Machine Learning-Based Receiver in Multiple Input Multiple Output Communications Systems

Zhou Zhou

Dissertation submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Electrical Engineering

Lingjia Liu, Chair

Richard M. Buehrer

Steven W. Ellingson

Yang Yi

Weijun Xie

June, 21, 2021

Blacksburg, Virginia

Keywords: Machine Learning, Neural Network, MIMO-OFDM, Symbol Detection, Channel Equalization, OTFS

Machine Learning-Based Receiver in Multiple Input Multiple Output Communications Systems

Zhou Zhou

(ABSTRACT)

Bridging machine learning technologies to multiple-input-multiple-output (MIMO) communications systems is a primary driving force for next-generation wireless systems. This dissertation introduces a variety of neural network structures for symbol detection/equalization tasks in MIMO systems configured with two different waveforms, orthogonal frequency-division multiplexing (OFDM) and orthogonal time frequency and space (OTFS). The former one is the major air interface in current cellular systems. The latter one is developed to handle high mobility. For the sake of real-time processing, the introduced neural network structures are incorporated with inductive biases of wireless communications signals and operate in an online training manner. The utilized inductive priors include the shifting invariant property of quadrature amplitude modulation, the time-frequency relation inherent in OFDM signals, the multi-mode feature of massive antennas, and the delay-Doppler representation of doubly selective channel. In addition, the neural network structures are rooted in reservoir computing - an efficient neural network computational framework with decent generalization performance for limited training datasets. Therefore, the resulting neural network structures can learn beyond observation and offer decent transmission reliability in the low signal-to-noise ratio (SNR) regime. This dissertation includes comprehensive simulation results to justify the effectiveness of the introduced NN architectures compared with conventional model-based approaches and alternative neural network structures.

Machine Learning-Based Receiver in Multiple Input Multiple Output Communications Systems

Zhou Zhou

(GENERAL AUDIENCE ABSTRACT)

An important topic for next-generation wireless systems is the integration of machine learning technologies with conventional communications systems. This dissertation introduces several neural network architectures to solve the transmission problems in wireless communications systems. The discussion focuses on the following major modern communications technologies: multiple-input-multiple-output (MIMO), orthogonal frequency-division multiplexing (OFDM), and orthogonal time frequency space (OTFS). In today's cellular networks, MIMO and OFDM are the major air-interface. OTFS is a novel technique that has been designed to work in a high-mobility setting. The implemented neural network structures are integrated with inductive biases of wireless communications signals and operate in an online training mode with limited training datasets. The neural network architectures, in particular, are based on reservoir computing, which is an efficient neural network computational system. A learning algorithm's inductive bias (also known as learning bias) is a collection of assumptions that the learner makes to infer outputs from unknown inputs. The dissertation introduces four different inductive priors from four different perspectives of MIMO communications systems. As a result, the neural network architectures can learn beyond observation and provide good generalization output in scenarios having model mismatch issues. The dissertation provides extensive simulation results to support the efficacy of the implemented NN architectures compared to alternative neural network models and traditional model-based approaches.

Contents

List of Figures

List of Tables

List of Abbreviations

4G	The Fourth Generation Cellular Network Technology
5G	The Fifth Generation Cellular Network Technology
AI	Artificial Intelligence
ANN	Artificial Neural Network
BER	Bit Error Rate
BPTT	Backpropagation Through Time
BS	Base Station
CP	Cyclic-Prefix
CSI	Channel State Information
DMRS	Demodulation Reference Signal
DoF	Degree of Freedom
DPD	Digital Pre-Distortion
FLOPS	Floating Point Operations per Second
LMMSE	Linear Minimum Mean Square Error
LTE	Long Term Evolution
MIMO	Multiple-Input and Multiple-Output

ML	Maximum Likelihood
MS	Mobile Station
NN	Neural Network
NR	New Radio
OFDM	Orthogonal Frequency-Division Multiplexing
OTFS	Orthogonal Time Frequency and Space
PAPR	Peak Average Power Ratio
PHY	Physical Layer
RB	Resource Block
RC	Reservoir Computing
RE	Resource Element
RNN	Recurrent Neural Network
Rx	Receiver
SD	Sphere Decoding
SISO	Single-Input and Single-Output
SNR	Signal to Noise Ratio
STM	Short Term Memory
Tx	Transmitter

Chapter 1

Introduction

Multiple-input-multiple-output (MIMO) is an essential radio access technology in current communications systems, such as 4G/5G cellular networks [2, 5]. A system with MIMO refers to its transmitter and receiver having multiple antennas. Moreover, MIMO systems transmit and receive multiple signal streams simultaneously over the same physical radio channel. By adopting appropriate transmitting and receiving processing techniques, the system's achievable rate/reliability can be improved as the number of antennas grows. Accordingly, operational modes of MIMO systems can be generally categorized as spatial multiplexing and spatial diversity coding, where the former scheme refers to transceiving multiple independent information streams via multiple wireless signal streams. In contrast, the latter one refers to a coded single information stream transceived by multiple signal streams. [95].

This dissertation develops receiving techniques for MIMO systems operating in the spatial multiplexing mode. In conventional processing frameworks of spatial multiplexing, channel state information (CSI) is often required at the transmitter or receiver to pre-eliminate or post-cancel out spatial interference, where the interference is caused by concurrent signal transceiving [76]. Accordingly, pilot symbols are associated with data symbols in the signal frames for CSI estimation. However, perfect CSI is impossible to acquire due to practical constraints: For instance, in the presence of nonlinearities in RF components, it is intractable to model the RF features precisely; with limited transmission power, the channel estimation has to operate under a low signal-to-noise ratio (SNR) regime. A consequence of using imperfect CSI is performance deterioration of link-level transmission. In addition, wireless communications systems are with modulation components converting information symbols to waveforms. For the sake of developing a more comprehensive receiver, this dissertation considers the primary air interface in current communications systems, orthogonal frequency division multiplexing (OFDM)[73] and one of its generalization – orthogonal time frequency and space (OTFS). More importantly, this dissertation utilizes a supervised-learning-based framework to train the receiver with compatible pilot overhead to current wireless communications systems. Signal features of adopted air-interface signals are incorporated into the neural networks (NNs) as inductive bias. Therefore, the resulting NN structures can offer desirable link-level performance using limited training datasets. The following sections in this chapter will discuss the motivations and challenges of using NNs as the receiver in MIMO systems.

1.1 MIMO Systems with Modulation Techniques

Modulation techniques are defined as converting/reconverting information symbols to carrier signals in communications systems. This dissertation considers using multi-carrier modulation as the communication air interface. By mounting up multiple antennas at the transceiver chain, communications systems encounter miscellaneous interference to handle, such as inter-spatial streams interference, inter-samples interference, and inter-carriers interference. The receiver is to mitigate the interference and reconstruct the transmitted symbols with low error probability.

1.1.1 OFDM

OFDM is a technique that modulates information symbols on multiple closely-spaced orthogonal sub-carriers. The spectra between two neighboring sub-carriers are overlapping to each other to maximize spectral efficiency. By leveraging fast Fourier transformation (FFT) and adding cyclic prefix (CP), the equalization of OFDM signals can be easily conducted in the frequency domain with significantly reduced complexity. Regarding MIMO systems with OFDM modulation, the spatial interference can also be handled via a sub-carrier basis. However, the interference cancellation at each sub-carrier is a very critical stage to achieve the link capacity.

The receiving technologies of MIMO systems can be sub-divided into coherent symbol detection and non-coherent symbol detection based on whether using instantaneous CSI. In the coherent detection, CSI is first obtained through pilot symbols allocated on the frequency domain. In the next step, a MIMO symbol detection stage is conducted by solving an inverse problem, where the solver relies on estimated CSI as the algorithm parameters. Conventional approaches are like minimum mean square error (MMSE) and maximum likelihood estimation (MLE). In non-coherent detection, an extra coding strategy is required, resulting in high computational complexity. Therefore, this dissertation considers coherent symbol detection for the sake of low costs in computation. However, OFDM systems have the following two disadvantages, which end up with a low receiving SNR environment:

- OFDM is sensitive to channel Doppler effects. When the system operates in a high mobility environment, the resulting Doppler shift deconstructs the orthogonality between sub-carriers. Thus, received signals are disturbed inter-carrier interference.
- The OFDM signals have a high value of peak to average ratio (PAPR). In practice, OFDM signals are often transmitted through the linear region of RF circuits to avoid

waveform distortion. However, ensuring hardware with linear transition features leads to poor energy efficiency. A trade-off solution is to set a small back-off region to the RF nonlinear region while handling the distortion using algorithms. In scenarios without pre-distortion, the received OFDM signals equivalently reside in a low SNR regime interfered by nonlinear noise.

To address the above issues, this dissertation is motivated to design robust receivers to handle the imperfectness.

1.1.2 OTFS

Since OFDM is sensitive to Doppler shift, an alternative waveform, orthogonal time-frequency space (OTFS), is introduced to tackle the Doppler effects. As a generalization to OFDM, OTFS utilizes 2-D time-frequency basis functions to spread information symbols in the entire time-frequency domain. Theoretically, it can offer full diversity gains in the time-frequency domain. In OTFS modulation, the doubly selective channel is simply approximated as a constant convolution kernel in the delay-Doppler domain. Thus, the receiving process in OTFS is to reverse the channel forward convolution. However, the performance of OTFS modulation exacerbates as well when model mismatch appears. For instance, the ideal channel kernel is often assumed to be strictly sparse. This hypothesis is leveraged as prior information to reduce the channel acquisition overhead and conduct low complexity symbol detection. However, practical systems utilize non-ideal transceiving impulse response filters. Therefore, the channel is no longer strictly sparse, which inevitably increases the receiver complexity rendering diminished transmission reliability for sparse prior-based receivers. In addition, the applications with a large number of sub-carriers and antennas make the problem further complicated and thereby fail to satisfy latency requirements in communications systems. [72, 102].

1.2 Neural Networks Enabled Receiver

Deep neural networks (NNs) have been applied to a wide range of scientific problems and achieved notable success [53]. Motivated by this driving force, neural network-enabled wireless communications systems become an important component for 5G-Advanced systems. Regarding recent research in AI to wireless communications systems, a wide range of directions of using AI to boost the design in wireless communications systems has been proposed. For instance,[12, 13, 22, 70] introduce an auto-encoder based framework to rebuild wireless communications systems. To this end, the system design is altered as learning the communications components via an end-to-end manner. In [24], recurrent neural network (RNN) is employed for a sequence equalization task. In addition, [25] introduces training an end-to-end communication system up to application layers, where a deep neural network-aided joint source and channel coding-decoding strategy is applied for the transmission of semantic information. Recent work in [17, 45, 80, 94] shows the effectiveness of using NNs for symbol detection with different channel conditions. [67] utilizes a neural network for channel decoding to reduce the complexity significantly. Another example of using deep neural networks for channel coding is from [48]. It adopts a neural network as an encoder for a Gaussian noise feedback channel. Furthermore, [87] uses a neural network to learn a power allocation strategy in interference channels. Other examples of using neural networks to solve wireless communications problems are presented in [8, 9, 16, 44, 71].

The power of AI is rooted in its generalization ability to perform inference beyond observation. One way to design a neural network-based symbol detector is unfolding an iteration algorithm to a deep neural network while adding free parameters to learn, such as a projected gradient descent-based neural network structure introduced in [81]. However, this framework often requires assigning explicit CSI as the neural network coefficients. The resulting neural network structure yields poor generalization performance when model mismatch appears. Therefore, it is imperative to find an alternative way to design symbol detection neural networks without using explicit CSI. On the other hand, directly applying generic NNs often results in poor performance due to the lack of training datasets in different transmission scenarios. Moreover, the convergence of generic neural networks through gradient descent-based algorithms can be very slow [37]. Therefore, to realize the full potential of NNs for symbol detection/equalization tasks, the following underfitting and overfitting issues have to be addressed for the sake of a balanced fitted neural network model:

- **Overfitting**:Training a symbol detection/equalization NN based on over-the-air feedback is prohibitively expensive. The infeasibility is because using gradient-based backpropagation across the wireless channel violates the control overhead defined in communications systems. Furthermore, training NNs compatible with physical layer (PHY) standards has to solve the limited training data issue. For example, the demodulation reference signals in 5G NR cellular systems with 2×2 multiplexing operation occupy approximately 15% of the bandwidth. In addition, cellular systems are anticipated to adapt the transmission modes on a millisecond basis. Therefore, it is desirable to have NN-based processing relying on online learning merely. Accordingly, overfitting is easy to happen due to the lack of an asymptotic large training dataset [92].

- **Underfitting**: RNNs are primarily designed to process sequential data. Wireless communications signals are often with structural features that can be analyzed via a transformation domain. For instance, OFDM signals are interleaved with time and frequency patterns. Simply applying generic RNNs to communications signals may face a learning floor of the loss function due to underfitting. Therefore, it requires either a more sophisticated NN model or a training dataset with good features to improve the learning efficiency.

1.3 Contributions and Attribution

This dissertation presents various neural network structures for the symbol detection/equalization tasks to optimize link-level reliability in MIMO systems. The introduced NN structures utilize reservoir computing (RC), a particular recurrent neural network (RNN), as the basic building blocks. The motivation of using RC is to exploit the causality inside the received communications signals presented by a sequential form. Meanwhile, RNNs can universally approximate dynamic systems [40]. According to the appealing generalization ability of RC, the following introduced neural network structures can train their coefficients using limited data. Beyond directly applying RC to the symbol detection task, the dissertation focuses on adding inductive bias of wireless signals to the NN architecture to circumvent the risks of a large model variance (overfitting). This combination implies the feasibility of training neural network-based symbol detectors/equalizers via online manners, thus obviating offline training stages. Accordingly, the NNs can immediately generate modulation symbols after the one-shot training is completed. Therefore, the aforementioned overfitting issues from the limited training dataset are addressed.

On the other hand, the underfitting issues are to be addressed by the following approaches: Employing a good pilot structure (good training data); Increasing the neural network complexity; Improving the convergence of learning algorithms. This dissertation will introduce the following structures to achieve a good tradeoff between the overfitting and underfitting based on their individual MIMO operation scenarios: (according to the order will be presented)

- A symmetric and decomposed neural network architecture is introduced in Chapter 2 for the symbol detection task in a MIMO-OFDM system. The neural network structure is motivated by the shifting invariant property of quadrature amplitude modulation (QAM). In detail, an M-ary modulation can be decomposed as a series of $M-1$ binary detection neural networks. In addition, the NN structure is integrated with RC to extract signals' time-domain features. The evaluation shows the introduced approach performs closely to the maximum likelihood estimation with imperfect channel state information (CSI) in the low SNR regime.

- A deep recurrent neural network structure for MIMO-OFDM system detection is presented in Chapter 3, namely, RCNet. It is incorporated with the time-frequency structure of OFDM signals. The structure-wise extension to standard RCs includes adding frequency processing layers after its time-domain output and stacking multiple basic RC units into deep versions. Experiments show that the added time-frequency inductive bias and deep architecture can provide more robust symbol detection performance than other NNs in the low SNR regime and PA's non-linear regime.

- As massive antennas are deployed at BSs, the multi-mode feature of the wireless channel (such as the elevation, azimuth directions) can be leveraged as an inductive bias for the neural network design. To this end, a tensorized neural network structure (multi-mode RC) is introduced in Chapter 4 for the uplink symbol detection task, where it utilizes tensor as the underlying data format for receiving signal processing. In addition, a theoretical condition on setting up an under-parameterized multi-mode RC is provided. The theory can guide the parameter configuration of NNs to avoid over-fitting issues sufficiently. Evaluation results reveal that the introduced framework can offer good generalization performance in the uplink transmission in massive MIMO systems.

- An RC-based equalization framework for an OTFS system is introduced in Chapter 5, where OTFS is an alternative waveform to OFDM in high mobility environments. OTFS operates the modulation/demodulation on a two-dimensional plane, the delay-Doppler domain, which goes beyond the frequency domain-based processing in OFDM. Along with particularly designed pilot patterns, the training of equalization neural network is exclusively conducted at each OTFS frame, where the pilot overhead is consistent with its definition in current cellular systems. The simulation results show that the introduced approach can offer more robust performance in different velocity settings. In addition, compared with its OFDM counterpart, OTFS requires less complexity to achieve the same link transmission reliability.

The implementations of the above neural networks are respectively presented from Chapter 2 to Chapter 5. Moreover, these Chapters are reproduced from the author's following papers,

1. ***Zhou Zhou****; Shashank Jere; Lizhong Zheng; et al. "Learning with Knowledge of Structure: A Neural Network-Based Approach for MIMO-OFDM Detection," in 2020 IEEE Asilomar Conference on Signals, Systems, and Computers.*

2. ***Zhou Zhou****; Shashank Jere; Lizhong Zheng; et al. "Learning for Integer-Constrained Optimization through Neural Networks with Limited Training," in NeurIPS 2020 Workshop on Learning Meets Combinatorial Algorithms.*

3. ***Zhou Zhou****; Lingjia Liu, Vikram Chandrasekhar; et al. "Deep Reservoir Computing Meets 5G MIMO-OFDM Systems in Symbol Detection," in AAAI-2020 Thirty-Fourth AAAI Conference on Artificial Intelligence.*

4. ***Zhou Zhou****; Lingjia Liu; Shashank Jere; et al. "RCNet: Incorporating Structural Information into Deep RNN for Online MIMO-OFDM Symbol Detection with Limited Training," in IEEE Transactions on Wireless Communications, 2021.*

5. ***Zhou Zhou****; Lingjia Liu; Jiarui Xu; et al. "Harnessing Tensor Structures – Multi-Mode Reservoir Computing and Its Application in Massive MIMO," in arXiv Preprint: 2102.09322.*

6. ***Zhou Zhou****; Lingjia Liu; Robert Calderbank; et al. "Learning to Equalize OTFS," in arXiv Preprint 2107.08236.*

where papers (1) and (2) are with respect to Chapter 2; papers (3) and (4) are reorganized as Chapter 3; paper (5) is presented in Chapter 4, and paper (6) is for in Chapter 5.

Chapter 2

MIMO-OFDM Symbol Detection via Shifting-RC

2.1 Introduction

MIMO-OFDM is the primary air interface in modern communication standards such as 4G (LTE-Advanced) and 5G New Radio (NR) [73]. As opposed to single-input single-output systems (SISO), MIMO-OFDM systems have symbols interfering in space and time due to the concurrent reception of signals at the receive antennas and the multipath propagation feature of wireless channels. In order to achieve the transmission (capacity) gains that MIMO offers, the choice of the symbol detection strategy plays a critical role. However, the following practical issues are present in conventional model-based symbol detection strategies: (1) Model Uncertainty - The exact characterization of the true end-to-end link from the transmitter to the receiver cannot be obtained. Instead, only a linear and static approximation of the wireless channel and a coarse approximation to the hardware imperfections in the transceiver are available. For e.g., Non-linearities in radio frequency (RF) circuits such as the power amplifier (PA) at the transmitter and finite quantization in baseband circuits such as analog-to-digital converter (ADC) at the receiver are hard to characterize accurately. Additionally, channel state information (CSI) is not directly accessible and needs to be estimated at the receiver using the received signal. Algorithms can thus only use the estimated

CSI, which yields diminishing detection success rate as the accuracy of the estimated CSI decreases, particularly in the low signal-to-noise ratio (SNR) regime [11]. (2) Implementation Complexity - The computational complexity of optimal detection algorithms have an exponential relationship with the number of variables to be detected. Therefore, applications with large number of OFDM sub-carriers, high modulation orders and large number of transmit/receive antennas render the problem non-deterministic polynomial-time hard (NP-hard), and thus unable to meet latency requirements [72].

The recent advent of neural networks (NNs) has led to the application of learning-based approaches to a wide array of scientific problems [53]. Specifically, the concept of artificial intelligence (AI)-enabled cellular networks has been introduced [85]. Such successful NN-based approaches motivate us to integrate NN strategies in symbol detection algorithms in order to mitigate the aforementioned issues from model-based approaches. However, this seemingly simple rationale is in fact extremely challenging due to the inherent constraints of wireless systems.

- *Data Deficit*: For the purpose of training, we can only access a limited training dataset sent over-the-air in the physical layer (PHY). For instance, in 3GPP LTE/LTE-Advanced and 5G systems the demodulation reference signals in a 2×2 MIMO-OFDM system occupy approximately 10% of the system resources in terms of resource elements (REs), whereas the remaining 90% of REs are occupied by data symbols. Accordingly, the variance-bias tradeoff is insufficient to be guaranteed without an asymptotic large training dataset [92].

- *Learning Deficit*: Generic NNs are often slow to configure due to the large number of hyper-parameters that need to be optimized. Gradient descent-based learning can be slow to converge or may converge towards a solution with poor generalization performance [37].

To address the challenges of applying neural networks to symbol detection tasks (as discussed in the introduction chapter), we consider incorporating inductive bias, i.e., prior knowledge, into the NN architecture by leveraging the structural properties of the MIMO-OFDM signal. This helps the network make inferences that go beyond the observed data, thus potentially solving the fundamental data deficit issue. The introduced NN structure leverages the following structural information of the MIMO-OFDM signal:

- Constellation of the digital modulation scheme.
- Time-domain convolution and superposition operation by the wireless channel.
- The presence of different frequencies inherent in the OFDM signal.

In the introduced NN, the constellation structure is reflected through a process which can generalize a binary detector to a multiple-class detector. The time-domain knowledge is captured through the dynamics of reservoir computing (RC), a special category of recurrent neural networks (RNNs) [63, 100, 101]. The frequency information is inherent in the weight initialization of the output layer of RC. Our evaluation results show that the introduced method performs close to maximum likelihood (ML) estimation in the low SNR regime while using imperfect (estimated) CSI.

The organization of this paper is as follows. In Sec. 2.2, we give an overview of the MIMO-OFDM symbol detection problem. In Sec. 2.3, we describe the technical details of our introduced method. Sec. 2.4 evaluates the performance of our approach in comparison to conventional symbol detection methods as well as less other general neural network structures. Finally, we conclude the paper in Sec. 5.6.

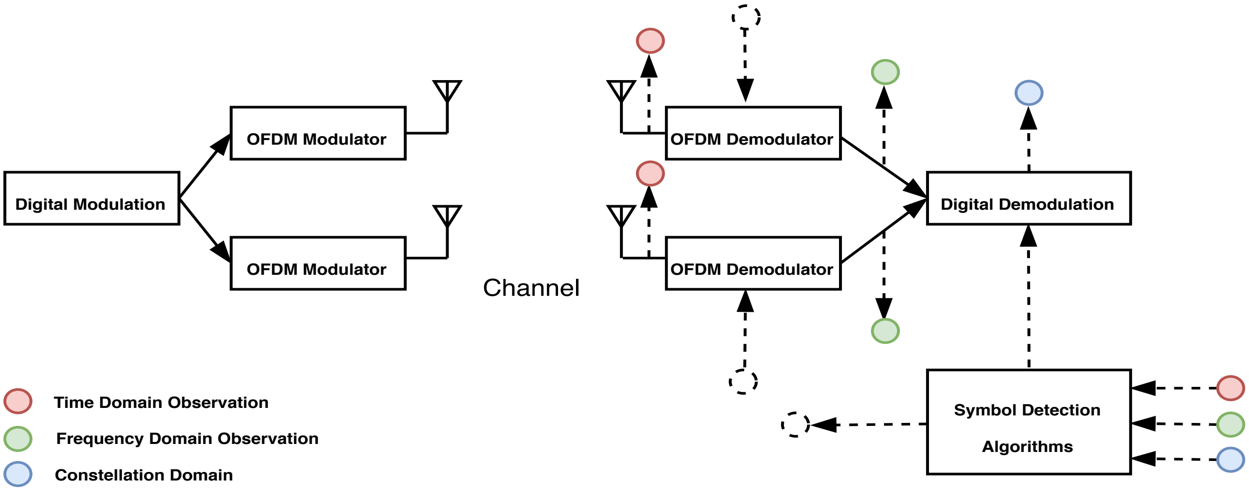


Figure 2.1: MIMO OFDM System

2.2 Problem Statement

2.2.1 MIMO-OFDM System

Let N_t and N_r represent the number of transmit and receive antennas respectively; let N_c represent the number of sub-carriers. Without loss of generality, we assume that the modulation scheme used is 16-QAM which is denoted by its alphabet as $\mathcal{A} = \{-3, -1, +1, +3\} \times \{-3j, -1j, +1j, +3j\}$. Let $\boldsymbol{X} \in \mathcal{A}^{N_t \times N_c}$ be the transmitted MIMO-OFDM symbol in the space-frequency domain, the received signal becomes

$$\boldsymbol{Y} = \sum_{\ell=0}^{L-1} \boldsymbol{H}[\ell]\boldsymbol{X}\boldsymbol{F}\boldsymbol{J}_\ell + \boldsymbol{N} \tag{2.1}$$

where $\boldsymbol{F} \in \mathbb{C}^{N_c \times N_c}$ is the inverse discrete Fourier transform (IDFT) matrix, $\boldsymbol{J}_\ell \in \mathbb{C}^{N_c \times N_c}$ is a cyclic permutation matrix with ℓ, $\boldsymbol{N}$ is the matrix of additive white Gaussian noise (AWGN), L is the number of multipath components of the wireless channel. For ease of discussion, the Cyclic Prefix (CP) is removed in the model.

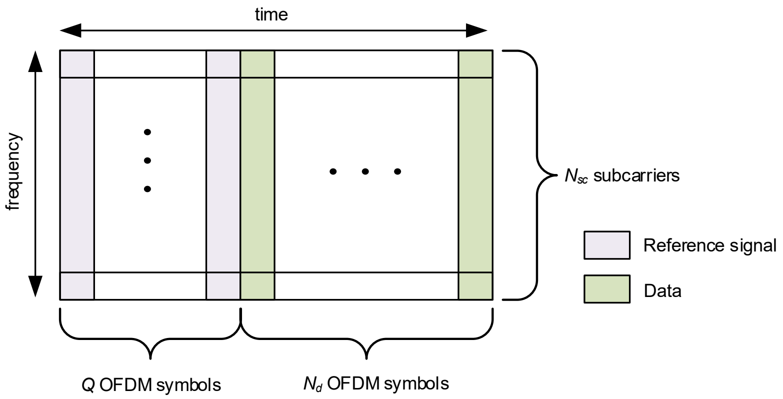


Figure 2.2: Pilot and data structure

2.2.2 Symbol Detection

The symbol detection task is to design an inverse map which infers $\boldsymbol{X}$ from the observation $\boldsymbol{Y}$. For simplicity, we consider using the real-valued version of $\boldsymbol{Y}$ and $\boldsymbol{X}$ in the inference task, which are respectively defined as

$$\tilde{\boldsymbol{Y}} := \begin{bmatrix} \mathrm{Re}(\boldsymbol{Y}) \\ \mathrm{Im}(\boldsymbol{Y}) \end{bmatrix} \text{ and, } \tilde{\boldsymbol{X}} := \begin{bmatrix} \mathrm{Re}(\boldsymbol{X}) \\ \mathrm{Im}(\boldsymbol{X}) \end{bmatrix}$$

where $\tilde{\boldsymbol{X}}$ and $\tilde{\boldsymbol{Y}}$ now represent 4-PAM symbols, the definition of the real-valued equivalent receive signal model to (2.1) can be found in the Appendix. The pilot and data structure of an OFDM subframe is shown in Fig. 2.2. In the receive processing, reference signals/pilots are used to train the NN of the RC for the learning-based symbol detection strategy while the remaining data symbols will go through the trained NN for symbol detection.

2.3 Introduced Approach

By casting the symbol detection problem in a Bayesian framework, it can be formulated as the following maximum likelihood estimation problem,

$$\arg\max_{\tilde{\boldsymbol{X}}} P(\tilde{\boldsymbol{X}}|\tilde{\boldsymbol{Y}}). \tag{2.2}$$

where $P(\cdot|\cdot)$ represents the joint posterior distribution. The joint posterior distribution can then be approximated via the naive Bayesian principle, i.e.,

$$P(\tilde{\boldsymbol{X}}|\tilde{\boldsymbol{Y}}) \sim \prod_{n_t \in 2N_t, n_c \in N_c} P_{n_t,n_c}(\tilde{x}_{n_t,n_c}|\tilde{\boldsymbol{Y}}) \tag{2.3}$$

where $P_{n_t,n_c}(\cdot|\tilde{\boldsymbol{Y}})$ stands for the marginal distribution of the $(n_t, n_c)^{\text{th}}$ entry of $\tilde{\boldsymbol{X}}$. Accordingly, we train NNs to approximate $P_{n_t,n_c}(\cdot|\tilde{\boldsymbol{Y}})$, i.e.,

$$f_{n_t,n_c}(\tilde{x}_{n_t,n_c}; \tilde{\boldsymbol{Y}}) = P(\tilde{x}_{n_t,n_c}|\tilde{\boldsymbol{Y}}) \tag{2.4}$$

where $f_{n_t,n_c}(\tilde{x}_{n_t,n_c}; \tilde{\boldsymbol{Y}})$ denotes the NN with $\tilde{x} \in \{-3, -1, +1, +3\}$ as the output and $\tilde{\boldsymbol{Y}}$ as the input. In the final step, the transmitted symbols can be estimated by choosing the maximum a posterior estimation,

$$\arg\max_{\tilde{x}_{n_t,n_c}} f_{n_t,n_c}(\tilde{x}_{n_t,n_c}; \tilde{\boldsymbol{Y}}). \tag{2.5}$$

Therefore, the symbol estimation task is completed by collecting the NN outputs corresponding to all the entries of $\tilde{\boldsymbol{X}}$.

