SICNN: Soft Interference Cancellation Inspired Neural Network Equalizers

STEFAN BAUMGARTNER [1,2] **(Graduate Student Member, IEEE),**
OLIVER LANG [1] **(Member, IEEE), AND MARIO HUEMER** [1,2] **(Senior Member, IEEE)**

[1]Institute of Signal Processing, Johannes Kepler University Linz, 4040 Linz, Austria
[2]JKU LIT SAL eSPML Laboratory, Johannes Kepler University Linz, 4040 Linz, Austria

CORRESPONDING AUTHOR: S. BAUMGARTNER (stefan.baumgartner@jku.at)

ABSTRACT In recent years data-driven machine learning approaches have been extensively studied to replace or enhance traditionally model-based processing in digital communication systems. In this work, we focus on equalization and propose a novel neural network (NN-)based approach, referred to as SICNN. SICNN is designed by deep unfolding a model-based iterative soft interference cancellation (SIC) method. It eliminates the main disadvantages of its model-based counterpart, which suffers from high computational complexity and performance degradation due to required approximations. We present different variants of SICNN. SICNNv1 is specifically tailored to single carrier frequency domain equalization (SC-FDE) systems, the communication system mainly regarded in this work. SICNNv2 is more universal and is applicable as an equalizer in any communication system with a block-based data transmission scheme. Moreover, for both SICNNv1 and SICNNv2, we present versions with highly reduced numbers of learnable parameters. Another contribution of this work is a novel approach for generating training datasets for NN-based equalizers, which significantly improves their performance at high signal-to-noise ratios. We compare the bit error ratio performance of the proposed NN-based equalizers with state-of-the-art model-based and NN-based approaches, highlighting the superiority of SICNNv1 over all other methods for SC-FDE. Exemplarily, to emphasize its universality, SICNNv2 is additionally applied to a unique word orthogonal frequency division multiplexing (UW-OFDM) system, where it achieves state-of-the-art performance. Furthermore, we present a thorough complexity analysis of the proposed NN-based equalization approaches, and we investigate the influence of the training set size on the performance of NN-based equalizers.

INDEX TERMS Equalization, neural networks, single carrier frequency domain equalization, soft interference cancellation, training set generation.

I. INTRODUCTION

DIGITAL communications at the physical layer level is traditionally a quite model-based discipline. That is, especially for the receiver processing blocks of digital communication systems, most algorithms have been developed based on physical and statistical models of the communication chain. With this established approach well interpretable methods can be obtained, their performance bounds can often be specified, and usually algorithms achieving optimal performance for the given models can be derived. Besides these advantageous properties, model-based approaches also have some downsides. Performance-optimal methods can in some cases exhibit an infeasible computational complexity, requiring the application of suboptimal algorithms in practice. Further, modeling errors, wrong (or oversimplified) assumptions, or insufficient model knowledge may lead to a considerable performance degradation. Since with data-driven machine learning methods many of the drawbacks of model-based approaches can be resolved, currently intensive research is conducted on machine learning approaches for several applications in communications engineering. This includes possible future scenarios like

communications assisted by reconfigurable intelligent surfaces (RISs) [2], molecular communications [3], or integrated sensing and communication [4]. However, also in traditional wireless communication systems promising results can be achieved by means of machine learning. This involves completely abandoning the block-based paradigm of current digital communication system design with the help of end-to-end learning [5], [6], or replacing / enhancing individual blocks of a standard communication chain [7], [8], [9]. The latter includes machine learning approaches for channel estimation [10], [11], channel decoding [12], and self-interference cancellation [13], [14], [15]. In this work, we regard another important processing block at the receiver, namely equalization. Equalization, also referred to as data estimation, is the task of reconstructing transmitted data – distorted during transmission over a channel – at the receiver side of a communication system. Typically, equalization is conducted by model-based methods. However, also with machine learning methods auspicious results have already been demonstrated [16], [17], [18], [19], [20], [21], [22], [23], [24], [25]. In current publications on machine learning approaches for data estimation, mainly neural networks (NNs) are employed. NNs are known to be universal function approximators [26] and thus are expected to approximate the optimal data estimators. Many of the presented results are promising, but there are also some new challenges arising. More specifically, standard NNs like a fully-connected feedforward NN (FCNN) are black-box approaches, i.e., their inference is not interpretable, performance bounds can hardly be derived, and domain knowledge is not exploited. Especially due to the latter fact most NNs suffer from requiring large amounts of training data and a high inference complexity. Optimally, one can fuse model-based and data-driven approaches by, e.g., incorporating existing model knowledge into NNs, which is expected to lead to less complex and better performing NNs than the standard black-box NNs. One possibility of incorporating model knowledge into NNs is to design their layer structure accordingly, which leads to NNs we refer to as model-inspired NNs. Currently, one of the most promising and most popular approaches for obtaining NNs with a model-inspired layer structure is deep unfolding [27]. The idea of deep unfolding is to take a model-based iterative algorithm, which is conceived for finding the solution of an optimization problem, fix its number of iterations, and unfold every iteration to a layer of an NN. Depending on the aspired abstraction level of the NN (i.e., the similarity between the model-based algorithm and the NN), only a few parameters of the model-based iterative algorithm (e.g., its step size) or even whole parts are replaced by learnable parameters or modules, respectively. Those can then be optimized with tools known from NN optimization by utilizing available training data. A number of NN-based data estimators, e.g., the NNs in [18], [20], [22], [23], [24], and [25] are designed by employing deep unfolding. In this work, we also apply deep unfolding for the design of our proposed NN-based equalizers.

Most NN-based data estimators are currently proposed for equalization in multiple-input multiple-output (MIMO) communication systems, often assuming data transmission over an uncorrelated Rayleigh fading channel. In this work, the developed NN-based equalizers are mainly evaluated for single carrier frequency domain equalization (SC-FDE) systems [28], [29]. In an SC-FDE system, a single carrier transmission scheme is utilized, but the payload data is transmitted in a block-wise manner with guard intervals between successive blocks, as it is the case in orthogonal frequency division multiplexing (OFDM) systems. The received blocks are transformed to frequency domain before conducting matched filtering, downsampling, and equalization. This allows an efficient receiver implementation [29] and results in a system model similar to that of an OFDM system. In this work, we regard employing both a cyclic prefix (CP) and a so-called unique word (UW), which is a known deterministic sequence, as guard interval. The UW can advantageously be utilized, e.g., for synchronization purposes [30], however, at the cost of equalization complexity. For a CP guard interval, the optimal linear equalizer is a low-complex single-tap equalizer, while for a UW guard interval, in turn, the optimal linear equalizer is more complex. In contrast to CP-OFDM systems, for SC-FDE optimal performance can only be obtained with computationally highly demanding nonlinear equalizers. This motivates developing NN-based data estimators for SC-FDE systems. Employing NN-based equalizers for SC-FDE systems necessitates, in contrast to MIMO systems over uncorrelated Rayleigh fading channels, an additional pre-processing step. As extensively described in [31], for the application of NN-based equalizers in SC-FDE systems a data normalization scheme is required for a well-behaved NN training and thus a satisfying performance of the NN equalizers. We also briefly review the necessary data normalization scheme in this work.

CONTRIBUTION

In this work, we propose the NN-based data estimators SIC-NNv1 and SICNNv2, which are designed by unfolding an iterative soft interference cancellation (SIC) method [32]. The main idea of iterative SIC is that in each iteration every single data symbol in the transmitted data vector is estimated on its own, by considering the influence of all other data symbols in the data vector as interference. This interference can be mitigated by incorporating estimates of the data symbols from the previous into the current iteration. By that, the data symbol estimates are refined from iteration to iteration. Although also DeepSIC [23] is inspired by the same iterative SIC method, SICNNv1 and SICNNv2 are fundamentally different from this NN. The idea of SIC is adopted by DeepSIC concerning its structure. That is, DeepSIC consists of multiple stages, where each stage is comprised of as many sub-FCNNs as there are data symbols in the transmitted data vector. Each of the sub-FCNNs is utilized to estimate one data symbol,

whereby the input data of a sub-FCNN is made up of the received vector as well as of estimates provided by the sub-FCNNs for the remaining data symbols from the last stage. All estimates are refined stage by stage. That is, Deep-SIC has similarities with the model-based SIC method only by refining the estimates of the posterior data symbol probabilities, but neither interference cancellation is conducted in a stage, nor model knowledge is utilized. In contrast, our proposed NN-based equalizers are far more similar to the underlying model-based method. More specifically, with SICNNv1 – an adapted version of an NN-based equalizer called SICNN proposed in our previous work [1] – we try to resemble the model-based iterative SIC method closely. However, we replace numerically demanding, computationally intensive operations, for which also approximations have to be made in the model-based approach, by low-complex NNs. This NN-based approach achieves significantly better performance than the corresponding model-based method, while exhibiting lower complexity. We tailor SICNNv1 for being employed as an NN-based equalizer in an SC-FDE system by exploiting some properties of this communication system in the NN architecture design. SICNNv2, in turn, is more abstracted from the model-based iterative SIC method, i.e., less model knowledge is utilized for the NN architecture design. However, it is more universal and can also be applied as an equalizer in any communication system with a block-based data transmission scheme. A further difference between DeepSIC and the proposed SICNNv1 and SICNNv2 is their generalization ability regarding different channels. As it is the case for, e.g., MMNet [21] or ViterbiNet [33], DeepSIC is trained for one specific channel. This generally allows lower complex NNs, but requires retraining as soon as the channel changes. SICNNv1 and SICNNv2 belong, like DetNet [18] or OAMP-Net [20], to the group of NN-based data estimators, which are trained with different channels sampled from a statistical channel model, and use the actual channel realization as an input. These NNs generally require an extensive offline training, and exhibit a higher computational inference complexity, but they do not have to be retrained as long as the specified statistical channel model is valid for the operating environment.

Since in every stage[1] of SICNNv1/SICNNv2 the same task has to be fulfilled, namely to refine estimated posterior data symbol probabilities, we additionally introduce two modified versions of SICNNv1 and SICNNv2. While in SICNNv1 and SICNNv2 for every stage different sub-NNs are utilized to estimate posterior data symbol probabilites, in SICNNv1Red and SICNNv2Red every stage uses the same sub-NNs, which drastically reduces the number of parameters to be trained.

We compare the proposed NN-based equalizers with state-of-the-art model-based and NN-based data estimators concerning achieved bit error ratio (BER) performance, and

regarding their computational complexity during inference. The evaluation is conducted for SC-FDE systems, either employing a UW or a CP as a guard interval, for both quadrature phase shift keying (QPSK) and 16-QAM (quadrature amplitude modulation) alphabets, and with perfect and imperfect channel knowledge at the receiver. We investigate the required amount of training data for achieving satisfying performance of selected NN-based equalizers, pointing out the advantage of reducing the number of learnable parameters of an NN. Further, we demonstrate the universal applicability of SICNNv2 by presenting its achieved performance for a communication system employing the so-called UW-OFDM signaling scheme.

As another important contribution of this paper, we present a novel approach to generate training sets for NN-based equalizers. In this approach, only those sample data transmissions are included in the training set for which the number of data symbol estimation errors made by a baseline equalizer exceeds a specified quantity. This greatly enhances the performance of NN-based data estimators at high signal-to-noise ratios (SNRs).

The remainder of this paper is structured as follows. In Sec. II, we review the SC-FDE signaling scheme and data transmission model, and we discuss state-of-the-art model-based equalizers, including in particular the iterative SIC method our NN-based equalizers are inspired by. In Sec. III, the novel NN-based equalizers are introduced and discussed in detail. Further, we propose a novel approach for generating training datasets for NN-based data estimators in Sec. IV. We present BER performance results, and an in-depth analysis of the computational complexity of the regarded model-based and NN-based equalizers in Sec. V.

NOTATION

Throughout this paper, we use lower-case bold face letters $\mathbf{x}$ for vectors and upper-case bold face letters $\mathbf{X}$ for matrices, x_k for the kth element of $\mathbf{x}$, $[\mathbf{X}]_{kj}$ for the element of $\mathbf{X}$ in row k and column j, and $[\mathbf{X}]_{k*}$ for the kth row of $\mathbf{X}$. Further, $(.)^T$, $(.)^H$, and $(.)^*$ indicate transposition, conjugate transposition, and conjugation respectively, while $|\mathbf{X}|$ is the determinant of the matrix $\mathbf{X}$. We denote the probability density function (PDF) of a continuous random variable as $p(.)$, the probability mass function (PMF) of a discrete random variable as $p[.]$, a conditional PMF of a random variable a given b as $p[a|b]$, and a PMF evaluated at the value $\tilde{a}$ as $p[a = \tilde{a}]$. We describe the expectation operator averaging over the PDF/PMF of a random variable a as $E_a[.]$, where the subscript of the expectation operator is omitted when the averaging PDF/PMF is clear from context.

II. PRELIMINARIES

In this section, we describe the system model for SC-FDE, and state-of-the-art model-based equalization approaches. Further, we discuss an iterative SIC approach for data estimation, and we highlight some properties of this method.

---

[1]In order to avoid any misunderstandings, we refer to one unfolded iteration of the model-based iterative SIC method as *stage* of SICNNv1/SICNNv2 instead of *layer*.

A. SINGLE CARRIER FREQUENCY DOMAIN EQUALIZATION

In an SC-FDE communication system [28], [29], [34], [35], a single carrier modulation scheme is employed for data transmission. At the transmitter, the data symbols to be transmitted, which are drawn from a modulation alphabet $\mathbb{S}$ (in this work, we mainly use QPSK as a modulation alphabet), are grouped into blocks of length N_d. These blocks of data symbols, which we refer to as data vectors $\mathbf{d} \in \mathbb{S}^{N_d}$, are strung together to generate a transmit data burst, whereby they are separated by guard intervals of length N_g. As a guard interval, in this work we consider using either a CP or a UW, which is a deterministic sequence known by the receiver. Depending on the employed guard interval, some processing steps in the receiver are different, which are described later in this section. The transmit data burst is upsampled and pulse shaped with a root-raised-cosine (RRC) filter, followed by transmitting the resulting signal over a multipath channel, which is additionally disturbed by additive white Gaussian noise (AWGN). At the receiver, the first processing step depends on the employed guard interval. While for a UW guard interval every received data vector including its succeeding guard interval is transformed individually to frequency domain, for a CP guard interval the CPs are removed first before transforming the remaining received data vectors individually to frequency domain. In frequency domain, the further processing steps matched filtering, downsampling, and equalization are conducted. Independent of the employed guard interval, the general model[2] of the transmission of a data vector up to the input of the equalizer in the equivalent complex baseband can be written as [36]

$$\mathbf{y}_{\mathrm{r}} = \widetilde{\mathbf{H}}\mathbf{F}_{N'}\mathbf{x} + \mathbf{w} . \tag{1}$$

Here, $\mathbf{y}_{\mathrm{r}} \in \mathbb{C}^{N'}$ is the received vector after matched filtering and downsampling in frequency domain, where N' depends on the employed guard interval and is being specified later in this section. $\widetilde{\mathbf{H}} \in \mathbb{R}^{N' \times N'}$ is a diagonal matrix containing the sampled frequency response of the cascade of upsampler, pulse shaping filter, multipath channel, matched filter, and downsampler on its main diagonal. Note, that $\widetilde{\mathbf{H}}$ is a real-valued matrix since we conduct optimal matched filtering in frequency domain, i.e., the filter is matched to the channel distorted transmit pulse (for further details on optimal matched filtering in SC-FDE systems, we refer to [37] and [38]). Furthermore, $\mathbf{F}_{N'} \in \mathbb{C}^{N' \times N'}$ is the N'-point discrete Fourier transform (DFT) matrix and $\mathbf{w} \sim \mathcal{CN}(\mathbf{0}, N'\sigma_{\mathrm{n}}^2\widetilde{\mathbf{H}})$ is circularly symmetric complex AWGN, with σ_{n}^2 being the variance of the AWGN in time domain. The structure of the transmitted vector $\mathbf{x} \in \mathbb{C}^{N'}$ as well as the final system model differ for UW and CP guard intervals, which we further detail in the following.

[2]Here we assume sufficiently long guard intervals such that each data block can be processed individually and independently of all other transmitted data blocks.

1) UNIQUE WORD GUARD INTERVAL

As already mentioned, in case of a UW guard interval [34], [35], at the receiver both a received data vector and its succeeding UW are transformed to frequency domain for the further processing steps. Hence, the vector $\mathbf{x} \in \mathbb{C}^{N'}$ in (1) has the form $\mathbf{x} = [\mathbf{d}^T, \mathbf{u}^T]^T$, where $\mathbf{d} \in \mathbb{S}^{N_d}$ is the transmitted data vector to be estimated, $\mathbf{u} \in \mathbb{C}^{N_g}$ is the UW, and $N' = N = N_d + N_g$. Inserting into (1) leads to

$$\mathbf{y}_{\mathrm{r}} = \widetilde{\mathbf{H}}\mathbf{F}_N \begin{bmatrix} \mathbf{d} \\ \mathbf{u} \end{bmatrix} + \mathbf{w} . \tag{2}$$

By assuming perfect channel knowledge on receiver side, the influence of the known UW $\mathbf{u}$ on the received vector $\mathbf{y}_{\mathrm{r}}$ can be removed according to

$$\mathbf{y}_{\mathrm{r}} - \widetilde{\mathbf{H}}\mathbf{M}'\mathbf{u} = \widetilde{\mathbf{H}}\mathbf{M}_{\mathrm{uw}}\mathbf{d} + \mathbf{w} , \tag{3}$$

with $\mathbf{F}_N = [\mathbf{M}_{\mathrm{uw}}\, \mathbf{M}']$, where $\mathbf{M}_{\mathrm{uw}} \in \mathbb{C}^{N \times N_d}$ and $\mathbf{M}' \in \mathbb{C}^{N \times N_g}$ are built by the first N_d columns and the remaining N_g columns of $\mathbf{F}_N$, respectively.

2) CYCLIC PREFIX GUARD INTERVAL

In case of a CP guard interval [28], [29], the guard intervals are removed at the receiver before transforming the received blocks of data to frequency domain, which means that $\mathbf{x}$ in (1) is realized as the transmitted data vector $\mathbf{d} \in \mathbb{S}^{N_d}$, and $N' = N_d$. Consequently, for a CP guard interval, the data transmission is modeled as

$$\mathbf{y}_{\mathrm{r}} = \widetilde{\mathbf{H}}\mathbf{F}_{N_d}\mathbf{d} + \mathbf{w} . \tag{4}$$

3) SYSTEM MODEL FOR SINGLE CARRIER FREQUENCY DOMAIN EQUALIZATION

As elaborated above, the model for data transmission in an SC-FDE system is given by (3) for a UW guard interval and by (4) for a CP. For the ease of notation, in the remainder of this work the SC-FDE system model is given for both guard intervals by

$$\mathbf{y} = \widetilde{\mathbf{H}}\mathbf{M}\mathbf{d} + \mathbf{w} = \mathbf{H}\mathbf{d} + \mathbf{w} , \tag{5}$$

with $\mathbf{y} \in \mathbb{C}^{N'}, \mathbf{M} \in \mathbb{C}^{N' \times N_d}$, and $\mathbf{H} = \widetilde{\mathbf{H}}\mathbf{M}$, where we employ for a

- UW guard: $N' = N = N_d + N_g$, $\mathbf{y} = \mathbf{y}_{\mathrm{r}} - \widetilde{\mathbf{H}}\mathbf{M}'\mathbf{u}$, $\mathbf{M} = \mathbf{M}_{\mathrm{uw}}$.
- CP guard: $N' = N_d$, $\mathbf{y} = \mathbf{y}_{\mathrm{r}}$, $\mathbf{M} = \mathbf{F}_{N_d}$.

4) MODEL-BASED EQUALIZATION

Based on (5), data estimation can be conducted for a given received vector $\mathbf{y}$ and a channel matrix $\mathbf{H}$. As thoroughly elucidated in [39], depending on the optimality criterion, there exist different optimal equalizers. The bit-wise maximum a-posteriori (MAP) estimator yields for every transmitted bit the bit value featuring the highest posterior probability. It is known to be the optimal estimator regarding the bit error probability. The vector MAP, in turn, is optimal regarding the error probability of the data vector estimate. However,

the computational complexity of both of the aforementioned estimators grows exponentially with the data vector length N_d, which makes them in general prohibitive for practical applications. Hence, one usually has to resort to suboptimal linear or nonlinear estimation methods.

