

Deep Reservoir Computing Meets 5G MIMO-OFDM Systems in Symbol Detection

Zhou Zhou,¹ Lingjia Liu,¹ Vikram Chandrasekhar,² Jianzhong Zhang,² Yang Yi¹

¹Department of Electrical and Computer Engineering, Virginia Tech, USA

{zhouzhou, ljliu, yangyi8}@vt.edu

²Samsung Research America, USA

{v.chandrasek, jianzhong.z}@samsung.com

Abstract

Conventional reservoir computing (RC) is a shallow recurrent neural network (RNN) with fixed high dimensional hidden dynamics and one trainable output layer. It has the nice feature of requiring limited training which is critical for certain applications where training data is extremely limited and costly to obtain. In this paper, we consider two ways to extend the shallow architecture to deep RC to improve the performance without sacrificing the underlying benefit: (1) Extend the output layer to a three layer structure which promotes a joint time-frequency processing to neuron states; (2) Sequentially stack RCs to form a deep neural network. Using the new structure of the deep RC we redesign the physical layer receiver for multiple-input multiple-output with orthogonal frequency division multiplexing (MIMO-OFDM) signals since MIMO-OFDM is a key enabling technology in the 5th generation (5G) cellular network. The combination of RNN dynamics and the time-frequency structure of MIMO-OFDM signals allows deep RC to handle miscellaneous interference in nonlinear MIMO-OFDM channels to achieve improved performance compared to existing techniques. Meanwhile, rather than deep feedforward neural networks which rely on a massive amount of training, our introduced deep RC framework can provide a decent generalization performance using the same amount of pilots as conventional model-based methods in 5G systems. Numerical experiments show that the deep RC based receiver can offer a faster learning convergence and effectively mitigate unknown non-linear radio frequency (RF) distortion yielding twenty percent gain in terms of bit error rate (BER) over the shallow RC structure.

Introduction

The early prosperity in academia of applying artificial neural networks to communications engineering can be traced back to 20 years ago (Patnaik et al. 2004). Due to the recent success of deep neural networks (DNNs), applications of DNNs in communications have attracted growing attention again. Although conventional wireless communications systems are established well through modeling methods, the advent of the 5th generation (5G) challenges the conventional physical layer transmission technologies and network architecture according to the great market growth: The

Copyright © 2020, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

global subscriptions of 5G is expected to reach 8.3 billion by the end of 2024, where 95 percent will be broadband services (Cerwall et al. 2019). In 5G new radio (NR), enhanced mobile broadband (eMBB) is one of the primary use cases defined for the high data rate transmission. To be specific, eMBB is expected to support stable wireless connections with very high data rates (tens of Gigabits per second) for low speed mobile users as well as moderate data rates for high speed mobile users. Meanwhile, 5G networks will support user mobility upto 500 km/h (Murara 2017) creating a large variation on the dynamics of the underlying wireless channel. Therefore, it is extremely important to have a 5G receiver that is able to conduct high speed symbol detection and is robust under different wireless propagation characteristics. Invoking neural networks into the 5G NR receiving process is a promising direction to improve the state of the art model-based receiving architecture for 5G and Beyond (Shafin et al. 2020).

In this work we aim to exploit the dynamic behaviors of deep recurrent neural networks (RNNs) for the task of symbol detection in a multiple-input multiple-output with orthogonal frequency division multiplexing (MIMO-OFDM) system. Note that MIMO-OFDM is the adopted waveform for 4G/5G (Liu et al. 2012) and is expected to be the dominant waveform for Beyond 5G systems as well. The main motivation of adopting RNNs instead of other neural network architectures is based on the fact that under fairly mild and general assumptions, RNNs are universal approximations of dynamic systems (Funahashi and Nakamura 1993). This feature is extremely important for 5G systems which are highly dynamic over time and frequency. However, to realize the full potential of RNNs for MIMO-OFDM symbol detection, new research challenges need to be addressed:

- From system design’s perspective, training a neural network (NN)-based receiver using over-the-air feedback — to update layer weights of the NN based on the back-propagation algorithm — is prohibitively expensive in terms of uplink control overhead. Over-fitting could happen when the selected NN model is too complicated.
- The underlying wireless environment is changing dynamically over time and frequency. This is especially true for 5G and future 6G systems which will utilize

high-frequency mmWave and Terahertz channels. Accordingly, the underlying NN model needs to be sophisticated enough to capture the time-frequency variation of the channel since otherwise under-fitting can result in a poor symbol detection performance.

- NNs, especially RNNs, are mainly designed to process time-domain data. Since data is transmitted in both time and frequency domains in MIMO-OFDM systems, it is critical to combine RNN with domain knowledge in an organic way to offer reliable and robust performance gains over conventional communication strategies.

In light of the challenges, we adopt the reservoir computing (RC) (Lukoševičius and Jaeger 2009) framework as our approach. RC is one category of RNNs which can avoid the gradient vanishing and exploding issues occurring during the training of conventional RNNs using back-propagation through time (BPTT) (Pascanu, Mikolov, and Bengio 2013). Furthermore, the training of the standard RC is only conducted on the output layer while the hidden layers and input layers are fixed according to a certain distribution. In this way, the training overhead will be significantly reduced making it a suitable NN platform for 5G receivers with reduced control overhead. To improve the processing capability of RC, we introduce the deep RC by making the following extensions to the shallow RC:

- Extend the output layer to multiple time-frequency layers. This extension is encouraged by the inherent time-frequency structure of the MIMO-OFDM signal.
- Stack shallow RCs into deep RCs. The motivation for this extension is rooted from the boosting mechanism.