2.3.1 Detection on a Binary Set

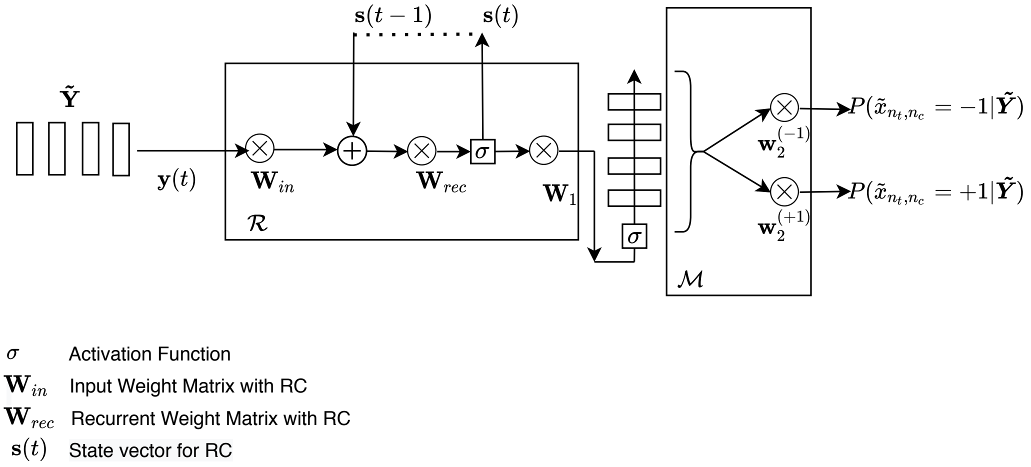

$\mathbf{s}(t-1)$
$\mathbf{s}(t)$
$\tilde{\mathbf{Y}}$
$\mathbf{y}(t)$
$\mathbf{W}_{in}$
$\mathbf{W}_{rec}$
$\mathbf{W}_1$
$\mathcal{R}$
σ
$\mathbf{w}_2^{(-1)}$
$\mathbf{w}_2^{(+1)}$
$P(\tilde{x}_{n_t,n_c} = -1|\tilde{\boldsymbol{Y}})$
$P(\tilde{x}_{n_t,n_c} = +1|\tilde{\boldsymbol{Y}})$
$\mathcal{M}$
σ Activation Function
$\mathbf{W}_{in}$ Input Weight Matrix with RC
$\mathbf{W}_{rec}$ Recurrent Weight Matrix with RC
$\mathbf{s}(t)$ State vector for RC

Figure 2.3: Binary detector $f_{n_t,n_c}(\tilde{x}_{n_t,n_c}; \tilde{\boldsymbol{Y}})$

For simplicity, we first consider the training of a binary decision NN for $\tilde{x}_{n_t,n_c}$, i.e., the NN only has two outputs. The underlying binary detector can be seen more clearly in Fig. 2.3. To be specific, the output ratio of this NN is an estimation of the marginal likelihood ratio of

$$\frac{P(\tilde{x}_{n_t,n_c} = +1|\tilde{\boldsymbol{Y}})}{P(\tilde{x}_{n_t,n_c} = -1|\tilde{\boldsymbol{Y}})} = \frac{f_{n_t,n_c}(\tilde{x}_{n_t,n_c} = +1; \tilde{\boldsymbol{Y}})}{f_{n_t,n_c}(\tilde{x}_{n_t,n_c} = -1; \tilde{\boldsymbol{Y}})}$$

For ease of discussion, we introduce $\mathcal{L}_{+-}$ as a notation of the likelihood ratio, i.e.,

$$\mathcal{L}_{+-}(\tilde{\boldsymbol{Y}}) := \frac{f_{n_t,n_c}(\tilde{x}_{n_t,n_c} = +1; \tilde{\boldsymbol{Y}})}{f_{n_t,n_c}(\tilde{x}_{n_t,n_c} = -1; \tilde{\boldsymbol{Y}})} \tag{2.6}$$

To capture the time domain features of MIMO-OFDM signals, we design the NN as

$$f_{n_t,n_c}(\tilde{x}_{n_t,n_c}; \tilde{\boldsymbol{Y}}) := \mathcal{M}_{\boldsymbol{W}_2}(\mathcal{R}_{\boldsymbol{W}_1}(\tilde{\boldsymbol{Y}})) \tag{2.7}$$

where $\mathcal{M}$ is a single-layer perceptron (feedforward NN) with a soft-max output, the coeffi-

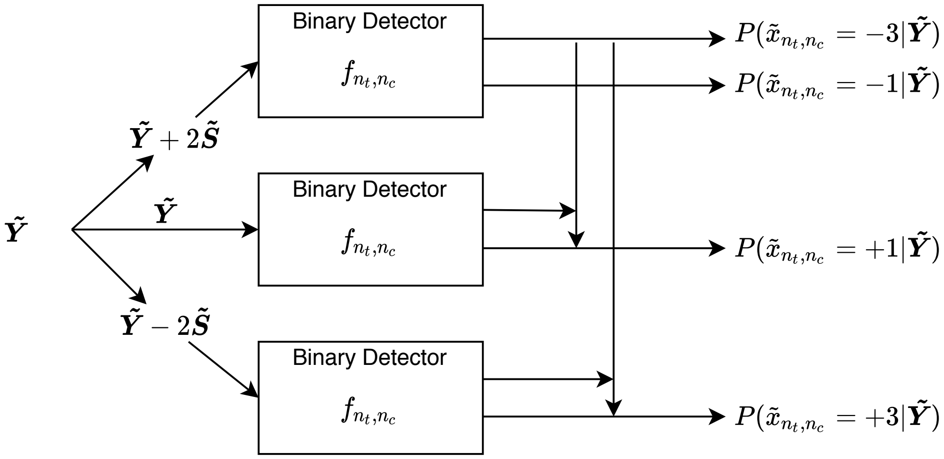

Binary Detector
f_{n_t,n_c}
Binary Detector
f_{n_t,n_c}
Binary Detector
f_{n_t,n_c}
$\tilde{\boldsymbol{Y}}$
$\tilde{\boldsymbol{Y}} + 2\tilde{\boldsymbol{S}}$
$\tilde{\boldsymbol{Y}}$
$\tilde{\boldsymbol{Y}} - 2\tilde{\boldsymbol{S}}$
$P(\tilde{x}_{n_t,n_c} = -3|\tilde{\boldsymbol{Y}})$
$P(\tilde{x}_{n_t,n_c} = -1|\tilde{\boldsymbol{Y}})$
$P(\tilde{x}_{n_t,n_c} = +1|\tilde{\boldsymbol{Y}})$
$P(\tilde{x}_{n_t,n_c} = +3|\tilde{\boldsymbol{Y}})$

Figure 2.4: Multiple-class detector generated by binary detector.

cients to learn are denoted as $\boldsymbol{W}_2 := [\boldsymbol{w}_2^{(-1)}, \boldsymbol{w}_2^{(+1)}]$, in which $\boldsymbol{w}_2^{(+1)}$, $\boldsymbol{w}_2^{(-1)}$ are the coefficients corresponding to the positive output and the negative output respectively. $\mathcal{R}$ represents a RNN. We fix the weights of the input layer and state transition layer of $\mathcal{R}$, and only train the output layer $\boldsymbol{W}_1$. This particular RNN structure follows the framework of reservoir computing (RC). The input and output dimension of the RNN is the exactly N_r. In general, RC is characterized by a **state equation** and an **output equation** [57]. In this paper, we choose the state equation as a first order Markov process, where the state are involved by input stimulus and initial states. The output equation is a linear mapping which readouts the internal states. Furthermore, we add a hyperbolic tangent (Tanh) function as the activation function between $\mathcal{R}$ and $\mathcal{M}$.

2.3.2 Detection on a Non-Binary Set

To conduct symbol detection on a non-binary set (e.g., $\{-3,-1,+1,+3\}$), we utilize the following shifting process to construct element-wise likelihood ratios introduced in [96],

$$\frac{P(\tilde{x}_{n_t,n_c}=-1|\tilde{\boldsymbol{Y}})}{P(\tilde{x}_{n_t,n_c}=-3|\tilde{\boldsymbol{Y}})}=\mathcal{L}_{+-}(\tilde{\boldsymbol{Y}}+2\tilde{\boldsymbol{S}}_{n_t,n_c})$$
$$\frac{P(\tilde{x}_{n_t,n_c}=+3|\tilde{\boldsymbol{Y}})}{P(\tilde{x}_{n_t,n_c}=+1|\tilde{\boldsymbol{Y}})}=\mathcal{L}_{+-}(\tilde{\boldsymbol{Y}}-2\tilde{\boldsymbol{S}}_{n_t,n_c})$$

where

$$\tilde{\boldsymbol{S}}_{n_t,n_c}:=\begin{bmatrix}\mathrm{Re}(\boldsymbol{S}_{n_t,n_c})\\ \mathrm{Im}(\boldsymbol{S}_{n_t,n_c})\end{bmatrix}$$

$$\boldsymbol{S}_{n_t,n_c}:=\begin{cases}\sum_{\ell=0}^{L-1}\boldsymbol{H}_{n_t,:}[\ell](\boldsymbol{F}\boldsymbol{J}_\ell)_{:,n_c}, \text{if } \tilde{x}_{n_t,n_c}=\mathrm{Re}(x_{n_t,n_c})\\ \sum_{\ell=0}^{L-1}j\boldsymbol{H}_{n_t,:}[\ell](\boldsymbol{F}\boldsymbol{J}_\ell)_{:,n_c}, \text{if } \tilde{x}_{n_t,n_c}=\mathrm{Im}(x_{n_t,n_c})\end{cases}$$

Note that the received signal $\tilde{\boldsymbol{Y}}$ corresponds to transmitted symbols modulated using the general QAM constellation. Therefore, the posterior estimation can be obtained by solving the equation below:

$$\sum_{a=\{-3,-1,+1,+3\}}P(\tilde{x}_{n_t,n_c}=a|\tilde{\boldsymbol{Y}})=1$$
$$P(\tilde{x}_{n_t,n_c}=-1|\tilde{\boldsymbol{Y}})=P(\tilde{x}_{n_t,n_c}=-3|\tilde{\boldsymbol{Y}})\mathcal{L}_{+-}(\tilde{\boldsymbol{Y}}+2\boldsymbol{S}_{ij})$$
$$P(\tilde{x}_{n_t,n_c}=+1|\tilde{\boldsymbol{Y}})=P(\tilde{x}_{n_t,n_c}=-1|\tilde{\boldsymbol{Y}})\mathcal{L}_{+-}(\tilde{\boldsymbol{Y}})$$
$$P(\tilde{x}_{n_t,n_c}=+3|\tilde{\boldsymbol{Y}})=P(\tilde{x}_{n_t,n_c}=+1|\tilde{\boldsymbol{Y}})\mathcal{L}_{+-}(\tilde{\boldsymbol{Y}}+-2\boldsymbol{S}_{ij})$$

Therefore, the complete NN for multi-class detection can be constructed as in Fig. 2.4.

2.3.3 Training Set

As outlined in the previous discussion, the symbol detection NN for a general QAM constellation can be constructed by shifting individual binary detection NNs. Therefore, training the NN in Fig. 2.4 is equivalent to training the binary decision neural network (2.7). Assume we have Q training samples,

$$\Big\{(\tilde{\boldsymbol{X}}_1, \tilde{\boldsymbol{Y}}_1), (\tilde{\boldsymbol{X}}_2, \tilde{\boldsymbol{Y}}_2), \cdots, (\tilde{\boldsymbol{X}}_Q, \tilde{\boldsymbol{Y}}_Q)\Big\} \tag{2.8}$$

where entries of $\tilde{\boldsymbol{X}}_q$ come from a general QAM constellation, the corresponding binary training sets for (n_t, n_c) can be created according to:

$$\begin{aligned} &\Big(+1, \tilde{\boldsymbol{Y}}_k - \tilde{x}_{n_t,n_c}\tilde{\boldsymbol{S}}_{n_t,n_c} + \tilde{\boldsymbol{S}}_{n_t,n_c}\Big) \\ &\Big(-1, \tilde{\boldsymbol{Y}}_k - \tilde{x}_{n_t,n_c}\tilde{\boldsymbol{S}}_{n_t,n_c} - \tilde{\boldsymbol{S}}_{n_t,n_c}\Big) \end{aligned} \tag{2.9}$$

through $(\tilde{\boldsymbol{X}}_k, \tilde{\boldsymbol{Y}}_k)$. For wireless systems where the training set is limited, we set $Q = N_t$ to be compatible with the standard orthogonal pilot structures adopted in modern cellular standards such as 3GPP LTE-Advanced and 5G.

2.3.4 Learning Tricks

To mitigate the issues of over-fitting and slow training convergence, the following specific techniques are employed in our experiments:

- The weight $\boldsymbol{W}_1$ is shared across all sub-carriers in the frequency domain.
- The weight $\boldsymbol{W}_2$ is initialized according to $\boldsymbol{w}_2^{(+1)} = \boldsymbol{w}_2^{(-1)} = (\boldsymbol{I}_{N_t} \otimes \boldsymbol{F})_{n_c+n_tN_c,:}$. This particular initialization essentially follows from the Fourier transformation embedded

in the generation of OFDM signals.

- The binary detector f_{n_t,n_c} can be trained by utilizing the training sets from the neighboring sub-carriers. This exploits the channel correlation between neighboring sub-carriers in frequency-selective fading channels.

2.4 Performance Evaluation

In this section, we provide performance evaluations of the introduced symbol detection approach. The modulation scheme used to generate $\boldsymbol{X}$ is 16-QAM. The underlying MIMO channel is assumed to be block fading. To be specific, the channel coefficients of the MIMO-OFDM system are generated using parameter configurations in the urban outdoor-to-indoor NLOS scenario of the WINNER II channel model [60] and are assumed to stay invariant in a block of $Q + N_d$ OFDM symbols. Uniform Linear Array (ULA) antennas are used at the transmitter and receiver with a half-wavelength spacing. The simulation parameters in the performance evaluation are set as follows: $L = 12$, $N_r = 2$, $N_t = 2$, $N_c = 256$, $N_{cp} = 25$, $Q = 2$, and $N_d = 98$. The number of neurons in the RC is set to 128 with an input buffer of length 32 as the windowed-ESN structure discussed in [100].

To evaluate the symbol detection performance, we first compare the introduced approach to conventional methods assuming perfect CSI. This result is illustrated in Fig. 2.5. As we can observe, the introduced method outperforms the conventional linear minimum mean square estimation (LMMSE) strategy while is strictly inferior to the optimal maximum likelihood (ML) detection strategy. In a more relevant scenario, we conduct the CSI estimation first using the available reference signals/pilots. The underlying performance evaluation is shown in Fig. 2.6 where we conduct LMMSE-based channel estimation to obtain the estimated CSI. The results suggest that the introduced method can offer detection performance that is close

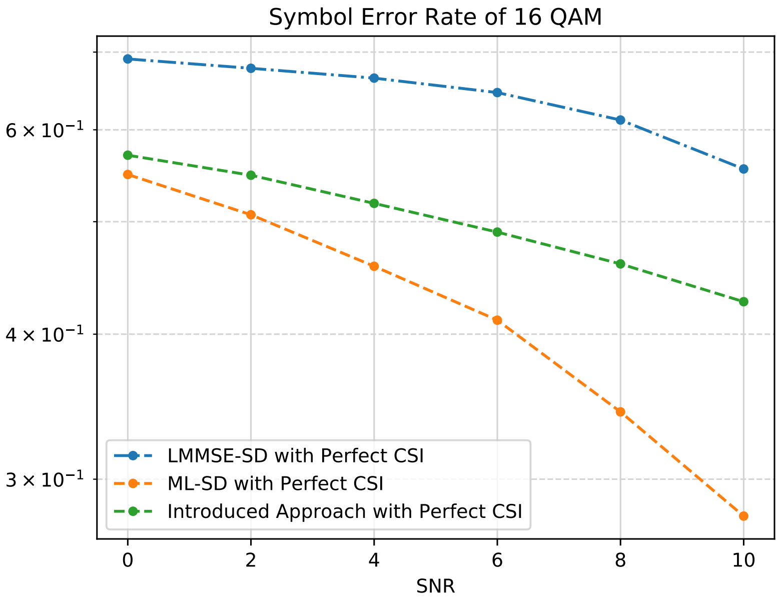


Figure 2.5: Symbol error rate comparison under perfect CSI.

to the optimum ML method especially in the low SNR regime. Note that the complexity of RC-based strategy is even lower than that of LMMSE-based strategy.

Finally, we present the performance comparison under alternate NN architectures. This comparison uses the metric of generalization error which is characterized by the symbol error rate (SER) in this case, as demonstrated in Fig. 2.7. The label 'shifting MLP' represents a structure similar to Fig. 2.7, but using a multilayer perceptron (MLP) in the time domain instead of the RC. The training scheme is the same as in the introduced network. The label 'pure MLP' represents an MLP that performs the symbol detection directly instead of employing the knowledge of structure to build the multi-class symbol detector. The results demonstrate that our method which includes a strong inductive bias can achieve a better generalization performance compared to other generic non structure-aware NN architectures illustrating the power of RC and structure knowledge in wireless systems.

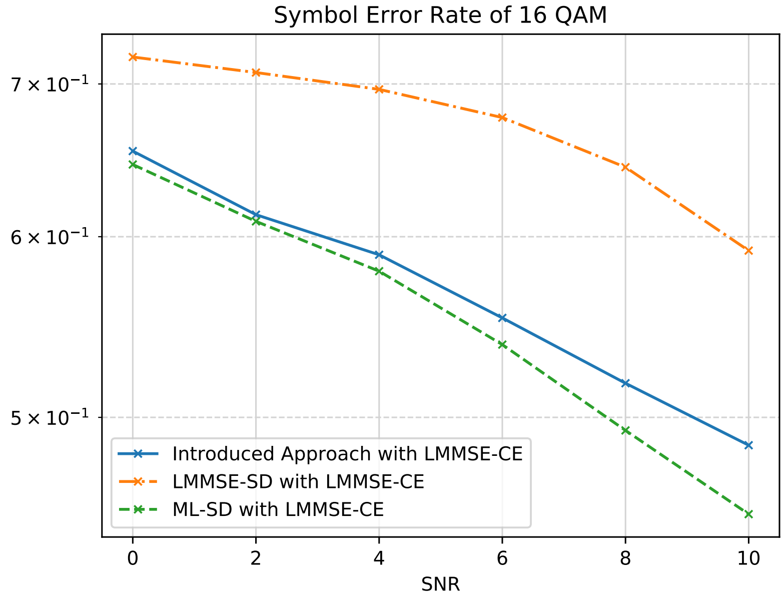


Figure 2.6: Symbol error rate comparison under LMMSE-based estimated CSI.

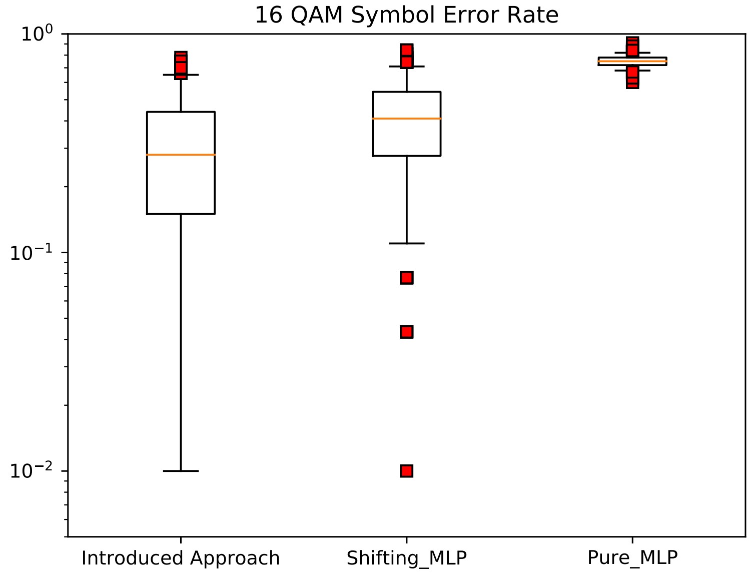


Figure 2.7: SNR = 10 dB, Training Set Size = 2 MIMO-OFDM Symbols

2.5 Conclusion

A NN-based symbol detector for MIMO-OFDM systems is introduced in this paper. The NN incorporates structural knowledge of MIMO-OFDM signals that characterizes features in time, frequency and constellation where the training overhead is significantly reduced due to the inductive bias. The time domain information is captured through the RC while the frequency domain information is reflected in the initialization of RC's output layer. The structure of the modulation scheme is incorporated by a shifting process which generalizes binary symbol detectors to multi-class detectors. The training of the associated network weights is conducted through backpropagation. Experimental results show that the introduced NN-based detection can perform close to ML detection under estimated CSI. Since the complexity of the introduced RC-based strategy is relatively low, it provides a promising symbol detection strategy for MIMO-OFDM systems.

Chapter 3

MIMO-OFDM Symbol Detection via Deep-RC

3.1 Introduction

This chapter presents an alternative symbol detection method without using explicit channel estimation. The implementation is fully based on reservoir computing (RC) [57] which is a special category of RNNs. RC is capable of avoiding the issues of vanishing and exploding gradients which occur during training of conventional RNNs using back-propagation through time (BPTT) [75]. Furthermore, the training of RC is only conducted on the output layer while the hidden layers and the input layers are fixed according to a certain distribution. In this way, the amount of training needed for MIMO-OFDM symbol detection can be significantly reduced making it an ***operationally feasible*** solution to address *Challenge 1*. This benefit can be clearly seen in Section 3.5.4 where we show a quantitative comparison of the training overhead for various learning-based strategies. The RC architecture makes applying NN techniques in the physical layer of cellular networks possible and feasible. It is also shown in [84] that RC-based symbol detection can significantly improve the underlying energy-efficiency of the system.

In this work, rather than directly applying the shallow RC structure, we attempt to address *Challenge 2* and *Challenge 3* by introducing *RCNet* through the following extensions to facilitate ***deep*** RC-based symbol detection methods that further improve detection performance:

- Extend the output layer of a shallow RC structure to a multiple-layer network promoting joint time-frequency processing. This method can effectively resolve *Challenge 2* by incorporating the structural information (time-frequency structure) of MIMO-OFDM signals into the output layer design of RCNet.
- Stack shallow RCs together into a "deep" RC to improve the processing capability of RCNet. In this way, *Challenge 3* can be addressed owing to the boosting mechanism of NNs.

The first extension on deepening the output layer is achieved by replacing the original single layer output of the shallow RC with a three-layer structure: a time-domain layer, a Fourier transform layer, and a frequency-domain layer, namely the "time-frequency RC". The time-domain layer attempts to reconstruct the transmitted time-domain signal. The Fourier transform layer is used to transform the time-domain signal to the frequency-domain. The frequency-domain layer attempts to extract frequency-domain features to further improve the detection performance. The second extension is achieved by concatenating multiple "time-frequency RCs" sequentially. Note that the output weights of each RC layer are trained in a consecutive fashion.

Through extensive experiments, we show that this deep structure of RCNet demonstrates appealing and robust performance when non-linear effects exist in the end-to-end wireless system. In other words, it outperforms conventional MIMO symbol detection strategies under channel non-linearities caused, for e.g., by the power amplifier (PA) at the transmitter (Tx) or the quantization error due to the low resolution of analog-to-digital converters (ADCs) at the receiver (Rx). The results suggest that the introduced MIMO-OFDM symbol detection framework can be a very promising enabling technology for Beyond-5G cellular systems where high-frequency spectrum and low-resolution ADCs would be frequently used. To the best of our knowledge, we believe this is the first recurrent neural network-based MIMO-OFDM symbol detector in the literature that can provide good symbol detection performance using a limited training set under relevant channel environments (e.g., the Winner II channel) in the presence of practical real-world constraints such as transmit-side non-linearity due to the PA and limited quantization resolution in the receiver.

3.1.1 Related Work

Deep RNN

Deep neural networks (DNN) can extract sophisticated features thereby providing improved classification performance over shallow NNs [38]. Hierarchical RNNs are shown to be capable of learning long-term dependencies of signals [23]. In [34], the deep long short-term memory network was introduced by consecutively stacking the hidden layers of multiple RNNs. This deep structure is shown to significantly improve performance in the task of speech recognition. The methods of extending a shallow RNN to deep RNNs have been summarized in [74]: multi-layer RNNs can be constructed by increasing input layers, hidden layers and output layers, as well as stacking multiple shallow RNNs into a deep form. Concatenating echo state networks (ESNs) into a chain by learning readout layers connecting to each ESN layer is introduced in [42]. The follow up work in [29, 30] extended this structure into a deep version which is demonstrated to be able to achieve a higher memory compared to the shallow one. However, these extensions are very general and do not consider specific domain knowledge and structural information for targeted applications.

NN-based Symbol Detection

Based on the convolution feature of wireless channels, convolutional neural networks (CNNs) are utilized for OFDM symbol detection in [94]. The work in [80] combines the CSI and received symbols together as the input of DNNs for MIMO symbol detection. In [17], it is shown that NN-based MIMO symbol detection can be generalized under imperfect CSI. Two frameworks, namely, data-driven and model-driven NNs, are introduced in [45] for the design of an OFDM receiver. However, all of the above methods are based on the NNs requiring a large training set. Therefore, it is almost impossible to adopt any of these methods in practical cellular networks.

On the other hand, our previous work [64, 100], showed that shallow RCs can achieve good performance for MIMO-OFDM symbol detection even with a very limited training set. In [64], an ESN, which is a special variant of RC, is introduced as an OFDM symbol detector without relying on obtaining explicit CSI. This scheme is evaluated under relevant scenarios for cellular networks where the training/reference signal overhead is comparable to that of current cellular networks. In our follow-up work, the windowed ESN (WESN) is introduced by adding a sliding window to the input of ESN to enhance the short-term memory (STM) [100]. Experimental results show that WESN can provide good performance over standard ESNs using the training/reference signal set adopted in 4G LTE-Advanced [5]. Furthermore, ESN-based symbol detectors can effectively compensate for the distortion caused by non-linear components in wireless transmission.

The organization of this paper is as follows. In Sec. 3.2, we briefly describe the transceiver architecture. In Sec. 3.3 and 3.4, we introduce the two extensions incorporated in our proposed RCNet structure and its associated learning algorithms. Sec. 3.5 evaluates the performance of RCNet as opposed to existing symbol detection strategies for MIMO-OFDM systems. The conclusion and future work is contained in Sec. 4.6.