The best linear estimator in the Bayesian sense is the linear minimum mean square error (LMMSE) estimator, which is given by [40]

$$\hat{\mathbf{d}} = \left(\mathbf{M}^H \widetilde{\mathbf{H}} \mathbf{M} + \frac{N\sigma_n^2}{\sigma_d^2}\mathbf{I}\right)^{-1} \mathbf{M}^H \mathbf{y} = \mathbf{E}_{\text{LMMSE}}\mathbf{y}, \quad (6)$$

with $\mathbf{I}$ and σ_d^2 being the identity matrix with appropriate dimensions and the variance of the symbol alphabet, respectively. In case of a CP guard interval ($\mathbf{M} = \mathbf{F}_{N_d}$), (6) can be simplified to [35]

$$\hat{\mathbf{d}} = \frac{1}{N'}\mathbf{M}^H \left(\widetilde{\mathbf{H}} + \frac{\sigma_n^2}{\sigma_d^2}\mathbf{I}\right)^{-1}\mathbf{y} = \frac{1}{N'}\mathbf{M}^H \mathbf{E}_{\text{LMMSE,dg}}\mathbf{y}, \quad (7)$$

where $\mathbf{E}_{\text{LMMSE,dg}} = \left(\widetilde{\mathbf{H}} + \frac{\sigma_n^2}{\sigma_d^2}\mathbf{I}\right)^{-1}$ is a diagonal matrix, and thus also the inversion required to compute this estimator matrix can be realized efficiently. Typically, instead of multiplying by $\frac{1}{N'}\mathbf{M}^H$, an inverse DFT is conducted. Note, that this low-complex equalizer can also be employed for a UW guard interval when applying LMMSE estimation to (2) instead of (3), i.e., the known UW $\mathbf{u}$ is not being removed before data estimation, but is estimated as well. Compared to the LMMSE estimator (6), this approximate[3] LMMSE estimator allows a lower-complex equalization, however, at the cost of performance degradation [35].

A popular suboptimal nonlinear estimator is the decision feedback equalizer (DFE), which is an iterative method. There, in every iteration LMMSE estimation of the data symbol with the smallest error variance is conducted, followed by removing the influence of the hard decision data symbol estimate on the received vector. However, in case of wrong data symbol estimates, this method suffers from error propagation deteriorating the estimation performance. For more details, we refer to [39], where the DFE is elaborated for a so-called unique word orthogonal frequency division multiplexing (UW-OFDM) system. In the following, we address another suboptimal nonlinear method, namely iterative soft interference cancellation (SIC), in more detail, since the proposed NN-based equalizers are deduced from this model-based approach.

B. ITERATIVE SOFT INTERFERENCE CANCELLATION

The idea of the iterative SIC method proposed in [32] is to estimate each data symbol d_k, $k \in \{0, \ldots, N_d-1\}$, in the data vector $\mathbf{d}$ separately, and refine the estimates from iteration to iteration. For the estimation of the kth data symbol d_k, all other data symbols d_l, $l \neq k$, are treated as interference, and thus their influence on the received vector $\mathbf{y}$ is cancelled as

---

[3]By neglecting the knowledge about the UW, only an approximate LMMSE estimator is obtained.

Algorithm 1 Model-Based Iterative SIC for SC-FDE

1: **function** IterativeSoftIC_SC-FDE($\mathbf{H}$, $\mathbf{y}$, σ_n^2, σ_d^2)
2:     $\hat{d}_k^{(-1)} \leftarrow 0$, $e_k^{(-1)} \leftarrow \sigma_d^2$ $\forall k = 0, \ldots, N_d - 1$
3:     **for** $q = 0, \ldots, Q-1$ **do**
4:         **for** $k = 0, \ldots, N_d-1$ **do**
5:             Compute $\mathbf{y}_{\text{ic},k}^{(q)}$ according to (11)
6:             Compute $\mathbf{C}_{\mathbf{vv},k}^{(q)}$ following (12) for $q = 0$, or (14) for $q > 0$
7:             Evaluate posterior PMF $p[d_k|\mathbf{y}_{\text{ic},k}^{(q)}]$
8:             Update soft information: $\hat{d}_k^{(q)}$ using (18) and $e_k^{(q)}$ via (19)
9:         **end for**
10:     **end for**
11:     **return** $\hat{\mathbf{d}}^{(Q-1)}$
12: **end function**

---

far as possible. Since the data symbols d_l are unknown, their currently available estimates $\hat{d}_l$ are utilized for interference cancellation. Interference cancellation reduces the unknown variable to be estimated from an N_d-dimensional data vector $\mathbf{d}$ to a single data symbol d_k, and thus nonlinear minimum mean square error (MMSE) estimation – which is generally computationally infeasible for estimating $\mathbf{d}$ – can easily be applied for estimating d_k. In order to prevent error propagation, instead of using hard decision data symbol estimates for interference cancellation, soft information of every data symbol estimate from the previous iteration is utilized in form of the MMSE estimate, which is the posterior mean, and the corresponding conditional mean square error (MSE). The soft estimates are refined iteratively. In [32], this approach is proposed for a MIMO system where all entries of the channel matrix $\mathbf{H}$ are independent of each other (all entries of the channel matrix are modeled as independent random variables following a normal distribution). This allows for simplifications in the iterative SIC method that cannot be applied in general. In the following, we present the iterative SIC method following the approach proposed in [32]. However, we adapt this method, which is also summarized in Algorithm 1, for an SC-FDE system, where the assumption of independent elements of $\mathbf{H}$ is not fulfilled.

Let us regard the qth iteration, $q = 0, \ldots, Q-1$, of Q total iterations of the iterative SIC method. We assume that for every data symbol d_k, $k \in \{0, \ldots, N_d-1\}$, a soft estimate from the previous iteration $(q-1)$ is available, namely, the MMSE data symbol estimate, which is the posterior mean

$$\hat{d}_k^{(q-1)} = E_{d_k|\mathbf{y}_{\text{ic},k}^{(q-1)}}\left[d_k|\mathbf{y}_{\text{ic},k}^{(q-1)}\right], \quad (8)$$

and the corresponding MSE given $\mathbf{y}_{\text{ic},k}^{(q-1)}$

$$e_k^{(q-1)} = E_{d_k|\mathbf{y}_{\text{ic},k}^{(q-1)}}\left[|d_k - \hat{d}_k^{(q-1)}|^2|\mathbf{y}_{\text{ic},k}^{(q-1)}\right]. \quad (9)$$

Here, $\mathbf{y}_{\text{ic},k}^{(q-1)}$ is the received vector without the interference of all but the kth data symbol estimates in iteration $(q-1)$. For

the estimation of a data symbol d_k, one can reformulate the system model (5) to

$$\mathbf{y} = \mathbf{h}_k d_k + \bar{\mathbf{H}}_k \bar{\mathbf{d}}_k + \mathbf{w}, \tag{10}$$

where $\bar{\mathbf{H}}_k$ is $\mathbf{H}$ after removing the kth column $\mathbf{h}_k$, and $\bar{\mathbf{d}}_k$ is the data vector without the kth data symbol d_k. The term $\bar{\mathbf{H}}_k \bar{\mathbf{d}}_k$ denotes the interference caused by all but the kth data symbol in the data vector, which should ideally be removed from $\mathbf{y}$ for the estimation of d_k. SIC can be conducted by removing $\bar{\mathbf{H}}_k \hat{\bar{\mathbf{d}}}_k^{(q-1)}$ from $\mathbf{y}$, leading to

$$\mathbf{y}_{\mathrm{ic},k}^{(q)} = \mathbf{y} - \bar{\mathbf{H}}_k \hat{\bar{\mathbf{d}}}_k^{(q-1)} = \mathbf{h}_k d_k \underbrace{- \bar{\mathbf{H}}_k \bar{\boldsymbol{\delta}}_k^{(q-1)} + \mathbf{w}}_{\mathbf{v}_k^{(q)}}, \tag{11}$$

where $\hat{\bar{\mathbf{d}}}_k^{(q-1)}$ consists of all but the kth data symbol estimates from iteration $(q-1)$, and $\bar{\boldsymbol{\delta}}_k^{(q-1)} = \hat{\bar{\mathbf{d}}}_k^{(q-1)} - \bar{\mathbf{d}}_k$ contains the (unknown) data symbol estimation errors from the previous iteration step. For estimating d_k based on (11), the statistics of the total noise vector $\mathbf{v}_k^{(q)}$, which is composed of the Gaussian noise vector $\mathbf{w}$ and the noise due to data symbol estimation errors, have to be specified. We start by considering the noise statistics for the first iteration ($q = 0$). As we will elaborate later in this section, initializing the data symbol estimates with the mean of the symbol alphabet is a rational choice, i.e., $\hat{\bar{\mathbf{d}}}_k^{(-1)} = \mathbf{0}$, leading to $\mathbf{y}_{\mathrm{ic},k}^{(0)} = \mathbf{y}$ and $\bar{\boldsymbol{\delta}}_k^{(0)} = -\bar{\mathbf{d}}_k$. Assuming independent and identically distributed (i.i.d.) data symbols with uniform prior probability and reasonably large N_d, following central limit theorem arguments, $\bar{\mathbf{H}}_k \bar{\mathbf{d}}_k$ can be considered to follow a circularly symmetric complex Gaussian distribution with zero mean, and to be independent of $\mathbf{w}$. Hence, $\mathbf{v}_k^{(0)}$ approximately also follows a circularly symmetric complex Gaussian distribution with zero mean and a covariance matrix

$$\mathbf{C}_{\mathbf{vv},k}^{(0)} = E\big[\mathbf{v}_k^{(0)} \mathbf{v}_k^{(0)H}\big] = \sigma_\mathrm{d}^2 \widetilde{\mathbf{H}} \bar{\mathbf{M}}_k \bar{\mathbf{M}}_k^H \widetilde{\mathbf{H}} + N \sigma_\mathrm{n}^2 \widetilde{\mathbf{H}}, \tag{12}$$

where $\bar{\mathbf{M}}_k$ is the matrix $\mathbf{M}$ without the kth column. For all further iterations ($q > 0$), we start by specifying the type of the statistical distribution of the vector $\mathbf{r}_k^{(q-1)} = \bar{\mathbf{H}}_k \bar{\boldsymbol{\delta}}_k^{(q-1)}$. Based on central limit theorem arguments and unbiased MMSE estimates (cf. Appendix I), $\mathbf{r}_k^{(q-1)}$ can be approximated to follow a circularly symmetric complex Gaussian distribution with zero mean, and thus the same assumption holds for $\mathbf{v}_k^{(q)}$. As shown in Appendix I, the noise covariance matrix $\mathbf{C}_{\mathbf{vv},k}^{(q)}$ is given by

$$\begin{aligned}
\mathbf{C}_{\mathbf{vv},k}^{(q)} &= \bar{\mathbf{H}}_k \widetilde{\mathbf{E}}_k^{(q-1)} \bar{\mathbf{H}}_k^H + N \sigma_\mathrm{n}^2 \widetilde{\mathbf{H}} \\
&\quad - \bar{\mathbf{H}}_k E_{\big(\bar{\mathbf{d}}_k | \mathbf{y}, \hat{\mathbf{d}}^{(q-2)}\big), \mathbf{w}} \big[\bar{\boldsymbol{\delta}}_k^{(q-1)} \mathbf{w}^H\big] \\
&\quad - E_{\big(\bar{\mathbf{d}}_k | \mathbf{y}, \hat{\mathbf{d}}^{(q-2)}\big), \mathbf{w}} \big[\mathbf{w} \bar{\boldsymbol{\delta}}_k^{(q-1)H}\big] \bar{\mathbf{H}}_k^H, \tag{13}
\end{aligned}$$

where $\widetilde{\mathbf{E}}_k^{(q-1)} = E_{\bar{\mathbf{d}}_k | \mathbf{y}, \hat{\mathbf{d}}^{(q-2)}} \big[\bar{\boldsymbol{\delta}}_k^{(q-1)} \bar{\boldsymbol{\delta}}_k^{(q-1)H} \big| \mathbf{y}, \hat{\mathbf{d}}^{(q-2)}\big]$ is the conditioned error covariance matrix. For the off-diagonal entries of $\widetilde{\mathbf{E}}_k^{(q-1)}$ and for the third and the fourth term

in (13) no exact closed form solution is available. A possible workaround is to employ the approximation

$$\mathbf{C}_{\mathbf{vv},k}^{(q)} \approx \bar{\mathbf{H}}_k \mathbf{E}_k^{(q-1)} \bar{\mathbf{H}}_k^H + N \sigma_\mathrm{n}^2 \widetilde{\mathbf{H}}, \tag{14}$$

where $\mathbf{E}_k^{(q-1)} = \mathrm{diag}\big(\big[e_0^{(q-1)}, \ldots, e_{k-1}^{(q-1)}, e_{k+1}^{(q-1)}, \ldots, e_{N_\mathrm{d}-1}^{(q-1)}\big]\big)$, i.e., correlations between the data symbol estimates as well as correlations between the estimation errors and the AWGN noise are neglected. However, this may lead to inaccuracies in the estimation process. Especially at high SNRs, for deep fading channels, and when q is increasing (then the $e_k^{(q-1)}$ are usually becoming smaller), $\mathbf{C}_{\mathbf{vv},k}^{(q)}$ can become ill-conditioned, which is – in combination with the occurring approximation errors – an issue for computing its inverse required for the next steps in the estimation process.

For computing a data symbol estimate $\hat{d}_k^{(q)}$, the posterior PMF $p\big[d_k | \mathbf{y}_{\mathrm{ic},k}^{(q)}\big]$ is needed, which can be obtained via the Bayesian rule by utilizing the likelihood function $p\big(\mathbf{y}_{\mathrm{ic},k}^{(q)} | d_k\big)$. Given the noise covariance matrix $\mathbf{C}_{\mathbf{vv},k}^{(q)}$, the likelihood function follows to

$$\begin{aligned}
p\big(\mathbf{y}_{\mathrm{ic},k}^{(q)} | d_k\big) &= \frac{1}{\pi^N |\mathbf{C}_{\mathbf{vv},k}^{(q)}|} \exp\big(-\mathbf{x}_k^{(q)H} \mathbf{C}_{\mathbf{vv},k}^{(q)-1} \mathbf{x}_k^{(q)}\big) \\
&= \alpha_k^{(q)} f_k^{(q)}(d_k), \tag{15}
\end{aligned}$$

with $\mathbf{x}_k^{(q)} = \mathbf{y}_{\mathrm{ic},k}^{(q)} - \mathbf{h}_k d_k$, a scaling factor

$$\alpha_k^{(q)} = \frac{1}{\pi^N |\mathbf{C}_{\mathbf{vv},k}^{(q)}|} \exp\big(-\mathbf{y}_{\mathrm{ic},k}^{(q)H} \mathbf{C}_{\mathbf{vv},k}^{(q)-1} \mathbf{y}_{\mathrm{ic},k}^{(q)}\big), \tag{16}$$

which is independent of d_k, and a function

$$\begin{aligned}
f_k^{(q)}(d_k) &= \exp\big(d_k^* \mathbf{h}_k^H \mathbf{C}_{\mathbf{vv},k}^{(q)-1} \mathbf{y}_{\mathrm{ic},k}^{(q)} + \mathbf{y}_{\mathrm{ic},k}^{(q)H} \mathbf{C}_{\mathbf{vv},k}^{(q)-1} \mathbf{h}_k d_k\big) \\
&\quad \cdot \exp\big(-d_k^* \mathbf{h}_k^H \mathbf{C}_{\mathbf{vv},k}^{(q)-1} \mathbf{h}_k d_k\big) \tag{17}
\end{aligned}$$

depending on d_k. With the above stated results at hand, the MMSE data symbol estimate $\hat{d}_k^{(q)}$ and its corresponding MSE $e_k^{(q)}$ can then be computed as

$$\begin{aligned}
\hat{d}_k^{(q)} &= E\big[d_k | \mathbf{y}_{\mathrm{ic},k}^{(q)}\big] = \sum_{s' \in \mathbb{S}} s' p\big[d_k = s' | \mathbf{y}_{\mathrm{ic},k}^{(q)}\big] \\
&= \frac{\sum_{s' \in \mathbb{S}} s' p\big(\mathbf{y}_{\mathrm{ic},k}^{(q)} | d_k = s'\big)}{\sum_{s' \in \mathbb{S}} p\big(\mathbf{y}_{\mathrm{ic},k}^{(q)} | d_k = s'\big)} = \frac{\sum_{s' \in \mathbb{S}} s' f_k^{(q)}(s')}{\sum_{s' \in \mathbb{S}} f_k^{(q)}(s')} \tag{18}
\end{aligned}$$

and

$$e_k^{(q)} = E\big[|d_k - \hat{d}_k^{(q)}|^2 | \mathbf{y}_{\mathrm{ic},k}^{(q)}\big] = \frac{\sum_{s' \in \mathbb{S}} |s' - \hat{d}_k^{(q)}|^2 f_k^{(q)}(s')}{\sum_{s' \in \mathbb{S}} f_k^{(q)}(s')}, \tag{19}$$

where a uniform prior PMF $p[d_k]$ is assumed.

Updating the data symbol soft estimates concludes one iteration. The succeeding iteration starts with conducting SIC following (11).

A quite interesting and not obvious result is verified in Appendix II, namely, that when using a zero vector as initialization for the estimated data symbol vector, after one

iteration the iterative SIC exhibits the same hard decision bit error probability as the LMMSE estimator. Hence, we employ the zero vector, which is also the prior mean of the data vector (since we assume a symmetric modulation alphabet and uniformly distributed data symbol probabilities), as initialization of the iterative SIC method.

III. SOFT INTERFERENCE CANCELLATION INSPIRED NEURAL NETWORK EQUALIZERS

As already mentioned in Sec. II-B, the model-based SIC method suffers from the issue that the computation of the inverse noise covariance matrix $\mathbf{C}_{\mathbf{vv},k}^{(q)^{-1}}$, also known as precision matrix, is computationally and numerically demanding while approximations have to be made in addition. With our proposed NN equalizers SICNNv1 and SICNNv2, whose layer structures are inspired by the iterative SIC method, we aim to overcome the weaknesses of the model-based SIC method. However, the SIC operation from the model-based method is preserved in the developed NNs, which is expected to help SICNNv1 and SICNNv2 to provide reliable soft estimates (required, e.g., to compute log-likelihood ratios in case of channel coded transmission), and allows to obtain interpretable intermediate quantities / variables inside the NN-based equalizers. While the structure of SICNNv1 is very similar to the model-based method and is specifically designed for SC-FDE communication systems, SICNNv2 is more general and can be applied for any communication system, where the system model can be formulated as in (5) with any matrix $\mathbf{H}$. For both SICNNv1 and SICNNv2, we additionally present a version with a reduced number of learnable parameters, since we exploit the knowledge that every stage of SICNNv1/SICNNv2 has to provide estimates of the posterior data symbol probabilities, given the estimates of the previous stage and the received vector, and thus should work with the same set of learnable parameters. The parameter-reduced versions are referred to as SICNNv1Red and SICNNv2Red.

