therefore, the alternative formulation (34) can be used where the vector $\tilde{\mathbf{w}}$ and matrix $\tilde{\mathbf{X}}$ are defined as

$$\tilde{\mathbf{w}} = \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \vdots \\ \mathbf{w}_N \end{bmatrix} \tag{35}$$

$$\tilde{\mathbf{X}} = [\mathbf{I}_N x_1, \mathbf{I}_N x_2, \ldots, \mathbf{I}_N x_N] \,, \tag{36}$$

with $\mathbf{w}_k$ and x_k being the k^{th} column vector of the matrix $\mathbf{W}$ and the k^{th} element of the vector $\mathbf{x}$, respectively, and $\mathbf{I}_N$ denotes the identity matrix of size N.

Of course the application of this trick is not always possible, e.g., for the kernel $\mathbf{K}$ of a convolutional layer or when using some framework's autograd[3] function. In this case one can apply another method called joint normalization [50], [51]. This method derives separate bounds for each element of the error vector (or separate bounds for its real and imaginary part), i.e.,

$$e_k[n+1] \approx e_k[n] + \left.\frac{\partial e_k}{\partial \boldsymbol{\theta}_i}\right|_{\boldsymbol{\theta}_i = \boldsymbol{\theta}_i[n-1]} (\boldsymbol{\theta}_i[n] - \boldsymbol{\theta}_i[n-1]), \tag{37}$$

such that the error is always a scalar, and also the derivative of it w.r.t. to a matrix is defined. In this case, the update equation becomes

$$\boldsymbol{\theta}_i[n+1] = \boldsymbol{\theta}_i[n] - \frac{\bar{\mu} \nabla_{\boldsymbol{\theta}_i} J\big(\mathbf{y}[n], \hat{\mathbf{y}}[n]\big)}{\epsilon + \sum_i \sum_k \left\| \nabla_{\boldsymbol{\theta}_i} e_k[n] \right\|_2^2}. \tag{38}$$

2) ALTERNATIVE FORMULATIONS

Alternatively to the formulation in (38), one can also use the maximal normalization term across all parameters $\boldsymbol{\theta}_i$ and all errors e_k, as this serves as a less pessimistic upper bound

$$\boldsymbol{\theta}_i[n+1] = \boldsymbol{\theta}_i[n] - \frac{\bar{\mu} \nabla_{\boldsymbol{\theta}_i} J\big(\mathbf{y}[n], \hat{\mathbf{y}}[n]\big)}{\epsilon + \max\limits_{i,k} \left\{ \left\| \nabla_{\boldsymbol{\theta}_i} e_k[n] \right\|_2^2 \right\}}. \tag{39}$$

Another way to formulate the normalization is to just take the sum over all errors e_k separately for each parameter $\boldsymbol{\theta}_i$, resulting in

$$\boldsymbol{\theta}_i[n+1] = \boldsymbol{\theta}_i[n] - \frac{\bar{\mu} \nabla_{\boldsymbol{\theta}_i} J\big(\mathbf{y}[n], \hat{\mathbf{y}}[n]\big)}{\epsilon + \sum_k \left\| \nabla_{\boldsymbol{\theta}_i} e_k[n] \right\|_2^2}. \tag{40}$$

This last formulation, however, has proven to be quite LR sensitive (i.e., the NN can become unstable if the LR has been chosen just a little bit too high).

B. RMSPROP^{MAX}

As SGD and other variants of this algorithm (e.g., AdaGrad [52]) all have their drawbacks despite, or because, being so simple in a computational sense, the authors of [53] came up with RMSProp, which aims to normalize the update equation on a per-entry basis, i.e., each and every entry of the parameter $\boldsymbol{\theta}_i$ has an individual normalization term in its

2. See, e.g., [49, eq. (20)], [48, eq. (14)] or [50, eqs. (21) and (26)].

3. i.e., automatic generation of the gradient by some framework.

update equation. Standard RMSProp updates the parameters according to

$$\mathrm{E}\left[\mathbf{g}_i^2\right]_n = \alpha \mathrm{E}\left[\mathbf{g}_i^2\right]_{n-1} + (1-\alpha)\left(\nabla_{\boldsymbol{\theta}_i} J\right)^2 \tag{41}$$

$$\boldsymbol{\theta}_i = \boldsymbol{\theta}_i - \frac{\mu}{\sqrt{\epsilon + \mathrm{E}\left[\mathbf{g}_i^2\right]_n}} \nabla_{\boldsymbol{\theta}_i} J\,, \tag{42}$$

where $\mathrm{E}[\,\cdot\,]$ denotes the expectation, $\mathbf{g}_i = \nabla_{\boldsymbol{\theta}_i} J$ and α is the forgetting factor. Note that all operations (especially the squaring and the division of μ) are element-wise, meaning that each and every element of the gradient (and $\boldsymbol{\theta}_i$) requires at least one such operation. Naturally, this can quickly skyrocket the required number of operations if the NN becomes big enough (see Section VI for details). As it turns out, this approach yields better results than its (less complex) predecessors, at the cost of a complexity explosion.

Since RMSProp tends to be the least computationally intensive one after SGD while still working quite well in practice, it was the next obvious choice for the problem at hand. In order to ease the computational burden, we propose a slight change to the update equation in order to bring down the number of operations needed to update the parameters each time-step.

1) ALGORITHM OVERVIEW

We propose a simple change to (41) which reduces the number of divisions to an absolute minimum. Similar to the idea of joint normalization, we only take the maximum element of the squared gradient and only consider this for the expectation calculation

$$\mathrm{E}\left[\mathbf{g}_i^2\right]_n = \alpha \mathrm{E}\left[\mathbf{g}_i^2\right]_{n-1} + (1-\alpha)\max\left\{\left(\nabla_{\boldsymbol{\theta}_i} J\right)^2\right\}. \tag{43}$$

This approach implies that if the algorithm converges for elements of $\mathrm{E}[\mathbf{g}_i^2]_n$ in (41) being smaller than the value obtained via (43), it will also has to converge if just the maximum element of $\mathrm{E}[\mathbf{g}_i^2]_n$ is taken to normalize the LR. Obviously, this could affect convergence speed compared to the vanilla implementation as the estimate of the squared gradient used to normalize the LR is always the highest possible value. Therefore, the LR is decreased (or increased if $\mathrm{E}[\mathbf{g}_i^2]_n < 1$) in a pessimistic way. This guarantees that the algorithm always converges if the original implementation would converge. By doing so, $\mathrm{E}[\mathbf{g}_i^2]_n$ reduces from a vector or matrix to a scalar. The update (42) remains unchanged. Note, however, that the division now only needs to be carried out once per parameter $\boldsymbol{\theta}_i$ instead of for each element. Another important observation is, that this change to the algorithm introduces absolutely no additional operations compared to the original one, unlike the normalization presented in the previous section. The gradient needs to be calculated anyway in order to do some form of gradient descent, and tracking some part of it is also done either way by the original RMSProp.