3.2 OFDM Transceiver Architecture and Structural Information

In this section, we briefly introduce the transceiver architecture of a MIMO-OFDM system and illustrate the structural information that we will utilize for the design of RCNet. The OFDM resource grid can be seen most clearly in Fig. 3.1. The total system bandwidth is divided into N_{sc} sub-carriers. In each sub-frame, the first Q OFDM symbols are the reference signals (the training set) and the rest N_d OFDM symbols are used to carry data (the testing set). Therefore, the reference signal overhead (training overhead) is defined as $\eta = Q/(Q + N_d)$. In 3GPP LTE/LTE-Advanced and 5G systems, $Q + N_d = 14$ for normal sub-frames and the overhead is typically below 20% to improve spectral-efficiency [4, 6]. For a single MIMO-OFDM symbol, we denote the modulation symbol as $\{\boldsymbol{z}(n)\}_{n=0}^{N_{sc}-1}$, where $\boldsymbol{z}(n) \in \mathbb{C}^{N_t \times 1}$ represents the modulation symbols at the nth sub-carrier of the underlying

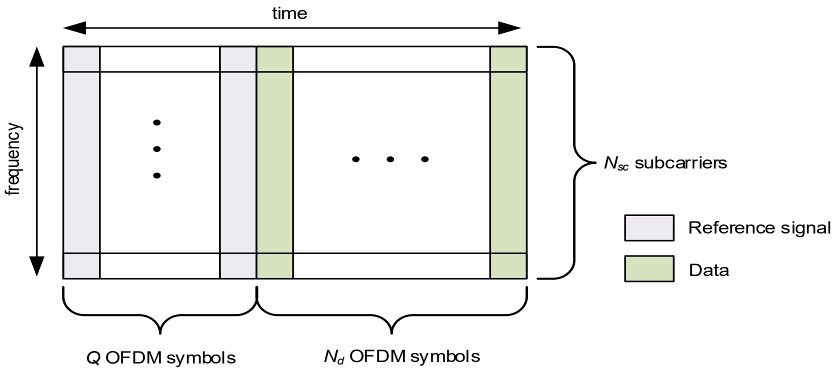


Figure 3.1: OFDM Resource Grid

MIMO-OFDM symbol in the **frequency domain**; N_t is the number of transmit antennas in the system. Each element of $\boldsymbol{z}(n)$ is modulated using quadrature amplitude modulation (QAM). A single MIMO-OFDM symbol in the frequency domain also can be lumped as a matrix,

$$\boldsymbol{Z} \triangleq [\boldsymbol{z}(0), \boldsymbol{z}(1), \cdots, \boldsymbol{z}(N_{sc}-1)]^T. \tag{3.1}$$

After the inverse fast Fourier transform (IFFT) operation and addition of the cyclic prefix (CP), we can obtain the **time domain** MIMO-OFDM symbol, denoted as $\tilde{\boldsymbol{X}}$. At the transmitter, the MIMO-OFDM symbol is distorted by a non-linear activation function $f(\cdot)$ due to the inherent non-linearity of transmitter-side radio circuits, such as the PA [46]. At the receiver, a single MIMO-OFDM symbol is expressed as

$$\boldsymbol{X} = q(h(f(\tilde{\boldsymbol{X}})) + \boldsymbol{N}), \tag{3.2}$$

where $\boldsymbol{X} \triangleq [\boldsymbol{x}(0), \boldsymbol{x}(1), \cdots, \boldsymbol{x}(N_{sc}+N_{cp}-1)]^T \in \mathbb{C}^{(N_{sc}+N_{cp})\times N_r}$; N_r is the number of receive antennas; $\boldsymbol{N}$ is the additive noise; $h(\cdot)$ represents the multi-path channel, such as the 3GPP spatial channel model (SCM) [1] and $q(\cdot)$ represents the non-linearity at the receiver.

For RC-based MIMO-OFDM symbol detection, the objective is to recover the **frequency domain** modulation symbols $\{\boldsymbol{z}(n)\}_{n=0}^{N_{sc}-1}$. On the other hand, the inputs to the RC are the **time domain** samples $\{\boldsymbol{x}(t)\}_{t=0}^{N_{cp}+N_{sc}-1}$. This structural information needs to be explored in the design of RCNet to improve its detection performance beyond that of the existing shallow RC-based symbol detection. In the supervised learning framework, the training set is defined as $\{d_q\}_{q=0}^{Q-1}$:

$$\begin{aligned} d_q &\triangleq \Big(\{\boldsymbol{x}_q(t)\}_{t=0}^{N_{cp}+N_{sc}-1}, \{\boldsymbol{z}_q(n)\}_{n=0}^{N_{sc}-1}\Big), \\ &\cong \Big(\{\boldsymbol{x}_q(t)\}_{t=0}^{N_{cp}+N_{sc}-1}, \{\tilde{\boldsymbol{x}}_q(t)\}_{t=0}^{N_{cp}+N_{sc}-1}\Big), \\ &\cong \big(\boldsymbol{X}_q, \boldsymbol{Z}_q\big) \cong \big(\boldsymbol{X}_q, \tilde{\boldsymbol{X}}_q\big). \end{aligned} \tag{3.3}$$

$\cong$ represents equivalently defined as; The subscript 'q' stands for the qth MIMO-OFDM symbol, i.e, the training set has Q batches in total. The notations used are summarized in Table 3.1.

Table 3.1: Notations

Symbols	**Definitions**
N_r	Number of receiver antennas
N_t	Number of transmitter antennas
N_{sc}	Number of sub-carriers
N_{cp}	Length of Cyclic Prefix (CP)
Q	Number of MIMO-OFDM symbols in the training set
N_d	Number of MIMO-OFDM symbols carrying data
η	Reference signal (training) overhead
$\boldsymbol{X}$	One MIMO-OFDM symbol at Rx in the time domain
$\tilde{\boldsymbol{X}}$	One MIMO-OFDM symbol at Tx in the time domain
$\boldsymbol{Z}$	One MIMO-OFDM symbol at Tx in the frequency domain
t	Sample index in the time domain
n	Sub-carrier index in the frequency domain
$\boldsymbol{z}(n)$	The nth modulation symbol of one MIMO-OFDM symbol at Tx in the frequency domain
$\tilde{\boldsymbol{x}}(t)$	The tth sample of one MIMO-OFDM symbol at Tx in the time domain
$\boldsymbol{x}(t)$	The tth sample of one MIMO-OFDM symbol at Rx in the time domain
$\boldsymbol{y}(t)$	The tth sample of one MIMO-OFDM symbol at the output of a time domain RC
$\tilde{\boldsymbol{y}}(n)$	The nth modulation symbol of one MIMO-OFDM symbol at the output of a time domain RC in the frequency domain
$\tilde{\boldsymbol{z}}(n)$	The nth modulation symbol of one MIMO-OFDM symbol at the output of a time-frequency domain RC

3.3 Time-Frequency RC – Incorporating Structural Information

To incorporate the time-frequency structural information inherent in the OFDM signal structure, we introduce in this section the new concept of "time-frequency RC" in addition to the shallow RC. For differentiation, the shallow RC is referred to as the "time domain RC" in this paper. To provide a systematic view of the RC-based design and to better articulate how the structural information is incorporated, we will first briefly discuss the "time domain RC" used in our previous work [64, 100] before describing the "time-frequency RC".

3.3.1 Time Domain RC – The Shallow RC

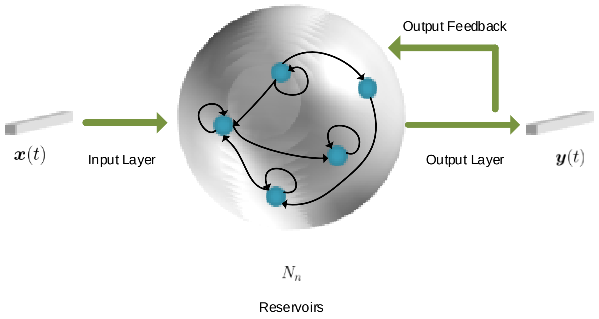


Figure 3.2: RCNet: Time Domain RC - Echo State Network (ESN) Realization

One realization of the time domain RC is illustrated in Fig. 3.2, namely an ESN [41]. In the figure, the collection of neurons is denominated as a *reservoir*. The ESN drives the input signal into a high dimensional dynamic response through a **fixed** random projection, where the response signal is represented by the trajectory of hidden neuron states. Meanwhile, non-linear activation functions are applied to the neuron states transition. The neurons in the reservoir are sparsely connected with fixed weights to satisfy certain distributions under which the response signals are asymptotically uncorrelated to the initial neuron states [41].

One justification for using fixed hidden states transition is from an experimental fact that the dominant changes of an RNN's weights during training happen at the output layer [82]. Finally, the desired outputs are obtained by learning a combination of the non-linear response signals.

Given the training set $\{d_q\}_{q=0}^{Q-1}$, the states of the reservoir generated by the qth batch are

$$\boldsymbol{s}_q^T(t+1) = f(\boldsymbol{s}_q^T(t)\boldsymbol{W}_s + \boldsymbol{x}_q^T(t)\boldsymbol{W}_{in} + \tilde{\boldsymbol{x}}_q{}^T(t)\boldsymbol{W}_{fb} + \boldsymbol{n}^T(t)) \tag{3.4}$$

where $\boldsymbol{s}_q(t) \in \mathbb{C}^{N_n \times 1}$ represents the neurons state vector, with N_n denoting the number of neurons in the reservoir; f is the activation function; $\boldsymbol{W}_s \in \mathbb{C}^{N_n \times N_n}$ denotes the state transition weights matrix; $\boldsymbol{W}_{in} \in \mathbb{C}^{N_r \times N_n}$ denotes the weights of the input layer; $\boldsymbol{n}(t) \in \mathbb{C}^{N_n \times 1}$ is an optional noise regularization term; $\boldsymbol{W}_{fb} \in \mathbb{C}^{N_r \times N_n}$ represents the weights on the feedback path which can be removed when teacher forcing is not required [41]. Correspondingly, the output signal of the time domain RC can be written as

$$\boldsymbol{y}_q^T(t) = \boldsymbol{s}_q^T(t)\boldsymbol{W}_{tout} \tag{3.5}$$

where $\boldsymbol{W}_{tout} \in \mathbb{C}^{N_n \times N_r}$ represents the readout layer. Note that in the training stage, if teacher forcing is enabled, the feedback signal associated with $\boldsymbol{W}_{fb}$ is $\tilde{\boldsymbol{x}}_q$. On the other hand, the feedback signal $\tilde{\boldsymbol{x}}_q$ is replaced by $\boldsymbol{y}_q$ after training.

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

$$\min_{\boldsymbol{W}_{tout}} \sum_{q=0}^{Q-1} \sum_{t=0}^{N_{cp}+N_{sc}-1} \left\| \boldsymbol{y}_q(t) - \tilde{\boldsymbol{x}}_q(t) \right\|_2^2 . \tag{3.6}$$

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

$$\boldsymbol{W}_{tout} = \left(\left[\boldsymbol{S}_0^T, \cdots, \boldsymbol{S}_{Q-1}^T \right]^T \right)^+ \left[\tilde{\boldsymbol{X}}_0^T, \cdots, \tilde{\boldsymbol{X}}_{Q-1}^T \right]^T \tag{3.7}$$

where $(\boldsymbol{S})^+$ is the pseudo-inverse of $\boldsymbol{S}$, and $\boldsymbol{S}_q$ is stacked by the trajectory of the states as

$$\boldsymbol{S}_q \triangleq [\boldsymbol{s}_q(0), \boldsymbol{s}_q(1), \cdots, \boldsymbol{s}_q(N_{sc} + N_{cp} - 1)]^T. \tag{3.8}$$

In addition, due to the feedback nature of RC, there exists a lag-effect on the generated state response [57]. A delay parameter can be introduced in the learning process such that the following slightly revised training set is utilized[39],

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

where p is the aforementioned delay parameter, and $\boldsymbol{O}$ represents the zero matrix. The time domain RC is trained with different values of p with $p^\star$ being the value of p generating the minimal objective value defined in (3.6). This input-output delay offset $p^\star$ will be used to configure the RC for testing. Overall, the training procedure of the time domain RC-based MIMO-OFDM symbol detection is summarized in Algorithm 5, where p is uniformly sampled from 0 to N_{cp} in steps of P. Symbol detection is conducted by feeding the received signal to the learned RC.

3.3.2 Time-Frequency RC – RC with Structural Information

Since the time domain RC focuses only on the received time domain signal without processing in the frequency domain, it is clear that it does not take advantage of the structural information of the underlying OFDM signalization. In OFDM systems, the training signal is

Algorithm 1 Time domain RC-based MIMO-OFDM symbol detection

```
Input: $\{d_q\}_{q=0}^{Q-1}$
Output: $\boldsymbol{W}_{tout}$, $p^\star$
  for $p \in [0, N_{cp}]$ with step size $P$ do
  Generate $p$-delayed training set $d^{(p)}$
  Generate the state matrix $\{\boldsymbol{S}_q^{(p)}\}_{q=0}^{Q-1}$ according to the dynamics equation (3.4)
  Calculate the output weights $\boldsymbol{W}_{tout}^{(p)}$ using (3.7) and the objective value $Obj_p$ using (3.6)
  end for
  Find the optimal $p^\star = \arg\min_p Obj_p$ and $\boldsymbol{W}_{tout} = \boldsymbol{W}_{tout}^{p^\star}$
```

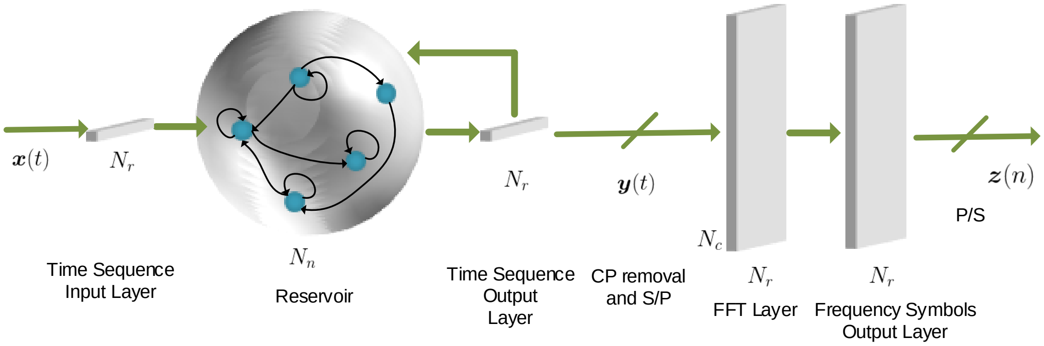


Figure 3.3: RCNet: Time-Frequency RC

sent over the frequency domain and the frequency domain provides a much cleaner view of the transmitted signal. Therefore, it is desirable for the NN-based symbol detector to conduct detection tasks in the frequency domain. Meanwhile, the underlying NN-based symbol detector should also take advantage of the time domain correlation.

To achieve these goals, we introduce the time-frequency RC whose structure is shown in Fig. 3.3. Following the same line of reasoning as the time domain RC, the reservoir first generates high dimensional response signals according to the input. The response signals are mapped to the desired signals through a time domain layer, a fixed FFT layer, and a frequency domain layer. Accordingly, the time domain layer output is the same as that in (3.5). After removing the CP and conducting an FFT on $\{\boldsymbol{y}_q(t)\}_{t=N_{cp}}^{N_{sc}+N_{cp}-1}$, the obtained frequency domain symbols are denoted as $\{\tilde{\boldsymbol{y}}_q(n)\}_{n=0}^{N_{sc}-1}$. Subsequently, the frequency domain layer output is defined as

$$\tilde{\boldsymbol{z}}_q^T(n) = \tilde{\boldsymbol{y}}_q^T(n)\mathrm{diag}\left(\boldsymbol{w}_{fout}(n)\right) \tag{3.9}$$

where $\boldsymbol{w}_{fout}(n) \in \mathbb{C}^{N_r \times 1}$ is the weight specified at the nth sub-carrier; diag($\boldsymbol{w}$) represents a diagonal matrix which has $\boldsymbol{w}$ as the main diagonal elements. Furthermore, we set the magnitude of each entry of $\boldsymbol{w}_{fout}(n)$ as 1. This allows the introduced frequency domain layer to compensate for the residual phase error after the time domain processing. It is important to note that the frequency domain processing can effectively tune the delay parameter discussed in Section 3.3.1. This is because the FFT layer converts a shift in the time domain to a phase variation in the frequency domain. Accordingly, the added output layers in the time-frequency RC essentially leverage the structural information of the MIMO-OFDM signal.

The learning objective of the time-frequency output layer is

$$\begin{aligned} &\min_{\substack{\boldsymbol{W}_{tout} \\ \{\boldsymbol{w}_{fout}(n)\}_{n=0}^{N_{sc}-1}}} \sum_{q=0}^{Q-1} \sum_{n=0}^{N_{sc}-1} \left\| \boldsymbol{z}_q(n) - \tilde{\boldsymbol{z}}_q(n) \right\|_2^2 \\ &s.t. \quad \mathrm{diag}\left(|\boldsymbol{w}_{fout}(n)|\right) = \boldsymbol{I}, \quad \forall n = 0, \cdots, N_{sc}-1 \end{aligned} . \tag{3.10}$$

which is equivalent to the time domain objective function defined in (3.6). In order to seek a proper solution to the above problem, we resort to alternative least squares (ALS) as the solver to obtain closed-form update rules for $\boldsymbol{W}_{tout}$ and $\boldsymbol{w}_{fout}(n)$. The detailed derivation can be found in the Appendix with their closed-form update rules outlined in (B.4) and (B.1) respectively. The learning algorithm of the time-frequency RC is summarized in Algorithm 2.

Note that there are certain implementation-related issues that need to be clarified when fully

Algorithm 2 Time-frequency RC based MIMO-OFDM symbol detection

```
Input: ${d_q}_{q=0}^{Q-1}$
Output: $\boldsymbol{W}_{tout}$, $\{\boldsymbol{w}_{fout}(n)\}_{n=0}^{N_{sc}-1}$
    Generate the state matrix $\{\boldsymbol{S}_q^{(p)}\}_{q=0}^{Q-1}$ according to the dynamics equation (3.4)
    Initialize $\boldsymbol{w}_{fout}(n) = \mathbf{1}$, $\forall n = 1, \cdots, N_{sc}$
    while (3.10) does not converge do
    Update $\boldsymbol{W}_{tout}$ using (B.4)
    Update $\boldsymbol{w}_{fout}(n)$ using (B.1)
    end while
```

connected layers are employed in the frequency domain, i.e. replacing $\mathrm{diag}\left(\boldsymbol{w}_{fout}(n)\right)$ with $\boldsymbol{W}_{fout}(n)$ in (3.9). In this case, the learning rule of the output layers becomes

$$\min_{\substack{\boldsymbol{W}_{tout} \\ \{\boldsymbol{W}_{fout}(n)\}_{n=0}^{N_{sc}-1}}} \sum_{n=0}^{N_{sc}-1} \left\| \tilde{\boldsymbol{F}}(n)\boldsymbol{S}\boldsymbol{W}_{tout}\boldsymbol{W}_{fout}(n) - \boldsymbol{Z}(n) \right\|_F^2$$

where $\tilde{\boldsymbol{F}}(n) \triangleq \boldsymbol{I} \otimes \boldsymbol{f}(n)$, $\boldsymbol{f}(n) \in \mathbb{C}^{N_{sc}\times 1}$ is the nth row of a Fourier matrix, $\otimes$ denotes the Kronecker product operation and

$$\boldsymbol{S} \triangleq [\boldsymbol{S}_0(N_{cp} : N_{sc} + N_{cp} - 1, :)^T,\ \boldsymbol{S}_{Q-1}(N_{cp} : N_{sc} + N_{cp} - 1, :)^T]^T$$

Based on ALS, the update rule of $\boldsymbol{W}_{fout}(n)$ is given by

$$\boldsymbol{W}_{fout}(n) = (\tilde{\boldsymbol{F}}(n)\boldsymbol{S}\boldsymbol{W}_{tout})^{+}\boldsymbol{Z}(n). \tag{3.11}$$

However, given $\boldsymbol{W}_{fout}(n)$, $\boldsymbol{W}_{tout}$ is updated by solving a Sylvester's equation which can introduce a heavy computational load. Meanwhile, this brings more parameters to learn which can lead to overfitting since the training set size is usually limited in practical systems.

3.4 RCNet – Stacking RCs for a Deep Network

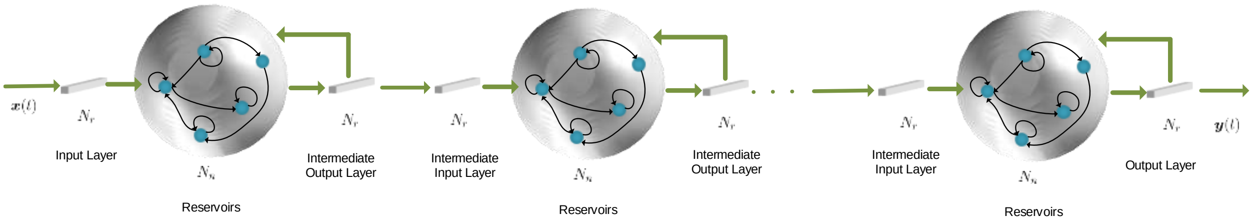


Figure 3.4: RCNet: Deep Time RC

We now introduce RCNet by stacking multiple RCs into a "deep" RC network. Intuitively, this deep structure can be interpreted as decomposing different levels of interference cancellation for the received MIMO-OFDM signal. Based on the discussion in Section 3.3, the basic building block of RCNet can be either the time domain RC or the time-frequency RC.

When the building block is the time domain RC, RCNet can be constructed by stacking them into a deep structure (Deep Time RC) as shown in Fig. 3.4. Let L denote the total number of building blocks in RCNet. Given the training set $d^{(p)}$, for the l^{th} RC, the state equations are

$$\boldsymbol{s}_q^{(l)T}(t+1) = f\left(\boldsymbol{s}_q^{(l)T}(t)\boldsymbol{W}_s^{(l)} + \boldsymbol{y}_q^{(l-1)T}(t)\boldsymbol{W}_{in}^{(l)} + \tilde{\boldsymbol{x}}_q^T(t)\boldsymbol{W}_{fb}^{(l)} + \boldsymbol{n}^T(t)\right)$$

where the superscript (l) represents the l^{th} RC; $\boldsymbol{y}^{(l)}$ follows the output equation from the previous layer which is defined in (3.5). The output layer and intermediate output layers of this RCNet are learned sequentially, i.e. the intermediate output layer closest to the input is learned first; the next RC is learned based on the results generated by the previous one. The input of the l^{th} RC is the output of the $(l-1)^{\text{th}}$ RC after training. The teacher forcing for different RCs is the same. The final output of this RCNet generates an estimate of the desired MIMO-OFDM symbol, i.e. $\boldsymbol{y}_q^{(L)}(t)$. In the state equation of the l^{th} layer, the feedback signal associated with $\boldsymbol{W}_{fb}^{(l)}$ is replaced by $\boldsymbol{y}_q$ after training. The learning algorithm of the deep time RC is summarized in Algorithm 3. This learning method can be interpreted via the boosting framework [28]: by sharing learned features among a set of weak learners, their ensemble can result in a stronger learning ability.

Algorithm 3 RCNet: Deep Time RC based MIMO-OFDM Symbol Detection

```
Input: $\{d_q\}_{q=0}^{Q-1}$
Output: $\{\boldsymbol{W}_{tout}^{(l)}\}_{l=0}^{L-1}$, $\{p^{(l)}\}_{l=0}^{L-1}$
  $\{\boldsymbol{x}_q^{(o)}(t)\}_{t=0}^{N_{cp}+N_{sc}-1} = \{\boldsymbol{x}_q(t)\}_{t=0}^{N_{cp}+N_{sc}-1}$
  for $l$ from 0 to $L-1$ do
  $d_q(l) \triangleq \left(\{\boldsymbol{x}_q^{(l)}(t)\}_{t=0}^{N_{cp}+N_{sc}-1}, \{\tilde{\boldsymbol{x}}_q(t)\}_{t=0}^{N_{cp}+N_{sc}-1}\right)$
  for $p \in [0, N_{cp}]$ with step size $P$ do
  Generate $p$-delayed training set $d^{(p)}(l)$
  Generate the state matrix $\{\boldsymbol{S}_q^{(p)}\}_{q=0}^{Q-1}$ according to the state equation (3.4)
  Calculate the output weights $\boldsymbol{W}_{tout}^{(p)}$ using (3.7) and the objective value $Obj_p$ using (3.6)
  end for
  Find the optimal $p^{(l)} = \arg\min_p Obj_p$ and $\boldsymbol{W}_{tout}^{(l)} = \boldsymbol{W}_{tout}^{p^{(l)}}$
  Use the learned $\boldsymbol{W}_{tout}^{(l)}$ and $p^{(l)}$ to generate $\{\boldsymbol{x}_q^{(l+1)}(t)\}_{t=0}^{N_{cp}+N_{sc}-1}$
  end for
```

Similarly, we can change the building block of RCNet from the time domain RC to the newly introduced time-frequency RC. The structure of the corresponding RCNet (Deep Time-Frequency RC) is shown in Fig. 3.5. The Deep Time-Frequency RC is expected to provide better performance than the Deep Time RC since it takes advantage of the structural information of OFDM signalization. Furthermore, it is important to note that compared to the Deep Time RC version of RCNet, additional IFFT layers need to be added between two adjacent time-frequency RCs to transform the former frequency domain output to a time domain signal for subsequent processing. A similar learning algorithm for this RCNet (Deep Time-Frequency RC) is summarized in Algorithm 4. Note that this learning methodology of RCNet is fundamentally different from conventional gradient-based learning in that the RC structures do not require learning of the internal state transition matrix and the input layer.

Using the l_2-norm as the loss function allows us to leverage the least squares framework to arrive at a closed-form solver, which is more efficient than conventional gradient-based objective descent. Training any of the deep RC structures via backpropagation through time (BPTT) is very inefficient since BPTT essentially unfolds the RC structure into a "very deep neural network" in both the time and space dimensions, which can cause severe gradient vanishing issues during training. On the other hand, the sequential learning methodology builds on the boosting mechanism, where results learned from previous layers offer a good starting point for the learning of the deeper layers.

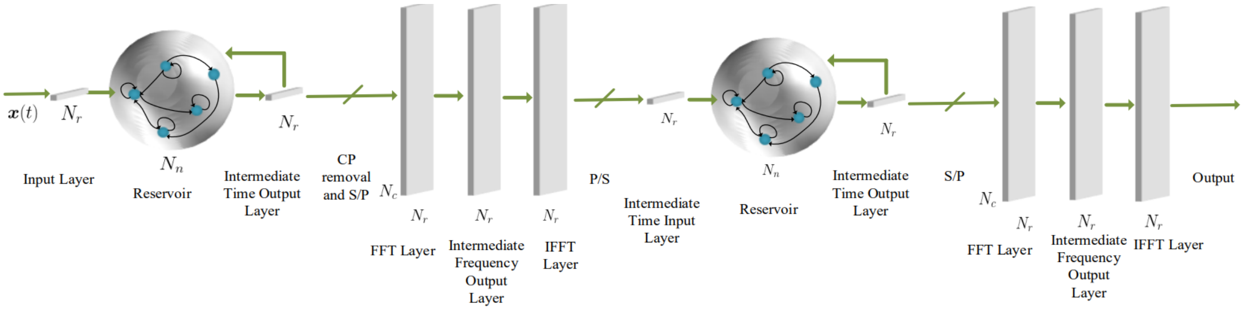


Figure 3.5: RCNet: Deep Time-Frequency RC structure

Algorithm 4 RCNet: Deep Time-Frequency RC based MIMO-OFDM Symbol Detection

```
Input: $\{d_q\}_{q=0}^{Q-1}$
Output: $\{\boldsymbol{W}_{tout}^{(l)}\}_{l=0}^{L-1}$, $\{p^{(l)}\}_{l=0}^{L-1}$
    $\{\boldsymbol{x}_q^{(o)}(t)\}_{t=0}^{N_{cp}+N_{sc}-1} = \{\boldsymbol{x}_q(t)\}_{t=0}^{N_{cp}+N_{sc}-1}$
    for $l$ from 0 to $L-1$ do
    $d_q(l) \triangleq \left(\{\boldsymbol{x}_q^{(l)}(t)\}_{t=0}^{N_{cp}+N_{sc}-1}, \{\tilde{\boldsymbol{x}}_q(t)\}_{t=0}^{N_{cp}+N_{sc}-1}\right) \cong \left(\{\boldsymbol{x}_q^{(l)}(t)\}_{t=0}^{N_{cp}+N_{sc}-1}, \{\boldsymbol{z}_q(n)\}_{n=0}^{N_{sc}-1}\right)$
    Generate the state matrix $\{\boldsymbol{S}_q^{(p)}\}_{q=0}^{Q-1}$ according to the state equation (3.4)
    Initialize $\boldsymbol{w}_{fout}(n) = \mathbf{1}$, $\forall n = 1, \cdots, N_{sc}$
    while (3.10) does not converge do
    Update $\boldsymbol{W}_{tout}$ using (B.4)
    Update $\boldsymbol{w}_{fout}(n)$ using (B.1)
    end while
    Use the learned $\boldsymbol{W}_{tout}$ and $\boldsymbol{w}_{fout}(n)$ to generate $\{\boldsymbol{x}_q^{(l+1)}(t)\}_{t=0}^{N_{cp}+N_{sc}-1}$
    end for
```

The number of RC components in RCNet is another tunable parameter. A validation set can be utilized for determining the proper value of L. We can simultaneously test the validation error while increasing L: Once the validation error stops decreasing, we can stop increasing L. Furthermore, the number of neurons in each RC can also be configured differently. How to optimize the number of neurons in each RC to achieve the best generalization performance can be treated as future work. The comparison of the complexity of time RC with conventional approaches can be found in Table I in our previous work [100]. Following a similar derivation, the additional number of FLOPS required for the time-frequency RC is on the order of $O(N_r N_{sc}(Q + 1 + Qlog(N_{sc})))$, with training the corresponding deep structures being L times more computationally expensive.