A. SICNNv1

The NN architecture is deduced by deep unfolding [27] the iterative SIC method described in Sec. II-B to Q stages of SICNNv1. That is, every iteration of the model-based SIC method (the outer loop of Algorithm 1), corresponds to one stage of SICNNv1. The steps conducted in one stage of SICNNv1, which is schematically shown in Fig. 1, are very similar to those of the model-based method described in Algorithm 1, however, the model-based computations in line 6 and line 7 of Algorithm 1, are accomplished using data-driven FCNNs. Let us describe the structure of stage q of SICNNv1, $q \in \{0, \dots, Q-1\}$, in more detail, starting at its input. The inputs of stage q are the received vector $\mathbf{y}$, the sampled frequency response $\operatorname{diag}(\widetilde{\mathbf{H}})$, the noise variance σ_{n}^2, and the vectors $\hat{\mathbf{p}}_k^{(q-1)} = [\hat{\mathbf{p}}_{k,\mathrm{Re}}^{(q-1)T}, \hat{\mathbf{p}}_{k,\mathrm{Im}}^{(q-1)T}]^T$, $\hat{\mathbf{p}}_k^{(q-1)} \in [0,1]^{2|\mathbb{S}'|}$, $k \in \{0, \dots, N_{\mathrm{d}} - 1\}$, where $\mathbb{S}' = \operatorname{Re}\{\mathbb{S}\} = \operatorname{Im}\{\mathbb{S}\}$

(assuming a symmetric alphabet $\mathbb{S}$). The elements $\hat{p}_{k,\mathrm{Re},l}^{(q-1)}$ and $\hat{p}_{k,\mathrm{Im},l}^{(q-1)}$ of the vectors $\hat{\mathbf{p}}_{k,\mathrm{Re}}^{(q-1)}$ and $\hat{\mathbf{p}}_{k,\mathrm{Im}}^{(q-1)}$, respectively, are the estimates of stage $(q-1)$ for the data symbol posterior probabilities $p[\operatorname{Re}\{d_k\} = s_l | \mathbf{y}_{\mathrm{ic},k}^{(q-1)}]$ and $p[\operatorname{Im}\{d_k\} = s_l | \mathbf{y}_{\mathrm{ic},k}^{(q-1)}]$, respectively, where $s_l \in \mathbb{S}'$ are the uniquely numbered symbols of $\mathbb{S}'$, $l \in \{0, \dots, |\mathbb{S}'| - 1\}$. For the inputs of the first stage ($q = 0$), $\hat{\mathbf{p}}_{k,\mathrm{Re}}^{(-1)} = \hat{\mathbf{p}}_{k,\mathrm{Im}}^{(-1)} = \frac{1}{|\mathbb{S}'|}\mathbf{1}$ is chosen, i.e., the estimated posterior probabilities are initialized uniformly, which represents the prior data symbol probability distribution. The values of the elements of $\hat{\mathbf{p}}_{k,\mathrm{Re}}^{(q)}$ and $\hat{\mathbf{p}}_{k,\mathrm{Im}}^{(q)}$ are updated in every stage following the procedure described below.

Similar to the model-based method, in the first step (① in Fig. 1) of the qth stage

$$\hat{d}_{k,\mathrm{Re}}^{(q-1)} = \sum_{l=0}^{|\mathbb{S}'|-1} s_l \hat{p}_{k,\mathrm{Re},l}^{(q-1)}, \qquad \hat{d}_{k,\mathrm{Im}}^{(q-1)} = \sum_{l=0}^{|\mathbb{S}'|-1} s_l \hat{p}_{k,\mathrm{Im},l}^{(q-1)},$$
(20)

and

$$e_k^{(q-1)} = \sqrt{\left(e_{k,\mathrm{Re}}^{(q-1)}\right)^2 + \left(e_{k,\mathrm{Im}}^{(q-1)}\right)^2},$$
(21)

with the estimated MSEs

$$e_{k,\mathrm{Re/Im}}^{(q-1)} = \sum_{l=0}^{|\mathbb{S}'|-1} \left(s_l - \hat{d}_{k,\mathrm{Re/Im}}^{(q-1)}\right)^2 \hat{p}_{k,\mathrm{Re/Im},l}^{(q-1)},$$
(22)

are computed, where $e_k^{(q-1)}$ is utilized as a reliability measure of the corresponding data symbol estimate. Note that these quantities can be computed independently for every data symbol index k, and thus the blocks ① are drawn isolated from each other in Fig. 1.

In a second step (② in Fig. 1), model-based interference cancellation is carried out by computing

$$\mathbf{y}_{\mathrm{ic},k}^{(q)} = \mathbf{y} - \bar{\mathbf{H}}_k \hat{\bar{\mathbf{d}}}_k^{(q-1)},$$
(23)

using the data symbol estimates $\hat{\bar{d}}_{k,l}^{(q-1)} = \hat{d}_{l,\mathrm{Re}}^{(q-1)} + j\hat{d}_{l,\mathrm{Im}}^{(q-1)}$, $l \in \{0, \dots, k-1, k+1, \dots, N_{\mathrm{d}} - 1\}$ from stage $(q-1)$ for $\hat{\bar{\mathbf{d}}}_k^{(q-1)}$.

Instead of computing (an approximate of) the precision matrix $\mathbf{C}_{\mathbf{vv},k}^{(q)^{-1}}$ in a model-based fashion, we estimate it by utilizing fully-connected feedforward layers, which we also refer to as sub-NNs. A straightforward approach is to estimate the real and the imaginary part of the N^2 elements of $\mathbf{C}_{\mathbf{vv},k}^{(q)^{-1}}$. We exploit two observations for both reducing the number of parameters to be estimated by a sub-NN and ensuring that the estimated precision matrix satisfies the properties implied by its definition. Firstly, the covariance matrix $\mathbf{C}_{\mathbf{vv},k}^{(q)}$, and thus also its inverse, has to be a Hermitian, positive definite matrix. That is, $\mathbf{C}_{\mathbf{vv},k}^{(q)^{-1}}$ can be decomposed into the matrix product $\mathbf{C}_{\mathbf{vv},k}^{(q)^{-1}} = \mathbf{B}_k^{(q)H}\mathbf{B}_k^{(q)}$, where $\mathbf{B}_k^{(q)}$ is the matrix to be estimated by the sub-NNs. Secondly, our empirical investigations showed, that for the regarded SC-FDE communication system a precision matrix $\mathbf{C}_{\mathbf{vv},k}^{(q)^{-1}}$ exhibits significant
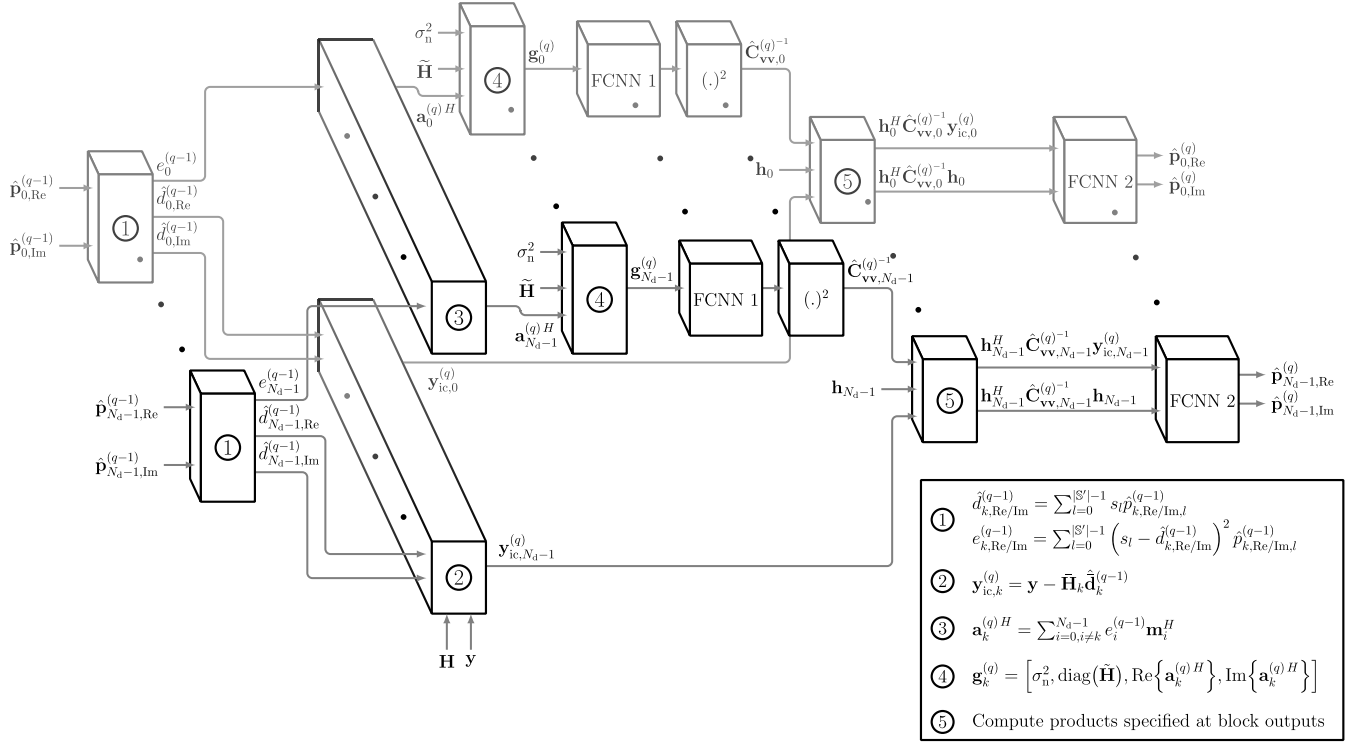
FIGURE 1. Schematic structure of one stage of SICNNv1.

non-zero values only on the major and the first few minor diagonals, and thus can be approximated as a band matrix. In the initial version of SICNNv1 described in [1] (where it is simply referred to as SICNN), we specify $\mathbf{B}_k^{(q)}$ to be a lower triangular matrix containing non-zero values only on the main diagonal and the first n_{md} minor diagonals, where $n_{\mathrm{md}} \in \mathbb{N}_0$ is a hyperparameter of SICNN, which tremendously reduces the number of non-zero elements of $\mathbf{B}_k^{(q)}$ to be estimated. The non-zero elements of the complex-valued matrix $\mathbf{B}_k^{(q)}$ are to be estimated by two separate sub-NNs, using one sub-NN for the real part and one for the imaginary part. As stated in [1], hyperparameter optimization turns out that the best equalization performance is achieved with $n_{\mathrm{md}} = 0$, i.e., the precision matrix is assumed to be a diagonal matrix. This insight motivates an adaption of the architecture of SICNN for the structure of SICNNv1. As depicted in Fig. 1, a single sub-NN FCNN 1 is employed to estimate the major diagonal of $\mathbf{C}_{\mathbf{vv},k}^{(q)}$. To ensure positive definiteness, the final estimates of the major diagonal of $\mathbf{C}_{\mathbf{vv},k}^{(q)^{-1}}$ are obtained by squaring the outputs of FCNN 1. FCNN 1 has $n_{\mathrm{L,C}}$ hidden layers with $n_{\mathrm{H,C}}$ neurons per hidden layer, ReLU activation functions, and a batch norm layer after the input layer. For determining the required inputs of FCNN 1, we reconsider the computation of $\mathbf{C}_{\mathbf{vv},k}^{(q)}$ in (13) and the quantities involved there. Besides the terms describing correlations between $\mathbf{w}$ and $\hat{\tilde{\mathbf{d}}}_k^{(q-1)}$, only terms consisting of σ_{n}^2, $\widetilde{\mathbf{H}}$, and $\bar{\mathbf{M}}_k \widetilde{\mathbf{E}}_k^{(q-1)} \bar{\mathbf{M}}_k^H$ occur in (13). When replacing $\widetilde{\mathbf{E}}_k^{(q-1)}$ by its approximation $\mathbf{E}_k^{(q-1)}$, the latter term can be expressed

as $\bar{\mathbf{M}}_k \mathbf{E}_k^{(q-1)} \bar{\mathbf{M}}_k^H = \sum_{i=0,i\neq k}^{N_{\mathrm{d}}-1} e_i^{(q-1)} \mathbf{m}_i \mathbf{m}_i^H =: \mathbf{A}_k^{(q)}$, with $\mathbf{m}_i$ as the ith column of $\mathbf{M}$. $\mathbf{A}_k^{(q)}$ is a Hermitian Toeplitz matrix, consequently it is already fully described by its first row $\mathbf{a}_k^{(q)H} = \left[\mathbf{A}_k^{(q)}\right]_{0*} = \sum_{i=0,i\neq k}^{N_{\mathrm{d}}-1} e_i^{(q-1)} \mathbf{m}_i^H$, where we exploit $m_{i,0} = 1$ for all i in the last step. The vectors $\mathbf{a}_k^{(q)H}$, which are computed in block ③, are concatenated in block ④ with σ_{n}^2 and $\mathrm{diag}(\widetilde{\mathbf{H}})$ to the input vector of FCNN 1

$$\mathbf{g}_k^{(q)} = \left[\sigma_{\mathrm{n}}^2,\ \mathrm{diag}(\widetilde{\mathbf{H}}),\ \mathrm{Re}\{\mathbf{a}_k^{(q)H}\},\ \mathrm{Im}\{\mathbf{a}_k^{(q)H}\}\right]. \quad (24)$$

With a given estimated precision matrix $\hat{\mathbf{C}}_{\mathbf{vv},k}^{(q)^{-1}}$, the posterior PMF $p[d_k|\mathbf{y}_{\mathrm{ic},k}^{(q)}]$ should be estimated. Experiments, where the posterior PMF is computed as described in Sec. II-B in a model-based fashion did not lead to satisfying performance. We assume a major reason for this issue is that the estimate $\hat{\mathbf{C}}_{\mathbf{vv},k}^{(q)^{-1}}$ provided by FCNN1 is not precise enough to be treated as the exact precision matrix. Hence, we utilize another sub-NN (FCNN 2 in Fig. 1), which is trained (jointly with FCNN1) to estimate the posterior PMF $p[d_k|\mathbf{y}_{\mathrm{ic},k}^{(q)}]$ and can cope with inaccuracies in the estimated precision matrix. More specifically, the output of FCNN2 is the vector $\hat{\mathbf{p}}_k^{(q)} = \left[\hat{\mathbf{p}}_{k,\mathrm{Re}}^{(q)\,T}, \hat{\mathbf{p}}_{k,\mathrm{Im}}^{(q)\,T}\right]^T$ (which is also the output of the qth SICNNv1 stage), containing estimates for the data symbol posterior probabilities, as introduced earlier in this section. To specify the required input quantities of FCNN 2 for estimating the posterior PMF $p[d_k|\mathbf{y}_{\mathrm{ic},k}^{(q)}]$, let us consider

its representation

$$p[d_k|\mathbf{y}_{\mathrm{ic},k}^{(q)}] = \frac{p(\mathbf{y}_{\mathrm{ic},k}^{(q)}|d_k)}{\sum_{s' \in \mathbb{S}} p(\mathbf{y}_{\mathrm{ic},k}^{(q)}|s')} = \frac{f_k^{(q)}(d_k)}{\sum_{s' \in \mathbb{S}} f_k^{(q)}(s')} , \quad (25)$$

by applying the Bayesian rule and assuming a uniform prior data symbol probability, where $f_k^{(q)}(.)$ is defined in (17). Due to the definition of $f_k^{(q)}(.)$, the posterior PMF $p[d_k|\mathbf{y}_{\mathrm{ic},k}^{(q)}]$ depends only on $\mathbf{h}_k^H \mathbf{C}_{\mathbf{vv},k}^{(q)^{-1}} \mathbf{y}_{\mathrm{ic},k}^{(q)}$, its complex conjugate, and $\mathbf{h}_k^H \mathbf{C}_{\mathbf{vv},k}^{(q)^{-1}} \mathbf{h}_k$. Therefore, the input vector of FCNN 2 is chosen to be

$$\mathbf{s}_k^{(q)} = \left[\mathrm{Re}\{\mathbf{h}_k^H \hat{\mathbf{C}}_{\mathbf{vv},k}^{(q)^{-1}} \mathbf{y}_{\mathrm{ic},k}^{(q)}\}, \ \mathrm{Im}\{\mathbf{h}_k^H \hat{\mathbf{C}}_{\mathbf{vv},k}^{(q)^{-1}} \mathbf{y}_{\mathrm{ic},k}^{(q)}\}, \right.$$
$$\left. \mathbf{h}_k^H \hat{\mathbf{C}}_{\mathbf{vv},k}^{(q)^{-1}} \mathbf{h}_k \right]^T , \quad (26)$$

where the elements of $\mathbf{s}_k^{(q)}$ are computed in block ⑤. FCNN 2 consists of $n_{\mathrm{L,Pr}}$ fully-connected hidden layers with $n_{\mathrm{H,Pr}}$ neurons per hidden layer, ReLU activation functions between each hidden layer, and a batch norm layer after the input layer. Further, two independent softmax functions are utilized as output activation functions to obtain $\hat{\mathbf{p}}_{k,\mathrm{Re}}^{(q)}$ and $\hat{\mathbf{p}}_{k,\mathrm{Im}}^{(q)}$, which are concatenated to the stage output $\hat{\mathbf{p}}_k^{(q)}$.

An investigation of the inputs of sub-NN FCNN 2 as defined in (26) reveals a large variation of the values of $\mathbf{s}_k^{(q)}$, which depends on both the data symbol index k and the SICNNv1 stage index q. Hence, in the qth SICNNv1 stage we suggest to multiply $\mathbf{y}_{\mathrm{ic},k}^{(q)}$ and $\mathbf{h}_k$ by $||\mathbf{y}_{\mathrm{ic},k}^{(q)}||_2^{-1/2}$, which turns out to lead to a more robust training procedure.

In one stage, we use the same sub-NNs for estimating the precision matrix and the posterior data symbol probabilities for all N_{d} data symbols to be estimated. However, different sub-NNs are utilized from stage to stage, i.e. their learnable parameters are in general different to those of the sub-NNs of the remaining stages, whereby the hyperparameters of the sub-NNs are the same for all stages.

We optimize the parameters of SICNNv1 by employing a custom loss function based on the cross entropy loss, which can be computed as

$$f_{\mathrm{CE}}(\boldsymbol{\alpha}, \boldsymbol{\beta}) := -\frac{1}{D} \sum_{d=0}^{D-1} \alpha_d \ln(\beta_d) ,$$

with any vectors $\boldsymbol{\alpha}, \boldsymbol{\beta} \in [0, 1]^D$ and $\sum_d \alpha_d = \sum_d \beta_d = 1$. More specifically, instead of utilizing only the final output of SICNNv1 for computing the loss value, the custom loss is based on Q partial cross entropy losses of all Q stage outputs, which are weighted by the corresponding stage index q. That is, the employed custom loss function is given by

$$\ell(\mathbf{d}_{\mathrm{oh}}, \hat{\mathcal{P}}) = \frac{1}{QN_{\mathrm{d}}} \sum_{q=0}^{Q-1} \sum_{k=0}^{N_{\mathrm{d}}-1} w_q \big(f_{\mathrm{CE}}(\mathbf{d}_{\mathrm{oh},k,\mathrm{Re}}, \hat{\mathbf{p}}_{k,\mathrm{Re}}^{(q)})$$
$$+ f_{\mathrm{CE}}(\mathbf{d}_{\mathrm{oh},k,\mathrm{Im}}, \hat{\mathbf{p}}_{k,\mathrm{Im}}^{(q)}) \big) , \quad (27)$$

where $\hat{\mathcal{P}} := \big\{ \{\hat{\mathbf{p}}_{k,\mathrm{Re/Im}}^{(q)}\}_{k=0}^{N_{\mathrm{d}}-1} \big\}_{q=0}^{Q-1}$ is the collection of all stage outputs, $w_q = (q+1)/\sum_{q=1}^{Q-1} q$ is the weighting factor of the

partial losses, and $\mathbf{d}_{\mathrm{oh},k,\mathrm{Re}}$ and $\mathbf{d}_{\mathrm{oh},k,\mathrm{Im}}$ are one-hot vectors corresponding to the real and imaginary part of a data symbol d_k, respectively. This custom loss function is inspired by the loss function employed for training DetNet [18] and by the auxiliary classifiers of GoogLeNet [41], and should lead to a faster converging training.

B. SICNNv2

The first operations conducted in a stage of SICNNv2 are equivalent to those in a stage of SICNNv1. That is, given the estimated posterior probabilities $\hat{\mathbf{p}}_k^{(q-1)}$ from the previous stage $(q-1)$, the corresponding data symbol estimates $\hat{d}_{k,\mathrm{Re/Im}}^{(q-1)}$ and estimated MSEs $e_{k,\mathrm{Re/Im}}^{(q-1)}$ are computed according to (20) and (22), respectively. With the data symbol estimates $\hat{d}_{k,\mathrm{Re/Im}}^{(q-1)}$ for all data symbols in the data vector at hand, interference cancellation is conducted as a next step according to (23) to obtain $\mathbf{y}_{\mathrm{ic},k}^{(q)}$. However, as shown in Fig. 2, the remaining structure of a stage of SICNNv2 differs from the stage structure of SICNNv1. Specifically, while the further inference steps conducted in a stage of SICNNv1 to obtain the stage output are similar to the steps of the model-based algorithm, in an SICNNv2 stage an FCNN is employed for directly estimating the posterior data symbol probabilities using the input vector

$$\mathbf{z}_k^{(q)} = \big[\rho_k^{(q)} \mathrm{Re}\{\mathbf{y}_{\mathrm{ic},k}^{(q)T}\}, \ \rho_k^{(q)} \mathrm{Im}\{\mathbf{y}_{\mathrm{ic},k}^{(q)T}\}, \ \rho_k^{(q)} \mathrm{Re}\{\mathbf{h}_k^T\},$$
$$\rho_k^{(q)} \mathrm{Im}\{\mathbf{h}_k^T\}, \ e_{k,\mathrm{Re}}^{(q-1)}, \ e_{k,\mathrm{Im}}^{(q-1)}, \ \rho_k^{(q)^2} \sigma_{\mathrm{n}}^2 \big]^T ,$$

with a normalization factor $\rho_k^{(q)} = ||\mathbf{y}_{\mathrm{ic},k}^{(q)}||_2^{-1/2}$. The estimated posterior data symbol probabilites are contained in the output vector $\hat{\mathbf{p}}_k^{(q)} = \big[\hat{\mathbf{p}}_{k,\mathrm{Re}}^{(q)T}, \hat{\mathbf{p}}_{k,\mathrm{Im}}^{(q)T} \big]^T$ of the FCNN, which is also the output of the qth stage of SICNNv2. The FCNN has n_{L} hidden layers, n_{H} neurons per hidden layer, a batch norm layer after the input layer and every third hidden layer, and ReLU activation. SICNNv2 is trained with the same loss function (27) as SICNNv1.