Our first extension on deepening the output layer is by replacing the original one layer output to a three layer structure which is constructed by a time domain layer, a Fourier transform layer, and a frequency domain layer, namely “time-frequency RC” tailoring towards the unique structure of MIMO-OFDM waveforms. The second extension of the deep RC is implemented by concatenating several RCs sequentially. The output weights of each RC layer are learned through a consecutive manner which differs from the methods proposed in (Jaeger 2007; Gallicchio and Micheli 2017; Gallicchio, Micheli, and Pedrelli 2017). Through numerical experiments, we observe that this multiple-layered structure shows an appealing performance when nonlinear radio frequency (RF) effect exists in the transmission link. That is, it achieves a high success rate of symbol detection when received signals are distorted by non-linear RF components which effectively improves the hardware efficiency. This learning-based method can be considered as an enabling technique for future communications systems.

Related Work

Deep Reservoir Computing

DNNs have been proven to be able to extract sophisticated features which lead to improved classification abilities over shallow NNs (Krizhevsky, Sutskever, and Hinton 2012). It is first revealed in (El Hiji and Bengio 1996) that hierarchical RNNs can learn long-term dependencies of signals.

In (Zhu et al. 2016), the deep long short-term memory network was introduced by stacking multiple RNNs hidden layers consecutively. It demonstrates that this deep structure can significantly improve the performance in the task of action recognition. Based on these evidence, we believe a similar advantage can be achieved by extending shallow RCs to multiple layered RCs. In general, the methods of extending a shallow RNN into deeper RNNs have been summarized in (Pascanu et al. 2014). It introduced four ways which are respectively increasing input layers, hidden layers, and output layers, as well as stacking multiple shallow RNNs into a deep form. As a special case of RNN, the extension of the shallow RC can follow the same way. Accordingly, specific training methods can be developed for the deep RC structure due to the free training of the hidden layers. In (Jaeger 2007), it introduced concatenating echo state networks (ESNs), a special RC, into a chain by learning readout layers connecting to each ESN layer. The follow-up work (Gallicchio and Micheli 2017; Gallicchio, Micheli, and Pedrelli 2017) extended this structure into a deep version which is able to achieve a higher memory compared to the shallow one.

Neural Networks based Symbol Detection

For symbol detection in MIMO-OFDM systems, the conventional method is based on modeling the transmission link and solving the formulated problem based on the model. Recent advances in NNs potentially offer a solution to the symbol detection task without relying on the model-based assumptions (Samuel, Diskin, and Wiesel 2017; Khani et al. 2019; Mosleh et al. 2016; 2018). Generally speaking, there are major ways to utilize the help of DNN. The first one use DNN to conduct parameter tuning based on existing symbol detection methods. For example, in (Samuel, Diskin, and Wiesel 2017) DNN is introduced to conduct MIMO symbol detection based on the input of the estimated channel state information (CSI) and the received symbols. A DNN is constructed in (Khani et al. 2019) based on iterative soft-thresholding algorithms to fine tune the parameters for MIMO symbol detection. However, these methods usually require large training datasets with explicit CSI available. Furthermore, they didn't consider the effect of OFDM on MIMO symbol detection. Alternatively, MIMO symbol detection can be formulated as a classification problem where DNNs can be directly utilized. In (Mosleh et al. 2016; 2018), ESN is utilized for the MIMO-OFDM symbol detector which does not rely on the explicit information of channel states. The effectiveness of this scheme is evaluated through the comparison with conventional methods over non-linear RF components. Furthermore, (Zhou, Liu, and Chang 2019) introduced the windowed echo state network (WESN) by adding a sliding window to the input of ESN to enhance the short-term memory (STM) of the underlying NN. WESN is shown to provide better performance over standard ESN using LTE/LTE-Advanced compatible reference signals as the training set. However, to the best of our knowledge, there is no existing DNN methods combining features of the MIMO-OFDM waveform and deep RC architecture to conduct symbol detection directly.

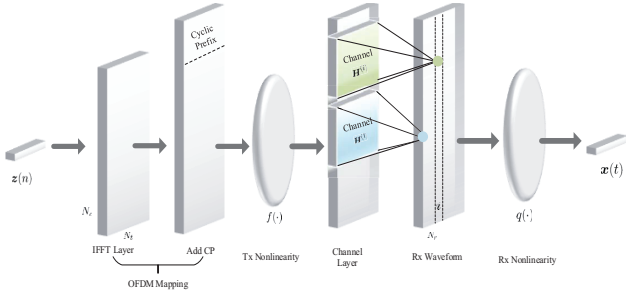


Figure 1: Overview of MIMO-OFDM signal generation.

MIMO-OFDM Data Structure

In this section, we briefly introduce the generation of a sequential MIMO-OFDM signal from the perspective of NNs. It offers an insight on embedding the intrinsic time-frequency structure to the design of RC based MIMO-OFDM receiver. The sequential MIMO-OFDM signal is concatenated by individual batches, where each batch is denominated a ‘‘MIMO-OFDM symbol’’. In our scenario, a transmitter (Tx) sends MIMO-OFDM symbols to a receiver (Rx) based on the unit of the MIMO-OFDM symbol. Every single MIMO-OFDM symbol at the Rx is formed by a chain of signal processing blocks which can be regarded as a multiple layered generative NN as illustrated in Figure 1. The input of this NN is denoted as a batch $\{z(n)\}_{n=0}^{N_c-1}$, where $z(n) \in \mathbb{C}^{N_t \times 1}$ represents the modulation symbols at the n th sub-carrier of OFDM in the **Frequency Domain**; N_t stands for the number of transmitting antennas in the system. Each entry of $z(n)$ is randomly selected from a pre-designed table according to modulation manners. For example, $\{1, -1\}$ is the table of binary phase-shift keying (BPSK), and $\{1+1j, 1-1j, -1+1j, -1-1j\}$ is the quadrature phase-shift keying (QPSK).