3.5 Performance Evaluation

In this section, we provide performance evaluations for the introduced RCNet framework under relevant scenarios. The modulation scheme used to generate $z(n)$ is set to be 16-QAM. The simulation parameters in the performance evaluation are configured as the follows: $N_r = 4$, $N_t = 4$, $N_{sc} = 1024$, $N_{cp} = 160$, $Q = 4$, and $N_d = 13$. Note that in this case the training overhead is only 23.5%, which is significantly lower from most other NN-based detection methods that use a prohibitively high training overhead. The channel model adopted in the evaluation is the Winner II channel model defined in [60], where the transmitter and receiver are configured with uniform linear arrays having half-wavelength antenna spacing. Furthermore, the communication scenario is chosen to be the outdoor-to-indoor case.

Recall from Fig. 3.1 that the first Q OFDM symbols constitute the training set and next N_d OFDM symbols are the data symbols that make up the testing set. Therefore, the first Q OFDM symbols will be used to train the RCNet while the bit error rate (BER) will be evaluated using the testing set (data symbols). This training and testing procedure is conducted for 100 consecutive sub-frames. The number of neurons for each layer of the RCNet (the component of the shallow RC) is set as 128. The number of layers L is set as 3 for the deep structures. Furthermore, a time window is added to the input layer of each RC unit as suggested in [100], where the length is set as 128. The state transition matrix $\boldsymbol{W}_s$ is generated randomly to satisfy the echo state property [41], where the spectral radius is chosen to be smaller than 1. The input weights matrix $\boldsymbol{W}_{in}$ is generated randomly from a uniform distribution. In our evaluations, adding teacher forcing does not show an improvement in performance. Therefore, $\boldsymbol{W}_{fb}$ is set to zero.

3.5.1 BER Performance under Tx Non-linearity

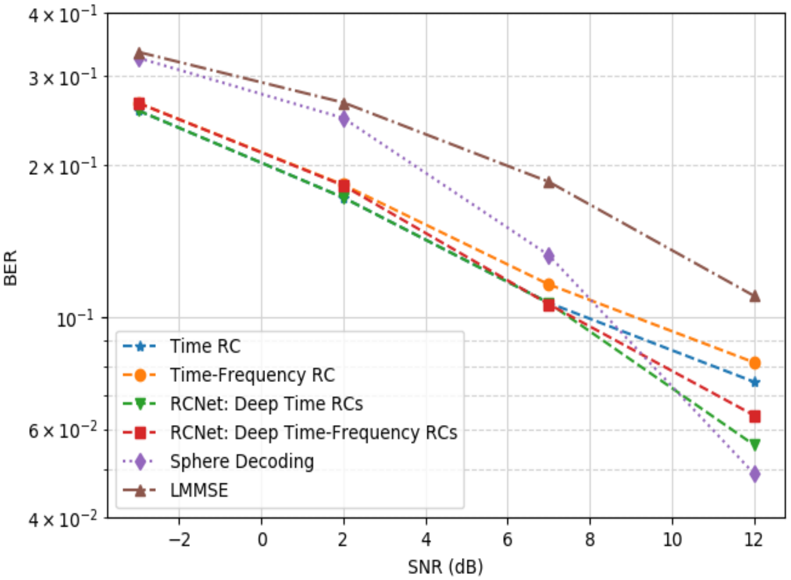


Figure 3.6: Average BER for RCNet-based methods and conventional methods in PA linear region

To evaluate the symbol detection performance, we compare RCNet to shallow RC-based strategies as well as conventional methods. To incorporate the Tx non-linearity in our eval-

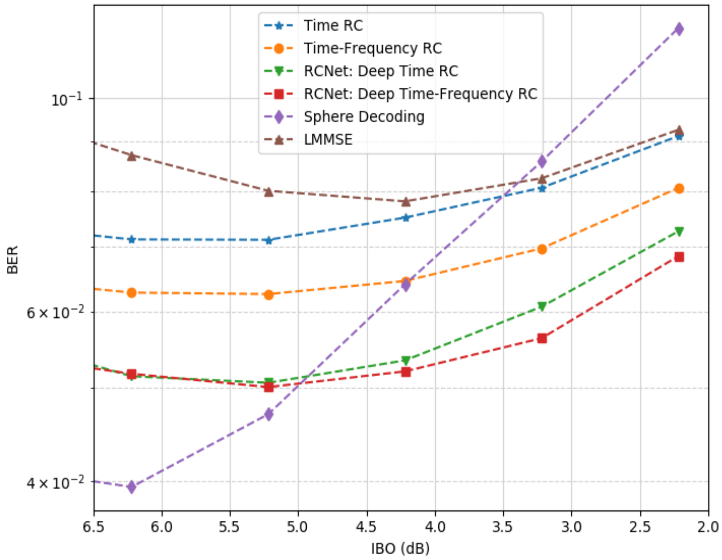


Figure 3.7: Average BER for RCNet-based methods and conventional methods by varying IBO to operate in PA's non-linear region

uation, the following RAPP model was adopted for the power amplifier (PA),

$$f(x) = \frac{x}{\left[1 + \left(\frac{|x|}{x_{sat}}\right)^{2\rho}\right]^{1/2\rho}} \tag{3.12}$$

where x represents the PA input signal, ρ is the smoothing parameter, and x_{sat} is the saturation level. When $x \ll x_{sat}$, we have $f(x) \approx x$, implying that the PA output signal has no distortion compared with the input. In our evaluation, we set $\rho = 3$ and $x_{sat} = 1$. As a benchmark, two conventional symbol detection methods, namely linear minimum mean squared error (LMMSE) and sphere decoding (SD) [32], are selected. Since these two methods rely on the knowledge of channel state information (CSI), we utilize LMMSE as the channel estimation strategy based on the Q OFDM symbols of reference signals (training set) assuming perfect knowledge of the noise variance and the linearity of the underlying wireless link.

We first consider the BER performance when the PA input power is in the linear region. In this case, the PA input power is backed off to be far from the PA's saturation region. Accordingly, 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 8 dB. In Fig. 3.6 we show the bit error rate (BER) plotted as a function of the received signal to noise ratio (SNR) in dB for various symbol detection schemes, where the SNR is calculated as the average signal to noise ratio across all subcarriers at the receiver side. From the results we can observe that all RC-based methods including the shallow structures and RCNet can perform better than conventional methods when the transmission power is low. This is because the estimated CSI is inaccurate in the low-SNR regime resulting in poor performance of the conventional model-based methods. For example, the performance of LMMSE symbol detection is greatly dependent on the accuracy of the channel state information (CSI), which is estimated using over-the-air pilots in practical systems. Therefore, inherent error in the CSI estimate can deteriorate the symbol detection performance, depending further on the modulation scheme used. On the other hand, RC-based methods are able to learn the underlying features of the channel without explicitly relying on the underlying CSI. Furthermore, it can be seen from the results that the two versions of RCNet provide slight performance improvement over their shallow counterparts demonstrating the benefits of the deep network structure.

In Fig. 3.7, we show the BER performance when the PA input power is close to the saturation region. In this case, the PA's output is distorted due to the compression effect. The distortion occurs when the peak-to-average-power-ratio (PAPR) of the PA input signal is higher than the IBO, 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 6 dB to 9 dB. Therefore, in order to investigate the BER performance under the PA's compression effect, we choose the IBO to be below 6.5 dB as shown in Fig. 3.7. The results clearly show that all RC-based methods perform relatively well when the IBO is low, especially when it is lower than 5 dB. Note that the PA efficiency is substantially higher when it is operating at a low IBO. This clearly suggests that RC-based methods can provide an improvement in PA efficiency by compensating for the transmitted waveform distortion at the receiver. Furthermore, the results in Fig. 3.7 demonstrate the benefits of utilizing the signal's structural information in the underlying NN design: the Time-Frequency RC performs significantly better than the Time RC and the Deep Time-Frequency RC performs better than the Deep Time RC. This is because the newly introduced Time-Frequency RC does take advantage of the OFDM signalization in the design of the underlying network structure of RC to address *Challenge 2*. On the other hand, the evaluation results also clearly show the power of the deep nature of the introduced RCNet to address *Challenge 3* without additional training overhead: RCNet performs substantially better than its shallow counterparts with the same training overhead. Note that in Fig. 3.7, as the IBO reduces, the received SNR will also increase leading to operation in the high SNR region at the receiver. Note that there are existing methods such as Digital Pre-Distortion (DPD) to mitigate the PA's non-linearity by pre-preprocessing the transmit data in the digital domain. However, such methods rely on accurate PA models and parameters, and cannot compensate for the loss when operating in the PA's compression region, but only when operating in its distortion region. Additionally, such mitigation methods rely on corrective processing at the transmitter, whereas RCNet can compensate for this distortion completely at the receiver, therefore, allowing PA operation at the transmitter in the non-linear regime for enhanced efficiency.

Furthermore, in Fig. 3.8, we investigate the distribution of symbol error rate (SER) over all subcarriers as a function of channel correlation. Accordingly, the x-axis is set to be the spatial channel condition numbers from all subcarriers. The SER is evaluated in the PA's nonlinear region with an IBO of 6 dB. Fig. 3.8 demonstrates that although the performance of all methods is affected by the channel condition number, both the Time RC and the Time-Frequency RC can maintain a given SER across a wider range of condition numbers, as compared to LMMSE.

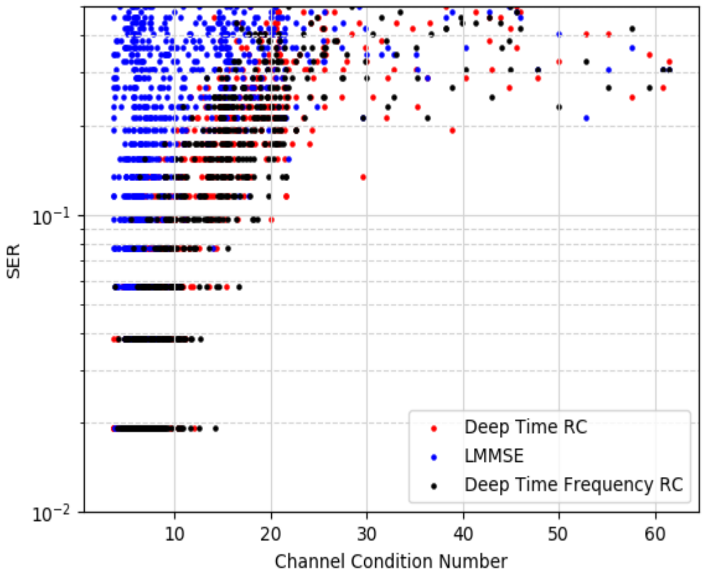


Figure 3.8: SER distribution over channel condition numbers

3.5.2 BER Performance under Rx Non-linearity

The non-linearity in the receiver stems primarily from the quantization of the received signal due to the finite resolution analog-to-digital conversion. For a MIMO-OFDM signal, the in-phase and quadrature components are quantized by a pair of analog-to-digital converters (ADCs), where the input-output relation of the ADCs can be defined as

$$q(x) = \begin{cases} \Delta\lceil x/\Delta \rceil - \Delta/2, & \text{if} |x| < A_{max} \\ A_{max} \cdot \text{sign}(x), & \text{otherwise} \end{cases} \tag{3.13}$$

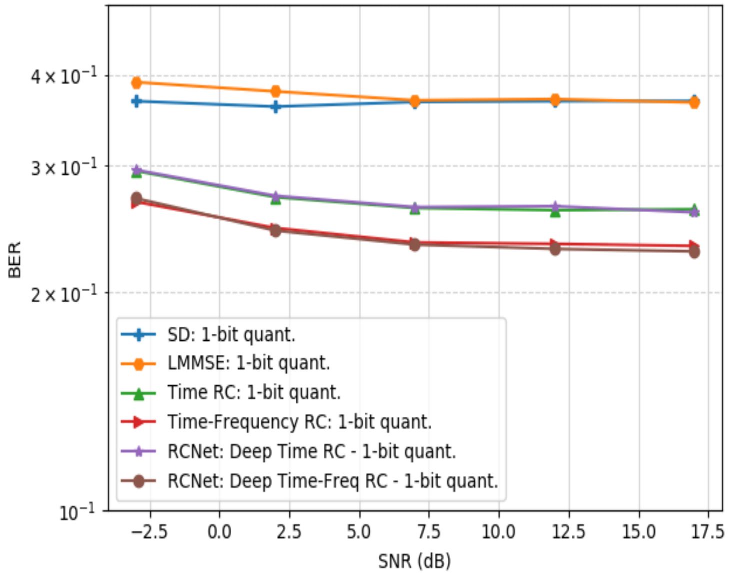


Figure 3.9: Average BER curves for RCNet-based methods and conventional methods using 1-bit resolution ADCs

in which x is the ADC's input, $\lceil \cdot \rceil$ is the ceiling function, $\Delta > 0$ represents the quantization interval, and A_{max} is the maximal input amplitude of ADC such that $A_{max} = (2^n - 1)\Delta/2$, where n is the number of quantization bits. To be specific, when $\Delta = 2A_{max}$, $q(x)$ represents a one-bit ADC. In the state-of-the-art, low-resolution ADCs are often utilized to digitize a signal with a large bandwidth at high frequencies (e.g., mmWave bands and Terahertz bands) or to reduce the power consumption of the underlying ADCs. Note that almost all existing work in the symbol detection domain with limited resolution ADCs makes system assumptions such as large number of antennas and ideal CSI, which are either not compatible with the setting in RCNet or not feasible in practice. Also, to the best of our knowledge, we did not find existing work that can generate stable channel estimation results as well as conduct symbol detection in a MIMO-OFDM system with quantized CSI.

In Fig. 3.9 and Fig. 3.10, we investigate the BER performance of uncoded MIMO-OFDM signals under 1-bit and 2-bit quantization respectively ($n = 1$ and $n = 2$) using QPSK mod-

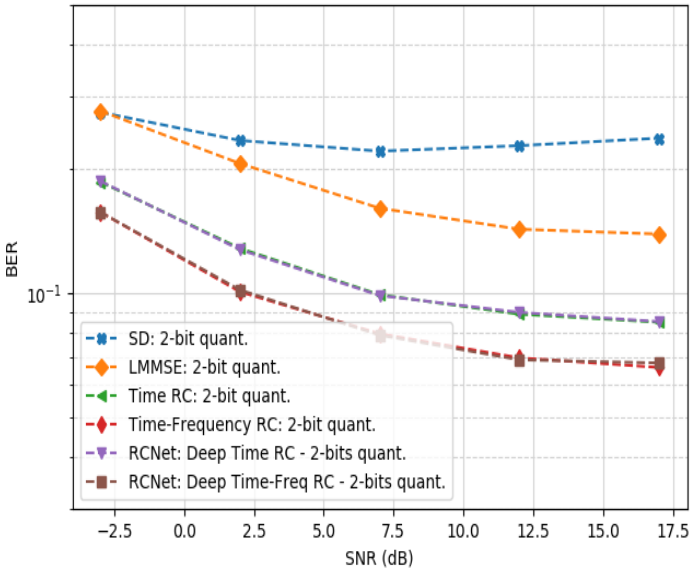


Figure 3.10: Average BER curves for RCNet-based methods and conventional methods using 2-bit resolution ADCs

ulation as a function of the received SNR. Since only 1-bit and 2-bit ADCs are used, the resulting quantization errors are usually large. From the figures, we can see that conventional methods are very sensitive to quantization errors, for e.g., the SD strategy completely collapses in both cases. On the other hand, all RC-based methods can handle the large quantization errors very well showing the benefits of adopting this particular learning-based approach. It is important to note that the RC-based strategies outperform the conventional LMMSE and SD in all SNRs of interest. Furthermore, a considerable gain can be achieved from Time-Frequency RC to Time RC showing the importance of incorporating structural information in the underlying NN design for MIMO-OFDM symbol detection. As for the comparison between the shallow RCs and the RCNet structures, we see very marginal BER performance improvement. This might be due to the fact that the shallow RC already takes the best advantage of the available limited information for symbol detection while additional iteration does not provide performance improvement. Identifying the detailed reasons behind this phenomenon can be treated as a future extension of this work. Note that even though Fig. 3.6 shows that deep time-frequency RC performs slightly worse than the time RC in the PA's linear region, i.e., in the absence of non-linearities, their performance is typically close to each other in the absence of system impairments. On the other hand, Fig. 3.7 and Figs. 3.9, 3.10 demonstrate that in the PA's non-linear operation region and with extremely low-resolution ADCs in the receiver respectively, both the time-frequency RC and the deep time-frequency RC provide better performance than their time RC counterparts. Finally, even though the uncoded BER range of 0.25 to 0.4 presented in Fig. 3.9 and Fig. 3.10 represents an infeasible operational range for a radio receiver, the performance gain of RCNet over model-based methods in this extremely challenging situation of 1-bit or 2-bit quantization and highly distorted CSI demonstrates its applicability and the advantage it offers over conventional methods in such challenging scenarios. By incorporating channel coding, it is possible to improve the RCNet performance further to a feasible operating point for LTE/LTE-Advanced systems.

3.5.3 Learning Convergence of RCNet

The evaluation results presented in previous sections show that RC-based methods are effective in the low SNR regime and under the effects of transmitter and receiver non-linearities. These evaluations use $L = 3$ for the RCNet structures. Intuitively, we can increase L to yield better training performance, however, a higher L may cause over-fitting. Fig. 3.11 shows the testing BER of RCNet as a function of L considering only the transmitter non-linearity: increasing L does ensure a decrease in the generalization error up to $L = 3$ in general. For implementation, a suitable value for L that minimizes the testing BER may be determined via fine-tuning.

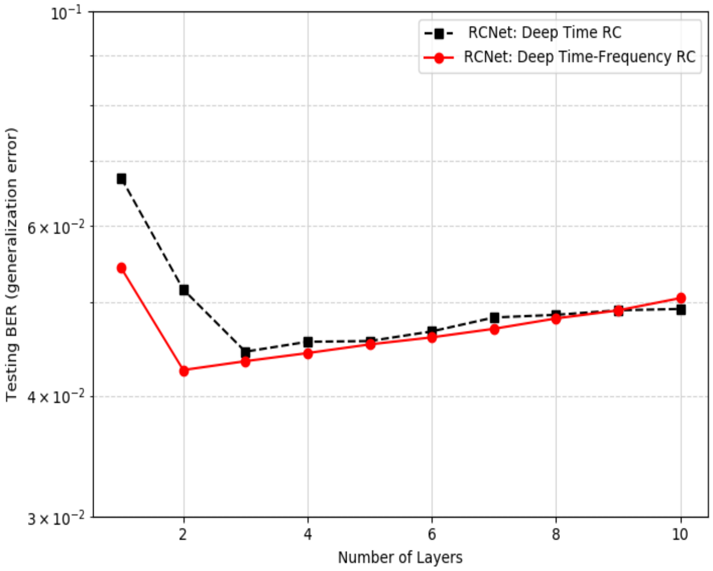


Figure 3.11: Testing BER (Generalization Error) for RCNet as a function of L

We now evaluate the convergence behavior of RCNet under transmit non-linearity. For an SNR of 12 dB, we train different RCs under the same channel realization. The objective function used for tracing the number of iterations is defined in (3.6) for the Time RC and in (3.10) for the Time-Frequency RC. Fig. 3.12 shows the learning curve of the time domain RCs where each iteration corresponds to a fixed delay parameter. For the delayed training of RCNet, we set $P = 5$ and choose the delay parameters uniformly from 0 to N_{cp} in Algorithm 3. Therefore, the training of each RC layer of RCNet requires 5 iterations and a total of $L = 10$ layers are considered in Fig. 3.12. The legend "entering next layers" represents a single iteration in the for-loop on L in Algorithm 3 and Algorithm 4. For instance, when the iterator l changes from $l = L_1$ to $l = L_1 + 1$, the output signal of layer L_1 "enters" layer $L_1 + 1$. For the shallow time RC, we set $P = 50$ in Algorithm 5, meaning that the resolution used in the search for the optimum delay is finer for the case of shallow RC than that for the case of RCNet. Therefore, there are a total of 50 iterations for the shallow

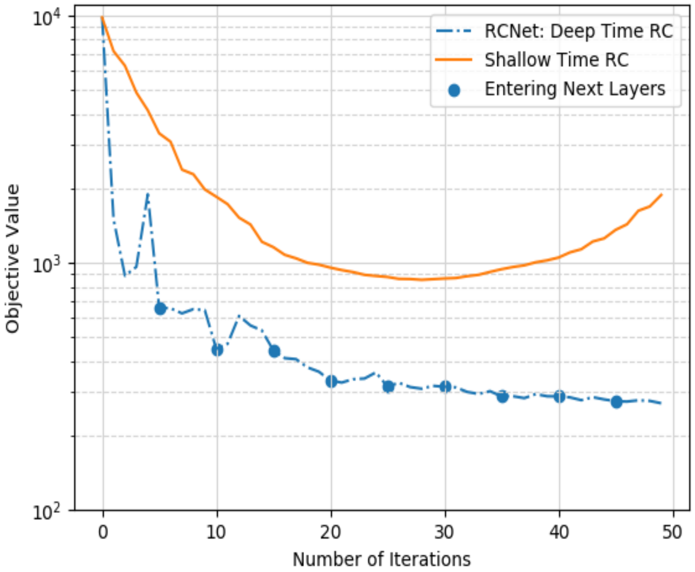


Figure 3.12: Learning curves of the Shallow Time RC and RCNet (Deep Time RC)

Time RC. From Fig. 3.12, we clearly observe that a finer delay parameter cannot result in a lower objective value, whereas adding extra layers of RCs on top of the shallow RC can decrease the objective value.

A similar conclusion can be drawn by observing the learning curve of the Time-Frequency RC and RCNet as depicted in Fig. 3.13. For RCNet (Deep Time-Frequency RC), we fix the number of iterations of the ALS for solving each RC layer to be 5 in Algorithm 4. Therefore, as in Fig. 3.12, the training for each RC layer of the RCNet requires 5 iterations and a total of $L = 10$ layers are considered in Fig. 3.13. As shown in Fig. 3.13, the objective value decreases significantly by adding one extra RC layer. Compared with the shallow Time-Frequency RC, the fitting error of RCNet (Deep Time-Frequency RC) is significantly smaller. To gain an intuition of the role of the different RC layers in RCNet, the layers can be thought of as learning different levels of features, with each layer producing a different output distance compared to the training target, so that the signals generated by the deeper layers are

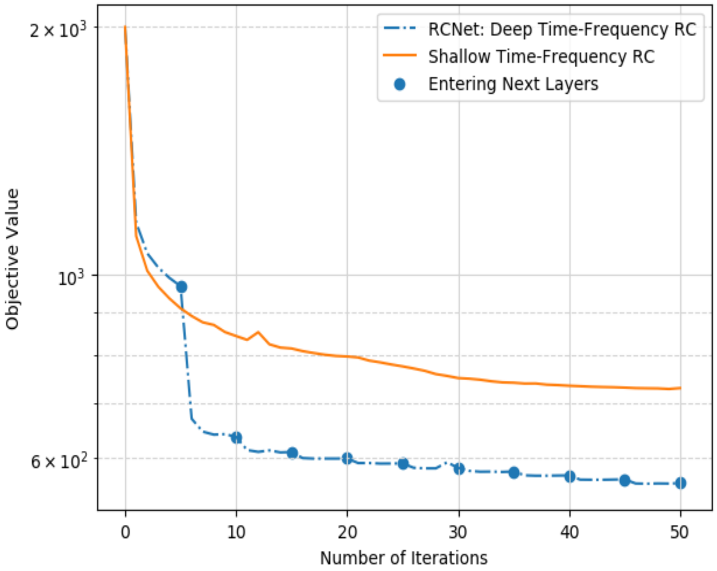


Figure 3.13: Learning curves of the Shallow Time-Frequency RC and RCNet (Deep Time-Frequency RC)

closer to the target. Therefore, deepening the RCNet structure can lead to a decrease in the training error, as compared to the shallow structure. The comparison between the two RCNet structures (Deep Time RC and Deep Time-Frequency RC) is presented in Fig. 3.14. In this comparison, we set the number of iterations of each RC layer to be 20 with L being swept from 1 to 10. Fig. 3.14 shows that the Deep Time-Frequency RC yields a stable objective value where increasing L no longer helps in deceasing the training error. On the other hand, increasing L always helps to reduce the training error for the Deep Time RC. From the generalization error on the testing set shown in Fig. 3.11, the stability characteristics of the learning curve for the Deep Time-Frequency RC provide an efficient way to determine a suitable value for L that can avoid overfitting issues.

Table 3.2: Bit Error Rate (BER) Comparison with alternate NN-based methods (PA IBO = 2.2 dB)

Detection Framework	**Training Symbols**	η **(%)**
Time RC [100]	4	23.5
Time-Frequency RC		
RCNet: Deep Time RC		
RCNet: Deep Time-Frequency RC		
LSTM	4	23.5
Bi-LSTM		
GRU		
LSTM	10	43.5
Bi-LSTM		
GRU		
LSTM	20	60.6
Bi-LSTM		
GRU		
LSTM	500	97.5
Bi-LSTM		
GRU		
LSTM	5000	99.7
LMMSE	4	23.5
Sphere Decoder		
DetNet [81] **(IID Gaussian Channel)**	3000	83.5
Training with Perfect CSI		
Training with Estimated CSI		
MMNet [47] **(3GPP MIMO Channel)**		
SNR = $13 - 18$ dB	3096	38.24

Table 3.3: Bit Error Rate (BER) Comparison with alternate NN-based methods (PA IBO = 2.2 dB)

Detection Framework	**Training BER**	**Testing BER**
Time RC [100]	2×10^{-3}	9×10^{-2}
Time-Frequency RC	8×10^{-3}	8×10^{-2}
RCNet: Deep Time RC	1.4×10^{-3}	7.3×10^{-2}
RCNet: Deep Time-Frequency RC	3.5×10^{-2}	6.9×10^{-2}
LSTM	3×10^{-2}	4.7×10^{-1}
Bi-LSTM	4.6×10^{-5}	4.7×10^{-1}
GRU	6×10^{-2}	4.7×10^{-1}
LSTM	2.2×10^{-1}	4.7×10^{-1}
Bi-LSTM	5.1×10^{-2}	4.7×10^{-1}
GRU	2.7×10^{-1}	4.8×10^{-1}
LSTM	2×10^{-1}	4×10^{-1}
Bi-LSTM	9.9×10^{-2}	4×10^{-1}
GRU	2.5×10^{-1}	4.1×10^{-1}
LSTM	4.5×10^{-1}	4.4×10^{-1}
Bi-LSTM	4.4×10^{-1}	4.3×10^{-1}
GRU	4×10^{-1}	4.4×10^{-1}
LSTM	4.4×10^{-1}	4.3×10^{-1}
LMMSE	N/A	9.2×10^{-2}
Sphere Decoder		1.3×10^{-1}
DetNet [81] **(IID Gaussian Channel)**		
Training with Perfect CSI	7×10^{-2}	1.7×10^{-1}
Training with Estimated CSI	1.3×10^{-1}	1.5×10^{-1}
MMNet [47] **(3GPP MIMO Channel)**	2.4×10^{-1}	2.55×10^{-1}

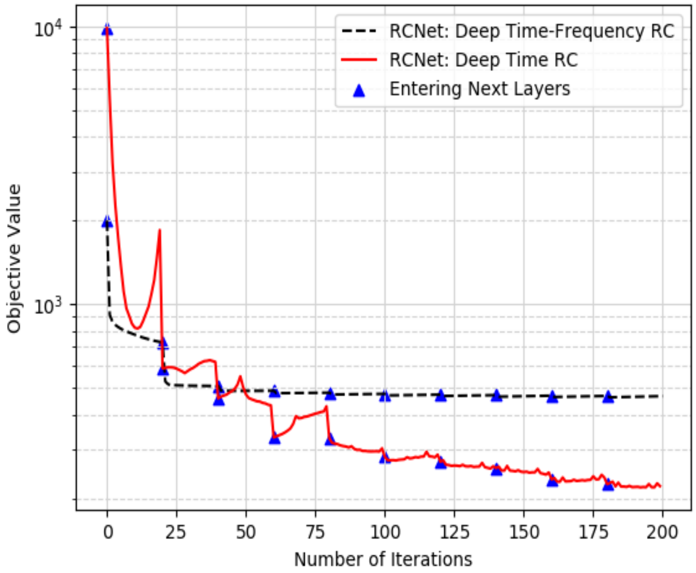


Figure 3.14: Learning curve of the Deep Time RC and Deep Time-Frequency RC structures in RCNet

3.5.4 Comparison with Other NNs

In Table 3.2, we present the performance comparison between the RC-based symbol detectors (shallow RCs and RCNet) and symbol detectors constructed using alternate NN architectures. Specifically, other popular RNN architectures are considered for performance comparison against RCNet. The variants include long short-term memory (LSTM), bidirectional-LSTM (Bi-LSTM) and gated recurrent unit (GRU) that are more robust against the short-term memory issue in vanilla RNNs. Bi-LSTM is an extension of traditional LSTMs, training on both the original input sequence as well as its reversed copy leading to a doubling of the training time for a given size of the reference signal set. To keep the comparison fair, the number of units in the recurrent hidden layer for the LSTM, Bi-LSTM and GRU structures is set to 128, in line with the 128 neurons used in each layer of RCNet. In this comparative analysis, only the PA non-linearities in the transmitter are exercised by choosing a low input back-off (IBO) of 2.2 dB while ignoring quantization effects of the ADCs in the receiver. It can be seen that in the case of all three RNN-variants, even a prohibitively large training set of 500 reference OFDM symbols, amounting to a reference signal (training) overhead of $\eta = 97.5\%$, is insufficient to achieve an acceptable BER for data transmission. On the other hand, with only 4 reference OFDM symbols, i.e., a reference signal overhead of $\eta = 23.5\%$, all four RCNet detection methods achieve better BER performance than conventional detection methods (SD and LMMSE) listed in Table 3.2. Note that SD performs worse than LMMSE since we are operating in the non-linear region of the PA, as can be confirmed from the results in Fig. 3.7.