The architecture of SICNNv2 is solely based on the idea of SIC, and is more universal than that of SICNNv1 since no properties of an SC-FDE system are utilized. Hence, SICNNv2 can be employed for equalization in other communication systems like, e.g., general MIMO systems or UW-OFDM systems. However, for SC-FDE we expect a higher equalization complexity and maybe a worse BER performance of SICNNv2 compared to SICNNv1 as less model knowledge is incorporated.

C. PARAMETER REDUCTION: SICNNv1Red and SICNNv2Red

For reducing the number of parameters to be trained, we exploit the fact that in every stage of SICNNv1 and SICNNv2 the same task is fulfilled, namely, the posterior data symbol probabilities are to be estimated given the estimated posterior data symbol probabilities from the previous stage, the received vector, the channel matrix, and the noise variance. Hence, we employ the same sub-NNs in
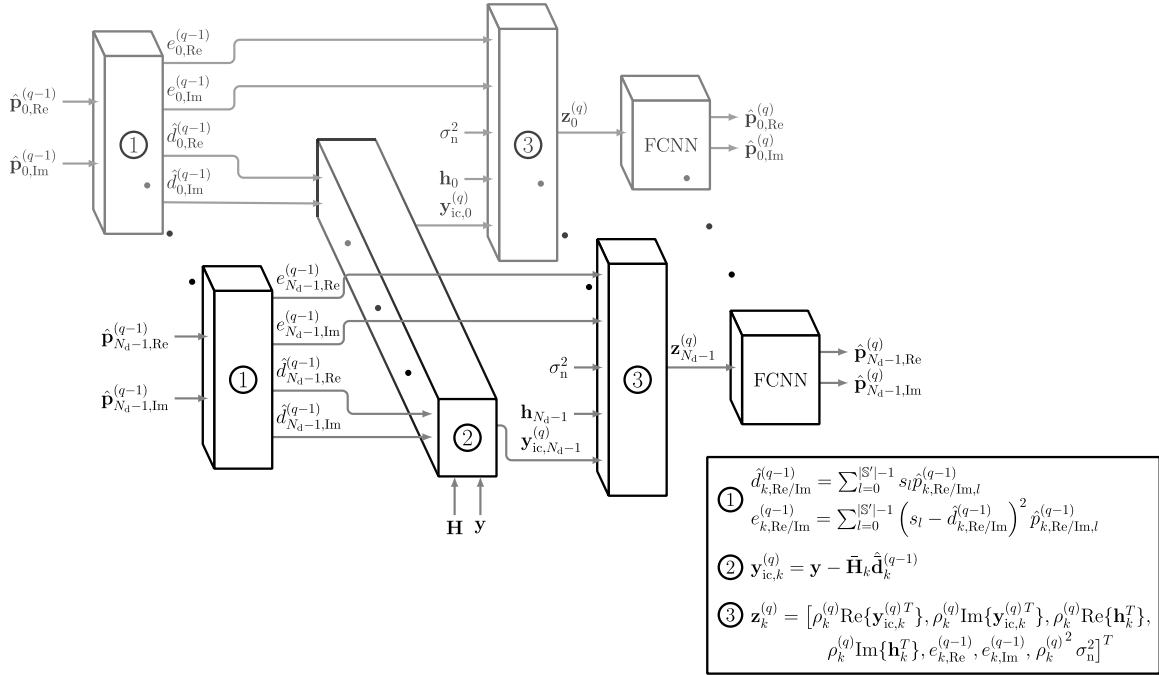
$$\boxed{\begin{aligned} \textcircled{1} \quad & \hat{d}_{k,\mathrm{Re/Im}}^{(q-1)} = \sum_{l=0}^{|\mathbb{S}'|-1} s_l \hat{p}_{k,\mathrm{Re/Im},l}^{(q-1)} \\ & e_{k,\mathrm{Re/Im}}^{(q-1)} = \sum_{l=0}^{|\mathbb{S}'|-1} \left(s_l - \hat{d}_{k,\mathrm{Re/Im}}^{(q-1)} \right)^2 \hat{p}_{k,\mathrm{Re/Im},l}^{(q-1)} \\ \textcircled{2} \quad & \mathbf{y}_{\mathrm{ic},k}^{(q)} = \mathbf{y} - \bar{\mathbf{H}}_k \hat{\bar{\mathbf{d}}}_k^{(q-1)} \\ \textcircled{3} \quad & \mathbf{z}_k^{(q)} = \big[\rho_k^{(q)} \mathrm{Re}\{\mathbf{y}_{\mathrm{ic},k}^{(q)\,T}\}, \rho_k^{(q)} \mathrm{Im}\{\mathbf{y}_{\mathrm{ic},k}^{(q)\,T}\}, \rho_k^{(q)} \mathrm{Re}\{\mathbf{h}_k^{T}\}, \\ & \qquad \rho_k^{(q)} \mathrm{Im}\{\mathbf{h}_k^{T}\}, e_{k,\mathrm{Re}}^{(q-1)}, e_{k,\mathrm{Im}}^{(q-1)}, {\rho_k^{(q)}}^2 \sigma_{\mathrm{n}}^2 \big]^{T} \end{aligned}}$$

FIGURE 2. Schematic structure of one stage of SICNNv2.

every stage of the NN-based equalizers, leading to the corresponding parameter-reduced versions SICNNv1Red and SICNNv2Red. These parameter-reduced NNs can also be viewed as a single stage where its output is fed back Q times.

This NN architecture distinctly reduces the number of parameters to be optimized, which reduces the computational effort for training, and is also supposed to lead to a more robust training procedure and a smaller amount of training data to be required. However, it turns out that the employed loss function for training the parameter-reduced NNs has to be slightly altered for obtaining a good performance. More specifically, the outputs of the stages with higher stage index q are given a higher importance by changing the weights w_q of the loss function (27) for training SICNNv1Red and SICNNv2Red to

$$w_q = \frac{(q+1)^r}{\sum_{q=1}^{Q-1} q^r}, \qquad (28)$$

where r is a hyperparameter, which we choose for SC-FDE systems to be $r = 4$.

IV. TRAINING SET GENERATION AND DATA NORMALIZATION

In this section, we describe a novel approach for generating training sets for NN-based equalizers. For the regarded SC-FDE systems, this approach considerably improves the performance of NN-based equalizers at high SNRs. Further, we briefly describe a data normalization scheme specifically tailored for SC-FDE, which was already presented in [31].

A. TRAINING SET GENERATION

The achieved BER performances of model-based and NN-based equalizers are generally evaluated in a specified E_{b}/N_0 interval, where E_{b} is the mean bit energy and N_0 is the noise power spectral density on receiver side, i.e., E_{b}/N_0 is a measure for the SNR. To generate the training set for NN-based equalizers, typically sample data transmissions over channel realizations[4] drawn from a statistical channel model are conducted, where E_{b}/N_0 for the data transmission is selected randomly with a uniform distribution within a specified range, short training SNR range. The upper and lower limits of the training SNR range are typically hyperparameters, which are to be selected carefully, since they have a significant influence on the performance of trained NN-based equalizers [39], [42], [43]. Despite a careful selection of the training SNR range, the issue of "flattening out" BER curves can occur. That is, although NN-based equalizers perform well for a wide E_{b}/N_0 range, at higher E_{b}/N_0 values, and thus low BERs (of, e.g., 10^{-5} or 10^{-6}), their BER curves do not fall as steeply as those of many model-based equalizers. Although this issue occurs for most NN-based equalizers (shown, e.g., in [44] and [31]), interestingly, there are only very few works like [43], where proper training of NN-based equalizers is addressed. In this work, we propose a novel approach for the generation of training sets for NN-based equalizers. By training NN-based data estimators with these specifically generated training sets, the issue of flattening out BER curves at high SNRs can be mitigated significantly.

---

[4]These channel realizations are not used for evaluation, but the same statistical channel model is used to generate channels used for evaluation.

Typically, even low complex equalizers like the LMMSE equalizer achieve low BERs at high SNRs (as it can be seen, e.g., in Fig. 4, where the LMMSE estimator achieves a BER of $5 \cdot 10^{-5}$ at $E_b/N_0 = 14$ dB). In other words, the decision boundaries of the baseline LMMSE equalizer and the optimal bit-wise MAP equalizer differ only slightly. That is, by randomly generating data transmissions for the training set at high SNRs, for those data transmissions even with the baseline LMMSE estimator only very few data symbol estimation errors occur. However, the NN-based equalizers are expected to approximate the optimal bit-wise MAP estimator. Hence, for NN training, exactly those few received data symbols are of interest, where with the baseline LMMSE estimator a wrong estimate is obtained for the corresponding transmitted data symbol, while the optimal estimator still achieves a correct data symbol estimate. Since only a few of those important received data symbols are contained in the training set when generating the training data randomly, their influence on the training loss of the NN is small, leading to the aforementioned issue of flattening out BER curves. With the aforementioned observations in mind, we suggest the following method for generating the training set of NN-based equalizers for SC-FDE systems: instead of randomly selecting an SNR value within the SNR training range for the transmission of every data burst that is contained in the training set, we define an evenly spread grid on the SNR training range (on a linear scale). The number of grid points coincides with the number of channels over which data transmissions are to be conducted to generate the training set. For every SNR grid point, a channel realization is drawn from the assumed statistical channel model, and a burst of N_{burst} data vectors $\mathbf{d}$ is transmitted over this channel. The corresponding received vectors $\mathbf{y}$ are equalized using a baseline LMMSE estimator. Instead of including all data vectors of the transmitted burst in the training set, only those are retained where the baseline equalizer produces at least N_{epd} errors per data vector. Since particularly at higher SNRs the number of retained data vectors is generally far lower than N_{burst}, another burst of data vectors is generated and transmitted over the same channel, again followed by keeping only the data vectors where at least N_{epd} errors per data vector are produced by the baseline estimator. This procedure is repeated until N_{burst} data vectors are found for the specific channel, which are then included in the training set. However, for flat channels, even with the baseline equalizer no or too few errors occur such that no data vectors are found which could be included in the training set. Therefore, a stopping criterion has to be introduced, where after N_{check} burst generations the number of retained data vectors is checked. If the number of retained data vectors is smaller than, e.g., $0.1 N_{burst}$, the current channel realization is discarded. While keeping the SNR value corresponding to the specified SNR grid point, a new channel realization is drawn from the statistical channel model, and the same data vector selection process as described above is carried out. The parameters for the training set generation depend on the communication system setup for which the NN-based

equalizers are trained, and thus they are specified in Sec. V-A individually for every setup.

B. DATA NORMALIZATION

Normalizing the input data of NNs is generally considered to be very important for training convergence when optimizing their learnable parameters via backpropagation [45], [46]. While in many current publications on NN-based data estimation in MIMO systems over uncorrelated Rayleigh fading channels no normalization of the NN input data is applied (cf., e.g., [17], [18], [19]), we showed in [39] and [47] that for a so-called UW-OFDM communication system a proper data normalization is of major importance for the performance of NN-based equalizers (for a visualization of the influence of data normalization on the performance of NN-based equalizers for UW-OFDM, we refer to [47]). With the same idea as for UW-OFDM in mind, namely to apply a normalization scheme leading to variances of the elements of the noise vector that are independent of the multipath channel, we implement a data normalization scheme for SC-FDE systems. This data normalization scheme for SC-FDE is elucidated in [31], and thus we repeat here only the result. To obtain channel-independent noise variances $\text{var}(w_i)$, the system model (5) has to be multiplied by $\mathbf{K} = \kappa \widetilde{\mathbf{H}}^{-1/2}$, where

$$\kappa = \sqrt{\text{tr}\{\widetilde{\mathbf{H}}\}/\text{tr}\{\widetilde{\mathbf{H}}\mathbf{M}\mathbf{M}^H\widetilde{\mathbf{H}}\}}. \tag{29}$$

The normalization of the input data of the NN-based equalizers is implemented by multiplying both $\mathbf{y}$ and $\widetilde{\mathbf{H}}$ by $\mathbf{K}$ as part of pre-processing, and is neglected in the remainder of this paper for the sake of readability.

V. RESULTS

In this section, we investigate the proposed versions of SICNN thoroughly by means of simulations of data transmission in an indoor frequency selective environment. To demonstrate the wide applicability the proposed NN-based approaches, we evaluate them for a number of different SC-FDE communication system setups. We show simulation results for SC-FDE systems with both UW and CP guard intervals. Besides simulations with a QPSK modulation alphabet, also results for a 16-QAM alphabet are provided. Most of the simulations are conducted assuming perfect channel knowledge on receiver side. The robustness of the proposed NN-based data estimators in case of imperfect channel knowledge, is proven by simulating their performance for estimated channel impulse responses. Further, we highlight the performance improvements of NN-based equalizers when being trained on a training set generated with our proposed approach, presented in Sec. IV-A. We compare SICNNv1 and SICNNv2, and their corresponding parameter-reduced versions SICNNv1Red and SICNNv2Red, with state-of-the-art model-based and NN-based equalizers in terms of both their achieved BER performance over a specified SNR range and their computational complexity. More specifically, for comparison with model-based equalizers, we use the LMMSE

estimator [35], the iterative DFE (implemented in the same way as described in [47] for UW-OFDM systems), and the iterative SIC method, where the approximation (14) is employed. We compare the proposed NNs with the state-of-the-art NN-based data estimators OAMP-Net2 [19] and DetNet [18], whereby we do not use DetNet as proposed in [18] for MIMO systems, but a better performing version that is adapted for SC-FDE systems [31]. Moreover, we show the BER performance and computational complexity of KAFCNN from [31], which is an FCNN that is designed for equalization in SC-FDE systems by using a layer conducting an inverse DFT as a last layer, i.e., the knowledge that the data symbols being defined in time domain are to be estimated given a received vector in frequency domain is incorporated.

Moreover, for an SC-FDE system with a UW guard interval we present the influence of a limited training set size on the BER performance of selected NN-based equalizers to investigate the ''data hunger'' of an NN depending on its number of learnable parameters.

Finally, we also present performance results for SIC-NNv2 being employed as an equalizer in communication system utilizing the so-called UW-OFDM signaling scheme. With these results we want the highlight the wide applicability and the versatility of the proposed NN-based equalizers.

A. SIMULATION SETUP AND NEURAL NETWORK TRAINING

The shown simulation results are obtained by simulating data transmission without channel coding. Apart from Sec. V-D, all simulation settings, results, interpretations, and conclusions in this work are given for SC-FDE systems. The simulation setup and the results for the simulation of SIC-NNv2 as an equalizer in a UW-OFDM system are detailed in Sec. V-D. For SC-FDE communications with a UW guard interval, simulations are conducted with the SC-FDE system parameters $N_d = 20$, $N_g = 12$ (i.e., $N = 32$), RRC roll-off factor $\alpha = 0.25$, a baseband sampling time $T_s = 52$ ns. Further, unless noted otherwise, a QPSK modulation alphabet is employed, and perfect channel knowledge is assumed on receiver side. The parameters of the simulations of SC-FDE systems with a CP guard interval differ from those with a UW guard interval by the data vector length $N_d = 32$, all other parameters are maintained.

The achieved BER performances of the different equalizers are plotted in a specified E_b/N_0 interval. The presented BER performances for SC-FDE systems are averaged results over 7000 different multipath channel realizations, which are modeled as described in [48] in form of tapped delay lines with uniformly distributed phase, Rayleigh distributed magnitude, and an exponentially decaying power profile with a root mean square delay spread of $\tau_{RMS} = 100$ ns. The data transmission is conducted in form of data bursts containing 1000 blocks of payload data per burst, where the channel is assumed to

TABLE 1. Parameters of the training set generation approach for different sc-fde system settings.

SC-FDE parameters	Baseline equalizer	N_{epd}	N_{burst}	E_b/N_0 train. range in dB
UW guard, QPSK	LMMSE, eq. (6)	3	100	$[2, 12.5]$
CP guard, QPSK	LMMSE, eq. (7)	2	100	$[5, 18]$
UW guard, 16-QAM	LMMSE, eq. (6)	3	100	$[6, 19]$
UW guard, QPSK, channel est.	approx. LMMSE, eq. (7)	3	100	$[3, 16]$

be stationary for one burst and changes independently of its previous realizations for every burst.

For training the NN-based equalizers, we generate training sets with the proposed approach described in Sec. IV-A. The selected parameters of the training set generation method N_{epd} and N_{burst}, the training E_b/N_0 range, as well as the employed baseline equalizer for selecting the data vectors for the training set are summarized for all SC-FDE system setups in Tab. 1. Unless stated otherwise, every training set consists of data transmissions over 30000 different channels. For training of all NNs early stopping is used. That is, the BER performance on a validation set is evaluated after every epoch, and the set of learnable NN parameters achieving the best validation performance is chosen after training for a pre-defined maximum number of epochs.

The hyperparameters of the NN-based equalizers are found using the hyperparameter optimization framework Optuna [49]. For SICNNv1, the best hyperparameter settings found are given in Tab. 2, and we train it for 25 epochs at most. For SICNNv1Red, the same hyperparameters as for SICNNv1 are used apart from a learning rate $\eta_{SICNNv1Red}$. The hyperparameters of SICNNv2 are also given in Tab. 2, and we train it for a maximum of 25 epochs. The hyerparameters of SICNNv2Red differ from those of SICNNv2 only by $\eta_{SICNNv2Red}$. For DetNet, OAMP-Net2, and KAFCNN the best hyperparameter found are shown in Tab. 3. Moreover, DetNet and KAFCNN are trained for 60 epochs at most, and OAMP-Net2 for a maximum of 15 epochs.

B. BIT ERROR RATIO PERFORMANCE FOR SC-FDE
1) UNIQUE WORD GUARD INTERVAL, QPSK

We start with investigations on an SC-FDE system with UW guard interval and QPSK modulation alphabet. First, we investigate the influence of the number of iterations Q of the model-based iterative SIC method as well as the number of stages Q of SICNNv1 on the achieved BER performance to highlight similarities and differences between the model-based and the NN-based approach. As shown in Fig. 3,

TABLE 2. Hyperparameter settings of the proposed NN equalizers. The hyperparameters of the parameter-reduced NNs SICNNv1Red and SICNNv2Red differ from the hyperparameters of SICNNv1 and SICNNv2 only by the learning rate $\eta_{\text{SICNNv1Red}}$ and $\eta_{\text{SICNNv2Red}}$, respectively.

SC-FDE parameters	SICNNv1 / SICNNv1Red							SICNNv2 / SICNNv2Red				
	η_{SICNNv1}	$\eta_{\text{SICNNv1Red}}$	Q	$n_{\text{L,C}}$	$n_{\text{H,C}}$	$n_{\text{L,pr}}$	$n_{\text{H,pr}}$	η_{SICNNv2}	$\eta_{\text{SICNNv2Red}}$	Q	n_{L}	n_{H}
UW guard, QPSK	$6 \cdot 10^{-4}$	$3 \cdot 10^{-5}$	7	3	70	2	10	$5 \cdot 10^{-4}$	$1 \cdot 10^{-4}$	7	4	200
CP guard, QPSK	$1 \cdot 10^{-3}$	$7 \cdot 10^{-5}$	7	3	100	2	10	$9 \cdot 10^{-4}$	$1 \cdot 10^{-4}$	7	4	250
UW guard, 16-QAM	$9 \cdot 10^{-4}$	$4 \cdot 10^{-5}$	7	3	70	3	20	$1 \cdot 10^{-3}$	$2.5 \cdot 10^{-4}$	7	4	230

TABLE 3. Hyperparameter settings of the State-of-the-art NN equalizers used for comparison. Similar to the original publication [18], for DetNet η denotes the learning rate, L the number of layers, d_{h} the number of hidden neurons in the single-hidden-layer FCNN, d_{v} the dimension of the auxiliary variable passing unconstrained information thourgh the network, and β the residual weighting factor. For OAMP-Net2 [19], η is the learning rate, and T is the number of layers. For KAFCNN [31], η is the learning rate, L the number of layers, d_{h} the number of neurons per hidden layer, and β the residual weighting factor.