To convey more information through one modulation symbol, the table pattern can be generalized to a higher-order modulation noted as M -quadrature amplitude modulation (M-QAM), where M is the number of symbols in the modulation class.

For ease of expression, the input batch can be equivalently lumped to a $N_c \times N_t$ matrix as

$$\mathbf{Z} \triangleq [z^T(0), z^T(1), \dots, z^T(N_c - 1)]^T. \quad (1)$$

Then, \mathbf{Z} is fed into an inverse fast Fourier transform (IFFT) layer which performs the IFFT on each column of \mathbf{Z} , where the output of this layer is denoted as $\tilde{\mathbf{Z}}$. The ‘Add CP’ layer combines $\tilde{\mathbf{Z}}$ with the cyclic prefix (CP) which is defined as the last N_{cp} rows of $\tilde{\mathbf{Z}}$. Accordingly, the resulting output of the ‘Add CP’ layer can be expressed as $\tilde{\mathbf{X}} = [\tilde{\mathbf{Z}}(N_c - N_{cp} : N_c, :)^T, \tilde{\mathbf{Z}}^T]^T$. Similarly, CP removal is defined as removing the first N_{cp} rows of $\tilde{\mathbf{X}}$. We further define

$$\tilde{\mathbf{X}} \triangleq [\tilde{\mathbf{x}}^T(0), \tilde{\mathbf{x}}^T(1), \dots, \tilde{\mathbf{x}}^T(N_c + N_{cp} - 1)]^T \quad (2)$$

which represents a MIMO-OFDM symbol at Tx in the **Time**

Domain. Note that \mathbf{Z} also can be reversely obtained from $\tilde{\mathbf{X}}$ by removing the added CP and conducting an FFT.

The MIMO-OFDM symbol is distorted by a nonlinear activation function $f(\cdot)$ due to the existing nonlinear RF components at the Tx, such as the power amplifier (PA) (Joung et al. 2014). The channel layer is expressed as multi-kernels convolutions without zero-padding. Therefore, the i th column of the output of the channel layer is calculated by

$$\mathbf{X}[:, i] = f(\tilde{\mathbf{X}}) \circledast \mathbf{H}^{(i)} + \mathbf{n}[:, i], \quad (3)$$

where $\mathbf{X} \in \mathbb{C}^{(N_c + N_{cp}) \times N_r}$ denotes the output signal of the channel layer; \mathbf{n} is the additive noise; \circledast stands for the convolution operator; $\mathbf{H}^{(i)} \in \mathbb{C}^{N_{cp} \times N_t}$ is the i th channel kernel which is generated according to the definition from the 3GPP spatial channel model. Note that the number of the channel kernels is equal to the number of receiving antennas, where the channel kernels are denoted as $\{\mathbf{H}^{(i)}\}_{i=0}^{N_r-1}$. Finally, a non-linear activation function $q(\cdot)$ is applied before the end of the output which represents non-linear RF components at the Rx, such as analog to digital converters (ADCs). With a slight abuse of the notation, we denote the output of the generative NN as

$$\mathbf{X} \triangleq [\mathbf{x}^T(0), \mathbf{x}^T(1), \dots, \mathbf{x}^T(N_c + N_{cp} - 1)]^T, \quad (4)$$

where $\mathbf{x}(t)$ represents the t th sample of one MIMO-OFDM symbol in the **Time Domain**.

For MIMO-OFDM symbol detection, we target to recover the **Frequency Domain** modulation symbols $\{z(n)\}_{n=0}^{N_c-1}$ through the **Time Domain** observations $\{\mathbf{x}(t)\}_{t=0}^{N_{cp}+N_c-1}$. Since symbol detection is essentially a classification problem, we can utilize NNs for symbol detection via supervised learning. To be specific, we denote the training set as $\{d_q\}_{q=0}^{Q-1}$, where d_q represents a tuple defined as

$$\begin{aligned} d_q &\triangleq \left(\{\mathbf{x}_q(t)\}_{t=0}^{N_{cp}+N_c-1}, \{z_q(n)\}_{n=0}^{N_c-1} \right) \\ &\cong \left(\{\mathbf{x}_q(t)\}_{t=0}^{N_{cp}+N_c-1}, \{\tilde{\mathbf{x}}_q(t)\}_{t=0}^{N_{cp}+N_c-1} \right); \quad (5) \\ &\cong (\mathbf{X}_q, \mathbf{Z}_q) \cong (\mathbf{X}_q, \tilde{\mathbf{X}}_q) \end{aligned}$$

where \cong represents ‘‘equivalently defined as’’; the subscript q stands for the q th MIMO-OFDM symbol, i.e, the training set has Q batches in total. The notations defined in this section are summarized in Table 1 for clarity.

The generation of the MIMO-OFDM dataset strictly follows the 5G NR specification in (3GPP TS 38.211 2019). For example, the OFDM signal generation follows OFDM baseband signal generation defined in section 5.3.1; the functional block of modulation follows the modulation mapper defined in section 7.3.1.2; MIMO operation follows the layer mapping and the antenna port mapping defined in section 7.3.1.3 and section 7.3.1.4. The wireless transmission channel follows the WINNER II channel (Meinilä et al. 2009) which is an established channel model based on practical measurement instead of analytical models.