As a benchmark to compare RCNet against, we also evaluate another deep learning-based MIMO symbol detection scheme *DetNet*, presented in [81]. A key requirement of DetNet is the availability of perfect CSI during testing for MIMO symbol detection. However, perfect CSI is either infeasible or extremely costly to obtain in practical wireless systems. In order to provide a fair comparison of DetNet against RCNet which does not rely on CSI for detection, we evaluate DetNet's performance in the 4×4 IID Gaussian channel in the following two cases: 1) Training with perfect CSI, and 2) Training with estimated CSI. In Case 1), we assume perfect CSI is available in the training phase of DetNet. Note that the perfect training CSI assumption is valid for DetNet since the amount of training data is abundant for the receiver to obtain the close-to-perfect CSI. On the other hand, during the testing phase of DetNet, like in the case of LMMSE and SD, we assume the LMMSE channel estimator is adopted to obtain the underlying CSI estimate. Accordingly, the estimated CSI is utilized in the testing phase. In Case 2), we assume the LMMSE estimator is also adopted in the training phase of DetNet to make sure the training and testing environments are the same. It is important to note that in both cases the training/channel estimation overhead is the same: the training overhead in the training phase + CSI estimation overhead in the testing phase. Further realizing that a PA backoff of 2.2 dB maps to a received SNR of 17 dB for RCNet, the same SNR value is used to train and test DetNet with its default configurations of parameters such as number of training iterations, learning rate and batch sizes. Table 3.2 clearly suggests that RCNet can outperform DetNet in both cases. To be specific, when training with perfect CSI, DetNet can achieve a testing BER of 0.17 with a training overhead of 83.5% (a training overhead of 60% and a pilot overhead of 23.5% for CSI estimation) On the other hand, the testing BER for DetNet is 0.15 when trained with estimated CSI. Both DetNet results in IID channels are significantly higher than 0.069, the corresponding testing BER of RCNet (Deep Time-Frequency RC). Furthermore, we faced significant difficulty ensuring the stability and convergence of the training process of DetNet with the spatially-correlated 3GPP SCM channel. This demonstrates that DetNet cannot be extended to realistic correlated MIMO channels. Additionally, for DetNet, a fixed training overhead is needed to train the underlying neural network in addition to the pilot overhead for channel estimation. This leads to a much higher training symbol overhead compared to RCNet. Furthermore, once the channel statistics have changed, the underlying neural network needs to be trained again under the new channel statistics.

For an exhaustive comparison with other NN-based detection methods, we also consider an iterative MIMO symbol detection framework such as MMNet [47]. From Table 3.2, it can be seen that for a 4 × 4 MIMO system, even with a training overhead of 38.24%, the BER performance is as high as 0.255 across the SNR range used for 3GPP MIMO channels in [47]. Note that this setting for evaluating MMNet is significantly different from [47], where massive MIMO channels (16×64 or 32×64) are considered. A best-effort hyperparameter tuning (learning rate and number of layers) is also performed in order to achieve the lowest possible SER for the 4×4 system, which is very different from the configurations for which MMNet parameters are originally trained. The reason behind the poor performance of MMNet in the

4×4 MIMO setting may be understood from the fact that iterative methods such as MMNet rely on the asymptotic property of the massive MIMO channel to achieve optimality [43], which does not hold for small number of transmit and receive antennas.

Overall, from the comparison with other RNN-based symbol detection strategies we can clearly see that RCNet offers advantages of requiring very limited training overhead to provide good performance of MIMO-OFDM symbol detection tasks. Compared with conventional model-based symbol detection strategies, we can see that RCNet provides advantages such as being robust to model mismatch and RF non-linearities (transmitter and/or receiver). On the other hand, the state-of-art NN-based symbol detection strategies, such as *DetNet* and *MMNet* either require extensive amount of training data, time, and computational resources compared to RCNet or are not well-suited to a MIMO configuration with limited antennas.

3.6 Conclusion and Future Work

RC-based symbol detectors have been introduced for MIMO-OFDM systems to work under very limited training sets. This paper introduced a deep RNN-based network called *RCNet* to 1) incorporate structural information of the OFDM signalization, and 2) deepen the original shallow RC-based symbol detection strategies to further improve the detection performance. Incorporating structural information has been achieved through the invention of Time-Frequency RC where the learning is done both in the time and frequency domains to take advantage of the time-frequency structure of the underlying OFDM signal. Meanwhile, the deep nature of RCNet has been achieved by extending a shallow RC structure to a deep RC structure in the following two ways: cascading time domain RCs and cascading Time-Frequency RCs. The associated learning algorithms have been developed for each of these extensions and extensive evaluation has been conducted. Experimental results showed that RCNet can outperform conventional methods using a limited training set under non-linear RF effects of the wireless link demonstrating the effectiveness of incorporating structural information and deepening RCs for symbol detection.

An important area of exploration for future research is how to determine the optimal L, i.e., the number of RCs in RCNet especially under receiver non-linearity. Connections to the boosting method may provide insights on designing the number of neurons in each layer. Another interesting direction for future work is how such RC-based detection methods can be combined with transmit-side precoding to jointly optimize the link performance with limited CSI feedback at the transmitter. The full potential of the RCNet symbol detection method is yet to be explored. From a theoretical standpoint, it would also be meaningful to analyze the functionality of each layer in the interference cancellation for a multi-user MIMO network.

Chapter 4

Massive MIMO Symbol Detection via Tensor-RC

4.1 Introduction

Adding more antennas to the base station is proven to be able to offer more favorable channel propagation conditions [59]. Therefore, massive multiple-input multiple-output (MIMO) becomes an essential physical layer technique for the 5th generation cellular networks (5G) [58]. By employing a large number of antennas at base stations (BSs), a "favorable propagation" channel condition can be achieved. It allows inter-user interference being effectively eliminated via fairly simple linear precoding or receiving methods, e.g., conjugate beamforming for downlink, or matched filtering for uplink [31].

However, the deployment of massive MIMO in practical systems encounters several implementation constraints. Primarily, for the sake of achieving the promised benefits by massive MIMO, highly accurate channel state information (CSI) is needed [90]. On the other hand, CSI with high precision is challenging to be obtained due to the low received signal-to-noise (SNR) before beamforming/precoding, as well as the limited pilot symbols defined in modern cellular networks due to control overhead [85]. Furthermore, theoretical analysis of massive MIMO systems usually assume ideal linearity and pleasing noise figures requiring exceptionally high costs on radio frequency (RF) and mixed analog-digital components [91].

This ends up with a compromise on the hardware selection which introduces imperfectness (e.g. dynamic non-linearity) to the transmission link. The resulting non-linearity, on the other hand, leads to waveform distortion and thereby diminishes the transmission reliability, which is challenging to be analytically tackled using model-based approaches.

Symbol detection is a critical stage in wireless communications. It is a process that accomplishes miscellaneous interference cancellation at receivers, such as inter-symbol, inter-stream, and inter-user inference, etc. Due to the potential model mismatch from the non-linearity caused by low-cost hardware devices, standard model-based signal processing approaches are no longer effective. With the advent of deep neural networks, there are growing interests in using neural networks (NNs) to handle the model mismatch [85]. In general, NN-based framework aims to compensate for the model mismatch through the non-linearity of NNs. This recent awareness of bridging learning-based approaches to the symbol detection task in massive MIMO systems has posed the following conceptual discussions in the NN design.

- Curse of Antenna Dimensionality: Since the input, hidden, and output layers of NNs are often configured as the same scale as the antennas to jointly extract and process spatial and time-frequency features, the growth of antenna numbers inevitably lead to the increase of the volume of underlying NN coefficients. As NNs essentially learn the underlying statistics of data, the corresponding increase in the parameter dimensionality often imposes an exponential need on the training data set to offer a reasonable generalization result. However, the availability of the training data for cellular networks (e.g.,4G or 5G) especially the online ones is extremely limited due to the associated control overhead [96]. Furthermore, the computational complexity is also evinced with an exponential relation to the NN scale. Learning neural weights through generic back-propagation can result in large computational complexities leading to severe processing delays which are not desirable especially for delay-sensitive applications.

- Blessings of Antenna Dimensionality: On the other hand, a large number of antennas is able to offer favorable propagation conditions [58]. Properly leveraging the asymptotic orthogonality of the wireless channels can often result in surprising outcomes, which conversely transforms the curse of dimensionality into "the blessings of dimensionality" [15]. These findings align with the measurement concentration phenomena which are widely applied to simplifying machine learning frameworks [33]. Therefore, to explore learning-based strategies for massive MIMO, we can incorporate inherent structures from the spatial channel as well as time-frequency features from the modulation waveform to the design of NNs, which is "blessed" to offer good generalization performance yet under very limited online training.

4.1.1 Previous Work

A commonly utilized approach for building learning-based symbol detectors is through unfolding existing optimization-based symbol detection methods to deep NNs, such as DetNet[81] and MMNet[47]. Since this framework is based on using explicit CSI, it usually suffers from performance drop or requires extensive hyper-parameters tuning when CSI is not perfect. Meanwhile, the resulting "very deep neural networks" are extremely demanding in computational resources which hinders their applications in practical scenarios. Alternatively, implicit CSI can be utilized to circumvent the above mentioned training issues. For example, [93] introduced a deep feedforward NNs for symbol detection in single-input-single-output (SISO) OFDM systems. Due to its independence from channel models, this approach can equalize the channel with nonlinear distortion (power amplifier (PA)). However, this method uses a less-structured deep NN which is yet too complicated to train in practice, since the guaranteed generalization performance is based on extensive training over large datasets that are impossible to obtain in over-the-air scenarios. Furthermore, "uncertainty in generalization" [85] will arise if the dataset used for training the underlying NN is not general enough to capture the distribution of data encountered in testing. This is especially true for 5G and Beyond 5G networks that needs to offer reliable service under vastly different scenarios and environments.

In 4G/5G MIMO-OFDM systems, there exists different operation modes with link adaptation, rank adaptation, and scheduling on a subframe basis [54]. Therefore, it is challenging, to adopt a complete offline training-based approach. Rather, it is critical to design an online NN-based approach to conduct symbol detection in each subframe only using the limited training symbols that are present in that particular subframe. In this way, the online-learning-based approach can be adaptive and robust to the change of operation modes, channel distributions, and environments. On the other hand, conducting effective and efficient learning only through the limited training symbols within a subframe is extremely challenging. To achieve this goal, more structural knowledge of the wireless channel and modulation waveform need to be incorporated as inductive priors to the NNs to significantly relieve the training overhead in each subframe basis[96, 98]. Reservoir computing (RC) and its deep version have been introduced for the MIMO-OFDM symbol detection task in [64, 99, 100, 101] to achieve learning on a subframe basis. To be specific, [64] is the first work using a vanilla RC structure to conduct MIMO-OFDM symbol detection, where the input and output are defined in the time domain. It shows this simple approach can achieve good symbol detection performance in short memory channels with limited training. [99, 100, 101] extended the RC-based symbol detection framework by adding units in width and deepth to handle channel with long taps as well as more severe non-linearity. Experiments show that the extended RC framework – RCNet can effectively compensate for the distortion caused by non-linear components in wireless systems as well as mitigate miscellaneous interference merely from receiver side using the training dataset only from each subframe.

4.1.2 Contributions

In this paper, we consider the uplink symbol detection in a massive MIMO system with OFDM waveform using a "subframe by subframe" learning framework. Uplink transmission is a typical low SNR scenario since mobile terminals often use relatively low transmission powers, and the RC framework has not yet been investigated under the scope of the massive MIMO systems. Being able to conduct receive processing – symbol detection on a subframe basis is extremely important for robust and adaptive communications in the 5G and beyond 5G massive MIMO networks. By referring to the multi-dimensional feature of massive MIMO signals (e.g. elevation and azimuth directions in the spatial domain, the time and frequency domain), we are motivated to incorporate this tensor structure into our symbol detection NN. Although the concept of tensor-driven NNs has been studied before, such as Tensorized NNs in [68], where NN weights are formulated as a tensor-train decomposition [69], and CANDECOMP/PARAFAC (CP) decomposition characterized convolutional layers for learning-acceleration from [52], our strategy is different to these techniques as tensor is utilized as the forward path data structure rather than NN coefficients. In its application to massive MIMO systems, instead of treating the input signal as a vector sequence, we define the received signal as a tensor sequence that is consistent with the intrinsic multiple mode property of the underlying massive MIMO signals. Such a signal processing perspective has been studied in [18, 97] to solve conventional massive MIMO channel estimation problems. However, a more accurate explicit CSI does not sufficiently lead to an improvement of the transmission reliability, since symbol detection is conducted on a separate stage without knowledge of the channel estimation errors. Our introduced method is to directly demodulate symbols avoiding the intermediate channel estimation stage. More importantly, our method accomplishes the symbol detection by using training dataset only from each subframe. The resulting multi-mode processing framework can in general be extended to process any other tasks with tensor structured sequence, such as video, social networks, and recommendation systems, etc.

We name our introduced RC-based NN structure as “multi-mode reservoir computing” (Multi-mode RC), as it inherits the dynamic mechanism of RC and processes input-output relation using a tensor format (multi-dimensional array). In our framework, a core-tensor is built as hidden features of the input tensor sequence. Desired output is then obtained through a multi-mode mapping. In terms of tensor algebra, the RC readout is learned through a Tucker decomposition with a deterministic core-tensor thanks to the aforementioned feature extraction. A theoretical analysis is then provided to show the uniqueness of the learned NN coefficients. Our experiments reveal that this uniqueness condition is related to the avoidance of over-fitting issues since it prevents a zero loss value which often results in poor generalization performance on the testing dataset. Compared to single-mode RC, Multi-Mode RC can achieve better symbol detection performance in terms of uncoded bit error rate in the low SNR regime with reduced computational complexity. In addition, the introduced method is shown to be effective to combat extreme waveform distortion, e.g. applying one-bit analog to digital converter (ADC) as the receiving quantization. The remainder of this paper is organized as follows: In Sec. 5.2, we briefly introduce math foundations of tensor and the background of reservoir computing which are utilized to build the concept of Multi-Mode RC in Sec. 4.3. In Sec. 4.4, the application of Multi-Mode RC to massive MIMO-OFDM symbol detection is discussed. Sec. 4.5 evaluates the performance of Multi-Mode RC as opposed to existing symbol detection strategies in massive MIMO-OFDM systems. The conclusion and future research directions are outlined in Sec. 4.6.

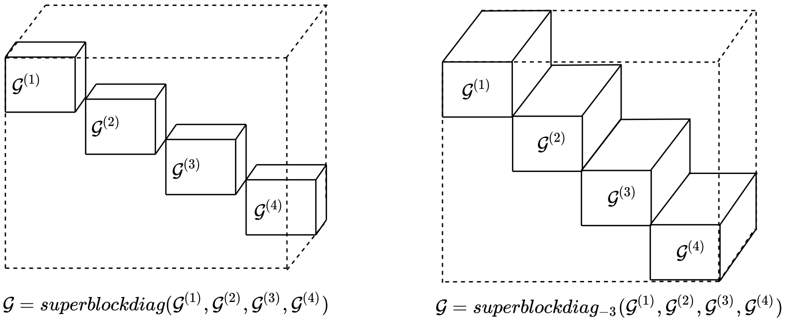
4.2 Preliminary

This section provides an overview of basic tensor algebra and reservoir computing which will be used in the rest of this paper as the methodology development. In our math notations: scalars, vector, matrix, and tensor are denoted by lowercase letters, boldface lowercase letters, boldface uppercase letters, and boldface Euler script letters respectively, e.g., x, $\boldsymbol{x}$, $\boldsymbol{X}$ and $\boldsymbol{\mathcal{X}}$.

4.2.1 Tensor Algebra

Tensor is an algebraic generalization to matrix. A tensor represents a multidimensional array, where the mode of a tensor is the number of dimensions, also known as ways and orders[49]. The $(i_1, i_2, \cdots, i_N)$th element of a N-mode tensor, or namely a Nth order tensor, $\boldsymbol{\mathcal{X}} \in \mathbb{C}^{I_1 \times I_2 \times \cdots \times I_N}$, is denoted as $x_{i_1, i_2, \cdots, i_N}$, where indices range from 1 to their capital versions.

By following matrix conventions, $rank(\boldsymbol{X})$ represents the rank of the matrix $\boldsymbol{X}$. $\boldsymbol{X}^T$, $\boldsymbol{X}^H$ and $\boldsymbol{X}^+$ respectively stands for the transpose, hermitian transpose, and Moore–Penrose pseudoinverse of the matrix $\boldsymbol{X}$. Analogously, the tensor transpose of a tensor $\boldsymbol{\mathcal{X}}$ is denoted as $\boldsymbol{\mathcal{X}}^{T_\Pi}$ which means the ith mode of $\boldsymbol{\mathcal{X}}^{T_\Pi}$ correspond to the mode numbered as $\Pi(i)$ of $\boldsymbol{\mathcal{X}}$, where Π is a permutation on set $\{1, 2, .., N\}$. Moreover, blockdiag$(\boldsymbol{A}_1, \cdots, \boldsymbol{A}_N)$ represents stacking $\boldsymbol{A}_1, \cdots, \boldsymbol{A}_N$ as a block-diagonal matrix. We denote superblockdiag$(\ \cdot\ )$ as a super-diagonal tensor by stacking its tensor arguments as illustrated in Fig. 4.1. superblockdiag$_{-n}(\cdot)$ forms a super diagonal tensor except on mode n.



$\mathcal{G} = superblockdiag(\mathcal{G}^{(1)}, \mathcal{G}^{(2)}, \mathcal{G}^{(3)}, \mathcal{G}^{(4)})$

$\mathcal{G} = superblockdiag_{-3}(\mathcal{G}^{(1)}, \mathcal{G}^{(2)}, \mathcal{G}^{(3)}, \mathcal{G}^{(4)})$

Figure 4.1: Illustration of the tensor diagonalization of 3-mode tensors: The left hand-side represents a super diagonal tensor superblockdiag($\boldsymbol{\mathcal{G}}^{(1)}, \boldsymbol{\mathcal{G}}^{(2)}, \boldsymbol{\mathcal{G}}^{(3)}, \boldsymbol{\mathcal{G}}^{(4)}$), the right hand side represents a super diagonal tensor only on the first two modes, i.e. superblockdiag_$_{-3}$($\boldsymbol{\mathcal{G}}^{(1)}, \boldsymbol{\mathcal{G}}^{(2)}, \boldsymbol{\mathcal{G}}^{(3)}, \boldsymbol{\mathcal{G}}^{(4)}$), where the diagonal elements are $\boldsymbol{\mathcal{G}}^{(1)}$, $\boldsymbol{\mathcal{G}}^{(2)}$, $\boldsymbol{\mathcal{G}}^{(3)}$ and $\boldsymbol{\mathcal{G}}^{(4)}$.

The definition of the mode-n matricization of a tensor $\boldsymbol{\mathcal{X}}$ is denoted as $\mathbf{X}_{(n)}$, where $(i_1, i_2, \ldots, i_N)$ of $\boldsymbol{\mathcal{X}} \in \mathbb{C}^{I_1 \times I_2 \times \cdots \times I_N}$ maps to the (i_n, j) entry of matrix $\mathbf{X}_{(n)} \in \mathbb{C}^{I_n \times I_{-n}}$, where $I_{-n} := \prod_{k \neq n} I_k$. According to[51],

$$j := 1 + \sum_{\substack{k=1 \\ k \neq n}}^{N} (i_k - 1) J_k \quad \text{with} \quad J_k = \prod_{\substack{m=k+1 \\ m \neq n}}^{N} I_m. \tag{4.1}$$

The n-mode product of a tensor $\boldsymbol{\mathcal{X}}$ with a matrix $\boldsymbol{U} \in \mathbb{C}^{J \times I_n}$ is defined as,

$$(\boldsymbol{\mathcal{X}} \times_n \mathbf{U})_{i_1 \ldots i_{n-1} j i_{n+1} \ldots i_N} = \sum_{i_n=1}^{I_n} x_{i_1 i_2 \ldots i_N} u_{j i_n}.$$

The Tucker decomposition of a tensor $\boldsymbol{\mathcal{X}}$ is defined as

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{G}} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \cdots \times_N \mathbf{A}^{(N)} \tag{4.2}$$

where $\boldsymbol{A}^{(n)}$ represents the nth factor matrix and $\boldsymbol{\mathcal{G}}$ is named as the core tensor. Accordingly, the mode-n unfolding of the tensor $\boldsymbol{\mathcal{X}}$ is given by

$$\boldsymbol{X}_{(n)} = \boldsymbol{A}^{(n)}\boldsymbol{G}_{(n)}(\boldsymbol{A}^{(1)} \otimes \cdots \boldsymbol{A}^{n-1} \otimes \boldsymbol{A}^{n+1} \cdots \otimes \boldsymbol{A}^{(N)})^T, \tag{4.3}$$

where $\boldsymbol{G}_{(n)}$ is the mode-n unfolding of $\boldsymbol{\mathcal{G}}$. Note the above unfolding tensor has a reverse order in the Kronecker products of factor matrices which differs to [49]. This is because we alter the way to pile up the indices of unfolding tensors according to (4.1).

We now consider a super diagonal core tensor $\boldsymbol{\mathcal{G}}$ with K blocks, i.e.,

$$\boldsymbol{\mathcal{G}} = \text{superblockdiag}(\boldsymbol{\mathcal{G}}^{(1)}, \boldsymbol{\mathcal{G}}^{(2)}, \cdots, \boldsymbol{\mathcal{G}}^{(K)})$$

where $\boldsymbol{\mathcal{G}}^{(k)} \in \mathbb{C}^{I_1^{(k)} \times I_2^{(k)} \times \cdots I_N^{(k)}}$ and a matrix $\boldsymbol{A}^{(n)}$ being partitioned as

$$[\boldsymbol{A}^{(n,1)}, \boldsymbol{A}^{(n,2)}, \cdots, \boldsymbol{A}^{(n,K)}].$$

Accordingly, the resulting n-mode product between $\boldsymbol{\mathcal{G}}$ and $\boldsymbol{A}^{(n)}$ can be written in terms of a super diagonal tensor except on the nth mode:

$$\boldsymbol{\mathcal{G}} \times_n \boldsymbol{A}^{(n)} = \text{superblockdiag}_{-n}(\boldsymbol{\mathcal{G}}^{(1)} \times_n \boldsymbol{A}^{(n,1)}, \boldsymbol{\mathcal{G}}^{(2)} \times_n \boldsymbol{A}^{(n,2)}, \cdots, \boldsymbol{\mathcal{G}}^{(K)} \times_n \boldsymbol{A}^{(n,K)}).$$

When we assume the core tensor of $\boldsymbol{\mathcal{X}}$ is super-diagonal, the Tucker decomposition defined in (4.2) can be alternatively expressed in terms of a summation of sub-Tucker decompositions:

$$\boldsymbol{\mathcal{X}} = \sum_{k=1}^{K} \boldsymbol{\mathcal{G}}^{(k)} \times_1 \mathbf{A}^{(1,k)} \times_2 \mathbf{A}^{(2,k)} \cdots \times_N \mathbf{A}^{(N,k)}. \tag{4.4}$$

An illustration for Tucker decomposition of a three mode tensor is depicted in Fig. 4.2.

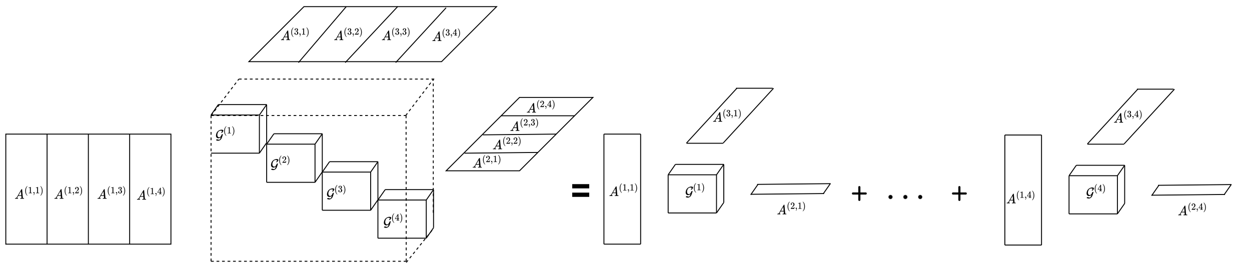


Figure 4.2: Illustration for a Tucker decomposition of a three mode tensor: Core tensor and factor matrices are with four partitions.

4.2.2 Reservoir Computing

Reservoir computing (RC) is defined as a framework for computation by using memory units. The 'reservoir' is composed of nonlinear components and recurrent loops, where the nonlinearity allows RC to process complex problems and the recurrent loops enable RC with memory. The 'computing' is achieved by reading out the states in the reservoir through learned NN layers. The training of this framework is conducted only on the readout layers which fundamentally circumvents gradient vanishing/explosion issues in back-propagation through time thanks to the fixed reservoir dynamics.

A vanilla discrete-time realization of RC is characterized by a state equation and an output equation. The state equation is formulated with time index t by,

$$\boldsymbol{s}(t+1) = \sigma\left(\boldsymbol{W}_{tran}\begin{bmatrix}\boldsymbol{s}(t)\\ \boldsymbol{y}(t)\end{bmatrix}\right) \tag{4.5}$$

where σ is a nonlinear function, $\boldsymbol{s}(t)$ is a vector representing the internal reservoir state, $\boldsymbol{y}(t)$ is the input vector, and $\boldsymbol{W}_{tran}$ stands for the reservoir weight matrix which is often chosen with a spectral radius smaller than 1 in order to asymptotically achieve a "similarity" to desired dynamic model[56]. The output equation is simply treated as

$$\boldsymbol{z}(t) = \boldsymbol{W}_{out} \begin{bmatrix} \boldsymbol{s}(t) \\ \boldsymbol{y}(t) \end{bmatrix}, \tag{4.6}$$

where $\boldsymbol{W}_{out}$ is the output weight matrix and $\boldsymbol{z}(t)$ stands for the output. As we can see, the output is with a skip-connection to the input which assimilates to a residual arrangement [10].

4.3 Multi-Mode Reservoir Computing

In this section, we introduce the framework of Multi-Mode RC. It processes a sequence-in and sequence-out task, where the time sequences are configured with more than one explicit modes, i.e., the input sequence is formulated as $\boldsymbol{Y}(t)$ or $\boldsymbol{\mathcal{Y}}(t)$ rather than a scalar-wise sequence $y(t)$ or a vector-wise one $\boldsymbol{y}(t)$.

4.3.1 Two-Mode Reservoir Computing

For ease of discussion, we begin by considering a two-mode RC. The architecture is comprised of a recurrent module, a feature queue, and an output mapping.