SC-FDE parameters	DetNet					OAMP-Net2		KAFCNN			
	η	L	d_{h}	d_{v}	β	η	T	η	L	n_{h}	β
UW guard, QPSK	$6 \cdot 10^{-4}$	15	200	20	0.5	$1 \cdot 10^{-3}$	8	$1 \cdot 10^{-3}$	12	250	0.8
CP guard, QPSK	$6 \cdot 10^{-4}$	15	250	30	0.5	$1 \cdot 10^{-3}$	10	$1 \cdot 10^{-3}$	12	300	0.8
UW guard, 16-QAM	$6 \cdot 10^{-4}$	15	220	25	0.5	$1 \cdot 10^{-3}$	8	$1 \cdot 10^{-3}$	12	280	0.8

our simulation result validates the proof of the equivalence of the bit error probabilities of the LMMSE hard decision estimates and the estimates of the iterative SIC method after one iteration. Moreover, the BER performance of the iterative SIC method considerably improves when conducting a second iteration ($Q = 2$), outperforming the DFE over a wide E_{b}/N_0 range. For $Q \geq 3$, two interesting effects are visible. Firstly, the BER performance flattens out at higher E_{b}/N_0 values. Secondly, although more iterations lead to an improvement of the BER performance at lower E_{b}/N_0 values (which is, however, rather small), at higher E_{b}/N_0 values the performance even slightly degrades the more iterations are conducted, which can be explained by the error caused by approximating the covariance matrix $\mathbf{C}_{\mathbf{vv},k}^{(q)}$. For SICNNv1, more stages than iterations of the model-based method are required to obtain good BER performance, however, for $Q = 7$ stages, the iterative SIC method is considerably outperformed by SICNNv1.

Next, we compare the proposed NN-based data estimators SICNNv1, SICNNv2, SICNNv1Red and SICNNv2Red with the aforementioned state-of-the-art model-based and NN-based equalizers in terms of achieved BER performance. Their training and hyperparameter optimization is conducted with the same training set as used for the proposed NN-based equalizers. As shown in Fig. 4, SICNNv1 is the best performing equalizer for a wide E_{b}/N_0 range, followed by SICNNv2 and OAMP-Net2. The parameter-reduced variant SICNNv1Red exhibits approximately the same performance as DetNet. All of the aforementioned NN-based equalizers can outperform or perform similarly as the model-based equalizers considered for comparison. SICNNv2Red, in turn,
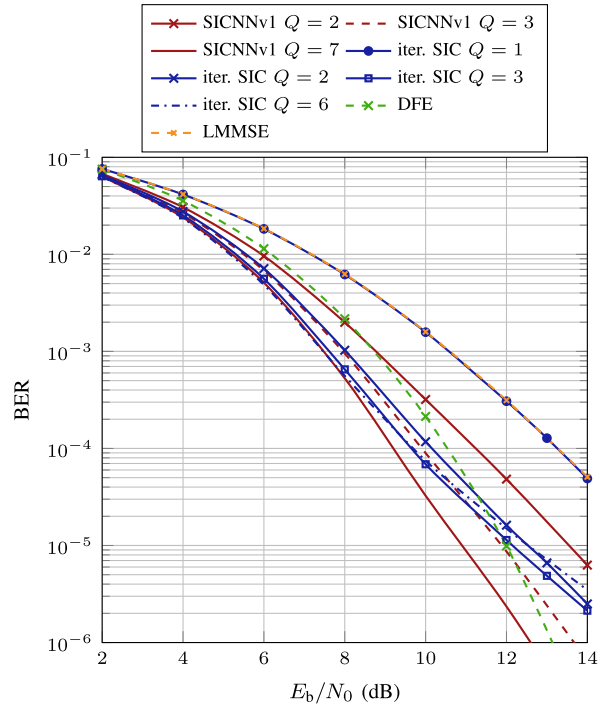

FIGURE 3. BER performance of the iterative SIC method and SICNNv1 for different numbers of iterations / stages Q (SC-FDE with UW guard, QPSK).

is the worst performing among the proposed NN-based equalizers, but still has a far better BER performance than KAFCNN. From this simulation result we can conclude that using the same sub-NNs in all stages of the proposed
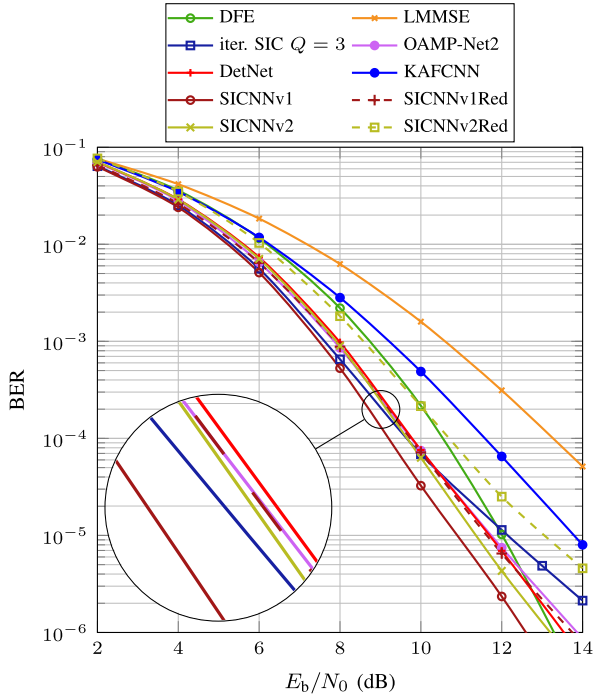
FIGURE 4. BER performance of NN-based and model-based equalizers for SC-FDE with a UW guard interval and QPSK alphabet.

FIGURE 5. BER performance comparison of NN-based equalizers for SC-FDE with a UW guard interval and QPSK alphabet, when being trained with randomly generated training data (rand. tr. data), or on a training set generated by the proposed approach.

NN-based equalizers leads to a reduction of the number of learnable parameters at the cost of a performance decrease. However, since fewer parameters have to be optimized, the reduction of learnable parameters decreases the computational effort for training the NNs.

For an SC-FDE system with a UW guard interval and QPSK modulation alphabet, we also show the influence of our proposed approach for training set generation, described in Sec. IV-A, on the BER performance of trained NN-based equalizers. Exemplary for SICNNv1, SICNNv2, and DetNet, we compare their performance when being trained on a dataset generated with our approach with the case that they are trained with randomly generated training data. For the randomly generated training data, the E_b/N_0 values of the sample transmissions contained in the training set are chosen randomly (with uniform distribution on the linear E_b/N_0 scale) in the range [3 dB, 14 dB], which is a state-of-the-art approach for the training of NN-based equalizers. As shown in Fig. 5, particularly at high E_b/N_0 values, the performance of the aforementioned NN-based equalizers can be significantly improved by training them on a training set generated by our proposed method. As elucidated in Sec. IV-A, we assume that the main reason for this performance improvement is, that at high SNRs distinctly more training samples lying close to the decision boundaries of the optimal equalizer are available, allowing the NNs to approximate the optimal decision boundaries. For all following results, the NNs are trained with datasets that are generated with our proposed method.
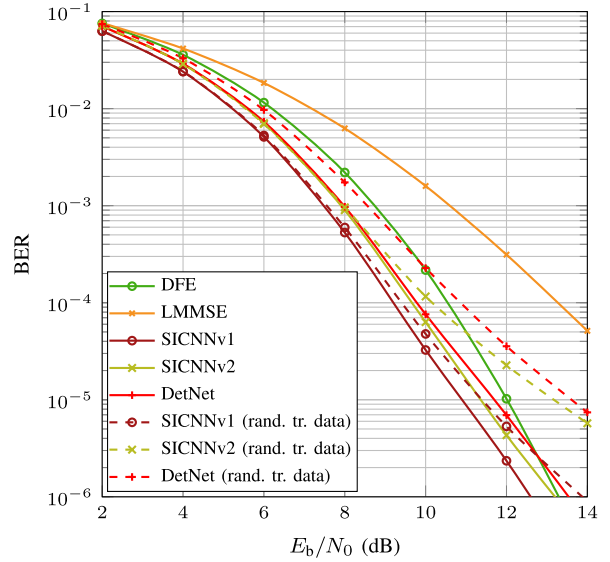
2) UNIQUE WORD GUARD INTERVAL, 16-QAM

To show that the proposed NN-based equalizers can also cope with higher order modulation alphabets, we present BER performance results for an SC-FDE system with a UW guard interval and 16-QAM modulation alphabet. As shown in Fig. 6, SICNNv1 is also the best performing among all considered equalizers for this system setup. Similar as for a QPSK modulation alphabet, the best performing equalizer behind SICNNv1 are SICNNv2 and OAMP-Net, outperforming the model-based DFE in lower E_b/N_0 regions and performing similarly in higher E_b/N_0 regions. SICNNv1Red, SICNNv2Red, DetNet, and KAFCNN, in turn, exhibit a significantly worse performance, where KAFCNN is the worst performing among all considered NN-based equalizers.

3) IMPERFECT CHANNEL KNOWLEDGE

We investigate the influence of imperfect channel knowledge on the performance of NN-based and model-based equalizers. To this end, the channel, which is assumed to be stationary for one transmitted data burst, is estimated as described in [50] using a known preamble. This preamble is transmitted prior to the data burst and contains two identical pilot vectors $\mathbf{x}_p$. Based on the two corresponding received pilot vectors $\mathbf{y}_p$, the channel frequency response is estimated with the best linear unbiased estimator (BLUE) [40]. For further details on the estimation of the channel frequency response, we refer to [50].

For this evaluation, the NN-based equalizers are trained in the same manner as for perfect channel knowledge, however, as an input a channel matrix is employed which is computed using the estimated channel frequency response. As shown in Fig. 7 for SICNNv1, SICNNv2, and OAMP-Net2,
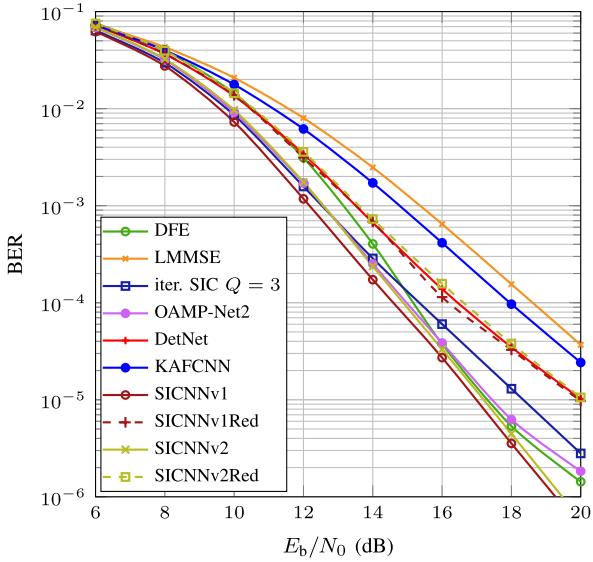
FIGURE 6. BER performance of NN-based and model-based equalizers for SC-FDE with a UW guard interval and 16-QAM alphabet.
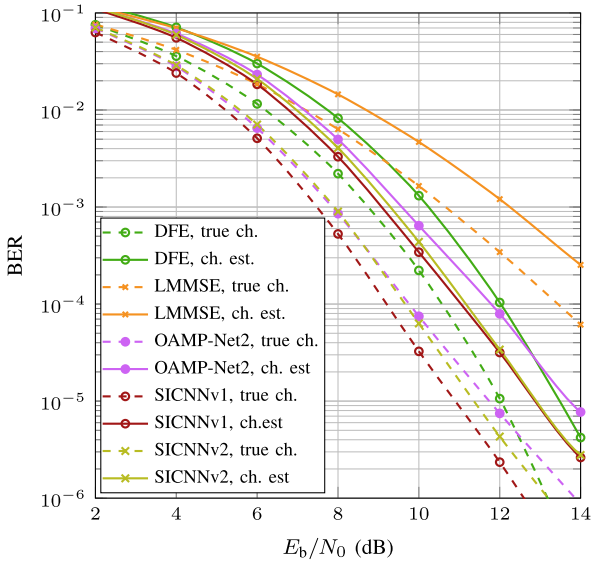

FIGURE 7. BER performance of NN-based and model-based equalizers for SC-FDE with a UW guard interval, QPSK alphabet, and with perfect and imperfect channel knowledge.

an imperfect channel knowledge has a similar influence on the BER performance as for model-based equalizers, demonstrating their robustness in terms of imperfect channel knowledge at the receiver.

4) CYCLIC PREFIX GUARD INTERVAL, QPSK

For all the previously shown results we have employed a UW as a guard interval. In this section, we investigate the influence of using a CP as a guard interval on the performance of the regarded NN-based and model-based equalizers. As presented in Fig. 8, SICNNv1 is also the best performing equalizer for this system setup, where its performance is very
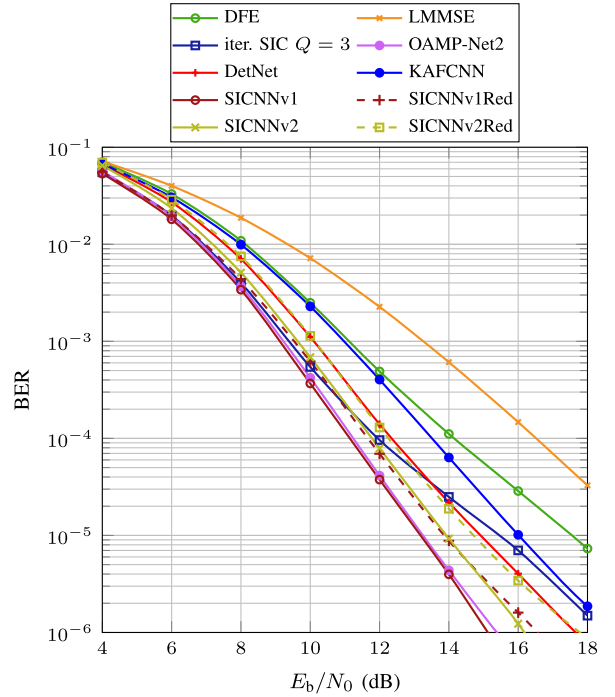

FIGURE 8. BER performance of NN-based and model-based equalizers for SC-FDE with a CP guard interval and QPSK alphabet.

similar to that of OAMP-Net2. SICNNv2 and SICNNv1Red exhibit a very similar BER performance and can clearly outperform DetNet, which achieves similar BER results as SICNNv2Red. The worst performing NN-based equalizer is KAFCNN, still outperforming the model-based DFE. The LMMSE equalizer performs worst, however, as mentioned in Sec. II-A, stands out due to its very low complexity for SC-FDE communications with CP guard intervals.

C. INFLUENCE OF A REDUCED TRAINING SET SIZE

In this section, we investigate the influence of a limited training set size on the BER performance of selected NN-based data estimators. That is, while for the BER performance results shown in Sec. V-B the NN-based equalizers are trained utilizing sample data transmissions over 30 000 different multipath channels, the regarded NNs are trained with training sets consisting of 10 000 or 3 000 channels. We evaluate the influence of a reduced training set size on the BER performance of selected NN-based equalizers for an SC-FDE system with UW-guard interval and QPSK modulation alphabet. The hyperparameters of the NN-based equalizers are the same as stated in Sec. V-A, apart from the number of training epochs which is adapted appropriately such that the number of update steps of the learnable NN parameters remains the same for all training set sizes.

We regard SICNNv1, its parameter-reduced variant SICNNv1Red, the OAMP-Net2, and the KAFCNN for performance comparison, whereby these NNs have for the chosen hyperparameter settings 135 590, 19 370, 32, and
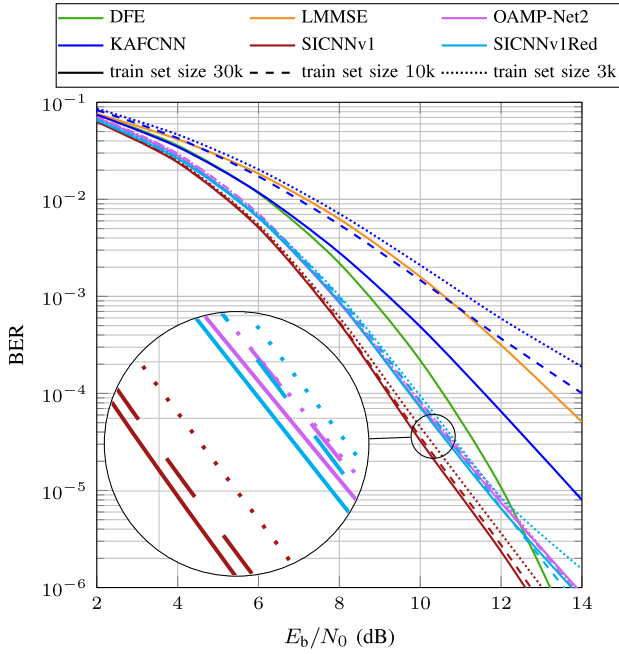
FIGURE 9. BER performance comparison for different training set sizes (SC-FDE with UW guard, QPSK alphabet).

746 628 learnable parameters, respectively. As shown in Fig. 9, the performance of SICNNv1 and SICNNv1red slightly degrades in case of a reduced training set size of 10 000 or 3 000 channels. The BER performance of OAMP-Net2 barely changes when reducing the training set size, while the performance of KAFCNN decreases most and is even worse than that of the LMMSE estimator when being trained with 3 000 different channels. That is, the fewer learnable parameters an NN contains, the less it suffers from a limited training set size. This result emphasizes the importance of parameter reduction, e.g., by incorporating model knowledge into the layer architecture of an NN.

D. BIT ERROR RATIO PERFORMANCE OF SICNNV2 FOR UW-OFDM

As mentioned in Sec. III-B, the layer architecture of SIC-NNv2 is inferred by deep unfolding iterative SIC, but no properties of any specific communication system are exploited. Hence, we expect SICNNv2 to be universally applicable for any communication system with system models similar to the system model (5) of SC-FDE systems. To demonstrate this claim, we apply SICNNv2 as an equalizer in a UW-OFDM system [51], [52], [53]. The data transmission in UW-OFDM systems can be modeled as [51], [52], and [39]

$$\mathbf{y} = \underbrace{\widetilde{\mathbf{H}}\mathbf{G}}_{\mathbf{H}}\,\mathbf{d} + \mathbf{w}\,, \qquad (30)$$

where $\mathbf{d} \in \mathbb{S}^{N_d}$ is the transmitted data vector of length N_d to be estimated, $\mathbf{y} \in \mathbb{C}^{N_d+N_u}$ the received vector at the input of the equalizer, and N_u the length of the UW

guard interval. Further, $\widetilde{\mathbf{H}} \in \mathbb{C}^{(N_d+N_u)\times(N_d+N_u)}$ denotes a diagonal matrix containing the sampled channel frequency response of the channel (excluding at positions of OFDM zero-subcarriers) on the main diagonal, $\mathbf{G} \in \mathbb{C}^{(N_d+N_u)\times N_d}$ the so-called generator matrix, which is a full, rectangular matrix, and $\mathbf{w} \sim \mathcal{NC}(\mathbf{0}, (N_d + N_u)\sigma_n^2\mathbf{I})$, where σ_n^2 is the variance of AWGN in time domain. For further details on UW-OFDM we refer to [51] and [52]. That is, the models of UW-OFDM and SC-FDE systems are very similar, allowing to employ SICNNv2 unaltered for UW-OFDM, apart from the used data normalization, which is described in [39]. We train (exactly in the same manner as all other state-of-the-art NNs used for comparison) and evaluate SICNNv2 for a UW-OFDM system referred to as system I in [39], where $N_d = 8, N_u = 4$, and the modulation alphabet is QPSK. The best hyperparameter combination found is a learning rate $\eta = 5 \cdot 10^{-4}$, $Q = 6$ stages, $n_L = 2$ hidden layers of the sub-NNs, and $n_H = 200$ neurons per hidden layer of the sub-NNs. SICNNv2 is compared to the state-of-the-art NN-based equalizers OAMP-Net2 [19], RE-MIMO [17], DetNet [18], and an improved version of DetNet, that employs a preconditioner in its layers [39]. Due to the small dimension of system I, even the optimal BER performance can be computed by applying the bit-wise MAP estimator. For all further details on the simulation setup, on the NNs used for comparison, or their training, we refer to [39]. As shown in Fig. 10, SICNNv2 can outperform the NN-based equalizers OAMP-Net2, DetNet in its original form [18], and performs similar as RE-MIMO. Further, its performance is very close to that of the improved version of DetNet and to the optimal BER performance achieved by the bit-wise MAP estimator. This result demonstrates the applicability of the proposed SICNN idea for different communication systems.