Table 1: Notations

Symbols	Definitions
N_r	The number of receiving antennas
N_t	The number of transmitting antennas
N_c	The number of sub-carriers
N_{cp}	The length of CP
Q	The number of MIMO-OFDM symbols in the training set (The number of batches in the training set)
\mathbf{X}	One MIMO-OFDM symbol at Rx in time domain (One batch of training input)
$\tilde{\mathbf{X}}$	One MIMO-OFDM symbol at Tx in time domain (One batch of training target in the time domain)
\mathbf{Z}	One MIMO-OFDM symbol at Tx in frequency domain (One batch of training target in the frequency domain)
$\mathbf{x}(t)$	The t th sample of one MIMO-OFDM symbol at Rx in the time domain
$\tilde{\mathbf{x}}(t)$	The t th sample of one MIMO-OFDM symbol at Tx in the time domain
$\mathbf{z}(n)$	The n th modulation symbol of one MIMO-OFDM symbol at Tx in the frequency domain

Multiple Output Layers

To describe the introduced time-frequency RC, we begin by introducing a RC structure which only has a single output layer. It can be directly applied to the symbol detection problem using the time domain MIMO-OFDM signal. Once the output layer is learned, the MIMO-OFDM symbols is obtained by a standard OFDM symbol de-mapping without further learning. For convenience, we denote this basic RC structure as ‘‘time-domain RC’’ to distinguish from the RC with time-frequency layers.

Time-Domain RC

One realization of RC with a single output layer is the ESN (Jaeger 2001). ESN drives an input signal into non-linear response signals lying in a high dimensional space through a fixed random projection. The non-linear dynamic response is represented by the trajectory of hidden neuron states. Meanwhile, non-linear activation functions are applied between consecutive states. The neurons in the reservoir are sparsely connected with fixed weights satisfying certain distributions under which the response signals are asymptotically uncorrelated to the initial neurons states. One evidence of using fixed states transition is from an experimental fact that the dominant changes of RNN’s weights happen at the output layer after the training (Schiller and Steil 2005). Finally, the desired output of ESN is obtained by a linear combination of the non-linear response signals.

For the time-domain RC based MIMO-OFDM symbol detection, given the training set $\{d_q\}_{q=0}^{Q-1}$, the states of the reservoir generated by the q th batch are calculated by

$$\mathbf{s}_q(t+1) = f(\mathbf{s}_q(t)\mathbf{W}_s + \mathbf{x}_q(t)\mathbf{W}_{in} + \tilde{\mathbf{x}}_q(t)\mathbf{W}_{fb} + \mathbf{n}(t))$$

where $\mathbf{s}_q(t) \in \mathbb{C}^{N_n \times 1}$ is the state vector of neurons in which N_n denotes the number of neurons in the reservoir; f is the activation function; \mathbf{W}_s is the states transition weights; \mathbf{W}_{in} is the weights of the input layer; $\mathbf{n}(t)$ is an optional noise regularization term; \mathbf{W}_{fb} is the weights on the feedback path, which can be removed when the teacher forcing is not required. Through a time domain readout layer \mathbf{W}_{tout} , the output sequence of the ESN is given by

$$\mathbf{y}_q(t) = \mathbf{s}_q(t)\mathbf{W}_{tout}. \quad (6)$$

To drive $\mathbf{y}_q(t)$ to the desired time domain MIMO-OFDM symbol, we can minimize the l_2 norm distance between $\{\mathbf{y}_q(t)\}_{q=0}^{Q-1}$ and $\{\tilde{\mathbf{x}}_q(t)\}_{q=0}^{Q-1}$, i.e.,

$$\min_{\mathbf{W}_{tout}} \sum_{q=0}^{Q-1} \sum_{t=0}^{N_{cp}+N_c-1} \|\mathbf{y}_q(t) - \tilde{\mathbf{x}}_q(t)\|_2^2. \quad (7)$$

Therefore, the readout weights can be updated by the following closed-form expression,

$$\mathbf{W}_{tout} = \left([\mathbf{S}_0^T, \dots, \mathbf{S}_{Q-1}^T]^T \right)^+ [\tilde{\mathbf{X}}_0^T, \dots, \tilde{\mathbf{X}}_{Q-1}^T]^T$$

where $(\mathbf{S})^+$ stands for the pseudo-inverse of \mathbf{S} ; and \mathbf{S}_q is stacked by the trajectory of the states as

$$\mathbf{S}_q \triangleq [\mathbf{s}_q^T(0), \mathbf{s}_q^T(1), \dots, \mathbf{s}_q^T(N_c + N_{cp} - 1)]^T. \quad (8)$$

Due to the feedback loop within the reservoir, there exists a lag-effect on the generated states response (Lukoševičius and Jaeger 2009). Accordingly, a delay parameter can be introduced in the learning (Holzmann and Hauser 2010) where the following slightly revised training set will be used:

$$d^{(p)} \triangleq \left([\mathbf{X}_0, \dots, \mathbf{X}_{Q-1}, \mathbf{O}_{N_r \times q}]^T, \right. \\ \left. [\mathbf{O}_{N_r \times q}, \tilde{\mathbf{X}}_0, \dots, \tilde{\mathbf{X}}_{Q-1}]^T \right)$$

where $p \in [0, P]$ is the delay parameter, and \mathbf{O} represents a zero matrix. Once the RC is trained with different values of p , we choose the p^* which generates the minimal training error defined in (7). In the testing stage, testing set is configured with the input-output delay offset p^* as well.

Time-Frequency RC

In time-frequency RC, the reservoir first generates high-dimensional response signals from the input. The response signals are then mapped to desired signals through a time-domain layer, an FFT layer, and a frequency-domain layer.