Recurrent Module: A recurrent module maps input sequence $\boldsymbol{Y}(t) \in \mathbb{C}^{N_{in-1} \times N_{in-2}}$ to a state sequence $\boldsymbol{S}(t) \in \mathbb{C}^{N_s \times N_s}$, where N_s represents the number of neurons defined on each mode of $\boldsymbol{S}(t)$[1]. Therefore, the total number of neurons is N_s^2. To equally obtain observations

[1] For simplicity, we assume $\boldsymbol{S}(t)$ as a square matrix. In general, it also can be designed as a non-square matrix. Meanwhile, the size of N_s is configured through experiments in order to maintain a balance between overfitting and underfitting according to the datasets. The RC structure based on non-square $\boldsymbol{S}(t)$ is left as our future work.

from the row-space and column-space of $\boldsymbol{Y}(t)$, we define the recurrent equation as,

$$\boldsymbol{S}(t+1) = \sigma\left(\boldsymbol{W}_{tran-1}\begin{bmatrix}\boldsymbol{S}(t), & \boldsymbol{O}\\ \boldsymbol{O}, & \tilde{\boldsymbol{Y}}(t)\end{bmatrix}\boldsymbol{W}_{tran-2}^{T}\right) \tag{4.7}$$

where

$$\tilde{\boldsymbol{Y}}(t) = \text{blockdiag}(\boldsymbol{Y}(t), \boldsymbol{Y}(t-1), \cdots, \boldsymbol{Y}(t-T')),$$

T' is a hyper-parameter representing the length of input window, σ is a non-linear function, $\boldsymbol{W}_{tran-1} \in \mathbb{C}^{N_s \times (N_s + T' N_{in-1})}$, $\boldsymbol{W}_{tran-2} \in \mathbb{C}^{N_s \times (N_s + T' N_{in-2})}$ are reservoir weight matrices applied on the row and column spaces respectively. Note that (4.7) also can be written as a sum as the form of (4.4). Accordingly, the state equation (4.7) can be regarded as an extension of the standard state equation (5.13) by incorporating independent mappings into the row and column spaces of state $\boldsymbol{S}(t)$ and input $\boldsymbol{Y}(t)$. It also can be equivalently written via the form of (5.13) through vectorizing the matrix-wise state and input. Rather than directly applying vectorized state and input to the standard RC, our introduced approach preserves the multi-mode feature of the input signal, where the advantages of using this strategy will be discussed in the analysis and evaluation sections of this paper.

Feature Queue: Our definition of the feature queue $\boldsymbol{G}(t)$ is a queue of sequence, i.e., at a given time t, the sample $\boldsymbol{G}(t)$ is a queue which is stacked up by current state sample $\boldsymbol{S}(t)$ and input sample $\tilde{\boldsymbol{Y}}(t)$. We opt for a simple formulation and use diagonalizing operation to write the queue as follows,

$$\boldsymbol{G}(t) = \text{blockdiag}(\boldsymbol{S}(t), \boldsymbol{S}^{T}(t), \tilde{\boldsymbol{Y}}(t), \tilde{\boldsymbol{Y}}^{T}(t)). \tag{4.8}$$

$\boldsymbol{G}(t)$ is also called extended state sequence as it is obtained with a skip connection to the input. The presence of $\tilde{\boldsymbol{Y}}^T(t)$ and $\boldsymbol{S}^T(t)$ is to create a fair treatment on the row and column space of $\tilde{\boldsymbol{Y}}(t)$ and $\boldsymbol{S}(t)$.

Output Mapping: An output layer ensures the feature queue can be identically mapped back to our desired output size. It is defined as:

$$\begin{aligned} \boldsymbol{Z}(t) &= \boldsymbol{W}_{out-1}\boldsymbol{G}(t)\boldsymbol{W}_{out-2}^T \\ &= \boldsymbol{G}(t) \times_1 \boldsymbol{W}_{out-1} \times_2 \boldsymbol{W}_{out-2} \end{aligned} \tag{4.9}$$

where $\boldsymbol{W}_{out-1} \in \mathbb{C}^{N_{out-1} \times N_{f-1}}$ and $\boldsymbol{W}_{out-2} \in \mathbb{C}^{N_{out-2} \times N_{f-2}}$; $N_{f-1} := 2N_s + T'(N_{in-1} + N_{in-2})$ and $N_{f-2} := 2N_s + T'(N_{in-1} + N_{in-2})$ respectively represent the size of row and column of $\boldsymbol{G}(t)$; Meanwhile, N_{out-1} and N_{out-2} respectively stand for the size of row and column of $\boldsymbol{Z}(t)$.

Loss Function: In this paper, the loss function is defined to handle sequence-to-sequence tasks. Given a set of $\{\boldsymbol{Y}_q(t), \boldsymbol{Z}_q(t)\}$ as the input-output pairs for training, where q stands for the batch index, our objective aims to generate $\boldsymbol{Z}_q(t)$ by using $\boldsymbol{Y}_q(t)$. Therefore, we use

$$\min_{\boldsymbol{W}_{out-1}, \boldsymbol{W}_{out-2}} \sum_{q=1}^{N_K} \sum_{t=1}^{N_T} \|\boldsymbol{Z}_q(t) - \boldsymbol{G}_q(t) \times_1 \boldsymbol{W}_{out-1} \times_2 \boldsymbol{W}_{out-2}\|_F^2, \tag{4.10}$$

where $\|\cdot\|_F$ is the Frobenius norm of a matrix. Although the loss function is simply formulated via a least square framework, it offers a connection between the RC readout learning and alternating least squares (ALS) algorithm which has been widely used and can be easily analyzed in the context of tensor decomposition[20]. We can further stack $\boldsymbol{Z}_q(t)$ and $\boldsymbol{G}_q(t)$ to 4-mode tensors along the time axis and the batches to have $\boldsymbol{\mathcal{Z}} \in \mathbb{C}^{N_{out-1} \times N_{out-2} \times N_T \times N_K}$ and $\boldsymbol{\mathcal{G}} \in \mathbb{C}^{N_{f-1} \times N_{f-2} \times N_T \times N_K}$ respectively. Accordingly, the loss objective (4.10) can be rewritten in concise way,

$$\min_{\boldsymbol{W}_{out-1}, \boldsymbol{W}_{out-2}} \|\boldsymbol{\mathcal{Z}} - \boldsymbol{\mathcal{G}} \times_1 \boldsymbol{W}_{out-1} \times_2 \boldsymbol{W}_{out-2}\|_F^2. \tag{4.11}$$

In the training stage, we feed a batch of sequences to RC and solve the problem (4.11) using alternating least squares, where $\boldsymbol{W}_{out-1}$ and $\boldsymbol{W}_{out-2}$ are iteratively updated by solving the following matrix-wise least square problems,

$$\boldsymbol{W}_{out-1} = \arg \min_{\boldsymbol{W}_{out-1}} \|\boldsymbol{Z}_{(1)} - \boldsymbol{W}_{out-1}\boldsymbol{G}_{(1)}(\boldsymbol{W}_{out-2} \otimes \boldsymbol{I}_{N_T} \otimes \boldsymbol{I}_{N_K})^T\|_F$$

$$\boldsymbol{W}_{out-2} = \arg \min_{\boldsymbol{W}_{out2}} \|\boldsymbol{Z}_{(2)} - \boldsymbol{W}_{out-2}\boldsymbol{G}_{(2)}(\boldsymbol{W}_{out-1} \otimes \boldsymbol{I}_{N_T} \otimes \boldsymbol{I}_{N_K})^T\|_F.$$

The iterative process continues until a certain stopping criterion is reached. In this ALS formulation, $\boldsymbol{G}_{(1)}$ and $\boldsymbol{G}_{(2)}$ represent the mode-1 and mode-2 unfoldings of tensor $\boldsymbol{\mathcal{G}}$. However, directly using this ALS calculation often requires large memory resources due to the Kronecker products. Therefore, we introduce an alternative approach to calculate the ALS which is discussed in Appendix.

Since reaction delays exist in RC systems[99], we often need to add another parameter τ, namely "Delay of RC states" in the loss objective to optimize. Therefore, we have the following augmented loss objective,

$$\min_{\tau < \tau_{max}} \min_{\boldsymbol{W}_{out-1}, \boldsymbol{W}_{out-2}} \sum_{q=1}^{N_K} \sum_{t=1}^{N_T} \|\boldsymbol{Z}_q(t) - \boldsymbol{G}_q(t+\tau) \times_1 \boldsymbol{W}_{out-1} \times_2 \boldsymbol{W}_{out-2}\|_F^2, \tag{4.12}$$

where τ_{max} represents the upper bound of τ to search. Accordingly, the samples of $\boldsymbol{G}_q(t)$ ranging from time index $t = 1$ to $t = N_T + \tau_{max}$ are obtained by using $\{\boldsymbol{Y}(t)\}_{t=1}^{N_T}$ with a τ_{max}-length zero paddings at the end as the RC input. At testing stage, the learned τ are applied to truncate the RC state sequence such that the output sequence becomes

Table 4.1: Notations of Multi-Mode RC

Notations	Definitions
T'	Input window length
N	Number of signal mode in Multi-Mode RC
N_{in-n}	Number of RC input at mode n
N_{out-n}	Number of RC output at mode n
N_{f-n}	Feature queue size at mode n
N_T	Input and output sequence length
N_K	Training batch size
N_s	Number of neurons on each mode of a state tensor
τ	Delay configuration in RC state response
$\boldsymbol{\mathcal{Y}}(t) \in \mathbb{C}^{N_{in1}\times N_{in-2}\times\cdots\times N_{in-N}}$	The input of Multi-Mode RC
$\boldsymbol{\mathcal{G}}(t) \in \mathbb{C}^{N_{f-1}\times N_{f-2}\times\cdots\times N_{f-N}}$	The internal feature of Multi-Mode RC
$\boldsymbol{\mathcal{Z}}(t) \in \mathbb{C}^{N_{out-1}\times N_{out-2}\cdots\times N_{out-N}}$	The output of RC
$\boldsymbol{\mathcal{Y}} \in \mathbb{C}^{N_{in1}\times N_{in2}\times\cdots\times N_{in-N}\times N_T\times N_K}$	A higher order tensor by stacking $\boldsymbol{\mathcal{Y}}(t)$
$\boldsymbol{\mathcal{G}} \in \mathbb{C}^{N_{f-1}\times N_{f-2}\times\cdots\times N_{f-N}\times N_T\times N_K}$	A higher order tensor by stacking $\boldsymbol{\mathcal{G}}(t)$
$\boldsymbol{\mathcal{Z}} \in \mathbb{C}^{N_{out-1}\times N_{out-2}\cdots N_{out-N}\times N_T\times N_K}$	A higher order tensor by stacking $\boldsymbol{\mathcal{Z}}(t)$

$\{\boldsymbol{G}(t+\tau)\times_1 \boldsymbol{W}_{out-1}\times_2 \boldsymbol{W}_{out-2}\}_{t=1}^{N_T}$, since the output is anticipated as a N_T-length sequence. In general, the output sequence can be truncated off more or less samples in order to match the desired sequence of the tasks.

4.3.2 Multi-Mode Reservoir Computing

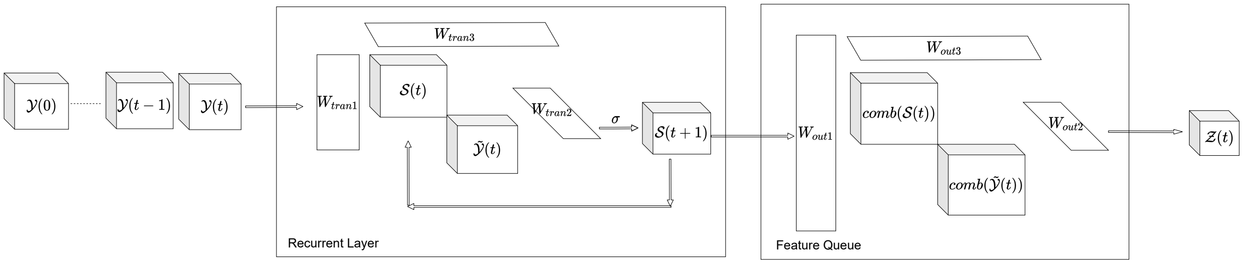


Figure 4.3: Illustration of Three-Mode Reservoir Computing Architecture

The structure of multi-mode (beyond 2-mode) reservoir computing is illustrated in Fig. 4.3.

As a general framework of the 2-mode RC, each component is respectively extended as

- **Recurrent Module:**

$$\begin{aligned} \boldsymbol{\mathcal{S}}(t+1) &= \sigma(\text{superblockdiag}(\boldsymbol{\mathcal{S}}(t), \tilde{\boldsymbol{\mathcal{Y}}}(t)) \times_1 \boldsymbol{W}_{tran-1} \times_2 \cdots \times_N \boldsymbol{W}_{tran-N}) \\ \tilde{\boldsymbol{\mathcal{Y}}}(t) &= \text{superblockdiag}(\boldsymbol{\mathcal{Y}}(t), \cdots, \boldsymbol{\mathcal{Y}}(t-T')) \end{aligned} \tag{4.13}$$

- **Feature Queue**:

$$\boldsymbol{\mathcal{G}}(t) = \text{superblockdiag}(\text{comb}(\boldsymbol{\mathcal{S}}(t)), \text{comb}(\tilde{\boldsymbol{\mathcal{Y}}}(t))) \tag{4.14}$$

  where $\text{comb}(\boldsymbol{\mathcal{S}}(t)) := \text{superblockdiag}(\boldsymbol{\mathcal{S}}(t), \boldsymbol{\mathcal{S}}^{T_{\Pi_1}}(t), \boldsymbol{\mathcal{S}}^{T_{\Pi_2}}(t) \cdots)$, Π_1, $\Pi_2 \cdots$ stand for permutation patterns which are up to $N!$ cases.

- **Output Layer**:

$$\boldsymbol{\mathcal{Z}}(t) = \boldsymbol{\mathcal{G}}(t) \times_1 \boldsymbol{W}_{out-1} \times_2 \boldsymbol{W}_{out-2} \cdots \times_N \boldsymbol{W}_{out-N} \tag{4.15}$$

- **Loss Function**:

$$\min_{\tau} \min_{\boldsymbol{W}_{out-1}, \boldsymbol{W}_{out-2}, \cdots, \boldsymbol{W}_{out-N}} \sum_{q=1}^{N_K} \sum_{t=1}^{N_T} \| \boldsymbol{\mathcal{Z}}_q(t) - \boldsymbol{\mathcal{G}}_q(t+\tau) \times_1 \boldsymbol{W}_{out-1} \cdots \times_N \boldsymbol{W}_{out-N} \|_F^2 \tag{4.16}$$

Similarly to the 2-mode case, the output weights are learned through alternating least squares. The optimization (4.16) can also be formulated as a high order tensor decomposition as defined in (4.11). Moreover, to further avoid model overfitting, regularization terms can be added in the loss function, such as ridge regression, i.e., $\|\boldsymbol{W}_{out-1}\|_F^2 + \|\boldsymbol{W}_{out-2}\|_F^2 + \cdots + \|\boldsymbol{W}_{out-N}\|_F^2$. In addition, we can observe that the optimization problem (4.16) is not the canonical Tucker decomposition defined in [19]. This is because the factor matrices are not designed as full column-rank in our framework. On the contrary, we choose the factor matrices with full row-rank to fulfill the mechanism of RC that is "yielding desired output through dimension reduction from internal memory states."

4.3.3 Theoretic Analysis

We now study the condition on the uniqueness of solving (4.16) via alternating least squares. Through our derivation as presented in Appendix, we can arrive at the following theorem.

Theorem 4.1. *Given* $\boldsymbol{\mathcal{Z}} \in \mathbb{C}^{N_{out-1} \times N_{out-2} \times \cdots \times N_{out-N} \times N_T \times N_K}$ *and* $\boldsymbol{\mathcal{G}} \in \mathbb{C}^{N_{f-1} \times N_{f-2} \times \cdots \times N_{f-N} \times N_T \times N_K}$ *with rank-*$(N_{out-1}, N_{out-2}, \cdots, N_{out-N}, N_T, N_Q)$ [2] *and rank-*$(N_{f-1}, N_{f-2}, \cdots, N_{f-N}, N_T, N_K)$ *respectively, and* $\forall n$, $N_{f-n} \geq N_{out-n}$, $N \geq 2$, *the achieved minimization of (4.16) is unique by using ALS when the initialization factor matrices are chosen as full rank and*

$$\sum_{i \neq n} N_{out-i} + N_T + N_K \geq N_{f-n}, \forall n. \tag{4.17}$$

The above theorem reveals that the uniqueness condition of the Multi-Mode RC learning is characterized by the shape of the output tensor and feature core-tensor. Alternatively, if we use a single batch for training, i.e., merge the last two modes of the tensors into one, the shape of the output tensor and core-tensor respectively become $\boldsymbol{\mathcal{Z}} \in \mathbb{C}^{N_{out-1} \times N_{out-2} \times \cdots \times N_{out-N} \times (N_T N_K)}$ and $\boldsymbol{\mathcal{G}} \in \mathbb{C}^{N_{f-1} \times N_{f-2} \times \cdots \times N_{f-N} \times (N_T N_K)}$. Therefore, the uniqueness condition can be rewritten as

$$\sum_{i \neq n} N_{out-i} + N_T N_K \geq N_{f-n}, \forall n. \tag{4.18}$$

[2] This stands for the multi-mode rank of a tensor. The definition can be found in Appendix.

In our experiments, we observe that when the uniqueness condition holds, the loss-value is often greatly larger than zero. The model thereby does not over-fit to the training dataset which can offer generalization on the unseen testing dataset. More related discussions on this observation are in the evaluation section.

On the other hand, Multi-Mode RC can be analyzed as an advance of conventional RC by imposing particular structures on the output layer, where the conventional RC refers to RC operating on single-mode data structures, i.e., scalars and vectors. To gain this insight, we consider vectorizing the output layer of a 2-mode RC. According to (4.9), the resulting output equation of the 2-mode RC via a single-mode RC based formulation is given by,

$$Vec(\boldsymbol{Z}(t)) = (\boldsymbol{W}_{out-2} \otimes \boldsymbol{W}_{out-1})\, Vec(\boldsymbol{G}(t)).$$

The above equation reveals that the output weight of multi-mode RC is forged as a Kronecker product of two sub-matrices to process a "vectorized" $\boldsymbol{G}(t)$. As opposed to single-mode RC using a fully connected layer, the resulting Kronecker output layer is with less freedom on parameters which requires a less amount of data to fit. Meanwhile, the Kronecker structure can further reduce time and space complexity. We present the complexity analysis results in Table D.1, where we assume that the conventional RC and Multi-Mode RC are with the same input-output size. In this table, the time complexity of the forward path is calculated by the matrix product between the output layer and RC state at each sample, while the complexity on output learning is from the matrix inverse operations involved in solving the loss objectives of the entire training data set. The memory costs in the forward path are calculated based on the size of internal states and output weights. Meanwhile, the memory spent on learning is on the same scale as the size of the internal state. Moreover, the input buffer length is ignored in this table for simplicity. However, it can be easily calculated by substituting N_{in-n} as $N_{in_n}T'$ in this table.

Table 4.2: Time Complexity and Memory Usage Comparison

NN Operations	Time
Standard RC forward	$\mathcal{O}(\prod_n N_{out-n}(\prod_n N_{in-n} + N_s^N))$
Multi-Mode RC forward	$\mathcal{O}(\sum_n N_{out-n}(\prod_n N_{in-n}N! + N_s^N))$
Standard RC learning	$\mathcal{O}((\prod_n N_{in-n} + N_s^N)^3 N_K)$
Multi-Mode RC learning	$\mathcal{O}(\sum_n N_{f-n}^3 \prod_n N_{out-n} N_K)$
	Memory
Standard RC forward	$\mathcal{O}(\prod_n N_{out-n}(\prod_n N_{in-n} + N_s^N))$
Multi-Mode RC forward	$\mathcal{O}(\sum_n N_{f-n} N_{out-n} + \prod_n N_{in-n}N! + N_s^N)$
Standard RC learning	$\mathcal{O}((\prod_n N_{in-n} + N_s^N) N_K)$
Multi-Mode RC learning	$\mathcal{O}(\prod_n N_{in-n}N! + N_s^N)$

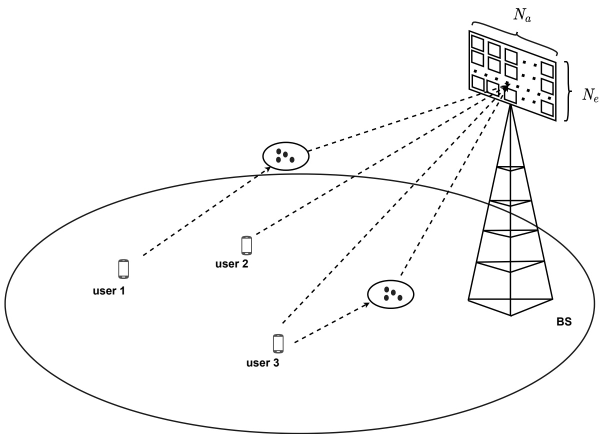


Figure 4.4: Uplink transmission in the multi-user massive MIMO system.

4.4 Application: Online Symbol Detection for Multi-user Massive MIMO

In this section, we will briefly review the transceiver architecture of multi-user massive MIMO-OFDM systems and elaborate on how to apply Multi-Mode RC to symbol detection of an **uplink** massive MIMO network.

4.4.1 Multi-user Massive MIMO-OFDM System

We assume N_u scheduled users are distributed in a cell communicating to a base station (BS) equipped with a massive rectangular array as shown in Fig. 4.4, where each user is mounted with N_q antennas. The transmitted signals from all users to BS can be written as $\boldsymbol{X}(t) \in \mathbb{C}^{N_u \times N_q}$. Let $\boldsymbol{x}(t) = \text{vec}(\boldsymbol{X}(t)) \in \mathbb{C}^{N_t \times 1}$, where $N_t = N_u N_q$. Each entry of $\boldsymbol{x}(t)$ is a time sequence which stands for a stream of OFDM signals. For convenience, the OFDM signal $\boldsymbol{x}(t)$ is organized as OFDM resource grids as illustrated in Fig. 4.5. In the OFDM resource grids, the horizontal direction represents OFDM symbols, while the vertical direction stands for sub-carriers indices. OFDM symbols are constructed into subframes where the time domain signal $\boldsymbol{x}(t)$ is obtained by applying an inverse Fourier transform (IFFT) on symbols across the subcarriers. Cyclic-prefix (CP) is added for each OFDM symbol. Let N_c denote the number of subcarriers and $N_D + N_K$ be the number of OFDM symbols within a subframe. Here, N_K is the number of OFDM symbols that are used as pilots/reference signals whereas N_D OFDM symbols within the same subframe are used for data transmission. Each element on the resource grids is modulated by quadrature amplitude modulation (QAM). Note that pilots/reference signals are used to conduct CSI estimation at the receiver in modern 4G and 5G networks. Furthermore, N_K **is configured to be smaller than** N_D to reduce over-the-air signaling overhead. Meanwhile, N_K is often designed to be proportional to the number of streams to offer a reliable CSI estimation.

The received signal at the BS can be expressed as the following,

$$\boldsymbol{Y}(t) = r\left(\sum_{\ell=0}^{L} \boldsymbol{\mathcal{H}}(\ell) \times_3 \boldsymbol{x}(t-\ell) + \boldsymbol{\mathcal{N}}(t)\right) \tag{4.19}$$

where $r(\cdot)$ is a function which characterizes the non-linearity at the receiver, such as ADCs, as well as model mismatch; $\boldsymbol{\mathcal{H}}(\ell) \in \mathbb{C}^{N_a \times N_e \times N_t}$ is a tensor which defines a spatial channel

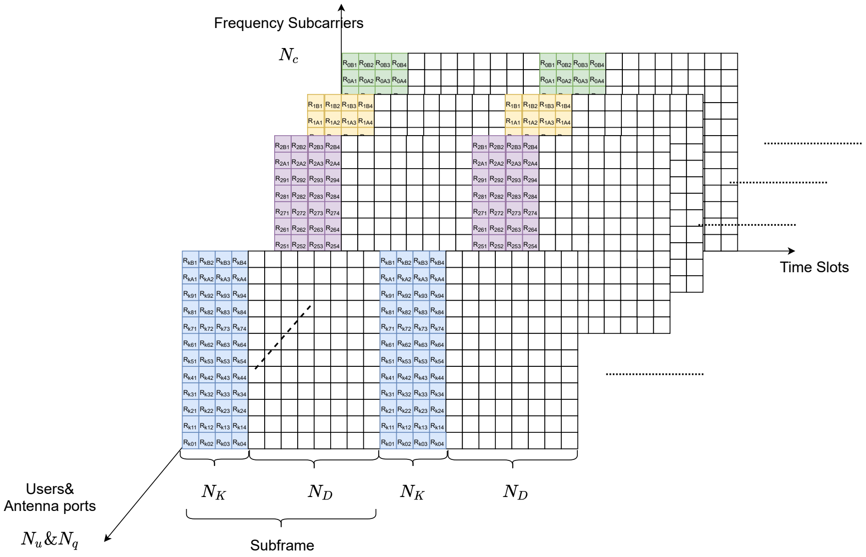


Figure 4.5: Massive MIMO-OFDM resource grids (subframe-subcarrier) structure for RC training and symbol detection

response at the ℓth-delay, where the total number of delays is denoted as L; N_a is the number of azimuth antennas, N_e is the number of elevation antennas of the massive MIMO BS antenna array. Our objective is to train a NN $\mathcal{D}$ which can recover $\boldsymbol{X}(t)$ by using $\boldsymbol{Y}(t)$, i.e.,

$$\mathcal{D}(\boldsymbol{Y}(t)) = \boldsymbol{X}(t), \tag{4.20}$$

such that the NN is learned by

$$\min_{\mathcal{D}} f\left(\{\mathcal{D}(\boldsymbol{Y}(t))\}, \{\boldsymbol{X}(t)\}\right). \tag{4.21}$$

4.4.2 Online Symbol Detection by Multi-Mode RC

Multi-Mode RC serves as the detection NN, $\mathcal{D}$, through the over-the-air pilots/reference signals on a subframe basis. The pilots defined in existing massive MIMO-OFDM systems of each subframe is directly utilized as the training dataset. The following data symbols in the same subframe is the testing dataset, i.e., we train the NN using N_K pilots to detect N_D data symbols in each subframe. This constraint makes the learning framework different from conventional NNs to enable online learning for robust and adaptive communications. The azimuth direction is set as the first mode of Multi-Mode RC and the elevation direction is set as the second mode. Each OFDM pilot symbol is considered as one training batch. Accordingly, the input sequence length equals the number of subcarriers, N_c, plus CP. The output is then truncated to be a N_c-length sequence following the process as described under equation (4.12). The symbols of each stream on each subcarrier are obtained through quantization and demodulation.

In massive MIMO systems, symbol detection can be conducted through either a joint or a decomposed approach:

Joint Processing

In the joint processing, data symbols are obtained through a single Multi-Mode RC with a multi-head output, where the size of the first and second mode of the RC output sequence are respectively N_q and N_u. The mode order of the output node also can be reversely configured. This is because Multi-Mode RC treats equally on each output mode according to the generation rule of the internal feature queue as shown in (4.14). A well trained joint model is anticipated to yield a good symbol detection performance since all interference and imperfect factors are handled jointly.

Table 4.3: Notations of Multi-Mode RC based MIMO-OFDM Symbol Detection

Notations	Definition	Corresponding notations in Multi-Mode RC
N_t	Number of antennas stacked from all users	N/A
N_u	Number of users	N_{out-1} in joint processing
N_q	Number of antennas at each user	N_{out-2} in joint processing
N_a	Number of azimuth antennas	N_{in1}
N_e	Number of elevation antennas	N_{in2}
N_c	Number of OFDM sub-carriers	N_T
N_K	Number of pilot symbols in a frame	N_K-Training batches
N_D	Number of data symbols in a frame	Testing batches
$\boldsymbol{x}(t) \in \mathbb{C}^{N_t \times 1}$	Transmitted Signal	Desired output
$\boldsymbol{X}(t) \in \mathbb{C}^{N_u \times N_s}$	Transmitted Signal	Desired output
$\boldsymbol{Y}(t) \in \mathbb{C}^{N_a \times N_e}$	Received Signal	Input
$\boldsymbol{\mathcal{Y}} \in \mathbb{C}^{N_a \times N_e \times T}$	Stacked tensor of received signal	Input

Decomposed Processing

The decomposed approach refers to learning the output weight through a decomposed way. For instance, in the case of 2-mode RC, it has $N_u \times N_q$ pairs of $(\boldsymbol{w}_{out-1}, \boldsymbol{w}_{out-2})$ to learn, where the vector weight $\boldsymbol{w}$ maps the internal states to a scalar entry of the output tensor sequence. In this framework, the training on each decomposed output weight is based on their individual loss allowing the training through a parallel manner which can significantly reduce the computation latency. On the other hand, the decomposed method takes extra resources on storage and computation compared to the joint approach. For instance, in 2-mode RC based joint approach, the size of output weight matrices $\boldsymbol{W}_{out-1}$ and $\boldsymbol{W}_{out-2}$ are respectively $N_{out-1} \times N_{f-1}$ and $N_{out-2} \times N_{f-2}$. In the decomposed way, the shapes of output mapping on each mode are respectively $1 \times N_{f-1}$ and $1 \times N_{f-2}$. Thus, there are $N_{out1}N_{out2} \times (N_{f-1} + N_{f-2})$ output weights in total for the decomposed approach which is higher than $N_{out-1} \times N_{f-1} + N_{out-2} \times N_{f-2}$ from the joint way. Overall, the map from MIMO-OFDM parameters to the Multi-Mode RC parameters is summarized in Table 4.3.