E. COMPUTATIONAL COMPLEXITY

Besides the BER performance of the regarded equalizers, also their computational complexity is an important aspect. We compare the inference complexity of the model-based and the NN-based equalizers regarded in this work in terms of the number of real-valued multiplications required for the equalization of a received vector, where four real-valued multiplications and two real-valued multiplications are accounted for a product of two complex values and a real and a complex value, respectively, and divisions are counted as multiplications. Since NN training can be carried out offline, we do not regard their training complexity. We assume that both $\widetilde{\mathbf{H}}$ and $\mathbf{H} = \widetilde{\mathbf{H}}\mathbf{M}$ are already available and thus the complexity of computing $\mathbf{H}$ is not considered for the following complexity analysis. Unless stated otherwise, we derive the computational complexities for a general matrix $\mathbf{M}$ and a length N' of the received vector, where, as described in Sec. II-A.3, both have to be replaced by the appropriate quantities $\mathbf{M}_{uw}$ and N, or $\mathbf{F}_{N_d}$ and N_d, when using a UW or a CP as a guard interval, respectively.

We start by investigating the complexity of the proposed SICNNv1. Let us first consider the operations conducted
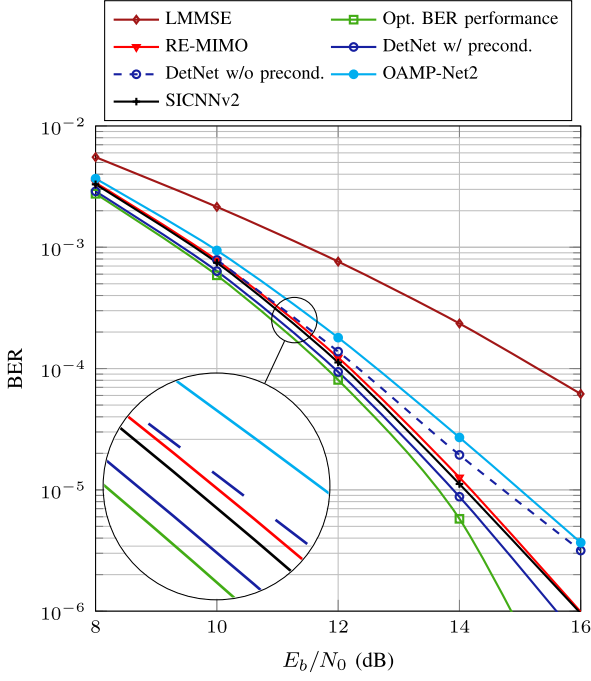
FIGURE 10. BER performance comparison for UW-OFDM (system I from [39]).

in a single stage q for estimating the data symbol d_k. The computation of $\hat{d}_{k,\text{Re}}^{(q)}$, $\hat{d}_{k,\text{Im}}^{(q)}$, $e_k^{(q)}$, $\mathbf{y}_{\text{ic},k}^{(q)}$, $\mathbf{a}_k^{(q)}$, takes

$$\underbrace{|\mathbb{S}'|}_{\hat{d}_{k,\text{Re}}^{(q)}} + \underbrace{|\mathbb{S}'|}_{\hat{d}_{k,\text{Im}}^{(q)}} + \underbrace{4|\mathbb{S}'| + 2}_{e_k^{(q)}} + \underbrace{4N'(N_\text{d}-1)}_{\mathbf{y}_{\text{ic},k}^{(q)}} + \underbrace{2N'(N_\text{d}-1)}_{\mathbf{a}_k^{(q)}}$$

real-valued multiplications, and the inference of FCNN 1 has a complexity of

$$\underbrace{2(3N'+1)}_{\text{batch norm.}} + (3N'+1)n_{\text{H,C}} + (n_{\text{L,C}}-1)n_{\text{H,C}}^2 + n_{\text{H,C}}N'.$$

Squaring the outputs of FCNN 1 takes another N' multiplications. For the terms $\mathbf{h}_k^H \hat{\mathbf{C}}_{\mathbf{vv},k}^{(q)^{-1}} \mathbf{y}_{\text{ic},k}^{(q)}$ and $\mathbf{h}_k^H \hat{\mathbf{C}}_{\mathbf{vv},k}^{(q)^{-1}} \mathbf{h}_k$, $\mathbf{h}_k^H \hat{\mathbf{C}}_{\mathbf{vv},k}^{(q)^{-1}}$ is computed first and multiplied by $\mathbf{y}_{\text{ic},k}^{(q)}$ and $\mathbf{h}_k$ subsequently, leading for these two terms in total to a complexity of

$$\underbrace{2N'}_{\mathbf{h}_k^H \hat{\mathbf{C}}_{\mathbf{vv},k}^{(q)^{-1}}} + \underbrace{4N'}_{\cdot \mathbf{y}_{\text{ic},k}^{(q)}} + \underbrace{4N'}_{\cdot \mathbf{h}_k}.$$

The inference of FCNN 2 and the normalization of $\mathbf{y}_{\text{ic},k}^{(q)}$ and $\mathbf{h}_k$ require another

$$6 + 3n_{\text{H,pr}} + (n_{\text{L,pr}}-1)n_{\text{H,pr}}^2 + 2n_{\text{H,pr}}|\mathbb{S}'|$$

and $8N'+1$ multiplications, respectively. Consequently, for estimating a single data symbol in a stage, in total

$$\begin{aligned} M_{\text{SICNNv1},kq} &= n_{\text{H,pr}}^2(n_{\text{L,pr}}-1) + n_{\text{H,pr}}(2|\mathbb{S}'|+3) \\ &\quad + n_{\text{H,C}}^2(n_{\text{L,C}}-1) + n_{\text{H,C}}(4N'+1) \\ &\quad + 19N' + 6|\mathbb{S}'| + 6N'N_\text{d} + 11 \end{aligned} \quad (31)$$

For data normalization, another

$$\underbrace{4N_\text{d}N'+1}_{\kappa} + \underbrace{N'}_{\kappa\tilde{\mathbf{H}}} + \underbrace{2N'}_{\mathbf{K}\mathbf{y}} + \underbrace{N'}_{\mathbf{K}\tilde{\mathbf{H}}}$$

real-valued multiplications are required, leading to a total complexity of SICNNv1

$$M_{\text{SICNNv1}} = QN_\text{d}M_{\text{SICNNv1},kq} + 4N_\text{d}N' + 4N' + 1, \quad (32)$$

with $M_{\text{SICNNv1},kq}$ as specified in (31).

For SICNNv2, the approach for deriving its complexity is similar. As already stated for SICNNv1, the total complexity for computing $\hat{d}_{k,\text{Re}}^{(q)}$, $\hat{d}_{k,\text{Im}}^{(q)}$, $e_{k,\text{Re}}^{(q)}$, $e_{k,\text{Im}}^{(q)}$, and $\mathbf{y}_{\text{ic},k}^{(q)}$ is given by $6|\mathbb{S}'| + 4N'(N_\text{d}-1)$. For computing the scaling factor $\rho_k^{(q)}$, scaling the quantities contained in $\mathbf{z}_k^{(q)}$ (and squaring $\rho_k^{(q)}$), and the inference of the FCNN, another

$$\underbrace{4N'}_{\rho_k^{(q)}} + \underbrace{4N'+4}_{\text{scaling}} + \left(1 + \left\lfloor \frac{n_\text{L}}{3} \right\rfloor\right) \underbrace{(6N'+2)}_{\text{batch norm.}} + (4N'+3)n_\text{H}$$
$$+ (n_\text{L}-1)n_\text{H}^2 + 2n_\text{H}|\mathbb{S}'|$$

multiplications are required. Consequently, the complexity of SICNNv2 follows to

$$M_{\text{SICNNv2}} = QN_\text{d}M_{\text{SICNNv2},kq} + 4N_\text{d}N' + 4N' + 1, \quad (33)$$

with

$$\begin{aligned} M_{\text{SICNNv2},kq} &= \left\lfloor \frac{n_\text{L}}{3} \right\rfloor (6N'+2) + n_\text{H}^2(n_\text{L}-1) + 4N'N_\text{d} \\ &\quad + 10N' + n_\text{H}(4N'+2|\mathbb{S}'|+3) + 6|\mathbb{S}'| + 6. \end{aligned} \quad (34)$$

The complexity of DetNet can be derived similarly as for UW-OFDM, which has been conducted in [39]. For all operations conducted for SC-FDE specific pre-processing and for the inference of DetNet, we refer to [31]. Here, we only state the final result for the inference complexity of DetNet. In the qth DetNet layer inference of a single hidden layer FCNN, one-hot demapping of the data vector estimate in the qth layer, and applying weighted residual connections are conducted, which entails

$$\begin{aligned} M_{\text{DetNet},q} &= 4N_\text{d}^2 + 6N_\text{d} + 2d_\text{h}(N_\text{d}(|\mathbb{S}'|+1) + d_\text{v}) \\ &\quad + 2N_\text{d}|\mathbb{S}'| + d_\text{v} \end{aligned} \quad (35)$$

real-valued multiplications, where d_h is the number of neurons in the hidden layer of the FCNN, and d_v is the dimension of an auxiliary variable passing unconstrained information from DetNet layer to DetNet layer [18], [31], [39]. In total, the inference complexity of DetNet is

$$M_{\text{DetNet}} = LM_{\text{DetNet},q} - 2N_\text{d}|\mathbb{S}'| + \underbrace{4N_\text{d}N' + 4N' + 1}_{\text{data normalization}}$$
$$+ \underbrace{6N_\text{d}N' + 4N_\text{d}^2N' + 2N'}_{\text{input pre-processing}}, \quad (36)$$

where the subtracted term accounts for no one-hot demapping in the last DetNet layer, L is the number of DetNet layers,

and for the input pre-processing the quantities $\mathbf{M}^H \widetilde{\mathbf{H}}^{1/2} \mathbf{M}$ and $\mathbf{M}^H \widetilde{\mathbf{H}}^{-1/2} \mathbf{y}$ are computed (cf. [31]).

The KAFCNN [31] with weighted residual connections and a multiplication by a partial inverse DFT matrix in the last layer requires

$$M_{\text{KAFCNN}} = \big(3N' + (n_{\text{L}} - 1)n_{\text{H}} + 2N'|\mathbb{S}'| + (n_{\text{L}} - 2)\big)n_{\text{H}}$$
$$+ \underbrace{4N'N_{\text{d}}|\mathbb{S}'| + 2N_{\text{d}}|\mathbb{S}'|}_{\text{part. IDFT + symb. scaling}} + \underbrace{4N_{\text{d}}N' + 4N' + 1}_{\text{data normalization}}$$

(37)

The OAMP-Net2 layer structure [19] also stems from unfolding an iterative model-based algorithm, such that determining its inference complexity is conducted in a similar fashion as for DetNet or SICNNv1/SICNNv2. For the matrix inverse that has to be computed in every layer of OAMP-Net2, we assume that a Cholesky decomposition [54] is employed for accomplishing this task. The computational complexity of OAMP-Net2 can thus be specified as (using the notation from [19])

$$M_{\text{ONet2}}$$
$$= T\Big(\underbrace{8N_{\text{d}}N' + 2N_{\text{d}}}_{\mathbf{r}_t} + \underbrace{4N_{\text{d}}^2(2N' + 1) + 8N_{\text{d}}N' + 5}_{\tau_t^2}$$
$$+ \underbrace{\frac{14}{3}N'^3 + 8N'^2(2N_{\text{d}} + 1) + 8N_{\text{d}}(N' + |\mathbb{S}'| + \frac{1}{2}) + 2}_{\hat{\mathbf{x}}_{\text{d},t+1}}$$
$$+ \underbrace{2N_{\text{d}}(2N' + 1) + 1}_{v_t^2} \Big) + \underbrace{4N_{\text{d}}N' + 4N' + 1}_{\text{data normalization}}$$

(38)

Let us now consider the complexity of the model-based equalizers. For the LMMSE estimator, one has to distinguish between UW and CP guard intervals. We start by regarding its complexity in case of a UW guard interval. Here, we first regard its complexity for determining the LMMSE estimator matrix $\mathbf{E}_{\text{LMMSE}}$, which is independent of the received vector $\mathbf{y}$ and thus has to be computed only once per data burst (the channel is assumed to be stationary for the whole data burst). By assuming that the inverse in (6) is computed utilizing a Cholesky decomposition [54],

$$M_{\text{LMMSE,burst}} = \underbrace{4N_{\text{d}}^2(N_{\text{d}} + N_{\text{g}})}_{\mathbf{M}^H \mathbf{H}} + \underbrace{\frac{14}{3}N_{\text{d}}^3 + 4N_{\text{d}}^2}_{\text{inverse (Cholesky)}}$$
$$+ \underbrace{4N_{\text{d}}^2(N_{\text{d}} + N_{\text{g}})}_{\cdot \mathbf{M}^H}$$
$$= \frac{38}{3}N_{\text{d}}^3 + 8N_{\text{d}}^2 N_{\text{g}} + 4N_{\text{d}}^2$$

(39)

real-valued multiplications are to be carried out for computing $\mathbf{E}_{\text{LMMSE}}$. Given $\mathbf{E}_{\text{LMMSE}}$, the complexity of equalizing one received vector is

$$M_{\text{LMMSE,eq}} = 4N_{\text{d}}(N_{\text{d}} + N_{\text{g}}).$$

(40)

In case of a CP guard interval, equalization becomes for the LMMSE estimator far less complex. As given in (7), the matrix for which an inverse has to be computed is a diagonal matrix. Hence, obtaining the estimator matrix $\mathbf{E}_{\text{LMMSE,dg}}$ requires $4N_{\text{d}}$ real-valued multiplications, which is already the number of multiplications that have to be carried out per burst

$$M_{\text{LMMSE,burst}} = 4N_{\text{d}}.$$

(41)

For equalization of a single received data vector, a multiplication with the diagonal estimator matrix $\mathbf{E}_{\text{LMMSE,dg}}$ is required, followed by conducting an inverse DFT, leading to a complexity of

$$M_{\text{LMMSE,eq}} = 4N_{\text{d}} + \underbrace{2N_{\text{d}}\log_2(N_{\text{d}})}_{\text{IDFT}},$$

(42)

real-valued multiplications.

For the DFE, using a CP guard interval does not reduce the complexity as for the LMMSE, since in every iteration the LMMSE error variances (which are the diagonal elements of the LMMSE error covariance matrix) have to be computed. We distinguish between operations to be carried out only once every data burst and those to be accomplished for every received vector. For a derivation of the complexity of the DFE we refer to [39], where an in-depth complexity analysis of the DFE for a similar system, namely a UW-OFDM system is conducted. Here, we only state the final results. The number of real-valued multiplications to be conducted once per data burst is

$$M_{\text{DFE,burst}} = \frac{7}{6}N_{\text{d}}^4 + \frac{11}{3}N_{\text{d}}^3 + \frac{19}{6}N_{\text{d}}^2 + 6N_{\text{d}}^2 N'$$
$$+ \frac{2}{3}N_{\text{d}} + 2N_{\text{d}}N' - \frac{14}{3},$$

(43)

while the inference complexity per received vector is given by

$$M_{\text{DFE,eq}} = 8N_{\text{d}}N'.$$

(44)

For the iterative SIC method, the same steps have to be conducted Q times, where – as for the LMMSE estimator – a Cholesky decomposition is utilized for inverting the (approximated) covariance matrices $\mathbf{C}_{\mathbf{vv},k}^{(q)}$ in every iteration. We assume that the estimated data symbols $\hat{d}_k^{(q-1)}$ are multiplied by the corresponding column of $\mathbf{H}$ only once per iteration, being available for interference cancellation required for estimating any data symbol d_k. Hence, the computational complexity of the iterative SIC method with Q iterations is

$$M_{\text{itSIC}} = QN_{\text{d}}\Big(\underbrace{6|\mathbb{S}'| + 2}_{\hat{d}_k^{(q)}, e_k^{(q)}} + \underbrace{(N_{\text{d}} - 1)(2N' + 4N'^2) + 4N'}_{\mathbf{C}_{\mathbf{vv},k}^{(q)}}$$
$$+ \underbrace{\frac{14}{3}N'^3 + 4N'^2}_{\text{Cholesky}} + \underbrace{4N'^2 + 8N' + 4}_{f_k^{(q)}(.)} + \underbrace{4N'}_{\mathbf{h}_k \hat{d}_k^{(q-1)}} \Big)$$
$$= QN_{\text{d}}\Big(\frac{14}{3}N'^3 + 4N'^2 N_{\text{d}} + 4N'^2 + 2N'N_{\text{d}}$$

TABLE 4. Number of required real-valued multiplications (Rounded to Hundreds) of evaluated equalizers for different SC-FDE system setups.

Equalizer		Nr. multiplications for different setups		
		UW guard, QPSK	UW guard, 16-QAM	CP guard, QPSK
SICNNv1 / SICNNv1Red	M_{SICNNv1}	3 288 600	3 409 300	8 929 500
SICNNv2 / SICNNv2Red	M_{SICNNv2}	6 836 900	31 552 200	50 600 900
DetNet	M_{DetNet}	565 100	911 400	1 153 200
KAFCNN	M_{KAFCNN}	753 900	976 900	1 072 700
OAMP-Net2	M_{ONet2}	4 892 300	4 894 800	9 815 100
LMMSE	$M_{\text{LMMSE,burst}}$	141 300	141 300	100
	$M_{\text{LMMSE,eq}}$	2 600	2 600	400
DFE	$M_{\text{DFE,burst}}$	295 400	295 400	1 545 400
	$M_{\text{DFE,eq}}$	5 100	5 100	8 200
Iterative SIC	M_{itSIC}	14 440 800	14 441 500	27 897 500

$$+ 14N' + 6|\mathbb{S}'| + 6\Big) \tag{45}$$

The numerical results of the complexities of the equalizers for the SC-FDE system setup specified in Sec. V-A are given in Tab. 4. For all system setups considered, DetNet and KAFCNN are the lowest complex NN-based equalizers. The inference complexity of SICNNv1 is distinctly lower than that of SICNNv2 and OAMP-Net2, and also than the model-based iterative SIC method it is deduced from. However, the LMMSE estimator and the DFE exhibit by far the lowest complexity.

VI. CONCLUSION

In this work, we proposed novel NN-based equalizers, called SICNNv1 and SICNNv2, inspired by a model-based soft interference cancellation scheme. SICNNv1 is tailored for an SC-FDE communication system, while SICNNv2 is also applicable for other communication systems with block-based data transmission. In addition, we presented a novel approach for generating training sets for NN-based equalizers, which considerably helps to improve their performance at high SNRs. We evaluated the proposed NN-based equalizers for a number of different SC-FDE system setups, and investigated their robustness with respect to imperfect channel knowledge at the receiver. In particular SICNNv1 exhibits a superior BER performance over all regarded state-of-the-art model-based and NN-based equalization approaches for all SC-FDE system setups considered. To highlight the universal applicability of SICNNv2, we exemplarily presented its state-of-the-art performance for a UW-OFDM system. Further, we investigated the influence of the size of the dataset used to train the NN-based equalizers, and we presented an in-depth complexity analysis.