The time domain output is the same as that in (6). After removing the CP and conducting the FFT on $\{\mathbf{y}_q(t)\}_{t=N_{cp}}^{N_c+N_{cp}-1}$, we can obtain the frequency domain symbols $\{\tilde{\mathbf{y}}_n(n)\}_{n=0}^{N_c-1}$. After applying a layer of neurons to the frequency symbols, we have

$$\tilde{\mathbf{z}}_q(n) = \tilde{\mathbf{y}}_n(n)\text{diag}(\mathbf{w}_{fout}(n)) \quad (9)$$

where $\mathbf{w}_{fout}(n)$ is the weight at the n th sub-carrier; $\text{diag}(\mathbf{w})$ represents a diagonal matrix in which the entries on

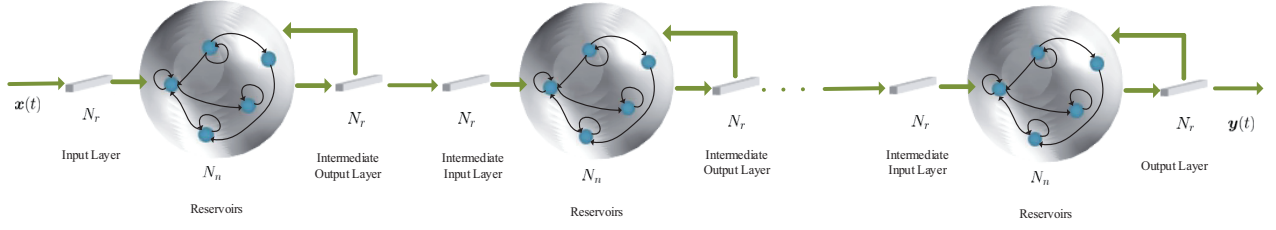


Figure 2: The structure of stacked time-domain RCs.

the main diagonal are \mathbf{w} , and elsewhere are zeros. Furthermore, we design the magnitude of each entry of $\mathbf{w}_{f_{out}}(n)$ as 1. This specific design allows a compensation for the residual phase error after the time domain processing. Note that this frequency domain compensation is equivalent to the delay learning procedure introduced in the time-domain RC since the Fourier transform converts the delay in the time domain to a phase shift in the frequency domain. By comparing the time-frequency RC structure to the MIMO-OFDM generative network defined in Figure 1, we can observe that the output layers of the time-frequency RC is symmetric to the first two layers in the MIMO-OFDM generative network. Therefore, this time-frequency RC essentially leverages the generative mechanism of the MIMO-OFDM signal.

The learning of the time-frequency output layer can be formulated as

$$\begin{aligned} \min_{\{\mathbf{w}_{f_{out}}(n)\}_{n=0}^{N_c-1}} \quad & \sum_{q=0}^{Q-1} \sum_{n=0}^{N_c-1} \|\mathbf{z}_q(n) - \tilde{\mathbf{z}}_q(n)\|_2^2 \\ \text{s.t.} \quad & \text{diag}(|\mathbf{w}_{f_{out}}(n)|) = \mathbf{I} \\ & \forall n = 0, \dots, N_c - 1 \end{aligned} \quad (10)$$

where the objective is equivalent to the time domain loss function defined in (7). To seek a proper solution of the above problem, we resort to alternative least squares (ALS) by which the closed-form updating rules of \mathbf{W}_{tout} and $\mathbf{w}_{f_{out}}(n)$ are derived as

$$\begin{aligned} \angle w_{f_{out},j}(n) &= -\angle(\mathbf{z}_j(n)^H \tilde{\mathbf{z}}_j(n)) \\ \mathbf{W}_{tout} &= \mathbf{S}^+ \tilde{\mathbf{F}}^H \hat{\mathbf{Z}} \end{aligned} \quad (11)$$

where $w_{f_{out},j}(n)$ is the j th entry of $\mathbf{w}_{f_{out}}(n)$; $\tilde{\mathbf{z}}_j(n)$ is the j th column of $\tilde{\mathbf{Z}}(n)$ which is defined as $\tilde{\mathbf{Z}}(n) \triangleq \tilde{\mathbf{F}}(n) \mathbf{S} \mathbf{W}_{tout}$, where $\tilde{\mathbf{F}}(n) \triangleq \mathbf{I} \otimes \mathbf{f}(n)$, $\mathbf{f}(n)$ is the n th row of the Fourier matrix; $\hat{\mathbf{Z}} := [\hat{\mathbf{Z}}_0^T, \dots, \hat{\mathbf{Z}}_{Q-1}^T]^T$, where $\hat{\mathbf{Z}}_q := [\hat{\mathbf{z}}_q^T(0), \hat{\mathbf{z}}_q^T(1), \dots, \hat{\mathbf{z}}_q^T(N_c - 1)]^T$ and $\hat{\mathbf{z}}_q(n) := \mathbf{z}_q(n) \text{diag}(\mathbf{w}_{f_{out}}^*(n))$.

Deep RC through Stacked RCs

Deep RC can be constructed by stacking multiple shallow RCs. We first stack time-domain RCs as depicted in Figure 2. Based on this structure, each single RC component (RC layer) conducts a different level of interference cancellation for the received MIMO-OFDM signal. Meanwhile, in

the introduced deep RC, the state transitions between different RCs are independent of each other.