2) ALTERNATIVE FORMULATIONS

A less pessimistic formulation of the simplification introduced in the previous section is to leave (41) unchanged and to update (42) to

$$\boldsymbol{\theta}_i = \boldsymbol{\theta}_i - \frac{\mu}{\sqrt{\epsilon + \max\left\{\mathrm{E}\left[\mathbf{g}_i^2\right]_n\right\}}} \nabla_{\boldsymbol{\theta}_i} J. \tag{44}$$

Doing so allows to track the importance of individual element's contribution to the gradient rather than tracking the overall maximum value of all elements of the gradient. This can then result in a lower overall maximum value than when tracking just the overall maximum, therefore possibly increasing the step-size (denominator is smaller) and increasing convergence speed.

If the NN, and therefore the parameters, are complex valued, this proposed approach clearly does not work as the $\max\{\cdot\}$ operation is not defined in the complex domain. In this case one can adapt either of the two proposed approaches to read as

$$\mathrm{E}\left[\mathbf{g}_i^2\right]_n = \alpha \mathrm{E}\left[\mathbf{g}_i^2\right]_{n-1} + (1-\alpha)\max\left\{\left|\left(\nabla_{\boldsymbol{\theta}_i} J\right)^2\right|^2\right\} \tag{45}$$

$$\boldsymbol{\theta}_i = \boldsymbol{\theta}_i - \frac{\mu}{\sqrt{\epsilon + \max\left\{\left|\mathrm{E}\left[\mathbf{g}_i^2\right]_n\right|^2\right\}}} \nabla_{\boldsymbol{\theta}_i} J\,, \tag{46}$$

where $|\cdot|^2 = \Re\{\cdot\}^2 + \Im\{\cdot\}^2$ is the element-wise squared absolute operation. Note that taking only one of the two equations is sufficient, the other one remains unchanged as in (41) and (42). In this case, the complexity clearly increases compared to the real-valued case, however, the number of divisions is still kept at a minimum.

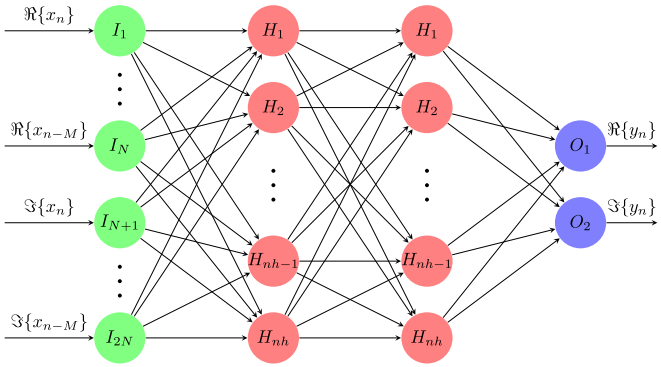

FIGURE 4. The neural network used for SI cancellation, with one input layer (green), two hidden layers (red) and one output layer (blue).

VI. COMPLEXITY

Before evaluating all approaches for their suitability for SIC, this section derives the computational complexity of the proposed architecture, NSGD and RMSProp$^{\text{Max}}$. This is firstly done in a general sense before the specific numbers used in the simulations are evaluated to get a more precise picture on how complex the proposed architectures are compared to current state-of-the-art approaches.

A. NEURAL NETWORK GENERAL

As this work only considers feed-forward (or dense) layers/NNs with the general architecture shown in Fig. 4, we

are only interested in the complexity of such an architecture as well. Of course the investigation can be expanded trivially to more complex layers like convolutional or LSTM [54], which are commonly found in CV and NLP tasks. As stated in Section V-A, a simple dense layer calculates

$$\hat{\mathbf{y}} = \mathbf{f}_{\text{act}}(\mathbf{W}\mathbf{x} + \mathbf{b}), \tag{47}$$

with $\mathbf{W} \in \mathbb{R}^{N\times M}$, $\mathbf{b} \in \mathbb{R}^N$ and $\mathbf{f}_{\text{act}} : \mathbb{R}^N \to \mathbb{R}^N$ being the weight, bias and activation function of the layer. The output can serve as the input for another layer, such that multiple layers can be followed one after another forming an NN. The activation function can take a wide variety of forms. For the sake of simplicity (and because that is the activation function used in the simulations in this work) we only consider the ReLU[4] in this derivation. Due to its simplicity it is widely used in practice and can be considered to result in no extra operations necessary to compute it (as it can be realized with a simple multiplexer in hardware). Therefore, the required number of operations for a single estimated signal are

$$n_{\text{mult,fp}} = NM \tag{48}$$

$$n_{\text{add,fp}} = NM, \tag{49}$$

for the multiplications and additions, respectively, per layer of the NN. Therefore, the total number of operations required for both, additions and multiplications in the forward path, is

$$n = \sum_i N_i N_{i+1}, \tag{50}$$

with $N_0 = M$ and $i = 0 \ldots L-1$, where L is the total number of layers.

B. NEURAL NETWORK PARAMETER UPDATE

In addition to this, the gradient of the cost function w.r.t. the input needs to be calculated at every time-step. Unlike the forward pass, this not only depends on the chosen NN architecture, but also on the cost function and is usually done using the chain rule (i.e., back propagation) to re-use as many computations as possible for each parameter. Note that this calculation needs to be done regardless of which optimization algorithm is used and therefore is the bare-minimum required if online learning is desired (i.e., for standard SGD). In case of a fully-connected NN with only ReLU as activation function, many operations from the forward path can be saved (as they appear the same in the gradient). The additional number of operations for calculating the gradients is

$$n_{\text{mult,bp}} = \sum_{i=1}^{L-1} N_L N_i + \sum_{i=0}^{L-1} N_i + N_{i+1} \tag{51}$$

$$n_{\text{add,bp}} = \sum_{i=1}^{L-1} N_i. \tag{52}$$

4. Calculating $\mathbf{f}_{\text{act}}(\mathbf{x}) = \mathbf{max}\{\mathbf{0}, \mathbf{x}\}$.

TABLE 1. Complexity in terms of multiplications, additions and divisions of the considered algorithms for the specific implementations considered in this work.