4.5 Performance Evaluations

This section provides performance evaluations of the introduced Multi-Mode RC for uplink symbol detection in a multi-user massive MIMO-OFDM scenario. We choose uncoded bit error rate (BER) as the quantitative metric to evaluate the reliability of the underlying link. Table 4.3 contains simulation parameters of the massive MIMO-OFDM system. The default system configuration is: $N_a = 8$, $N_e = 8$, $N_c = 512$, $N_{cp} = 32$, $N_q = 2$, $N_u = 2$, $N_K = 4$ and $N_D = 12$. Note that in this setting, the pilots/reference signals overhead is 25% which is inline with 5G standards [4]. The channel coefficients are generated according to the clustered delay line (CDL) model defined in 3GPP Technical Report (TR) 38.901, where the transmitter and receiver are configured with uniform linear arrays having half-wavelength antenna spacing, and the power delay profile is configured with a cluster delay rate of 3. The maximum delay spread of the channel is set as the length of the CP in OFDM. Each obtained BER point is collected over 100 consecutive subframes. SNR is defined as the average power ratio between noise-free received signal and the additive noise. The configuration of the Multi-Mode RC is set as follows: $T' = N_{CP}$, $N_s = 8$, the number of ALS iterations is set as 6, the state transition matrix $\boldsymbol{W}_{tran}$ on each mode is independently generated with its spectral radius less than 1. Meanwhile, the input weights matrix $\boldsymbol{W}_{in}$ is generated independently for each mode from a uniform distribution on $[-1, 1]$.

4.5.1 Parsing Multi-Mode RC

We first parse various components of a Multi-Mode RC to offer more insights on the underlying NN structure. We consider three types of Multi-Mode RC in the evaluation: 1) Our introduced one; 2) Our introduced one without using the tensor permutation to construct the feature queue in (4.14); 3) Our introduced one using a large number of N_s, where $N_s = 128$.

Fig. 4.6 and Fig. 4.7 respectively show the training and testing BERs under different numbers of iteration when $SNR = 15$dB. As we can observe that without tensor permutation in the feature queue, the RC performs underfitting to the task. This is because the feature from different modes of the input tensor sequence has not been equally extracted. Meanwhile, if we increase the number of neurons, the NN model complexity increases. Accordingly, it “overfits” the training data as the training BER dramatically decreases, whereas the testing BER increases.

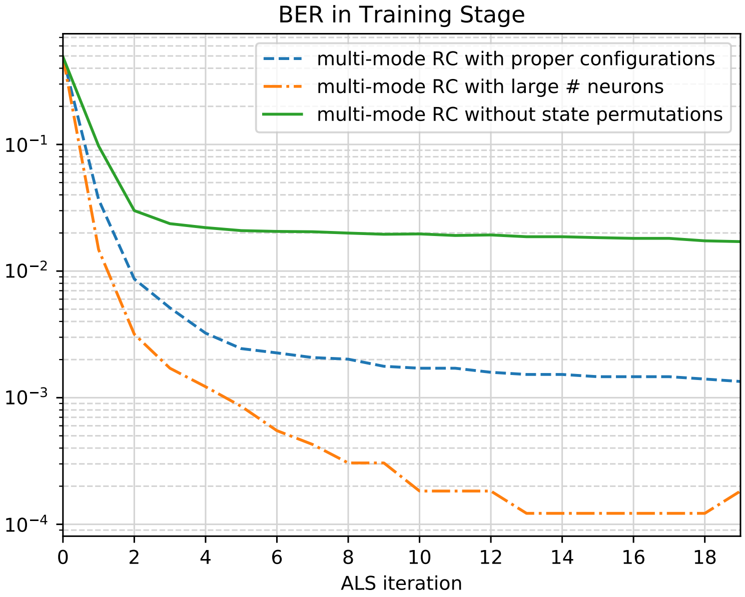


Figure 4.6: Training BER of multi-mode RC with respect to iterations in ALS.

4.5.2 Uniqueness Conditions

Now, we investigate how the uniqueness condition defined in Theorem 1 determines the training and testing performance. For convenience, we set $N_c = 64$ and use single-batch based training in this evaluation. According to (4.18), the critical conditions for this task

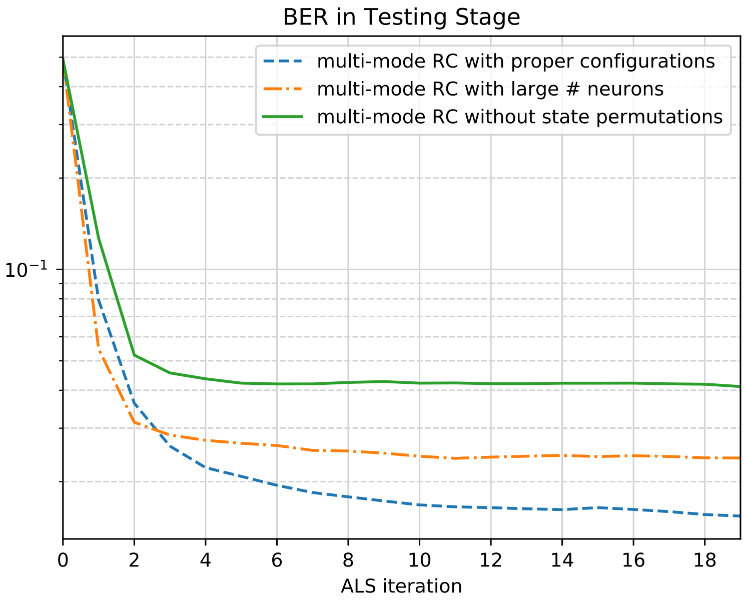


Figure 4.7: Testing BER of Multi-Mode RC with respect to iterations in ALS.

becomes,

$$N_u + N_T N_K == 2N_s + N_a T' + N_e T'$$

$$N_q + N_T N_K == 2N_s + N_a T' + N_e T'$$

When we only change parameters T' and N_s while fixing the rest based on our default setup, the critical conditions become $130 == 8T' + N_s$. This condition is plotted as the dashed red line in Fig. 4.8 and Fig. 4.9. Meanwhile, we also plot the contours of the log-loss as well as the BER in the same (N_s, T') plane. As shown in Fig. 4.8, the loss is guaranteed to be greater than a threshold, e.g., -4.00, when the uniqueness condition holds. On the other hand, when the condition is violated, the loss tends to be close to zero. In this case, the RC model overly fits to the training data which brings a high risk of over-fitting. This result is also consistent with the BER contour plotted in Fig. 4.8. Note that even though satisfying the uniqueness condition can potentially avoid overfitting, it may cause underfitting as we can observe the high BER below the condition line in Fig. 4.9.

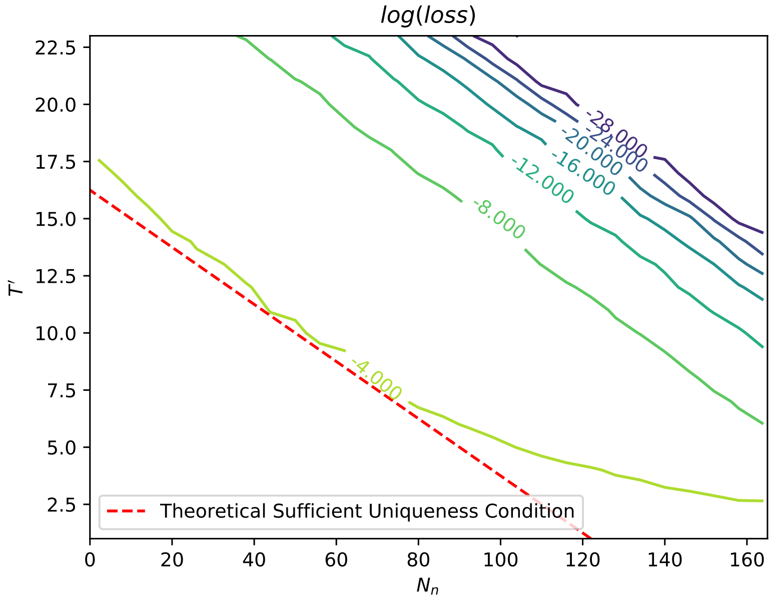


Figure 4.8: Log loss contour in (N_s, T') plane in training stage

4.5.3 Comparison with State-of-Art Detection Strategies

We now investigate the BER versus SNR using different approaches. In Fig. 4.10, the compared methods are: 1) LMMSE+LMMSE-CSI which uses linear minimum mean square error (LMMSE) based symbol detection under the LMMSE estimated CSI. 2) SD+LMMSE-CSI which uses sphere decoding for symbol detection based on the LMMSE estimated CSI. 3) Large-window RC refers to the windowed echo state network (WESN) introduced in [100] by vectorizing the input as a vector and setting the input buffer size as 52. 4) Ridge-large-window RC refers to the same WESN but using l_2 norm as a penalty term to the output weights in the loss function. 5) Multi-Mode RC is the introduced method. 6) Ridge-Multi-Mode RC standards for the same Multi-Mode RC but also adding a l_2 norm as the regularization on the output weights in the loss objective. Fig. 4.10 clearly demon-

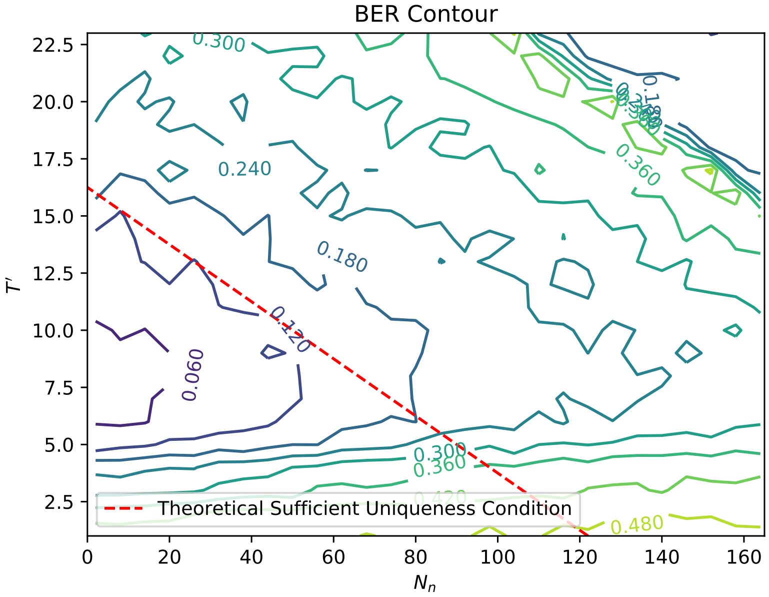


Figure 4.9: BER contour in (N_s, T') plane in testing stage

strates the performance gain of the Multi-Mode RC over the signal processing-based methods (LMMSE+LMMSE-CSI and SD+LMMSE-CSI). Meanwhile, we can see that the Multi-Mode RC is more robust than the single-mode RC as the multi-mode feature of MIMO-OFDM signals is leveraged for the symbol detection.

In addition, we investigate the BER performance by increasing the array size. Fig. 4.11 shows that when the number of antennas increases (e.g improve the antenna array from 8 × 8 to 10 × 10), the BER curves of all methods are improved. On the other hand, the Multi-Mode RC continues showing its advantage over other methods.

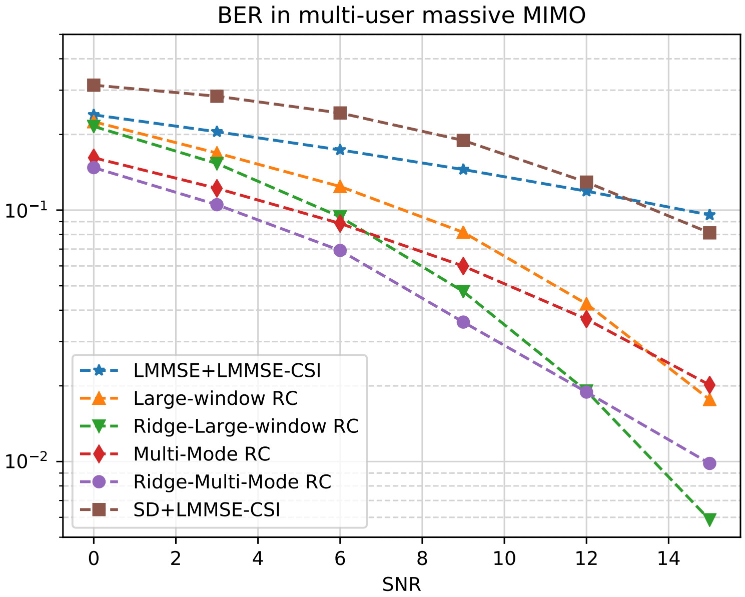


Figure 4.10: BER in a multi-user massive MIMO system with 64 antennas at the BS (8 × 8 antenna array).

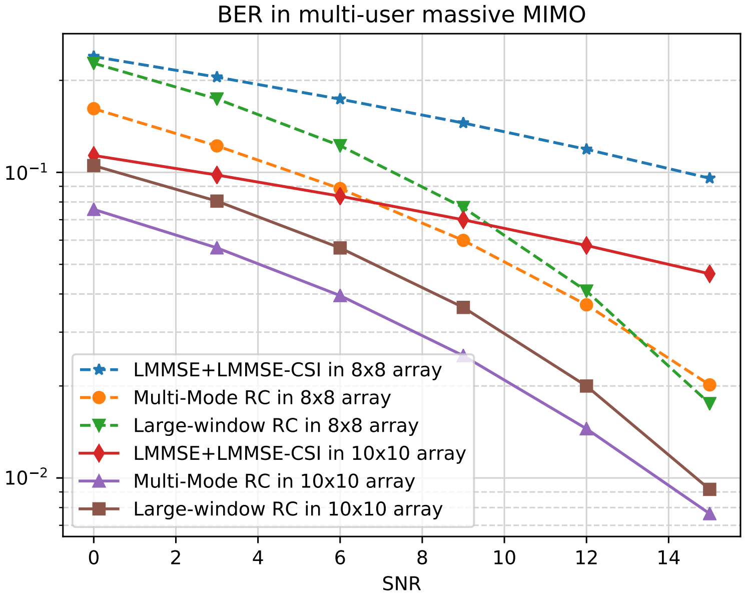


Figure 4.11: BER in multi-user massive MIMO with 64 (8 × 8) and 100 (10 × 10) antennas at the BS.

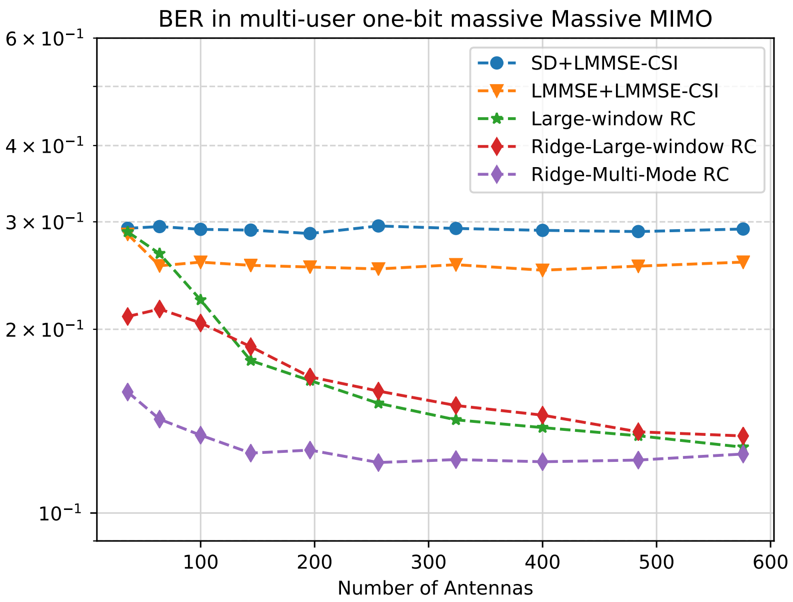


Figure 4.12: BER in one-bit multi-user massive MIMO systems with different antenna numbers.

4.5.4 Performance Evaluation under Receiving Non-linearity —- Low Resolution ADCs

To show the advantage of RC-based approach in other model mismatch scenarios, we evaluate the BER performance when low precision quantization is added in the link which is extremely relevant to massive MIMO systems. We consider the extreme case of using one-bit ADC which quantizes the in-phase and quadrature components to 1 or -1. The definition of the quantizer for any one of the components is,

$$q(x) = A_{max} \cdot \text{sign}(x) \tag{4.22}$$

where A_{max} is the maximum magnitude of the quantizer where we set it as 0.6 in the evaluation. Fig. 4.12 clearly shows that the Multi-Mode RC is the most robust method in

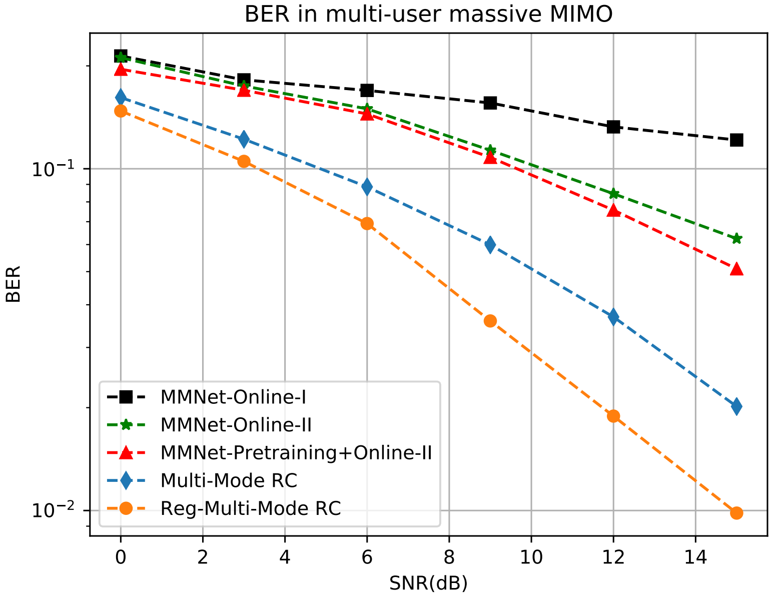


Figure 4.13: BER in a multi-user massive MIMO system with 64 antennas at the BS (8 × 8 antenna array).

this scenario. Meanwhile, we can observe the saturation phenomena of the BER curve when the antenna number increases. This is because the quantization level on each antenna is set as a fixed value. Intuitively, the performance can be further improved by optimizing the quantization level on each antenna. This will be considered as our future work.

4.5.5 Comparison with Model-based Learning Approaches

In this section, we compare the Multi-Mode RC against a state-of-the-art model-based symbol detection NN, MMNet [47]. MMNet is a deep NN structure based on unfolding iterative soft-thresholding algorithms, which adds degrees of flexibility on certain parameters in the NN for training. In our evaluation, MMNet is configured using the aforementioned training dataset associated with the LMMSE-estimated CSI. As the legends shown in Fig. 4.13, we choose three MMNet operation modes as the benchmark methods: 1) MMNet-Online-I contains only scalar trainable parameters. It assumes the channel additive noise in both testing and training stages are with homogeneous distributions. The "Online" scheme refers to a training framework described in the paper, by which a NN for conducting symbol detection on the first subcarrier is trained from scratch using 1000 iterations and N_K pilot symbols, while other NNs with respect to the remaining sub-carriers are fine-tuned based on the first NN with 3 additional iterations using their individual training symbols. 2) MMNet-Online-II has matrix-form parameters as noise variance estimators per layer, which is considered as an advanced structure of MMNet-Online-I using the same training strategy. 3) MMNet-Pretraining-Online-II adds a pre-trained training stage to MMNet-Online II. In this way, the NNs are initialized with pre-trained weights using 256 pilot symbols from historical training datasets with different channel realizations.

Note that, while MMNet achieves notable performance on both i.i.d Gaussian channels and spatially-correlated channels in [47], the utilized training symbols are relatively larger than the setup in this paper (e.g. 500 pilot symbols associated with perfect CSIs which is not consistent to the online over-the-air scenario as in our paper). Furthermore, the computation and memory requirements on MMNet are significantly higher than the introduced method. To be specific, MMNet requires N_c NNs to estimate symbols on all subcarriers, each of which stacks 10 layers of neurons. Therefore, the number of training iterations significantly increases. In our method, we only use 4 pilot symbols and a single NN to jointly accomplish the symbol detection task. Our evaluation results demonstrate that the Multi-Mode RC outperforms MMNet in the multi-user massive MIMO scenario with a steep performance improvement slope. Overall, Reg-Multi-Mode RC shows $5-6$ dB gain in SNR compared with MMNet.

4.6 Conclusion

In this paper, we presented a NN structure, Multi-Mode RC, for symbol detection in massive MIMO-OFDM systems. We elaborated on the NN architecture and its configuration for the symbol detection task. The introduced Multi-Mode RC framework is shown to be able to effectively cope with the model mismatch, waveform distortion as well as interference in the systems. Numerical results demonstrated the advantages of Multi-Mode RC in the following aspects: It can offer lower BER than conventional single-mode RC frameworks in low SNR regime while achieving reduced computational complexity. Compared to other model-based learning approaches, the introduced method can operate on a subframe-basis thus completely relying on the limited over-the-air pilots/reference symbols. This attractive feature enables us to train the symbol detection task using a compatible signal overhead as modern cellular networks.

In our future work, we will consider the optimization of the quantization thresholds at each antenna port. Since quantization is an irreversible process, adaptive quantization strategies are a promising approach to preserve the waveform information. Furthermore, incorporating gradient-free learning algorithms into the RC framework is another interesting direction.

Chapter 5

OTFS Symbol Detection via RC

5.1 Introduction

Telecommunication vendors are expanding their services to various new scenarios, such as the Internet of Things (IoT), the vehicle-to-everything (V2X), and the global coverage by non-terrestrial networks (NTN) including both high altitude platforms (HAPs) and low earth orbit (LEO) satellites in the 5G-Advanced era [26]. For many of those high mobility scenarios such as V2X and NTN, it is imperative to ask: Will the current 5G air-interface design be sufficient for these new scenarios? Will 5G-advanced and 6G benefit from applying an alternative waveform design to accommodate these new scenarios? The breakthrough by Orthogonal Time Frequency Space (OTFS) modulation [36] has shed light on these questions. It circumvents the fundamental bottlenecks in the time-frequency processing framework of Orthogonal Frequency Division Modulation (OFDM) which is the key physical layer waveform of 4G LTE-Advanced [55] and 5G NR [86]. Meanwhile, OTFS is a signal design framework/-control plane that improves network layer functions like scheduling that are essential to realizing the promise of 5G/6G. As an alternative to OFDM, OTFS is rooted in an "invariant expression" of the doubly-selective wireless channel via the delay-Doppler domain [36]. The resulting channel impulse response exhibits sparsity and persists for longer time than the time frequency expression. These channel features enable the algorithm and protocol design of OTFS systems in channel estimation/tracking and equalization with reduced signaling overhead. As opposed to the equalization benefits, it also can offer waveform benefits: the cyclic prefix (CP) of each OFDM symbol can be completely abolished in OTFS, significantly improving spectral efficiency especially for high mobility scenarios where CP is a significant system overhead for OFDM systems.

For high mobility scenarios with high Doppler shifts, transmitted symbols reside in the delay-Doppler domain experience a channel that changes more slowly compared with the counterparts in the time-frequency domain [35]. Motivated by this, OTFS is introduced by allocating modulation symbols in the delay-Doppler domain. Accordingly, the time-varying multi-path wireless channel becomes a near-constant impulse response in the delay-Doppler domain via the symplectic Fourier transformation (i.e., Zak transform). Channel representation on the delay-Doppler domain behaves slow variation since channel reflective scatters are relatively static in the surrounding environment. On the other hand, OTFS can be viewed as a spreading modulation scheme in the time-frequency domain, where the basis functions are selected from a two-dimensional Fourier series. In this perspective, OTFS can be regarded as a 2D-CDMA scheme, whereas OFDM is essentially a one-dimensional spreading scheme using a single-mode Fourier basis. OTFS can achieve enhanced channel diversity associated with the use of appropriate equalization methods [35, 89].

5.1.1 Signal Processing Challenges in OTFS Systems

In OTFS, the channel is modeled as a 2D multi-path response spread over the dimension of the delay-tap and the Doppler-shift. Hence, equalization techniques at the receiver are available to exploit the diversity over these two dimensions. However, the equalizer has to be adopted to address the following practical challenges:

- **Challenge 1: Pilot Overhead**. Conducting equalization in the Delay-Doppler domain requires pilots or reference signals for either explicit channel state information (CSI) estimation or equalizer coefficients adaptation. Due to the multi-tap nature of the delay-Doppler channel representation, guard intervals between pilot and data symbols are often inserted to avoid the underlying interference [78]. When OTFS systems are configured with multiple-input-and-multiple-output (MIMO) antennas, the overhead resulting from the guard intervals becomes a bottleneck, especially for massive MIMO systems. Therefore, it is important to resolve the pilot overhead issue by designing pilot patterns that support associated low complexity equalization methods.

- **Challenge 2: Accurate Channel Knowledge**. Acquiring accurate CSI in the delay-Doppler domain is critical for model-based equalizers in OTFS systems. Although time-varying multi-path channels can be ideally approximated as a constant impulse response in the delay-Doppler domain, high received SNRs of the corresponding pilots or reference signals are needed to acquire accurate 2D channel spreading profiles. However, obtaining high received SNRs for pilots or reference signals is challenging in OTFS systems due to power and hardware constraints. For example, when synchronization is not perfect, the transmitting and receiving pulse shaping filters in OTFS will fail to be biorthogonal. This introduces additional inter-carrier interference (ICI) [1] and inter-symbol interference (ISI) [14] leading to low SNRs at the receiver. Furthermore, the motion of channel scatters can cause misalignment to the channel-prior assuming a strict sparsity in the delay-Doppler domain. Therefore, it will result in a model mismatch to sparsity-based channel estimation algorithms.

[1] ICI happens when the OTFS system is implemented as an overlay to OFDM systems which will be discussed in Sec. 5.2.3

5.1.2 Related Work

The equalization of OFDM in doubly selective channels has been extensively studied. In [79], a low complexity minimum mean-squared error (MMSE) based equalization method for OFDM systems over a time-varying channel has been introduced. A two-stage MMSE equalization method was considered in [83] where the first step is to restrict the support of inter-carrier interference followed by eliminating the interference using successive MMSE. However, all these signal processing techniques rely on explicit accurate CSI. It is important to note that OFDM can be viewed as the projection of OTFS onto the time-frequency domain. In fact, this projection will change more abruptly making the tracking of channel more challenging. For example, the above introduced OFDM-based schemes require channel tracking operating on a sub-millisecond basis under high mobility environments for 5G NR networks. This makes it difficult to adopt neural network-based receive processing strategies.

To handle the wireless channels with high Doppler shifts, state-of-the-art equalization methods in OTFS systems often assume the explicit CSI in the delay-Doppler domain is ideally available [35, 36]. On the other hand, various channel estimation methods have been introduced to estimate explicit CSI using the pilot/reference signals. For example, the work in [27] introduced a channel estimation method with almost linear complexity. Embedded pilot-aided channel estimation schemes for OTFS are investigated in [78]. In each OTFS frame, guard intervals are placed between pilot and data symbols on the delay-Doppler domain. The underlying CSI is estimated by performing hard thresholding and the estimated CSI is utilized for symbol detection. Note that the scenario considered is relatively simple as each delay tap is associated with single Doppler shift generated using Jakes' formula. An alternative to using impulse signals as channel estimation pilots in the delay-Doppler domain is to use pseudo-noise (PN) sequences in the delay-Doppler domain [66].