Let L be the number of RCs of the deep RC. Given the training set, $d^{(p)}$, the state equation of the l th RC is given by

$$\begin{aligned} \mathbf{s}_q^{(l)}(t+1) &= f\left(\mathbf{s}_q^{(l)}(t) \mathbf{W}_s^{(l)}\right. \\ &\quad \left. + \mathbf{y}_q^{(l-1)}(t) \mathbf{W}_{in}^{(l)} + \tilde{\mathbf{x}}_q(t) \mathbf{W}_{fb}^{(l)} + \mathbf{n}(t)\right) \end{aligned}$$

where $\mathbf{y}^{(l)}$ follows the output equation from the previous layer which is defined in (6). All unknown weights of the deep RC are learned through a sequential manner: The training input of any RC is the fitting results generated from the previous RC, i.e., after the $l-1$ th RC is trained, the output is fed into the l th RC as its input. The teacher forcing target of each RC is the same and the output of the final RC is the detection of the underlying MIMO-OFDM signal. Effectively, this learning strategy is a boosting method (Freund 1995) where each RC is a weak learner while their connected ensembles result in a robust learning outcome.

Additionally, the selection of the number of RCs requires a cross-validation test: Increase the number of RC layers in training, meanwhile, calculate the validation loss; once the validation loss stops decreasing, we can stop increasing the number of RCs. Furthermore, the number of neurons in each RC can be different and how to optimize the number of neurons of each RC to achieve the best generalization performance is left as our future work. Since the way of constructing the deep RC is offered through a generic framework, we can also stack the time-frequency RCs into a stacked structure as depicted in Figure 3. Rather than stacking time-domain RCs, an extra IFFT layer needs to be added between every two consecutive time-frequency RCs to transform the previous frequency domain output to a time-domain input to the next RC layer.

Numerical Experiments

In this section, we provide numerical evaluations for our introduced deep RCs. The modulation we adopted to generate $z(n)$ is 16QAM and the parameters related to the training set according to Table 1 are configured as follows: $N_r = 4$, $N_t = 4$, $N_c = 1024$, $N_{cp} = 160$, $Q = 4$. The underlying wireless channel is the Winner II channel where Tx and Rx are configured to be uniform linear arrays with half-wavelength antennas spacing. The specific scenario we chose is outdoor-to-indoor. For each obtained BER point, we

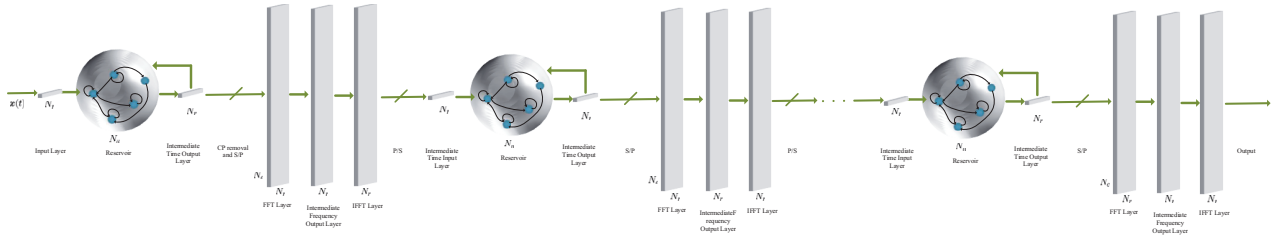


Figure 3: The structure of stacked time-frequency RCs.

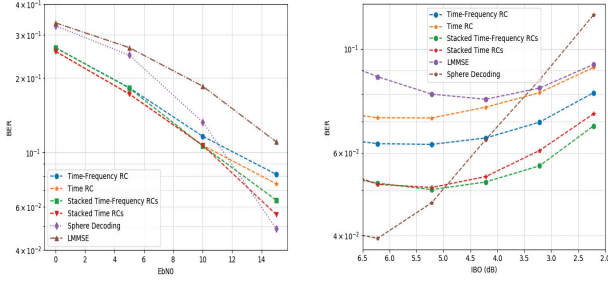


Figure 4: The average BER curve for RC based methods and conventional methods, where the left one is in PA linear region by varying the PA input power; while the right one is by varying IBO in PA non-linear region.

simulate the following training-testing procedure: We first send the training set through the channel, then we test the BER using a testing set which is chosen as the 13 MIMO-OFDM symbols next to the training set; We run 100 times of this training and testing procedure while using channels generated on the fly.

For the configuration of the RCs, the number of neurons is set as 128 for both shallow RC and each RC in the deep RC structure. Furthermore, a time window is added to the input layer of each RC unit, where the length is set to be 128. The state transition matrix \mathbf{W}_s is generated randomly to satisfy the echo state property (Jaeger 2001), where we choose $\rho(\mathbf{W}_s) < 1$. The input weight \mathbf{W}_{in} is generated randomly from a uniform distribution. Meanwhile, we ignore the usage of the teacher-forcing in the state update, therefore, \mathbf{W}_{fb} is set as zero.

BER Performance

To evaluate symbol detection performance, we compare different RC based methods to conventional ones using BER. As discussed in Section , RF non-linear effects through PA non-linearity is incorporated in the performance evaluation. To be specific, the following RAPP model is adopted

$$f(x) = \frac{x}{\left[1 + \left(\frac{|x|}{x_{sat}}\right)^{2p}\right]^{1/2p}} \quad (12)$$

where x represents the amplitude of PA input signal, p is the smooth parameter and x_{sat} is the saturation level. When

$x \ll x_{sat}$, we have $f(x) \approx x$ which means the PA output signal has no distortion compared to the input. In our evaluation, we set $p = 3$ and $x_{sat} = 1$. As a benchmark, two standard symbol detection methods, linear minimum mean squared error (LMMSE) and sphere decoding (SD) (Ghasemmehdi and Agrell 2011), are conducted. These two methods are model-based approaches: LMMSE is a linear and low-complexity method that heavily relies on the prior knowledge of noise variance as well as channel statistics of the underlying wireless link. It is widely utilized in modern wireless systems due to the low-complexity. SD is a non-convex solver (a variant of the branch-bound method) which performs to the optimal maximum likelihood (ML) detection under ideal assumptions. The complexity of SD is much higher than that of LMMSE making it difficult to be used in modern systems. The CSI estimation method used for these two symbol detection methods is the LMMSE channel estimation assuming the transmission link is perfectly linear and the underlying noise variance is perfectly known.