Algorithm	Mult.	Add.	Div.
SGD	870	435	–
NSGD	1341	860	1
RMSProp$^{\text{Max}}$	878	1325	6
RMSProp	2183	1360	437
IM2LMS [26]	332	274	2
IM2RLS [24]	10458	7602	52
KRLS [41]	> 67000	> 71000	> 125
SW-KRLS [42]	$4.5 \cdot 10^6$	$5 \cdot 10^6$	2
H2-NLMS [27]	160	160	1
MB-LMS [23]	96	144	9
Mixed-Signal [35][4]	384	352	1
Hybrid [17][4]	100	96	1
LSNN-base [7][5]	476	492	–
Poly-base [7][5]	741	492	–

Lastly, the cost of updating the parameters according to one of the two presented algorithms is

$$n_{\text{mult,nsgd}} = \sum_i N_i + \sum_i N_i N_{i+1} \tag{53}$$

$$n_{\text{add,nsgd}} = N_L + 2L + \sum_i (N_i - 1)(N_{i+1} - 1) + \sum_i (N_i - 1) \tag{54}$$

$$n_{\text{div,nsgd}} = 1. \tag{55}$$

for NSGD via (38), assuming that the derivative of the errors w.r.t. the parameters has already been computed and stored during the gradient calculations. Lastly, the update for RMSProp$^{\text{Max}}$ requires

$$n_{\text{mult,rmspm}} = 6N_L + \sum_i N_i - 1N_{i+1} \tag{56}$$

$$n_{\text{add,rmspm}} = 2N_L + \sum_i N_i - 1N_{i+1} \tag{57}$$

$$n_{\text{div,rmspm}} = 2N_L. \tag{58}$$

operations with (44). The complexity for the specific architectures considered in the simulation section are given in Table 1 (for IMD2 in case of the proposed architecture and the respective cases for problem specific algorithms).

It can be seen that the proposed approach, especially via RMSProp$^{\text{Max}}$, can save a tremendous amount of complexity and is comparable to current state-of-the-art architectures. Regardless of the update algorithms chosen, the proposed solution still requires less operations per sample than the IM2RLS and KRLS approaches.

VII. SIMULATIONS

Having derived the proposed algorithms and corresponding architectures, this section evaluates them on their performance on IMD2 and Tx harmonics cancellation, and further compares them to current state-of-the-art methods for SIC. The main metric of performance is the normalized

TABLE 2. Simulation settings used for all considered algorithms and test cases.

Algorithm	μ_1	μ_2	ϵ	$\mathbf{P}_0$	λ	α	M_w	P	$M+L$	n_h
SGD	0.001325	–	–	–	–	–	–	–	–	–
NSGD LLR	0.01325	–	10^{-6}	–	–	–	–	–	–	–
NSGD HLR	0.1325	–	10^{-6}	–	–	–	–	–	–	–
RMSProp$^{\mathrm{Max}}$ FLR	0.0022	–	10^{-6}	–	–	–	–	–	–	–
RMSProp$^{\mathrm{Max}}$ LRS	0.001325	–	10^{-6}	–	–	–	–	–	–	–
RMSProp$^{\mathrm{Max}}$ HO	0.001325	10^{-6}	–	–	–	–	–	–	–	–
IM2LMS [26]	0.02	–	0.1	–	–	–	–	–	–	–
IM2RLS [24]	–	–	–	$10^5\mathbf{I}$	0.9999	–	–	–	–	–
KRLS [41]	–	–	–	–	0.999	10^{-5}	–	–	–	–
SW-KRLS [42]	–	–	–	–	0.999	10^{-5}	1000	–	–	–
H2-NLMS [27]	0.0025	–	10^{-4}	–	–	–	–	–	–	–
MB-LMS [23]	$[0.003, 0.025]$	–	10^{-4}	–	–	–	–	–	–	–
Mixed-Signal [35]	–	–	–	$10^6\mathbf{I}$	0.9998	–	–	–	–	–
Hybrid [17]	0.03	–	10^{-4}	$10^5\mathbf{I}$	0.999	–	–	–	–	–
LSNN-base [7]	0.004	–	10^{-4}	–	–	–	–	7	13	17
Poly-base [7]	–	–	–	–	–	–	–	7	13	–

mean squared error (NMSE), which is defined as

$$\mathrm{NMSE}_{\mathrm{dB}} = 10\log_{10}\frac{\mathrm{E}\left[\left|y_{\mathrm{BB}}^{\mathrm{Int}}[n]-\hat{y}_{\mathrm{BB}}^{\mathrm{AC}}[n]\right|^2\right]}{\mathrm{E}\left[\left|y_{\mathrm{BB}}^{\mathrm{Int}}[n]\right|^2\right]}, \tag{59}$$

where $y_{\mathrm{BB}}^{\mathrm{Int}}[n]$ and $\hat{y}_{\mathrm{BB}}^{\mathrm{AC}}[n]$ represent the true SI signal disturbing the receiver and the estimated replica, respectively. Further, the signal to noise and interference ratio (SNIR) is defined as

$$\mathrm{SNIR}_{\mathrm{dB}} = 10\log_{10}\frac{\mathrm{E}\left[\left|y_{\mathrm{BB}}^{\mathrm{Rx}}[n]\right|^2\right]}{\mathrm{E}\left[\left|y_{\mathrm{BB}}^{\mathrm{Tot}}[n]-y_{\mathrm{BB}}^{\mathrm{Rx}}[n]-\hat{y}_{\mathrm{BB}}^{\mathrm{AC}}[n]\right|^2\right]} \tag{60}$$

which is evaluated for multiple interference-to-carrier-plus-noise (ICN) ratios, the ICN is defined as

$$\mathrm{ICN}_{\mathrm{dB}} = 10\log_{10}\frac{\mathrm{E}\left[\left|y_{\mathrm{BB}}^{\mathrm{Int}}[n]\right|^2\right]}{\mathrm{E}\left[\left|y_{\mathrm{BB}}^{\mathrm{Rx}}[n]+\eta_{\mathrm{BB}}[n]\right|^2\right]}. \tag{61}$$

The simulation parameters for all considered architectures are combined in Table 2 for all simulations and testcases. These have been hand-tuned to reach the best possible performance for the different algorithms for the considered problems. The NNs and algorithms have been implemented in PyTorch [55] and the weights and biases of the NNs have been initialized using a uniform distribution [55]. Similar to the NN depicted in Fig. 4, a NN with two hidden layers and one output layer with 15, 10 and 2 neurons, respectively, has been proven[5] to achieve excellent performance while keeping the complexity at a minimum. The hidden layers both use ReLU as an activation function. Further, while the ICN is variable for all scenarios, the Rx SNR is kept constant at 10 dB meaning that the Rx signal lies 10 dB above the noise signal which resides at −100 dBm. This constitutes the worst case for the receiver as it operates at the sensitivity limit. Additionally, the Rx signal always is allocated the maximum number of resource blocks, which is 50 for LTE10 signals, and the Tx signal allocation varies.