APPENDIX I. NOISE STATISTICS IN A SOFT INTERFERENCE CANCELLATION STEP

Let us consider the system model (11) for estimating the kth data symbol d_k, $k \in \{0, \ldots, N_{\text{d}} - 1\}$, for any but the first iteration ($q = 0$), i.e., $0 < q < Q$,

$$\mathbf{y}_{\text{ic},k}^{(q)} = \mathbf{y} - \bar{\mathbf{H}}_k \hat{\bar{\mathbf{d}}}_k^{(q-1)} = \mathbf{h}_k d_k \underbrace{- \overbrace{\bar{\mathbf{H}}_k \bar{\boldsymbol{\delta}}_k^{(q-1)}}^{\mathbf{r}_k^{(q)}} + \mathbf{w}}_{\mathbf{v}_k^{(q)}}, \tag{46}$$

which we repeat here for readability. Following central limit theorem arguments, the total noise $\mathbf{v}_k^{(q)}$ is assumed to feature a multivariate Gaussian distribution, i.e., $p(\mathbf{v}_k^{(q)}) = \mathcal{CN}(\boldsymbol{\mu}_{\mathbf{v},k}^{(q)}, \mathbf{C}_{\mathbf{vv},k}^{(q)})$, where $\boldsymbol{\mu}_{\mathbf{v},k}^{(q)}$ and $\mathbf{C}_{\mathbf{vv},k}^{(q)}$ are to be specified. We start by computing the statistics of $\bar{\boldsymbol{\delta}}_k^{(q-1)}$, followed by those of $\mathbf{r}_k^{(q)}$ to obtain finally the distribution of $\mathbf{v}_k^{(q)}$. It is important to note, that for computing noise statistics in iteration q, the estimation errors from the previous iteration $(q-1)$ have to be specified, whereby the interference canceled vectors $\mathbf{y}_{\text{ic},k}^{(q-1)}$ are fixed and available, i.e., the PDFs/PMFs of $\mathbf{v}_k^{(q)}$, $\mathbf{r}_k^{(q)}$, and $\bar{\boldsymbol{\delta}}_k^{(q-1)}$ are not unconditional, but are conditioned on a given $\mathbf{y}_{\text{ic},k}^{(q-1)}$.

A data symbol estimation error $\delta_k^{(q-1)} = \hat{d}_k^{(q-1)} - d_k$, where $\mathbf{y}_{\text{ic},k}^{(q-1)}$ is given, can only attain a finite number of different values (as many as the cardinality $|\mathbb{S}|$ of the symbol alphabet), and thus its statistics is described by a PMF $p_{d_k|\mathbf{y}_{\text{ic},k}^{(q)}}\big[\delta_k^{(q-1)}|\mathbf{y}_{\text{ic},k}^{(q-1)}\big]$. In order to compute the statistics of an estimation error $\delta_k^{(q-1)} = \hat{d}_k^{(q-1)} - d_k$, we start by reconsidering the MMSE estimate of the preceding iteration

$$\hat{d}_k^{(q-1)} = E_{d_k|\mathbf{y}_{\text{ic},k}^{(q-1)}}\big[d_k|\mathbf{y}_{\text{ic},k}^{(q-1)}\big]$$
$$= \sum_{s' \in \mathbb{S}} s' p\big[d_k = s'|\mathbf{y}_{\text{ic},k}^{(q-1)}\big]. \tag{47}$$

Since

$$\mathbf{y}_{\text{ic},k}^{(q-1)} = \mathbf{y} - \bar{\mathbf{H}}_k \hat{\bar{\mathbf{d}}}_k^{(q-2)}, \tag{48}$$

we can reformulate $p\big[d_k|\mathbf{y}_{\text{ic},k}^{(q-1)}\big]$ as $p\big[d_k|\mathbf{y}_{\text{ic},k}^{(q-1)}\big] = p\big[d_k|\mathbf{y}, \hat{\bar{\mathbf{d}}}_k^{(q-2)}\big]$, where we consider $\hat{\bar{\mathbf{d}}}_k^{(q-2)}$ to be a fixed vector

in iteration $(q-1)$, which does not feature a statistical distribution. It can be observed from the system model for iteration step $(q-1)$

$$\mathbf{y}_{\text{ic},k}^{(q-1)} = \mathbf{y} - \bar{\mathbf{H}}_k \hat{\bar{\mathbf{d}}}_k^{(q-2)} = \mathbf{h}_k d_k - \bar{\mathbf{H}}_k \bar{\boldsymbol{\delta}}_k^{(q-2)} + \mathbf{w} \quad (49)$$

that the PMF $p[d_k | \mathbf{y}, \hat{\bar{\mathbf{d}}}_k^{(q-2)}]$ does not depend on the estimate $\hat{d}_k^{(q-2)}$ for a given $\mathbf{y}$ and $\hat{\bar{\mathbf{d}}}_k^{(q-2)}$. Hence, $\hat{d}_k^{(q-2)}$ can be included in the condition of $p[d_k | \mathbf{y}, \hat{\bar{\mathbf{d}}}_k^{(q-2)}]$ without altering the PMF, i.e., $p[d_k | \mathbf{y}, \hat{\bar{\mathbf{d}}}_k^{(q-2)}] = p[d_k | \mathbf{y}, \hat{\bar{\mathbf{d}}}_k^{(q-2)}, \hat{d}_k^{(q-2)}] = p[d_k | \mathbf{y}, \hat{\mathbf{d}}^{(q-2)}]$. Consequently, the MMSE estimate $\hat{d}_k^{(q-1)}$ and its corresponding conditional MSE can be rewritten as

$$\hat{d}_k^{(q-1)} = E_{d_k | \mathbf{y}_{\text{ic},k}^{(q-1)}} [d_k | \mathbf{y}_{\text{ic},k}^{(q-1)}]$$
$$= E_{d_k | \mathbf{y}, \hat{\mathbf{d}}^{(q-2)}} [d_k | \mathbf{y}, \hat{\mathbf{d}}^{(q-2)}] \quad (50)$$

and

$$e_k^{(q-1)} = E_{d_k | \mathbf{y}_{\text{ic},k}^{(q-1)}} [|d_k - \hat{d}_k^{(q-1)}|^2 | \mathbf{y}_{\text{ic},k}^{(q-1)}]$$
$$= E_{d_k | \mathbf{y}, \hat{\mathbf{d}}^{(q-2)}} [|d_k - \hat{d}_k^{(q-1)}|^2 | \mathbf{y}, \hat{\mathbf{d}}^{(q-2)}], \quad (51)$$

respectively. Hence, the conditional PMF of an estimation error δ_k can be written as

$$p_{d_k | \mathbf{y}_{\text{ic},k}^{(q-1)}} [\delta_k^{(q-1)} | \mathbf{y}_{\text{ic},k}^{(q-1)}] = p_{d_k | \mathbf{y}, \hat{\mathbf{d}}^{(q-2)}} [\delta_k^{(q-1)} | \mathbf{y}, \hat{\mathbf{d}}^{(q-2)}],$$

and the PMF of $\bar{\boldsymbol{\delta}}^{(q-2)}$ as $p_{\bar{\mathbf{d}}_k | \mathbf{y}, \hat{\mathbf{d}}^{(q-2)}} [\bar{\boldsymbol{\delta}}_k^{(q-1)} | \mathbf{y}, \hat{\mathbf{d}}^{(q-2)}]$.

Following central limit theorem arguments, we assume the distribution of $\mathbf{r}_k^{(q)}$ to be multivariate Gaussian. The mean of $\mathbf{r}_k^{(q)}$ follows to

$$\boldsymbol{\mu}_{\mathbf{r},k}^{(q)} = E_{\bar{\mathbf{d}}_k | \mathbf{y}, \hat{\mathbf{d}}^{(q-2)}} [\bar{\mathbf{H}}_k (\hat{\bar{\mathbf{d}}}_k^{(q-1)} - \bar{\mathbf{d}}_k) | \mathbf{y}, \hat{\mathbf{d}}^{(q-2)}]$$
$$= \bar{\mathbf{H}}_k (\hat{\bar{\mathbf{d}}}_k^{(q-1)} - \hat{\bar{\mathbf{d}}}_k^{(q-1)}) = \mathbf{0}. \quad (52)$$

Its covariance matrix, in turn, is given by

$$\mathbf{C}_{\mathbf{rr},k}^{(q)} = E_{\mathbf{r}_k^{(q)}} [\mathbf{r}_k^{(q)} \mathbf{r}_k^{(q)H}]$$
$$= \bar{\mathbf{H}}_k \underbrace{E_{\bar{\mathbf{d}}_k | \mathbf{y}, \hat{\mathbf{d}}^{(q-2)}} [\bar{\boldsymbol{\delta}}_k^{(q-1)} \bar{\boldsymbol{\delta}}_k^{(q-1)H} | \mathbf{y}, \hat{\mathbf{d}}^{(q-2)}]}_{\widetilde{\mathbf{E}}_k^{(q-1)}} \bar{\mathbf{H}}_k^H. \quad (53)$$

For computing an element in the mth row and the nth column of $\widetilde{\mathbf{E}}_k^{(q-1)}$, $m, n \in \{0, \dots, N_d - 2\}$, let us first consider to which data symbol d_i and d_j in the data vector $\mathbf{d}$, $i, j \in \{0, \dots, N_d - 1\}$, the matrix element $[\widetilde{\mathbf{E}}_k^{(q-1)}]_{mn}$ belongs. Since $\bar{\boldsymbol{\delta}}_k^{(q-1)}$ contains the deviations of the data symbol estimates to the corresponding true data symbols for all but the kth data symbols in the data vector, the index mapping follows to

$$i = \begin{cases} m & m < k \\ m + 1 & m \geq k \end{cases} \quad \text{and} \quad j = \begin{cases} n & n < k \\ n + 1 & n \geq k \end{cases}.$$

An off-diagonal element of $\widetilde{\mathbf{E}}_k^{(q-1)}$ can be written as

$$[\widetilde{\mathbf{E}}_k^{(q-1)}]_{mn} = E_{\bar{\mathbf{d}}_k | \mathbf{y}, \hat{\mathbf{d}}^{(q-2)}} [\delta_i^{(q-1)} \delta_j^{(q-1)*} | \mathbf{y}, \hat{\mathbf{d}}^{(q-2)}]$$

$$= E_{d_i, d_j | \mathbf{y}, \hat{\mathbf{d}}^{(q-2)}} [d_i d_j^* | \mathbf{y}, \hat{\mathbf{d}}^{(q-2)}]$$
$$- E_{d_i, d_j | \mathbf{y}, \hat{\mathbf{d}}^{(q-2)}} [d_i \hat{d}_j^{(q-1)*} | \mathbf{y}, \hat{\mathbf{d}}^{(q-2)}]$$
$$- E_{d_i, d_j | \mathbf{y}, \hat{\mathbf{d}}^{(q-2)}} [\hat{d}_i^{(q-1)} d_j^* | \mathbf{y}, \hat{\mathbf{d}}^{(q-2)}]$$
$$+ E_{d_i, d_j | \mathbf{y}, \hat{\mathbf{d}}^{(q-2)}} [\hat{d}_i^{(q-1)} \hat{d}_j^{(q-1)*} | \mathbf{y}, \hat{\mathbf{d}}^{(q-2)}]$$
$$= E_{d_i, d_j | \mathbf{y}, \hat{\mathbf{d}}^{(q-2)}} [d_i d_j^* | \mathbf{y}, \hat{\mathbf{d}}^{(q-2)}]$$
$$- E_{d_i, d_j | \mathbf{y}, \hat{\mathbf{d}}^{(q-2)}} [d_i | \mathbf{y}, \hat{\mathbf{d}}^{(q-2)}] \hat{d}_j^{(q-1)*}$$
$$- \hat{d}_i^{(q-1)} E_{d_i, d_j | \mathbf{y}, \hat{\mathbf{d}}^{(q-2)}} [d_j^* | \mathbf{y}, \hat{\mathbf{d}}^{(q-2)}]$$
$$+ \hat{d}_i^{(q-1)} \hat{d}_j^{(q-1)*}$$
$$= E_{d_i, d_j | \mathbf{y}, \hat{\mathbf{d}}^{(q-2)}} [d_i d_j^* | \mathbf{y}, \hat{\mathbf{d}}^{(q-2)}]$$
$$- \hat{d}_i^{(q-1)} \hat{d}_j^{(q-1)*},$$

and a diagonal element of $\widetilde{\mathbf{E}}_k^{(q-1)}$ has the value

$$[\widetilde{\mathbf{E}}_k^{(q-1)}]_{mm} = E_{\bar{\mathbf{d}}_k | \mathbf{y}, \hat{\mathbf{d}}^{(q-2)}} [\delta_i^{(q-1)} \delta_i^{(q-1)*} | \mathbf{y}, \hat{\mathbf{d}}^{(q-2)}]$$
$$= E_{d_i | \mathbf{y}, \hat{\mathbf{d}}^{(q-2)}} [\delta_i^{(q-1)} \delta_i^{(q-1)*} | \mathbf{y}, \hat{\mathbf{d}}^{(q-2)}]$$
$$= e_i^{(q-1)}.$$

With the above results at hand, $\mathbf{v}_k^{(q)}$ can be specified to be a zero mean Gaussian distributed vector with a covariance matrix

$$\mathbf{C}_{\mathbf{vv},k}^{(q)} = E_{\mathbf{v}_k^{(q)}} [\mathbf{v}_k^{(q)} \mathbf{v}_k^{(q)H}]$$
$$= E_{\mathbf{r}_k^{(q)}, \mathbf{w}} [\mathbf{r}_k^{(q)} \mathbf{r}_k^{(q)H} - \mathbf{r}_k^{(q)} \mathbf{w}^H - \mathbf{w} \mathbf{r}_k^{(q)H} + \mathbf{w} \mathbf{w}^H]$$
$$= \mathbf{C}_{\mathbf{rr},k}^{(q)} - \bar{\mathbf{H}}_k E_{(\bar{\mathbf{d}}_k | \mathbf{y}, \hat{\mathbf{d}}^{(q-2)}), \mathbf{w}} [\bar{\boldsymbol{\delta}}_k^{(q-1)} \mathbf{w}^H]$$
$$- E_{(\bar{\mathbf{d}}_k | \mathbf{y}, \hat{\mathbf{d}}^{(q-2)}), \mathbf{w}} [\mathbf{w} \bar{\boldsymbol{\delta}}_k^{(q-1)H}] \bar{\mathbf{H}}_k^H + \mathbf{C}_{\mathbf{ww}}$$
$$= \bar{\mathbf{H}}_k \widetilde{\mathbf{E}}_k^{(q-1)} \bar{\mathbf{H}}_k^H - \bar{\mathbf{H}}_k E_{(\bar{\mathbf{d}}_k | \mathbf{y}, \hat{\mathbf{d}}^{(q-2)}), \mathbf{w}} [\bar{\boldsymbol{\delta}}_k^{(q-1)} \mathbf{w}^H]$$
$$- E_{(\bar{\mathbf{d}}_k | \mathbf{y}, \hat{\mathbf{d}}^{(q-2)}), \mathbf{w}} [\mathbf{w} \bar{\boldsymbol{\delta}}_k^{(q-1)H}] \bar{\mathbf{H}}_k^H + N \sigma_n^2 \widetilde{\mathbf{H}}, \quad (54)$$

with $\widetilde{\mathbf{E}}_k^{(q-1)}$ containing

$$[\widetilde{\mathbf{E}}_k^{(q-1)}]_{mn} = \begin{cases} e_i^{(q-1)} & m = n \\ E_{d_i, d_j | \mathbf{y}, \hat{\mathbf{d}}^{(q-2)}} [d_i d_j^* | \mathbf{y}, \hat{\mathbf{d}}^{(q-2)}] & \\ \quad - \hat{d}_i^{(q-1)} \hat{d}_j^{(q-1)*} & m \neq n \end{cases}$$

in the mth row and nth column, and the index mapping as defined above.

APPENDIX II. ESTIMATES OF ITERATIVE SOFT INTERFERENCE CANCELLATION METHOD AFTER FIRST ITERATION

We show for a QPSK modulation alphabet that the bit error probability of a hard decision estimate produced by the iterative SIC method described in Sec. II-B after the first iteration is equivalent to the bit error probability of an LMMSE hard decision estimate when initializing all data symbol estimates

of the iterative SIC method with 0, i.e., $\hat{d}_k^{(-1)} = 0$, $k \in \{0, \ldots, N_d - 1\}$. To this end, we show that the decision criteria of both estimation methods for the jth bit b_{jk} of the kth data symbol, $k \in \{0, \ldots, N_d - 1\}$, $j \in \{0, \ldots, \log_2(|\mathbb{S}|) - 1\}$, being 0 or 1 are the same, and thus also their bit error probability must coincide. The QPSK bit-to-symbol mapping $(b_{1k}b_{0k}) \mapsto d_k$ is assumed to map b_{0k} to the real part and b_{1k} to the imaginary part of d_k. The bit values 0 and 1 are mapped to the symbol values $-\rho$ and ρ, respectively, with $\rho = 1/\sqrt{2}$ being an energy normalization factor.

Let us start with the LMMSE data estimator. The SC-FDE system model is given by (cf. (5))

$$\mathbf{y} = \mathbf{H}\mathbf{d} + \mathbf{w}, \tag{55}$$

where $\mathbf{w} \sim \mathcal{CN}(\mathbf{0}, N\sigma_n^2 \widetilde{\mathbf{H}})$, and the corresponding LMMSE data estimator follows to

$$\hat{\mathbf{d}} = \sigma_d^2 \mathbf{H}^H (\sigma_d^2 \mathbf{H}\mathbf{H}^H + N\sigma_n^2 \widetilde{\mathbf{H}})^{-1} \mathbf{y}, \tag{56}$$

which is expressed in a different way as in (6), but can be shown to be mathematically equivalent. Based on the LMMSE estimates $\hat{d}_k$, the hard decision estimate for bit b_{0k} is

$$\hat{b}_{0k} = \begin{cases} 1 & \Pr(b_{0k} = 1|\hat{d}_k) > \Pr(b_{0k} = 0|\hat{d}_k) \\ 0 & \text{otherwise} \end{cases}. \tag{57}$$

It can be shown [55] that for the LMMSE estimator (56) and the model (55) $\Pr(b_{0k} = 1|\hat{d}_k) = \kappa \Pr(b_{0k} = 1|\mathbf{y})$ and $\Pr(b_{0k} = 0|\hat{d}_k) = \kappa \Pr(b_{0k} = 0|\mathbf{y})$, where κ is a proportionality constant. Hence, the condition on $\hat{d}_k$ in criterion (57) can be replaced by a condition on $\mathbf{y}$, leading to a decision criterion

$$\hat{b}_{0k} = \begin{cases} 1 & \Pr(b_{0k} = 1|\mathbf{y}) > \Pr(b_{0k} = 0|\mathbf{y}) \\ 0 & \text{otherwise} \end{cases}. \tag{58}$$

By using the Bayesian rule and assuming a uniform prior PMF $p[d_k]$, we can rearrange the criterion in (58) to

$$\Pr(b_{0k} = 1|\mathbf{y}) > \Pr(b_{0k} = 0|\mathbf{y})$$

$$\sum_{s' \in \mathcal{S}_0^{(1)}} p[d_k = s'|\mathbf{y}] > \sum_{s' \in \mathcal{S}_0^{(0)}} p[d_k = s'|\mathbf{y}]$$

$$\sum_{s' \in \mathcal{S}_0^{(1)}} p(\mathbf{y}|d_k = s') > \sum_{s' \in \mathcal{S}_0^{(0)}} p(\mathbf{y}|d_k = s'), \tag{59}$$

where $\mathcal{S}_0^{(0)}$ and $\mathcal{S}_0^{(1)}$ are the sets of data symbols containing a bit with value 0 and 1 at the 0th position, respectively. For determining the PDF $p(\mathbf{y}|d_k = s')$, we reformulate (55) as

$$\mathbf{y} = \bar{\mathbf{H}}_k \bar{\mathbf{d}}_k + \mathbf{h}_k d_k + \mathbf{w}. \tag{60}$$

Due to central limit theorem arguments, $\bar{\mathbf{H}}_k \bar{\mathbf{d}}_k$ can be assumed to be Gaussian distributed, and thus $p(\mathbf{y}|d_k)$ is approximated to be a multivariate complex Gaussian PDF, i.e.,

$$p(\mathbf{y}|d_k) = \zeta_{\mathbf{y}|d_k} \exp\left(-(\mathbf{y} - \boldsymbol{\mu}_{\mathbf{y}|d_k})^H \mathbf{C}_{\mathbf{yy}|d_k}^{-1} (\mathbf{y} - \boldsymbol{\mu}_{\mathbf{y}|d_k})\right), \tag{61}$$

with the scaling factor $\zeta_{\mathbf{y}|d_k} = \frac{1}{\pi^N |\mathbf{C}_{\mathbf{yy}|d_k}|}$, the conditional mean

$$\boldsymbol{\mu}_{\mathbf{y}|d_k} = E[\mathbf{y}|d_k] = \mathbf{h}_k d_k, \tag{62}$$

and the conditional covariance matrix

$$\begin{aligned} \mathbf{C}_{\mathbf{yy}|d_k} = \mathbf{C}_k &= E[(\mathbf{y} - \boldsymbol{\mu}_{\mathbf{y}|d_k})(\mathbf{y} - \boldsymbol{\mu}_{\mathbf{y}|d_k})^H |d_k] \\ &= E[(\bar{\mathbf{H}}_k \bar{\mathbf{d}}_k + \mathbf{w})(\bar{\mathbf{H}}_k \bar{\mathbf{d}}_k + \mathbf{w})^H |d_k] \\ &= \sigma_d^2 \bar{\mathbf{H}}_k \bar{\mathbf{H}}_k^H + N\sigma_n^2 \widetilde{\mathbf{H}}. \end{aligned} \tag{63}$$

Since $\mathbf{C}_{\mathbf{yy}|d_k} = \mathbf{C}_k$ does not depend on the realization of d_k, the scaling factor $\zeta_{\mathbf{y}|d_k}$ is a constant for any symbol $s' \in \mathbb{S}$. Consequently, by inserting (61) into (59) we arrive at the LMMSE hard decision criterion

$$\hat{b}_{0k} = \begin{cases} 1 & \sum_{s' \in \mathcal{S}_0^{(1)}} g(s') > \sum_{s' \in \mathcal{S}_0^{(0)}} g(s') \\ 0 & \text{otherwise} \end{cases}, \tag{64}$$

with

$$g(s') = \exp\left(-(\mathbf{y} - \mathbf{h}_k s')^H \mathbf{C}_k^{-1} (\mathbf{y} - \mathbf{h}_k s')\right) \tag{65}$$

and $\mathbf{C}_k$ as defined in (63).