We first consider the BER performance via transmitting signals through the linear region of PA. In this case, we need to back-off the PA input power away from the saturation region. To meet this goal, the input back-off (IBO) which is defined as the ratio between the PA's saturation power to the input power, is chosen to be greater than 8dB. From the result shown in the left sub-figure of Figure 4, we can see all RC based methods outperform conventional methods especially when the transmission power is low (denoted as E_b/N_0 — energy per bit over the noise). This is because the estimated CSI is inaccurate when E_b/N_0 is low resulting in a poor performance for conventional model-based methods. On the other hand, RC can learn the underlying features of the channel instead of the explicit CSI.

Next, we consider the BER performance when the PA input power is close to the saturation region. In this case, the PA output signal is distorted due to PA's compression effects. The distortion happens when the peak-to-average-power-ratio (PAPR) of the input signal is higher than the region IBO can offer, where the PAPR of an OFDM signal $x(t)$ is defined as $\|x(t)\|_\infty^2 / \|x(t)\|_2^2$. In our evaluation, the signal's PAPR is controlled in the range from 6dB to 9dB. Therefore, to investigate the BER performance under the effect of the compression from the PA, we test the BER performance by choosing the IBO below 6.5dB as shown in the right sub-figure of Figure 4. In this figure, we observe that the deep RC methods perform relatively well when the back-off is low, especially when IBO is lower than 5 dB.

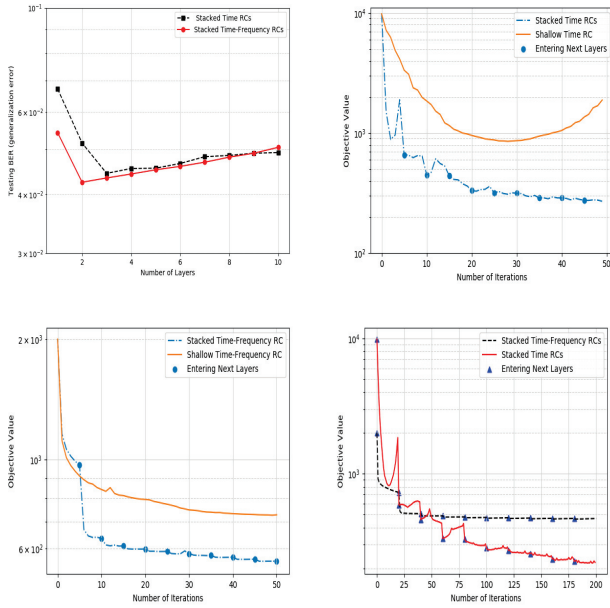


Figure 5: Learning curve of the shallow RC and deep RC.

More importantly, when we operate at a low IBO, the PA efficiency becomes high. Thus, RC-based methods offer alternative improvement for the PA efficiency by compensating for the waveform distortion at the receiver.

Learning Convergence

In this section, we will evaluate the convergence of the deep RC. To be specific, we set $E_b/N_0 = 15$ dB and train different RCs under the same channel realization. The objective for tracing the iterations is defined in (7).

Section demonstrates the BER performance of RC based methods when E_b/N_0 is low and under non-linear RF effects. In those evaluation we set $L = 3$ in deep RC. Intuitively, more RCs should yield better fitting performance. However, well-fitting in the training may bring over-fitting issues. As shown in the first sub-figure of Figure 5 that deep RCs do ensure a decreasing on the generalization error, however the issue related to over-fitting needs to be checked.

The second sub-figure of Figure 5 shows the learning curve of the time-domain RC. For the training set of each RC in the deep RC structure, we uniformly choose 5 delay parameters from 0 to N_{cp} . While for the shallow time-domain RC, we uniformly choose 50 delay parameters from 0 to N_{cp} . This means the resolution of the delay searching for the shallow RC is finer than that for the deep RC. From the figure we see that a finer delay searching space cannot lead to a lower objective value while adding extra RCs on top of a shallow RC can decrease the objective value.

A similar conclusion is also supported by observing the learning curve of stacked time-frequency RCs as depicted in the third sub-figure of Figure 5. Here we fix the number of iterations for the ALS in solving each RC layer to be 5. As shown in the figure, the objective value can be signifi-

cantly decreased by adding one extra RC. Compared with the shallow time-frequency RC, the fitting error of deep RC decreases much faster.

Furthermore, the comparison between the stacked time-domain RCs and stacked time-frequency RCs is presented in the fourth sub-figure of Figure 5. We see the stacked time-frequency RC can yield a stable output of the objective once increasing the number of RC layers does not decrease the generalization error/BER on the testing set (see the first sub-figure of Figure 5). Therefore, the stability on the learning curve of the stacked time-frequency RCs provides an efficient way to characterize a good L for the deep RC structure to avoid using extra data for validation.