5. The exact number of layers, neurons and other aspects of the NN architecture generally have to be found via trial and error to find the best performing NN on any given problem. For the considered cases in this work, we use the same architecture for all problems.

A. IMD2

The first considered scenario is IMD2 interfering with the Rx signal. Fig. 5(a) shows the performance of the proposed and state-of-the-art algorithms for an ICN of 20 dB. Naturally, KRLS [41] reaches the best performance in the least amount of samples. However, the next best performers are the NN with NSGD and a high LR (HLR), RMSProp$^{\mathrm{Max}}$ with a fixed LR (FLR) and the proposed architecture II using RMSProp$^{\mathrm{Max}}$ with a variable LR (HO). In terms of steady-state performance, IM2RLS [24] reaches approximately the same NMSE value as the proposed algorithms, even though it takes at least twice as long to converge to an acceptable value of −20 dB. Further, standard SGD clearly takes longer to converge but still manages to reach better performance than IM2LMS [26]. If the LR is chosen too low (LLR), NSGD cannot reach its optimal potential and clearly performs worst in this scenario. It is worth noting that, while the underlying interference effect as well as other parameters of the specific problem might change, the proposed architecture is the only one – besides the KRLS – that stays exactly the same for all scenarios (the NN does not change and neither do other parameters). Further, it can be seen that architecture II clearly reaches one of the best performances despite its LRs not being tuned at all.

The evaluation of these methods is then carried out for multiple ICN values in order to examine the steady-state SNIR performance (for the last LTE10 slot of the simulation) in Fig. 5(b). It can be seen that while for lower ICN values the proposed architectures perform slightly worse than SOTA algorithms (within 1 dB), their advantage is clearly visible for higher interference powers, where they outperform both IM2RLS and IM2LMS in terms of SNIR improvement.

B. TX HARMONICS

The performance of the proposed architectures is shown in Fig. 5(c) for an ICN of 20 dB. It can be seen that RMSProp$^{\mathrm{Max}}$

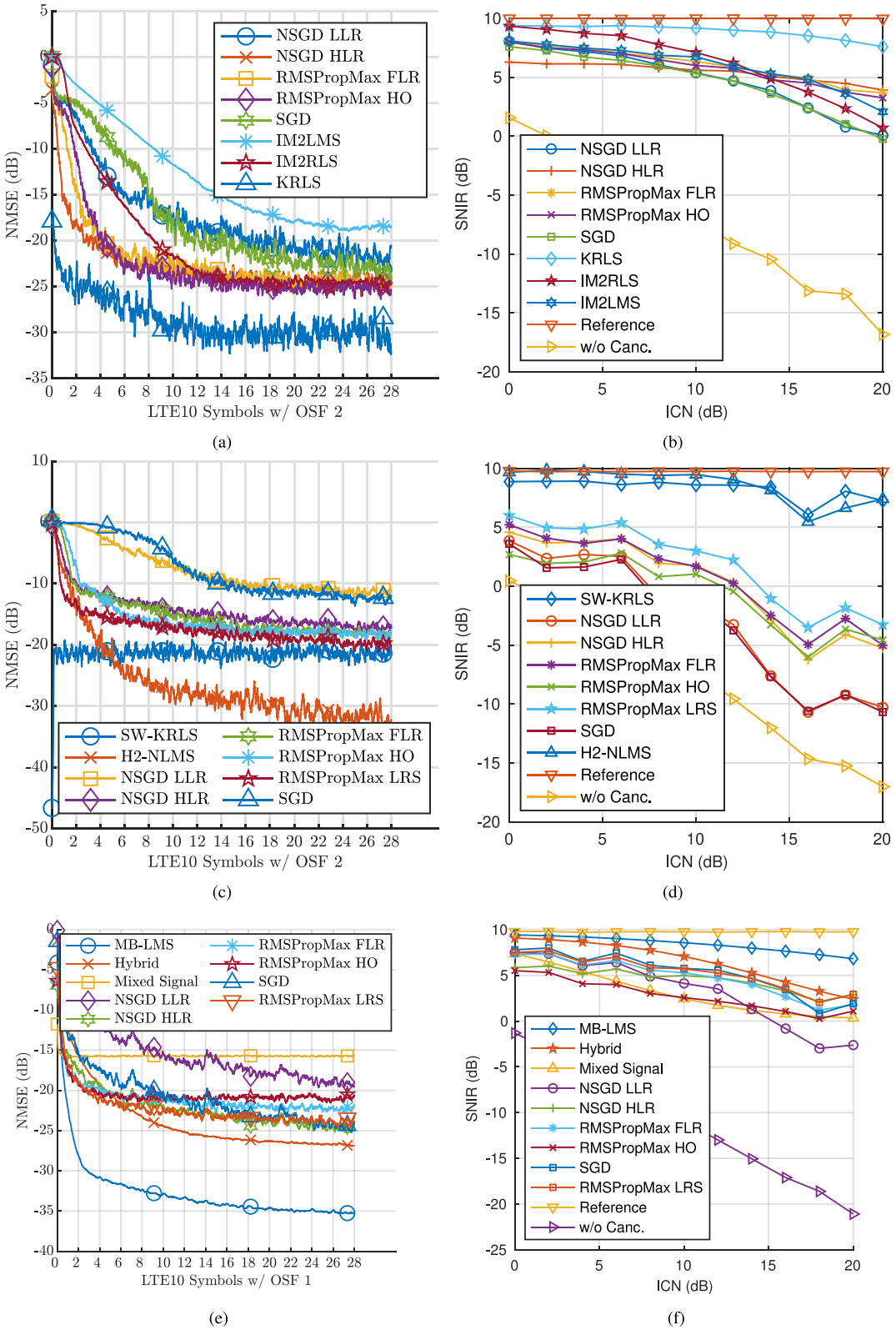

FIGURE 5. **NMSE performance of all considered architectures with an ICN of 20 dB for IMD2 (a), Tx harmonics (c) and modulated spurs (e), SNIR performance over varying ICN powers for IMD2 (b), Tx harmonics (d) and modulated spurs (f).**

with a problem specific LR (LRS) performs best among the NN based algorithms, followed by RMSProp$^{\text{Max}}$ utilized in architecture II (HO) and with a fixed LR. NSGD with a high LR reaches slightly worse NMSE values in the steady state while the low LR version yields the worst performance only slightly worse than vanilla SGD.

In terms of SNIR performance (cf. Fig. 5(d)), the problem specific LR yields the best improvement for the proposed algorithms with the other algorithms and configurations resulting in more or less the same performance as seen for the NMSE case. Again, SW-KRLS [42] and the model based variant [27] clearly perform best. Note however, that the SW-KRLS requires tremendously more operations per sample than the proposed architecture and the model based solution requires perfect model knowledge and, again, is only usable for this specific problem (second order Tx harmonics). If the interference changes (e.g., to IMD2) the model based approach would fail.