Let us now consider the iterative SIC method, starting with the system model in its first iteration ($q = 0$), which is given by

$$\mathbf{y} = \mathbf{h}_k d_k + \mathbf{v}_k^{(0)}, \tag{66}$$

where $\mathbf{v}_k^{(0)} = \bar{\mathbf{H}}_k \bar{\mathbf{d}}_k + \mathbf{w}$ is assumed to be zero mean Gaussian noise with a covariance matrix $\mathbf{C}_{\mathbf{vv},k}^{(0)} = \sigma_d^2 \bar{\mathbf{H}}_k \bar{\mathbf{H}}_k^H + N\sigma_n^2 \widetilde{\mathbf{H}}$. The MMSE estimate for data symbol d_k is the mean of the posterior PMF $E_{d_k|\mathbf{y}}[d_k|\mathbf{y}]$ for the model (66), which is given by (cf. (18))

$$\hat{d}_k = \frac{\sum_{s' \in \mathbb{S}} s' p(\mathbf{y}|d_k = s')}{\sum_{s' \in \mathbb{S}} p(\mathbf{y}|d_k = s')} \tag{67}$$

Considering the QPSK bit-to-symbol mapping defined above, the MMSE hard decision estimate for bit b_{0k} is

$$\hat{b}_{0k} = \begin{cases} 1 & \text{Re}\{\hat{d}_k\} > 0 \\ 0 & \text{otherwise} \end{cases}. \tag{68}$$

Since the denominator of the MMSE estimate given in (67) is always positive, the MMSE hard decision estimate $\hat{b}_{0k}$ is estimated to be 1 if

$$\begin{aligned} 0 &< \text{Re}\Big\{\sum_{s' \in \mathbb{S}} s' p(\mathbf{y}|d_k = s')\Big\} \\ &= \sum_{s' \in \mathbb{S}} \text{Re}\{s'\} p(\mathbf{y}|d_k = s') \\ &= \sum_{s' \in \mathcal{S}^{(+)}} \text{Re}\{s'\} p(\mathbf{y}|d_k = s') + \sum_{s' \in \mathcal{S}^{(-)}} \text{Re}\{s'\} p(\mathbf{y}|d_k = s') \\ &= \rho \sum_{s' \in \mathcal{S}^{(+)}} p(\mathbf{y}|d_k = s') - \rho \sum_{s' \in \mathcal{S}^{(-)}} p(\mathbf{y}|d_k = s'), \end{aligned} \tag{69}$$

or, equivalently formulated, if

$$\sum_{s' \in \mathcal{S}^{(+)}} p(\mathbf{y}|d_k = s') > \sum_{s' \in \mathcal{S}^{(-)}} p(\mathbf{y}|d_k = s'), \qquad (70)$$

with $\mathcal{S}^{(+)} \subset \mathbb{S}$ and $\mathcal{S}^{(-)} \subset \mathbb{S}$ being the sets of symbols containing solely symbols with positive and negative real part, respectively. For the given QPSK bit-to-symbol mapping, these two sets of symbols $\mathcal{S}^{(+)}$ and $\mathcal{S}^{(-)}$ coincide with the sets $\mathcal{S}_0^{(1)}$ and $\mathcal{S}_0^{(0)}$ as defined above, respectively. The PDF $p(\mathbf{y}|d_k)$ for the system model (66), in turn, is given by

$$p(\mathbf{y}|d_k) = \zeta_{\mathbf{v}_k^{(0)}|d_k} \exp\left(-(\mathbf{y} - \mathbf{h}_k d_k)^H \mathbf{C}_{\mathbf{vv},k}^{(0)^{-1}} (\mathbf{y} - \mathbf{h}_k d_k)\right),$$
(71)

with $\zeta_{\mathbf{v}_k^{(0)}|d_k} = \frac{1}{\pi^N \left|\mathbf{C}_{\mathbf{vv},k}^{(0)}\right|}$ being constant for every data symbol realization and $\mathbf{C}_{\mathbf{vv},k}^{(0)} \mathbf{C}_k = \sigma_{\mathrm{d}}^2 \bar{\mathbf{H}}_k \bar{\mathbf{H}}_k^H + N\sigma_{\mathrm{n}}^2 \widetilde{\mathbf{H}}$. Combining (68), (70), and (71) leads to the MMSE hard decision criterion

$$\hat{b}_{0k} = \begin{cases} 1 & \displaystyle\sum_{s' \in \mathcal{S}_0^{(1)}} g(s') > \sum_{s' \in \mathcal{S}_0^{(0)}} g(s') \\ 0 & \text{otherwise} \end{cases}, \qquad (72)$$

with $g(s')$ as defined in (65). This is the same decision criterion as that of the LMMSE (64), which concludes the proof.

REFERENCES

[1] S. Baumgartner, O. Lang, and M. Huemer, "A soft interference cancellation inspired neural network for SC-FDE," in *Proc. IEEE 23rd Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Jul. 2022, pp. 1–5.

[2] B. Sagir, E. Aydin, and H. Ilhan, "Deep-learning assisted IoT based RIS for cooperative communications," *IEEE Internet Things J.*, vol. 10, no. 12, pp. 10471–10483, Jun. 2023.

[3] X. Qian and M. Di Renzo, "Receiver design in molecular communications: An approach based on artificial neural networks," in *Proc. 15th Int. Symp. Wireless Commun. Syst. (ISWCS)*, Aug. 2018, pp. 1–5.

[4] J. M. Mateos-Ramos et al., "End-to-end learning for integrated sensing and communication," in *Proc. IEEE Int. Conf. Commun.*, May 2022, pp. 1942–1947.

[5] T. Oshea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cognit. Commun. Netw. Trans. Cogn. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, Dec. 2017.

[6] F. A. Aoudia and J. Hoydis, "Model-free training of end-to-end communication systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 11, pp. 2503–2516, Nov. 2019.

[7] H. He, S. Jin, C.-K. Wen, F. Gao, G. Y. Li, and Z. Xu, "Model-driven deep learning for physical layer communications," *IEEE Wireless Commun.*, vol. 26, no. 5, pp. 77–83, Oct. 2019.

[8] A. Balatsoukas-Stimming and C. Studer, "Deep unfolding for communications systems: A survey and some new directions," in *Proc. IEEE Int. Workshop Signal Process. Syst. (SIPS)*, Oct. 2019, pp. 266–271.

[9] J. Hoydis, F. A. Aoudia, A. Valcarce, and H. Viswanathan, "Toward a 6G AI-native air interface," *IEEE Commun. Mag.*, vol. 59, no. 5, pp. 76–81, May 2021.

[10] H. Ye, G. Y. Li, and B.-H. Juang, "Power of deep learning for channel estimation and signal detection in OFDM systems," *IEEE Wireless Commun. Lett.*, vol. 7, no. 1, pp. 114–117, Feb. 2018.

[11] Q. Hu, F. Gao, H. Zhang, S. Jin, and G. Y. Li, "Deep learning for channel estimation: Interpretation, performance, and comparison," *IEEE Trans. Wireless Commun.*, vol. 20, no. 4, pp. 2398–2412, Apr. 2021.

[12] E. Nachmani, E. Marciano, L. Lugosch, W. J. Gross, D. Burshtein, and Y. Be'ery, "Deep learning methods for improved decoding of linear codes," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 119–131, Feb. 2018.

[13] O. Ploder, C. Auer, C. Motz, T. Paireder, O. Lang, and M. Huemer, "SICNet—Low complexity sample adaptive neural network-based self-interference cancellation in LTE—A/5G mobile transceivers," *IEEE Open J. Commun. Soc.*, vol. 3, pp. 958–972, 2022.

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

[15] S. Baumgartner, C. Böck, M. Huemer, and A. Balatsoukas-Stimming, "Joint self-interference cancellation and data estimation for OFDM based full-duplex communication systems," in *Proc. 57th Asilomar Conf. Signals, Syst., Comput.*, 2023, pp. 345–352.

[16] M. Mohammadkarimi, M. Mehrabi, M. Ardakani, and Y. Jing, "Deep learning-based sphere decoding," *IEEE Trans. Wireless Commun.*, vol. 18, no. 9, pp. 4368–4378, Sep. 2019.

[17] K. Pratik, B. D. Rao, and M. Welling, "RE-MIMO: Recurrent and permutation equivariant neural MIMO detection," *IEEE Trans. Signal Process.*, vol. 69, pp. 459–473, 2021.

[18] N. Samuel, T. Diskin, and A. Wiesel, "Learning to detect," *IEEE Trans. Signal Process.*, vol. 67, no. 10, pp. 2554–2564, May 2019.

[19] H. He, C. Wen, S. Jin, and G. Y. Li, "Model-driven deep learning for MIMO detection," *IEEE Trans. Signal Process.*, vol. 68, pp. 1702–1715, 2020.

[20] J. Liao, J. Zhao, F. Gao, and G. Y. Li, "A model-driven deep learning method for massive MIMO detection," *IEEE Commun. Lett.*, vol. 24, no. 8, pp. 1724–1728, Aug. 2020.

[21] M. Khani, M. Alizadeh, J. Hoydis, and P. Fleming, "Adaptive neural signal detection for massive MIMO," *IEEE Trans. Wireless Commun.*, vol. 19, no. 8, pp. 5635–5648, Aug. 2020.

[22] N. Shlezinger, N. Farsad, Y. C. Eldar, and A. J. Goldsmith, "ViterbiNet: A deep learning based Viterbi algorithm for symbol detection," *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3319–3331, May 2020.

[23] N. Shlezinger, R. Fu, and Y. C. Eldar, "DeepSIC: Deep soft interference cancellation for multiuser MIMO detection," *IEEE Trans. Wireless Commun.*, vol. 20, no. 2, pp. 1349–1362, Feb. 2021.

[24] T. Van Luong, N. Shlezinger, C. Xu, T. M. Hoang, Y. C. Eldar, and L. Hanzo, "Deep learning based successive interference cancellation for the non-orthogonal downlink," *IEEE Trans. Veh. Technol.*, vol. 71, no. 11, pp. 11876–11888, Nov. 2022.

[25] Y. Wei, M.-M. Zhao, M. Hong, M.-J. Zhao, and M. Lei, "Learned conjugate gradient descent network for massive MIMO detection," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–6.

[26] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Netw.*, vol. 4, no. 2, pp. 251–257, 1991.

[27] J. R. Hershey, J. Le Roux, and F. Weninger, "Deep unfolding: Model-based inspiration of novel deep architectures," 2014, *arXiv:1409.2574*.

[28] F. Pancaldi, G. M. Vitetta, R. Kalbasi, N. Al-Dhahir, M. Uysal, and H. Mheidat, "Single-carrier frequency domain equalization," *IEEE Signal Process. Mag.*, vol. 25, no. 5, pp. 37–56, Sep. 2008.

[29] M. Huemer, A. Koppler, R. Weigel, and L. Reindl, "A review of cyclically extended single carrier transmission with frequency domain equalization for broadband wireless transmission," *Eur. Trans. Telecommun.*, vol. 14, no. 4, pp. 329–341, 2003.

[30] M. Huemer, H. Witschnig, and J. Hausner, "Unique word based phase tracking algorithms for SC/FDE-systems," in *Proc. IEEE Global Telecommun. Conf.*, Dec. 2003, pp. 70–74.

[31] S. Baumgartner, O. Lang, and M. Huemer, "Neural network based single-carrier frequency domain equalization," in *Computer Aided Systems Theory—EUROCAST 2022* (Lecture Notes in Computer Science), 2022. Cham, Switzerland: Springer, pp. 295–302.

[32] W.-J. Choi, K.-W. Cheong, and J. M. Cioffi, "Iterative soft interference cancellation for multiple antenna systems," in *Proc. IEEE Wireless Commun. Netw. Conf. Rec.*, Sep. 2000, pp. 304–309.

[33] N. Shlezinger, R. Fu, and Y. C. Eldar, "Deep soft interference cancellation for MIMO detection," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 8881–8885.

[34] H. Witschnig, T. Mayer, A. Springer, L. Maurer, M. Huemer, and R. Weigel, "The advantages of a known sequence versus cyclic prefix in a SC/FDE system," in *Proc. 5th Int. Symp. Wireless Pers. Multimedia Commun.*, vol. 3, Oct. 2002, pp. 1328–1332.

[35] M. Huemer, C. Hofbauer, and J. B. Huber, "Unique word prefix in SC/FDE and OFDM: A comparison," in *Proc. IEEE Globecom Workshops*, Dec. 2010, pp. 1296–1301.

[36] S. Reinhardt, T. Buzid, and M. Huemer, "Receiver structures for MIMO-SC/FDE systems," in *Proc. IEEE 63rd Veh. Technol. Conf.*, Sep. 2006, pp. 1401–1405.

[37] M. Huemer, L. Reindl, A. Springer, and R. Weigel, "Implementation aspects on single carrier transmission with frequency domain equalization," in *Proc. 4th Int. OFDM Workshop*, 1999, p. 18.

[38] Y. Yang, T. Ihalainen, M. Rinne, and M. Renfors, "Frequency-domain equalization in single-carrier transmission: Filter bank approach," *EURASIP J. Adv. Signal Process.*, vol. 2007, no. 1, pp. 1–16, Dec. 2007.

[39] S. Baumgartner, G. Bognár, O. Lang, and M. Huemer, "Neural network approaches for data estimation in unique word OFDM systems," *IEEE Trans. Veh. Technol.*, vol. 73, no. 3, pp. 3690–3706, Mar. 2024.

[40] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*, vol. 1. Upper Saddle River, NJ, USA: Prentice-Hall, 1993.

[41] C. Szegedy et al., "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Oct. 2015, pp. 1–9.

[42] H. Xu, Y. Li, B. Tan, J. Wu, and D. Hu, "A neural network empowered belief propagation algorithm optimized for short-cycles in tanner graph," *IEEE Trans. Mach. Learn. Commun. Netw.*, vol. 1, pp. 31–42, 2023.

[43] R. Wiesmayr, G. Marti, C. Dick, H. Song, and C. Studer, "Bit error and block error rate training for ML-assisted communication," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2023, pp. 1–5.

[44] M. Goutay, F. A. Aoudia, and J. Hoydis, "Deep hypernetwork-based MIMO detection," in *Proc. IEEE 21st Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, May 2020, pp. 1–5.

[45] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, *Efficient BackProp*. Berlin, Germany: Springer, 2012, pp. 9–48.

[46] C. M. Bishop, *Neural Networks for Pattern Recognition*. London, U.K.: Oxford Univ. Press, 1995.

[47] 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.*, Oct. 2021, pp. 381–388.

[48] J. Fakatselis, *Criteria for 2.4 GHz PHY Comparison of Modulation*, document IEEE P802.11-97/157r1, 1997.

[49] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, 2019, pp. 2623–2631.

[50] M. Huemer and O. Lang, "On component-wise conditionally unbiased linear Bayesian estimation," in *Proc. 48th Asilomar Conf. Signals, Syst. Comput.*, Nov. 2014, pp. 879–885.

[51] M. Huemer, C. Hofbauer, and J. Huber, "The potential of unique words in OFDM," in *Proc. 15th Int. OFDM Workshop*, 2010, pp. 140–144.

[52] M. Huemer, C. Hofbauer, and J. B. Huber, "Non-systematic complex number RS coded OFDM by unique word prefix," *IEEE Trans. Signal Process.*, vol. 60, no. 1, pp. 285–299, Jan. 2012.

[53] O. Lang, C. Böck, M. Huemer, and C. Hofbauer, "Increasing the bandwidth efficiency in UW-OFDM," in *Proc. 53rd Asilomar Conf. Signals, Syst., Comput.*, Nov. 2019, pp. 1857–1861.

[54] G. H. Golub and C. F. van Loan, *Matrix Computations*. Baltimore, MD, USA: Johns Hopkins Univ. Press, 2013.

[55] W. Haselmayr and A. Springer, "Extrinsic LLR computation by the SISO LMMSE detector: Four different approaches," in *Computer Aided Systems Theory—EUROCAST 2015* (Lecture Notes in Computer Science). Cham, Switzerland: Springer, 2015, pp. 529–536.

STEFAN BAUMGARTNER (Graduate Student Member, IEEE) was born in Linz, Austria, in 1996. He received the Dipl.-Ing. (M.Sc. equivalent) degree in electronics and information technology, specialized in signal and information processing, from Johannes Kepler University (JKU), Linz, in 2021, where he is currently pursuing the Ph.D. degree with the Institute of Signal Processing. His research interests include machine learning approaches in communications engineering.

OLIVER LANG (Member, IEEE) received the bachelor's degree in electrical engineering and information technology and the master's degree in microelectronics from Vienna University of Technology, Vienna, Austria, in 2011 and 2014, respectively, and the Ph.D. degree from Johannes Kepler University (JKU) Linz, Linz, Austria, in 2018. From 2014 to 2018, he was a member of the Institute of Signal Processing (ISP), JKU Linz. From 2018 to 2019, he worked at 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 the ISP, JKU. He is the main inventor of several patents in the field of automotive radar systems and the main author of several publications in the field of estimation theory and adaptive filtering.

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 was 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. From 2012 to 2013, he was the Dean of the Faculty of Technical Sciences. In September 2013, he moved back to Linz, where he is currently heading the Institute of Signal Processing, JKU Linz, as a Full Professor. Since 2017, he has been the Co-Head of the Christian Doppler Laboratory for Digitally-Assisted RF Transceivers for Future Mobile Communications. His research interests include statistical and adaptive signal processing, signal processing architectures and implementations, as well as mixed-signal processing with applications in information and communications engineering, radio frequency, and baseband integrated circuits, sensor, and biomedical signal processing. Within these fields, he has published more than 300 scientific articles. He is a member of the IEEE Signal Processing, the Circuits and Systems, the Microwave Theory and Techniques, the Communications Societies, German Society of Information Technology (ITG), and Austrian Electrotechnical Association (OVE). He received the dissertation awards of the ITG and Austrian Society of Information and Communications Technology (GIT) in 2000, Austrian Cardinal Innitzer Award in Natural Sciences in 2010, and German ITG Award in 2016. From 2009 to 2015, he was a member of the editorial board of the *International Journal of Electronics and Communications* (AEU). From May 2017 to April 2019, he served as an Associate Editor for the IEEE Signal Processing Letters.