Conclusion and Future Work

This paper presents two novel approaches toward extending the shallow RC structure to deep RCs. The associated learning algorithms are developed for each extension. Experimental results demonstrate the effectiveness of deepening RCs for the task of symbol detection in MIMO-OFDM systems. However, an important exploration for future research is to determine the optimal size of RCs in each layer for the task of MIMO-OFDM symbol detection. Connections to the boosting method may provide insights on designing the number of neurons in each layer. Another intriguing question for future work is how does this RC based receiving method can be combined with transmitter precoding to jointly optimize the link performance with limited feedback. The full potential of the deep RC based symbol detection method is yet to be explored. From theoretical perspective it would be meaningful to analyze the functionality of each layer for interference cancellation.

Acknowledgement

This work was supported by the U.S. National Science Foundation under grants ECCS-1802710, ECCS-1811497, CNS-1811720, and CCF-1937487.

References

- 3GPP TS 38.211. 2019. NR; Physical channels and modulation.
- Cerwall, P.; Jonsson, P.; Torres, A.; Lindberg, P.; Öhman, K.; and Karapantelakis, A. 2019. Mobile subscriptions outlook. *Ericsson mobility report*.
- El Hihi, S., and Bengio, Y. 1996. Hierarchical recurrent neural networks for long-term dependencies. In *Advances in neural information processing systems*, 493–499.
- Freund, Y. 1995. Boosting a weak learning algorithm by majority. *Information and computation* 121(2):256–285.
- Funahashi, K., and Nakamura, Y. 1993. Approximation of dynamical systems by continuous time recurrent neural networks. *Neural Networks* 6(6):801 – 806.
- Galicchio, C., and Micheli, A. 2017. Echo state property of deep reservoir computing networks. *Cognitive Computation* 9(3):337–350.

- Gallicchio, C.; Micheli, A.; and Pedrelli, L. 2017. Deep reservoir computing: a critical experimental analysis. *Neurocomputing* 268:87–99.
- Ghasemmehdi, A., and Agrell, E. 2011. Faster recursions in sphere decoding. *IEEE Transactions on Information Theory* 57(6):3530–3536.
- Holzmann, G., and Hauser, H. 2010. Echo state networks with filter neurons and a delay&sum readout. *Neural Networks* 23(2):244–256.
- Jaeger, H. 2001. The “echo state” approach to analysing and training recurrent neural networks—with an erratum note. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report* 148(34):13.
- Jaeger, H. 2007. Discovering multiscale dynamical features with hierarchical echo state networks. Technical report, Jacobs University Bremen.
- Joung, J.; Ho, C. K.; Adachi, K.; and Sun, S. 2014. A survey on power-amplifier-centric techniques for spectrum-and energy-efficient wireless communications. *IEEE Communications Surveys & Tutorials* 17(1):315–333.
- Khani, M.; Alizadeh, M.; Hoydis, J.; and Fleming, P. 2019. Adaptive neural signal detection for massive mimo. *arXiv preprint arXiv:1906.04610*.
- Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 1097–1105.
- Liu, L.; Chen, R.; Geirhofer, S.; Sayana, K.; Shi, Z.; and Zhou, Y. 2012. Downlink MIMO in LTE-Advanced: SU-MIMO vs. MU-MIMO. *IEEE Commun. Mag.* 50(2):140–147.
- Lukoševičius, M., and Jaeger, H. 2009. Reservoir computing approaches to recurrent neural network training. *Computer Science Review* 3(3):127–149.
- Meinilä, J.; Kyösti, P.; Jämsä, T.; and Hentilä, L. 2009. Winner II channel models. *Radio Technologies and Concepts for IMT-Advanced* 39–92.
- Mosleh, S.; Sahin, C.; Liu, L.; Zheng, R.; and Yi, Y. 2016. An energy efficient decoding scheme for nonlinear mimo-ofdm network using reservoir computing. In *2016 International Joint Conference on Neural Networks (IJCNN)*, 1166–1173. IEEE.
- Mosleh, S.; Liu, L.; Sahin, C.; Zheng, Y. R.; and Yi, Y. 2018. Brain-inspired wireless communications: Where reservoir computing meets MIMO-OFDM. *IEEE Trans. Neural Nets. & Learning Syst.* 29(10):4694–4708.
- Murara, B. 2017. IMT-2020 Network High Level Requirements, How African Countries can cope. *ITU*.
- Pascanu, R.; Gulcehre, C.; Cho, K.; and Bengio, Y. 2014. How to construct deep recurrent neural networks. In *Proceedings of the Second International Conference on Learning Representations (ICLR 2014)*.
- Pascanu, R.; Mikolov, T.; and Bengio, Y. 2013. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, 1310–1318.
- Patnaik, A.; Anagnostou, D. E.; Mishra, R. K.; Lyke, J.; et al. 2004. Applications of neural networks in wireless communications. *IEEE Antennas and Propagation Magazine* 46(3):130–137.
- Samuel, N.; Diskin, T.; and Wiesel, A. 2017. Deep MIMO detection. In *2017 IEEE 18th Intl Workshop on Signal Process. Advances in Wireless Commun. (SPAWC)*, 1–5.
- Schiller, U. D., and Steil, J. J. 2005. Analyzing the weight dynamics of recurrent learning algorithms. *Neurocomputing* 63:5–23.
- Shafin, R.; Liu, L.; Chandrasekhar, V.; Chen, H.; Reed, J.; and Zhang, J. 2020. Artificial intelligence-enabled cellular networks: A critical path to beyond-5G and 6G. *IEEE Wireless Commun.*
- Zhou, Z.; Liu, L.; and Chang, H.-H. 2019. Learning for detection: MIMO-OFDM symbol detection through downlink pilots. *arXiv:1907.01516*.
- Zhu, W.; Lan, C.; Xing, J.; Zeng, W.; Li, Y.; Shen, L.; and Xie, X. 2016. Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks. In *Thirtieth AAAI Conference on Artificial Intelligence*.