C. MODULATED SPURS

In terms of modulated spurs, the NMSE performance for all considered algorithms is shown in Fig. 5(e) for an ICN of 20 dB. For this case we compare ourselves to a mixed signal approach [35], a fully digital model based LMS (MB-LMS) [23] as well as a hybrid approach [17]. As can be seen the MB-LMS is able to surpass all other methods significantly,[6] while the proposed NN architectures result in a performance somewhere between the mixed signal approach (worst performance) and the hybrid architecture. Note that in this case, the proposed algorithms with a high LR (NSGD) and the best LR for Tx harmonics (RMSProp$^{\text{Max}}$) yield on-par performance with SGD.

With regard to the SNIR performance (see Fig. 5(f)), the same observation holds, the MB-LMS results in the best SNR recovery while the proposed architectures can only somewhat keep up with the hybrid approach. Note however, that the reference architectures are tailored to the specific problem and cannot be used to combat any other SI type, whereas our proposed NN is exactly the same as for the previous cases without any modification to the number of layers, neurons or the activation functions. Further, the chosen LRs remain the same as well. Overall, the SNR recovery is sufficient for all algorithms except NSGD with a low LR at higher ICN values.

D. IN-BAND FULL DUPLEX SI

To showcase the ability of our proposed architecture to not only being able to cope with SI effects in FDD transceivers but also being able to deal with in-band full duplex SI problems, we used the same overall setup as [7] and compare ourselves to the NN presented in this work (LSNN-base), as well as with a conventional polynomial cancellation architecture (Poly-base). Further, unlike the previous cases we do not use simulated data but measured data that has been provided by the authors of [7]. As both architectures from [7] have a training and evaluation phase, no convergence is shown in Fig. 6, but rather their performance once trained on a specific system. It can be seen that, with the exception of

6. Note however, that this approach needs a different step-size for each ICN value.

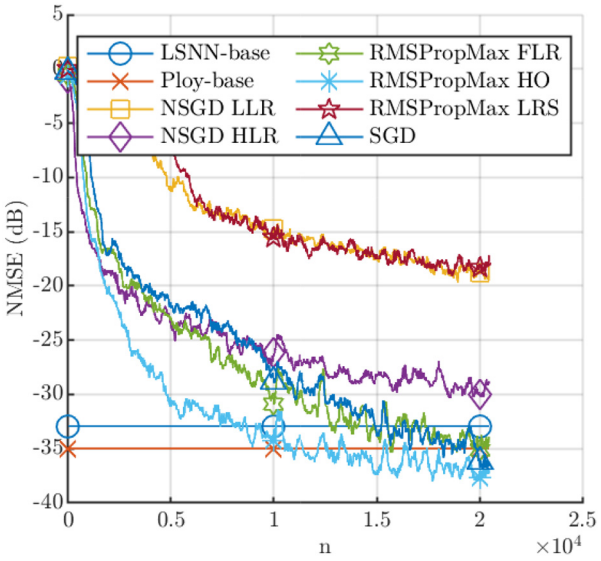

FIGURE 6. NMSE performance of all considered architectures for the in-band full duplex case.

NSGD with a low LR and RMSProp$^{\text{Max}}$ with the Tx harmonics LR, the proposed approaches can yield excellent performance and, in the case of RMSProp$^{\text{Max}}$ with adaptive LR, even surpass both reference algorithms. It is worth noting however, that it seems that most of the proposed algorithms still have not reached their steady state performance after one OFDM frame, as – unfortunately – the measurements provided by [7] do not contain more than that. Further, it has to be highlighted that the LSNN-base architecture requires a least squares estimate of the interference channel to cancel the linear part of the SI signal, while the NN afterwards only works on the nonlinear part. Our proposed architecture does not require this additional processing step and – once again – is the exact same as for the previous SI cases with the exception that the input dimension has been increased to 26 to accommodate the measurement data.

E. STEADY-STATE VARIANCE

While standard SGD is commonly used in literature [56] and even shows moderately good performance in the considered test-cases in this work, it does not constitute the most stable solution. Fig. 7(a) shows the mean NMSE performance (over the last LTE10 slot of the simulation) of all proposed algorithms for different ICN values. The LR dependency for NSGD and SGD can be seen clearly. Lower LR values work very well for low ICN values, while higher LR values yield better performance for high powered interferers. Therefore, a variable LR would be the best solution (and hence the motivation for architecture II).

Fig. 7(b) shows two box plots for different ICN values for 20 different duplexer realizations and 10 runs each. The impulse responses are based on real world duplexers for LTE communication devices, where S-parameter measurements of the stop-band have been acquired in order to fit

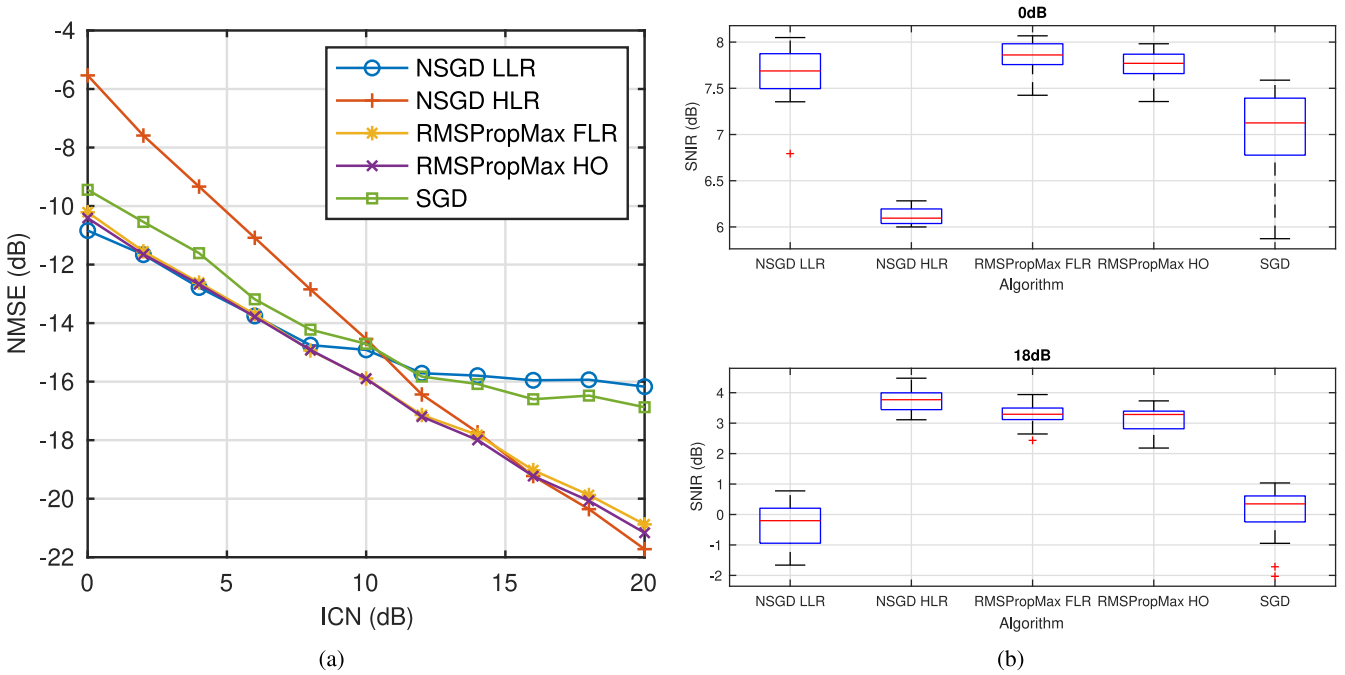

FIGURE 7. Mean NMSE performance over a wider variety of ICN values for all considered online learning algorithms for IMD2 (a); variance of the achieved NMSE value for different ICN values and online learning algorithms for IMD2 (b).

impulse responses. It can be clearly seen, that SGD, while performing reasonably good in the mean, results in a quite high variance in terms of performance, while NSGD with a high LR and both RMSProp$^{\text{Max}}$ realizations yield the best, stable solutions. Therefore, while SGD may result in less complex updates, its performance can be all over the place, being somewhat unpredictable.

VIII. CONCLUSION

We presented a novel NN based approach to cancel different self-interference effects, namely modulated spurs, Tx Harmonics , IMD2 and in-band full duplex SI. We further extended and modified existing training algorithms to yield stable, low-cost solutions, therefore finally enabling sample adaptive NNs in the field of SIC. The proposed solutions reach a considerably higher cancellation performance than all other considered competing architectures for the investigated SI interference scenarios with the exception of modulated spurs, where the proposed solution still manages to achieve satisfying performance. Especially for IMD2, the proposed methods are able to not only outperform state-of-the-art solutions in terms of NMSE and SNIR performance, but also surpass their convergence speed significantly. We further showed, that the same architecture used for three different SI effects, occurring in FDD transceivers, can be used to mitigate transceiver induced SI in an in-band full duplex scenario, surpassing current state-of-the-art solutions without the requirement to be trained offline. Lastly, it has been shown that simpler algorithms tend to result in a less predictable performance, i.e., their steady-state NMSE values can vary significantly. This issue is reduced if more complex, and stable algorithms like NSGD and RMSProp$^{\text{Max}}$ are used.

REFERENCES

[1] D. Zhang, M. M. Islam, and G. Lu, "A review on automatic image annotation techniques," *Pattern Recognit.*, vol. 45, no. 1, pp. 346–362, 2012.

[2] M. Pak and S. Kim, "A review of deep learning in image recognition," in *Proc. 4th Int. Conf. Comput. Appl. Inf. Process. Technol. (CAIPT)*, 2017, pp. 1–3.

[3] D. W. Otter, J. R. Medina, and J. K. Kalita, "A survey of the usages of deep learning for natural language processing," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 2, pp. 604–624, Feb. 2021.

[4] S. Wang and R. M. Summers, "Machine learning and radiology," *Med. Image Anal.*, vol. 16, no. 5, pp. 933–951, 2012.

[5] O. Ploder, O. Lang, T. Paireder, and M. Huemer, "An adaptive machine learning based approach for the cancellation of second-order-intermodulation distortions in 4G/5G transceivers," in *Proc. IEEE 90th Veh. Technol. Conf. (VTC Fall)*, Sep. 2019, pp. 1–7.

[6] O. Ploder, C. Motz, T. Paireder, and M. Huemer, "A neural network approach for the cancellation of the second-order-intermodulation distortion in future cellular RF transceivers," in *Proc. 53rd Asilomar Conf. Signals Syst. Comput.*, Nov. 2019, pp. 1144–1148.

[7] 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. Advances Wireless Commun. (SPAWC)*, 2018, pp. 1–5.

[8] 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.

[9] 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.

[10] F. A. Aoudia and J. Hoydis, "Towards hardware implementation of neural network-based communication algorithms," in *Proc. IEEE 20th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, 2019, pp. 1–5.

[11] E. Wang *et al.*, "Deep neural network approximation for custom hardware: Where we've been, where we're going," *ACM Comput. Surv.*, vol. 52, no. 2, pp. 1–39, May 2019. [Online]. Available: https://doi.org/10.1145/3309551

[12] S. Jung and S. S. Kim, "Hardware implementation of a real-time neural network controller with a DSP and an FPGA for nonlinear systems," *IEEE Trans. Ind. Electron.*, vol. 54, no. 1, pp. 265–271, Feb. 2007.

[13] N. Botros and M. Abdul-Aziz, "Hardware implementation of an artificial neural network using field programmable gate arrays (FPGA's)," *IEEE Trans. Ind. Electron.*, vol. 41, no. 6, pp. 665–667, Dec. 1994.

[14] S. Sadjina, C. Motz, T. Paireder, M. Huemer, and H. Pretl, "A survey of self-interference in LTE-advanced and 5G new radio wireless transceivers," *IEEE Trans. Microw. Theory Techn.*, vol. 68, no. 3, pp. 1118–1131, Mar. 2020.

[15] "REFSENS with one UL carrier for NC intra-band CA," Ericsson and ST-Ericsson, New Orleans, LA, USA, Rep. R4-126964, Nov. 2012.

[16] A. Gebhard, "Self-interference cancellation and rejection in FDD RF-transceivers," Ph.D. dissertation, Dept. Doktor der technischen Wissenschafte, Johannes Kepler Univ., Linz, Linz, Austria, 2019.

[17] O. Ploder, C. Motz, T. Paireder, C. Auer, H. Pretl, and M. Huemer, "Digitally intensive mixed-signal approach for self-interference cancellation in LTE-A/5G-transceivers," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2021, pp. 1–5.

[18] A. Kiayani, L. Anttila, and M. Valkama, "Modeling and dynamic cancellation of TX-RX leakage in FDD transceivers," in *Proc. 56th Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Columbus, OH, USA, Aug. 2013, pp. 1089–1094.

[19] D. Korpi, L. Anttila, and M. Valkama, "Nonlinear self-interference cancellation in MIMO full-duplex transceivers under crosstalk," *EURASIP J. Wireless Commun. Netw.*, vol. 2017, no. 1, pp. 1–15, 2017.

[20] L. Anttila, D. Korpi, E. Antonio-Rodríguez, R. Wichman, and M. Valkama, "Modeling and efficient cancellation of nonlinear self-interference in MIMO full-duplex transceivers," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, 2014, pp. 777–783.

[21] T. Paireder, C. Motz, and M. Huemer, "Spline-based adaptive cancellation of even-order intermodulation distortions in LTE-A/5G RF transceivers," *IEEE Trans. Veh. Technol.*, vol. 70, no. 6, pp. 5817–5832, Jun. 2021.

[22] C. Motz, O. Ploder, T. Paireder, and M. Huemer, "Enhanced transform-domain LMS based self-interference cancellation in LTE carrier aggregation transceivers," in *Proc. 17th Int. Conf. Comput.-Aided Syst. Theory (EUROCAST)*, Cham, Switzerland, Apr. 2019, pp. 28–35.

[23] C. Motz, T. Paireder, and M. Huemer, "Improving digital interference cancellation in LTE-A/5G-transceivers by statistical modeling," in *Proc. 54th Asilomar Conf. Signals Syst. Comput.*, 2020, pp. 887–894.

[24] A. Gebhard *et al.*, "A robust nonlinear RLS type adaptive filter for second-order-intermodulation distortion cancellation in FDD LTE and 5G direct conversion transceivers," *IEEE Trans. Microw. Theory Techn.*, vol. 67, no. 5, pp. 1946–1961, May 2019.

[25] A. Gebhard, R. S. Kanumalli, B. Neurauter, and M. Huemer, "Adaptive self-interference cancelation in LTE-A carrier aggregation FDD direct-conversion transceivers," in *Proc. IEEE Sens. Array Multichannel Signal Process. Workshop (SAM)*, Rio de Janerio, Brazil, Jul. 2016, pp. 1–5.

[26] A. Gebhard, C. Motz, R. S. Kanumalli, H. Pretl, and M. Huemer, "Nonlinear least-mean-squares type algorithm for second-order interference cancellation in LTE-A RF transceivers," in *Proc. 51st Asilomar Conf. Signals Syst. Comput. (ACSSC)*, Pacific Grove, CA, USA, Oct. 2017, pp. 802–807.

[27] C. Motz, T. Paireder, and M. Huemer, "Low-complex digital cancellation of transmitter harmonics in LTE-A/5G transceivers," *IEEE Open J. Commun. Soc.*, vol. 2, pp. 948–963, 2021.

[28] A. Kiayani *et al.*, "Adaptive nonlinear RF cancellation for improved isolation in simultaneous transmit–receive systems," *IEEE Trans. Microw. Theory Techn.*, vol. 66, no. 5, pp. 2299–2312, May 2018.

[29] S. Lahti, P. P. Campo, V. Lampu, L. Anttila, M. Valkama, and T. D. Hämäläinen, "Implementation of a nonlinear self-interference canceller using high-level synthesis," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2020, pp. 1–5.

[30] T. Paireder, C. Motz, S. Sadjina, and M. Huemer, "A robust mixed-signal cancellation approach for even-order intermodulation distortions in LTE-A/5G-transceivers," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 68, no. 3, pp. 923–927, Mar. 2021.

[31] R. S. Kanumalli, A. Gebhard, A. Elmaghraby, A. Mayer, D. Schwartz, and M. Huemer, "Active digital cancellation of transmitter induced modulated spur interference in 4G LTE carrier aggregation transceivers," in *Proc. 83rd Veh. Technol. Conf. (VTC Spring)*, Nanjing, China, May 2016, pp. 1–5.

[32] R. S. Kanumalli *et al.*, "Digitally-intensive transceivers for future mobile communications—Emerging trends and challenges," *ei Elektrotechnik und Informationstechnik*, vol. 135, no. 1, pp. 30–39, Feb. 2018.

[33] R. S. Kanumalli *et al.*, "Mixed-signal based enhanced widely linear cancellation of modulated spur interference in LTE-CA transceivers," in *Proc. 52nd Asilomar Conf. Signals Syst. Comput. (ACSSC)*, Pacific Grove, CA, USA, Oct. 2018, pp. 1382–1388.

[34] S. Sadjina, D. Krzysztof, R. S. Kanumalli, M. Huemer, and H. Pretl, "A circuit technique for blocker-induced modulated spur cancellation in 4G LTE carrier aggregation transceivers," in *Proc. Austrochip Workshop Microelectron. (Austrochip)*, Linz, Austria, Oct. 2017, pp. 23–28.

[35] A. Elmaghraby *et al.*, "A mixed-signal technique for TX-induced modulated spur cancellation in LTE-CA receivers," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 65, no. 9, pp. 3060–3073, Sep. 2018.

[36] A. T. Kristensen, A. Burg, and A. Balatsoukas-Stimming, "Advanced machine learning techniques for self-interference cancellation in full-duplex radios," in *Proc. 53rd Asilomar Conf. Signals Syst. Comput.*, 2019, pp. 1149–1153.

[37] A. T. Kristensen, A. Burg, and A. Balatsoukas-Stimming, "Identification of non-linear RF systems using backpropagation," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, 2020, pp. 1–6.

[38] C. Auer *et al.*, "Kernel adaptive filters: A panacea for self-interference cancellation in mobile communication transceivers?" in *Proc. 17th Int. Conf. Comput.-Aided Syst. Theory (EUROCAST)*, Cham, Switzerland, Apr. 2019, pp. 36–43.

[39] C. Auer, K. Kostoglou, T. Paireder, O. Ploder, and M. Huemer, "Support vector machines for self-interference cancellation in mobile communication transceivers," in *Proc. IEEE 91st Veh. Technol. Conf. (VTC Spring)*, May 2020, pp. 1–6.

[40] C. Auer, T. Paireder, O. Lang, and M. Huemer, "Kernel recursive least squares based cancellation for receiver-induced self-interference," in *Proc. 54th Asilomar Conf. Signals Syst. Comput.*, 2020, pp. 1–5.

[41] C. Auer, T. Paireder, and M. Huemer, "Kernel recursive least squares algorithm for transmitter-induced self-interference cancellation," in *Proc. IEEE Veh. Technol. Conf. (VTC-Spring)*, 2021, pp. 1–5.

[42] C. Auer, T. Paireder, and M. Huemer, "Self-interference cancellation in LTE/5G transceivers with sliding window kernel recursive least squares filters," in *Proc. 55th Asilomar Conf. Signals Syst. Comput.*, Nov. 2021, pp. 976–982.

[43] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," Dec. 2014, *arXiv:1412.6980*.

[44] M. D. Zeiler, "ADADELTA: An adaptive learning rate method," Dec. 2012, *arXiv:1212.5701*.

[45] S. Baumgartner, G. Bognár, O. Lang, and M. Huemer, "Neural network based data estimation for unique word OFDM," in *Proc. 55th Asilomar Conf. Signals Syst. Comput.*, Nov. 2021, pp. 381–388.

[46] K. Chandra *et al.*, "Gradient descent: The ultimate optimizer," 2019, *arXiv:1909.13371*.

[47] S. Haykin, *Adaptive Filter Theory*, 3rd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 1996.

[48] A. I. Hanna and D. P. Mandic, "A fully adaptive normalized nonlinear gradient descent algorithm for complex-valued nonlinear adaptive filters," *IEEE Trans. Signal Process.*, vol. 51, no. 10, pp. 2540–2549, Oct. 2003.

[49] A. I. Hanna, D. P. Mandic, and M. Razaz, "A normalised back-propagation learning algorithm for multilayer feed-forward neural adaptive filters," in *Proc. Neural Netw. Signal Process. 11th IEEE Signal Process. Soc. Workshop*, 2001, pp. 63–72.

[50] A. I. Hanna and D. P. Mandic, "A normalised complex backpropagation algorithm," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, vol. 1, 2002, pp. 977–980.

[51] T. Paireder, C. Motz, and M. Huemer, "Normalized stochastic gradient descent learning of general complex-valued models," *Electron. Lett.*, vol. 57, no. 12, pp. 493–495, 2021.

[52] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, no. 61, pp. 2121–2159, 2011.

[53] T. Tieleman and G. Hinton, "Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude," in *Neural Networks for Machine Learning*. Mountain View, CA, USA: COURSERA, 2012.
[54] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [Online]. Available: https://doi.org/10.1162/neco.1997.9.8.1735
[55] A. Paszke *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Red Hook, NY, USA: Curran Assoc., Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf
[56] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org

OLIVER PLODER (Graduate Student Member, IEEE) was born in Graz, Austria, in 1992. He received the bachelor's degree in information electronics from Johannes Kepler University Linz in 2016, and the master's degree in telecommunication engineering with a focus on wireless communications from the Universitat Politècnica de Catalunya (BarcelonaTech) in 2018. He is currently pursuing the Ph.D. degree with the Institute of Signal Processing, Johannes Kepler University Linz focusing his research on receiver interference cancellation for LTE and LTE-A RF transceiver systems by means of machine learning. His master thesis was written in cooperation with the Networked and Embedded Systems Laboratory, UCLA on the topic of secure state estimation in cyber–physical systems. Since September 2018, he has been a member of the Institute of Signal Processing, Johannes Kepler University Linz.

CHRISTINA AUER (Graduate Student Member, IEEE) was born in Linz, Austria, in 1987. She studied applied mathematics from Johannes Kepler University Linz in 2012, where she is currently pursuing the Ph.D. degree with the CD Laboratory for Digitally Assisted RF Transceivers for Future Mobile Communications. The topic of her master thesis, she partly wrote at UCLA, was image denoising. After graduation, she joined the company Linz Center of Mechatronics, where she was working in the research and development, especially with Kalman filtering for localization. Since September 2017, she has been a member of the Institute of Signal Processing, Johannes Kepler University Linz. Her research activities focus on self-interference cancellation using kernel adaptive filtering and machine learning.

CHRISTIAN MOTZ was born in Vöcklabruck, Austria, in 1990. He received the bachelor's degree (Hons.) in hardware software design and the master's degree in embedded systems design from the School of Informatics, Communications and Media, University of Applied Sciences Upper Austria, Hagenberg, Austria, in 2013 and 2015, respectively. He is currently pursuing the Ph.D. degree with the Institute of Signal Processing, Johannes Kepler University Linz (JKU), Linz, Austria.

His master's thesis was focused on pattern matching in cooperation with the Software Competence Center Hagenberg GmbH, Hagenberg. His thesis was on efficient implementation of Weyls discrepancy measure for 2-D image data. From 2010 to 2015, he was with the University of Applied Sciences Upper Austria. In 2016, he joined the Research Center for Non Destructive Testing GmbH, Linz. Since 2017, he has been a member of the Institute of Signal Processing, JKU, and the CD Laboratory for Digitally Assisted RF Transceivers for Future Mobile Communications, where he is involved in receiver interference cancellation by means of adaptive signal processing methods.

THOMAS PAIREDER received the bachelor's degree (Hons.) in information electronics and the master's degree (Hons.) in electronics and information technology from Johannes Kepler University Linz, Linz, Austria, in 2016 and 2018, respectively, where he is currently pursuing the Ph.D. degree with the Institute of Signal Processing. His topic is receiver interference cancellation by means of adaptive signal processing methods. His master's thesis was focused on signal processing in cooperation with the Research Center for Nondestructive Testing GmbH. In his thesis, he implemented a real-time processing system for laser-ultrasonic signals.

OLIVER LANG (Member, IEEE) received the bachelor's degree in electrical engineering and information technology and the master's degree in microelectronics from the Vienna University of Technology, Austria, in 2011 and 2014, respectively, and the Ph.D. degree from the Institute of Signal Processing, Johannes Kepler University (JKU), Linz, Austria, in 2018, where he was a Member from 2014 to 2018. From 2018 to 2019, he worked with DICE GmbH, Linz, which was a subsidiary company of Infineon Austria GmbH. During this period, he worked on automotive radar MMICs and systems. Since March 2019, he has been a University Assistant with Ph.D. at the Institute of Signal Processing, JKU. He is the main inventor of several patents and patent applications in the field of automotive radar systems and main author of several publications in the field of estimation theory and adaptive filtering. He was holding exercise courses in the field of digital signal processing, discrete-time signals and systems, information technology, optimal and adaptive signal processing systems, and circuit design.

MARIO HUEMER (Senior Member, IEEE) received the Dipl.-Ing. and Dr.techn. degrees from Johannes Kepler University (JKU) Linz, Austria, in 1996 and 1999, respectively. After holding positions in industry and academia, he became an Associate Professor with the University of Erlangen–Nuremberg, Germany, from 2004 to 2007, and a Full Professor with Klagenfurt University, Austria, from 2007 to 2013. Since September 2013, he has been heading the Institute of Signal Processing, JKU Linz as a Full Professor, and since 2017, he has been the Co-Head of the Christian Doppler Laboratory for Digitally Assisted RF Transceivers for Future Mobile Communications. His research focuses on statistical and adaptive signal processing, signal processing architectures, as well as mixed signal processing with applications in information and communications engineering, radio frequency transceivers for communications and radar, sensor, and biomedical signal processing. He has received the dissertation awards of the German Society of Information Technology (ITG) and the Austrian Society of Information and Communications Technology (GIT), respectively, the Austrian Kardinal Innitzer Award in Natural Sciences, and the German ITG Award.