Assuming ideal explicit CSI is available, a message-passing-based iterative algorithm for the equalization on the delay-Doppler domain is introduced in [77] where the extension to MIMO-OTFS systems has not yet been explored. [88] presents a rake receiver in conjunction with decision feedback to conduct the symbol detection. Ideal channel knowledge is assumed for turbo-boosting the success rate. Similar concepts of leveraging decision feedback have been widely studied in MIMO systems with multi-path channels, such as [7, 21, 50]. However, feedback errors can propagate to amplify the detection error especially in the low SNR regime. When these methods are generalized to equalization of OTFS in the 2D delay-Doppler domain, their efficiency has not been studied.

5.1.3 One-Shot Online Learning for OTFS through Reservoir Computing

To address the above mentioned challenges for OTFS systems, we consider using neural networks (NNs) to conduct equalization. In 4G and 5G networks where OFDM is used as the underlying waveform, there exists many MIMO transmission modes with link adaptation, rank adaptation, and scheduling operating on a subframe basis [55]. For high mobility scenarios that are of interest to 5G-Advanced and 6G, it is critical to design an online learning-based equalization method for OTFS systems that is adaptive and robust to the change of wireless environments, such as channel distributions, operation modes, scheduling decisions, and inter-user interference. However, wireless networks, especially cellular networks, have unique features and constraints. For example, the online over-the-air (OTA) training dataset (e.g., reference signals) of cellular networks is extremely limited since it is the system overhead instead of the data. In 5G NR, the overhead defined for the demodulation reference signal is at most 20% [3]. Therefore, NN-based online equalization for OTFS with very limited OTA training datasets is extremely challenging. To address this challenge, we consider a “one-shot learning” solution: In contrast to most learning-based algorithms, which require training on extensive datasets, one-shot learning aims to learn information from very few training samples. Intuitively, the superior generalization performance of the one-shot learning concept is achieved by incorporating NNs with inductive prior of unseen data [62].

This paper considers using reservoir computing (RC, a special recurrent neural network where the structure is embedded with a sequential inductive bias) for equalization in OTFS systems. RC has been previously utilized for spatial and temporal interference cancellation tasks in MIMO-OFDM systems [65, 96, 99, 101], and has proven to be efficient in handling high Doppler shift in an online fashion even with extremely limited OTA training datasets. In the RC framework for MIMO-OFDM, the temporal-spatial signal features are read out to conduct interference cancellation in consecutive OFDM symbols. In this paper, we generalize the concept by training the RC using a subset of received symbols (designed as specific pilot patterns) and applying the trained network on the remaining symbols (to equalize interference residing in data symbols). Our technical contributions in applying RC to equalization in OTFS are the following

- **Novel Pilot Structures**: We investigate two OTFS pilot structures the delay-Doppler domain for OTFS: interleaved pilot and superimposed pilot. Different from conventional pilot designs, the considered pilot and data symbols are allowed to interfere with each other without guard intervals between them. The training framework only utilizes pilot symbols within the same OTFS frame as the data symbols instead of those in previous OTFS frames or offline training dataset. Therefore, it offers a one-shot learning objective in terms of optimizing the NN weights learning through each individual OTFS frame. Since our design and approach is guard interval free and a sole online-learning based approach, it can effectively address *Challenge 1* listed in Sec.

5.1.1. Furthermore, the pilot overhead in our approach can be adaptively adjusted, providing a set of feasible configurations in the trade-off space of control overhead and system performance.

- **Robust Learning Algorithms**: Novel RC-based learning algorithms are developed for the two types of pilot pattern. As a result, the learning algorithms can jointly cope with the interference between the pilot and data symbols regarding the interleaved pilot pattern in the first part. By contrast, it is the signal power split between pilot and data for the superimposed pilot that enables the equalization NN to generalize. More important, the learning algorithm does not rely on explicit CSI providing robust equalization performance, even in low SNR regimes. This provides us an effective method to address *Challenge 2* raised in Sec. 5.1.1.

The remainder of this paper is organized as follows: In Section 5.2, we briefly introduce the preliminaries of OTFS systems. In Section 5.3, we develop a learning-based framework for OTFS equalization in the delay-Doppler domain. The discussion of systems considerations includes the pilot design (training dataset), the neural network structure, and the loss objective for the equalization. In Section 5.4, we extend the learning framework to MIMO-OTFS. We also introduce an approach where we apply multiple RCs in the equalization training of each OTFS frame. Section 5.5 presents the comparison between conventional equalization methods and our proposed method in both SISO and MIMO scenarios. We conclude the paper in Section 5.6.

5.2 Preliminaries

A wireless channel exhibiting both time and frequency selectivity is referred to as a doubly-selective channel. This section first provides a mathematical representation of wireless communications signals over the doubly-selective channel. Then, we discuss the OTFS modulation and its connection to the OFDM system. Finally, we present the end-to-end relation between transmitted symbols and received symbols in the OTFS system in the delay-Doppler domain.

5.2.1 Doubly-Selective Channel

We consider a wireless communication system with a transmitter and a receiver. Given the time domain transmitted signal $x(t)$, the received signal $r(t)$ is given by

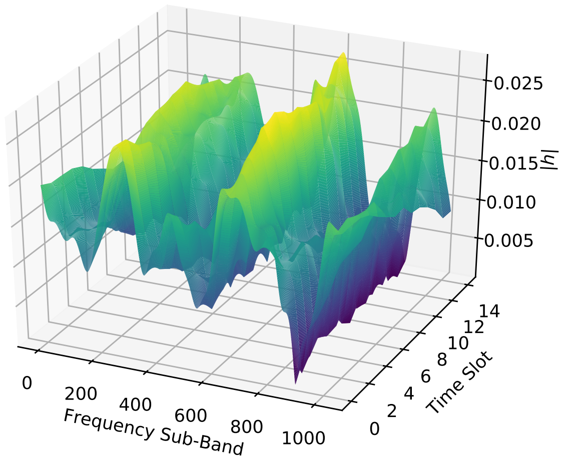
$$r(t) = \int \tilde{h}(\tau, t)x(t-\tau)d\tau, \tag{5.1}$$

where τ denotes delay and t denotes time in the doubly selective channel response $\tilde{h}(\tau, t)$. Using $\tilde{h}(\tau, t)$, the frequency selectivity of the wireless channel is characterized by having multi-path coefficients on τ. The time selectivity of the wireless channel is shown as $\tilde{h}(\tau, t_1) \neq \tilde{h}(\tau, t_2)$ when $t_1 \neq t_2$. Alternatively, the input-output relation in (5.1) can be expressed via a Doppler-variant channel impulse response as follows,

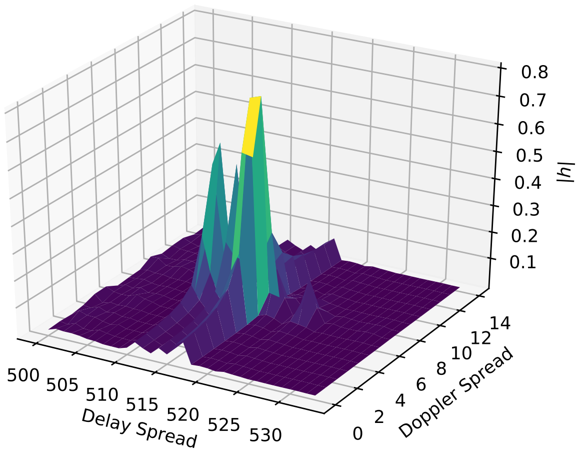
$$r(t) = \iint h(\tau, \nu)x(t-\tau)e^{j2\pi\nu t}d\tau d\nu \tag{5.2}$$

where

$$h(\tau, \nu) := \int \tilde{h}(\tau, t)e^{-j2\pi\nu t}dt,$$



(a)



(b)

Figure 5.1: Channel representations in different domain: (a) in the time-frequency domain, (b) in the delay-Doppler domain.

is the so-named Doppler-variant impulse response. It represents the Doubly-selective channel via the delay-Doppler domain. Note that the above formulation has a missing internal term

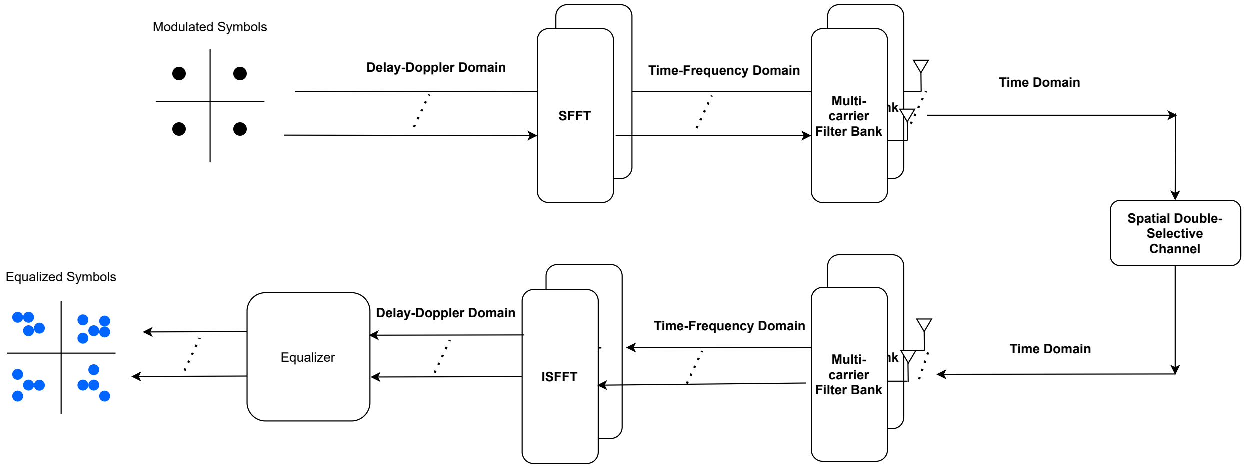


Figure 5.2: A link level schematic of a MIMO-OTFS system.

$e^{-2\pi\nu\tau}$ compared to the equation (5.1) in [36]. This is because the missing term is part of $h(\tau, \nu)$ in our notation. Representations of a doubly selective channel on these two domains are plotted in Fig. 5.1.

5.2.2 OTFS Modulation

An implementation of OTFS modulation is shown in Fig. 5.2, where OTFS is realized by adding pre-and post-processing blocks to a standard multi-carrier modulation system. The pre-processing block converts modulation symbols from the delay-Doppler domain to the time-frequency domain and vice versa. The multi-carrier filter bank (Heisenberg transform) at the Tx converts a 2D time-frequency domain signal to a 1D time-domain waveform. In our notation, we define the time domain waveform of one **OTFS frame** as $x(t)$. It is written as,

$$x(t) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} X[n,m] g_{\text{tx}}(t - n\Delta T) e^{j2\pi m \Delta f (t - n\Delta T)}, \tag{5.3}$$

where $X[n,m]$ is its time-frequency representation, $g_{tx}(t)$ is the pulse shaping filter, M stands for the number of frequency tones, ΔT is the time duration of one transmitting pulse, Δf

is the frequency spacing, and N is the number of transmitting pulses in one OTFS frame. Therefore, an OTFS frame occupies $N\Delta T$ seconds in time and $M\Delta f$ Hz bandwidth.

The pre-processing block which transforms modulation symbols from the delay-Doppler domain to the time-Frequency domain is defined as,

$$X[n,m] = \frac{1}{\sqrt{NM}} \sum_{k=0}^{N-1} \sum_{l=0}^{M-1} x[k,l] e^{j2\pi\left(\frac{nk}{N} - \frac{ml}{M}\right)}. \tag{5.4}$$

where $x[k,l]$ are the modulation symbols residing in the delay–Doppler domain. The transformation (5.4) is also referred to as the inverse symplectic finite Fourier transform (ISFFT) since it has Fourier basis functions. In addition, $x[n,m]$ can be interpreted as the superposition of QAM symbols spreading over the full time-frequency grid using $e^{j2\pi\left(\frac{nk}{N} - \frac{ml}{M}\right)}$ as the basis functions. Therefore, OTFS is able to exploit the diversity across the full-time and frequency domain. At the receiver, the received signals at the time-frequency domain, and delay-Doppler domain are respectively denoted as $Y[n,m]$ and $y[k,l]$. The relation between $Y[n,m]$ and $y[k,l]$ is characterized by the SFFT. $Y[n,m]$ is obtained by using receiving filter bank, which has a similar formulation as (5.3). For simplicity, we summarized the notations of the OTFS system in Table 5.1.

5.2.3 Integration with OFDM Systems

As shown in Fig. 5.2, an OTFS system can be implemented as an overlay of an OFDM system. This implementation expedites prototyping since OFDM systems are highly optimized in current wireless systems, such as 5G NR and WiFi. However, building an OTFS system on top of an existing transceiver system brings extra complexity. Alternatively, an OTFS system can be realized via its own "standalone". This simplification is because half of the operations on the pre-and post-processing operations (i.e., ISFFT and SFFT) can be

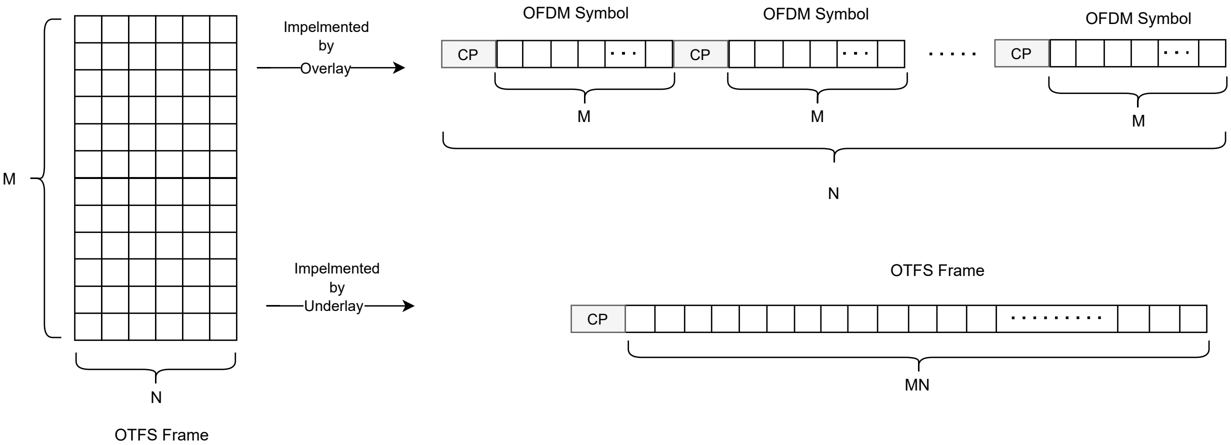


Figure 5.3: The time-domain frame structure of two different OTFS implementations: the top one overlays an OFDM system, and the bottom one is a standalone version.

canceled out by the FFT and IFFT operations from the multi-carrier filter bank. Therefore, the resulting standalone implementation is more lightweight, since it only needs one pair of FFT and IFFT in the transceiver chain. Meanwhile, the cyclic prefix (CP) of each OFDM symbol can be eliminated in OTFS, significantly improving spectral efficiency. Without adding CP inside the OTFS frame, OTFS handles the inter-symbol interference via a frame-based manner, where the interference "between OFDM symbols" is lumped to a 2D basis to equalize. However, a CP still needs to be added between two consecutive OTFS frames to avoid the "frame interference". As a summary, Fig. 5.3 depicts the time frame structures of these two different schemes for OTFS system implementation under the same sampling rate (an overlay on an OFDM system and a standalone version).

5.2.4 End-to-End Channel Model in Delay-Doppler Domain

For simplicity, we assume ideal bio-orthogonal transmit and receive pulse shaping filters[36]. After substituting (5.3) and (5.4), as well as their counterparts at the receiver, into (5.1), we arrive at an end-to-end model connecting $x[k,l]$ and $y[k,l]$ within one OTFS frame. The relation is given below [35],

$$y[k,l] = \frac{1}{NM}\sum_{k'=0}^{N-1}\sum_{l'=0}^{M-1}\bar{x}\left[k',l'\right]h\left[k-k',l-l'\right] \tag{5.5}$$
$$k=0,\cdots,N-1; l=0,\cdots,M-1$$

where $\bar{x}[k',l']$ is the periodized version of $x[k',l']$ with periods $(1/\Delta T, 1/\Delta f)$ in the delay-Doppler domain, and

$$h\left[k',l'\right] := h(\nu,\tau)|_{\nu=\frac{k'}{N\Delta T},\tau=\frac{l'}{M\Delta f}}\ . \tag{5.6}$$

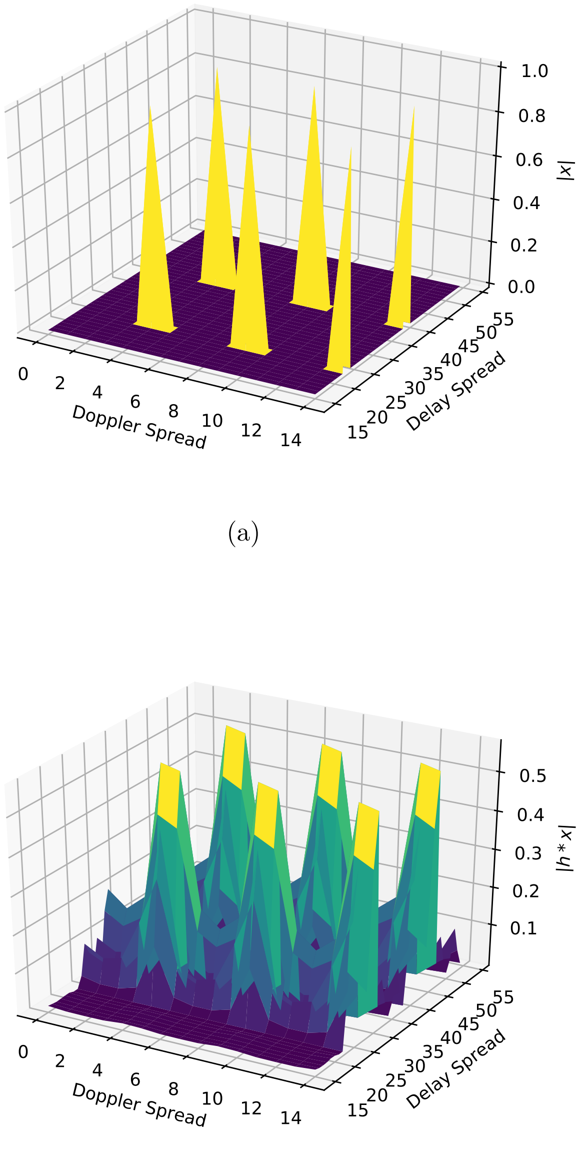
Note that CP added between two consecutive OTFS frames accomplishes this 2D circular convolution relation. In addition, any imperfectness in transmitting and receiving pulse shaping can be absorbed into the channel kernel $h[k,l]$ for a more comprehensive formulation.

Table 5.1: Notation in OTFS Systems

Notations	Definitions
$x[k,l]$	Modulation Symbols in Delay-Doppler Domain at Tx
$X[n,m]$	Time-Frequency Domain representation of the modulated symbols (via ISFFT) at Tx
$x(t)$	Time Domain Waveform of OTFS Signal at Tx
$\tilde{h}(\tau,t)$	Doubly-Selective Channel in delay-time domain
$h(\tau,\nu)$	Doubly-Selective Channel in delay-Doppler domain
$r(t)$	Time Domain Waveform of OTFS Signal at Rx
N	Number of transmitted pulses in one OTFS frame
M	Number of sub-carriers in one OTFS frame
$y[k,l]$	Modulation Symbols in Delay-Doppler Domain at Rx
$Y[n,m]$	Time-Frequency Domain representation of the modulated symbols (via ISFFT) at Rx

5.3 RC Meets OTFS in Equalization

We first introduce two types of online training dataset in an OTFS system with single antennas at the transmitter and receiver. We then present a neural network structure and



(a)

(b)

Figure 5.4: Distribution of symbols on the delay-Doppler domain at the transmitter (a) and the receiver (b).

the associated learning algorithms for equalization at the receiver.

5.3.1 Training and Testing Dataset

Since one OTFS frame can be discretized as a grid with N intervals along the Doppler domain and M intervals along the delay domain, we denote the modulated symbols $x[k, l]$ over one OTFS frame as a matrix $\boldsymbol{X} \in \mathcal{C}^{M \times N}$, where $\mathcal{C}$ is the predefined constellation for digital modulation, such as $\mathcal{C} = \{+1, -1\}$ for BPSK, and $\mathcal{C} = \{+1 + 1j, +1 - 1j, -1 + 1j, -1 - 1j\}$ for QPSK. We define the online training dataset (i.e., pilot symbols) and testing dataset allocated within the same OTFS frame. Accordingly, the equalization operation is based on each OTFS frame: The training method only uses the pilot symbols from one OTFS frame; Then, it applies the learned neural network to equalize the data symbols in the remainder of the same OTFS frame. This operation mode is robust to sudden channel changes, such as when the base station adopts a new transmission rank and schedules a new user. This paper considers two ways to construct pilots on the delay-Doppler domain: interleaved pilot and superimposed pilot patterns.

Interleaved Pilot

Here pilot symbols interleave with data symbols on the delay-Doppler domain as shown in Fig. 5.5. The training dataset is denoted as

$$\{\boldsymbol{X}_{train} := \boldsymbol{\Omega} \odot \boldsymbol{X}, \boldsymbol{Y}_{train} := \boldsymbol{Y}\}, \tag{5.7}$$

where *inside any of the curly brackets above and below, the left-hand element represents the NN desired output, while the right-hand component stands for the NN input,* and $\odot$ stands for the Hadamard product between two matrices. Accordingly, the testing dataset is given by,

$$\{\boldsymbol{X}_{test} := \bar{\boldsymbol{\Omega}} \odot \boldsymbol{X}, \boldsymbol{Y}_{test} := \boldsymbol{Y}\}, \tag{5.8}$$

where $\boldsymbol{X}$ is a symbol matrix representing one OTFS frame at the transmitter, $\boldsymbol{\Omega}$ is an indication matrix with value 1 at the location assigned as pilot symbols, and value 0 elsewhere, and $\boldsymbol{Y}$ is the corresponding received OTFS frame at the receiver. Fig. 5.5 only shows one possible realization of the interleaved pilot pattern, where the stairwise green spots in the first sub-figure stand for pilots (training dataset), and the remaining blank blocks represent data symbols for transmission (testing dataset). Alternative groupings of pilot symbols are shown in Fig. 5.6. The pilot overhead η for the interleaved pilot is given by,

$$\eta = \frac{|\boldsymbol{\Omega}|}{M \cdot N}. \tag{5.9}$$

Note that both the training and testing dataset use the observation $\boldsymbol{Y}$, rather than $\boldsymbol{Y} \odot \boldsymbol{\Omega}$ and $\boldsymbol{Y} \odot \bar{\boldsymbol{\Omega}}$ respectively. This design is motivated by the objective of learning the mutual interference between pilot and symbols to further reduce training overhead. It is possible to add guard symbols between pilot and data symbols, and this approach has been adopted in conventional channel estimation frameworks for OTFS, such as [78][2].

Superimposed Pilot

Here pilot symbols are spread over the 2D delay-Doppler domain, and are superimposed on data symbols as illustrated in Fig. 5.7. Therefore, symbols in each transmitted OTFS frame can be written as $\boldsymbol{X} = \boldsymbol{X}_{train} + \boldsymbol{X}_{test}$, where $\boldsymbol{X}_{test} \in \mathcal{C}^{M \times N}$, and $\boldsymbol{X}_{train} \in \mathbb{C}^{M \times N}$. Since the

[2] The guard symbols defined in [78] can be absorbed into the support of Ω when calculating the pilot overhead.

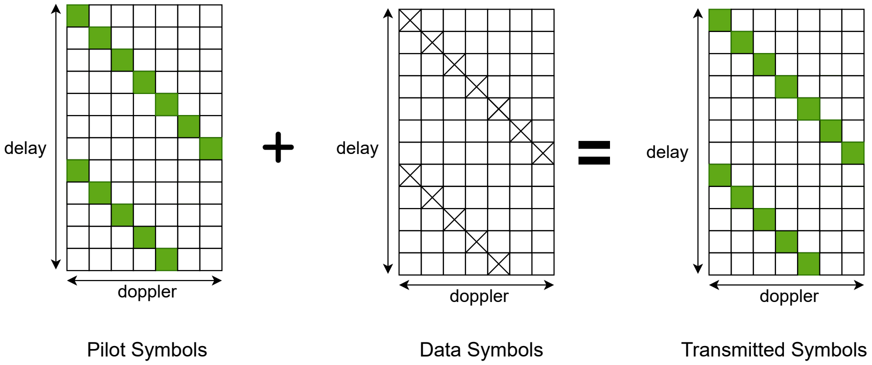


Figure 5.5: Delay-Doppler representation of interleaved pilots associated with the data symbols in one OTFS frame.

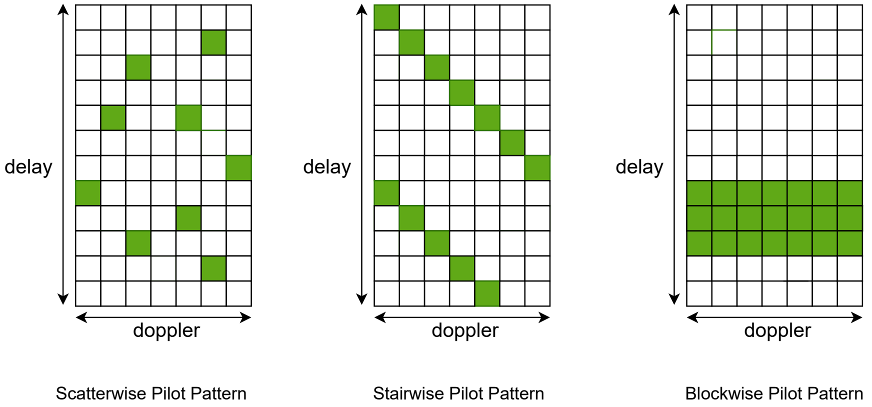


Figure 5.6: Different interleaved pilot patterns over one OTFS frame represented in the delay-Doppler Domain.

ideal end-to-end relation in OTFS system can be formulated as a twisted convolution (a linear mapping) on the 2D plane, the received OTFS symbols are given by $\boldsymbol{Y} = \boldsymbol{Y}_{train} + \boldsymbol{Y}_{test}$, where $\boldsymbol{Y}_{train}$ corresponds to the receiving components contributed only from $\boldsymbol{X}_{train}$, and $\boldsymbol{Y}_{test}$ is from $\boldsymbol{X}_{test}$.

However, simply adding superimposed pilots over data symbols may not easy for the receiving process. Regarding a supervised learning-based equalization framework, the receiver has to first identify the pilot components from received symbols $\boldsymbol{Y}$. Rather than using the support of interleaved pilots as discussed in the previous subsection, we consider resorting to the time-frequency domain to distinguish pilot symbols and data symbols. To this end, we design our superimposed pilot symbols as follows,

$$\boldsymbol{X}_{train} = \mathrm{SFFT}(c \cdot \boldsymbol{\Omega}),$$

SFFT represents the inverse mapping of (5.4) operating on matrices, and c is a constant which is expected to suppress the value of any entries of $\mathrm{ISFFT}(\boldsymbol{X}_{test})$ to make pilots distinguishable from data symbols.

However, a large constant c leads to a high peak to average ratio (PAPR) of the transmitted signal. We can add an interference term to diminish the energy from data symbols at the support of $\boldsymbol{\Omega}$ instead. To this end, we incorporate an aided interference term defined as follows,

$$\boldsymbol{X}_{aid-int} = -\mathrm{SFFT}(\mathrm{ISFFT}(\boldsymbol{X}_{test}) \odot \boldsymbol{\Omega}).$$

The newly formed transmitted signals on the delay-Doppler domain are,

$$\boldsymbol{X} = \boldsymbol{X}_{train} + \boldsymbol{X}_{test} + \boldsymbol{X}_{aid-int}, \tag{5.10}$$

and the corresponding received signals are $\boldsymbol{Y}$. Therefore, the training dataset has the following compact expression,

$$\{\boldsymbol{X}_{train} := \mathrm{SFFT}(\boldsymbol{\Omega}), \boldsymbol{Y}_{train} := \mathrm{SFFT}(\mathrm{ISFFT}(\boldsymbol{Y}) \odot \boldsymbol{\Omega})\}, \tag{5.11}$$

The testing dataset is,

$$\{\boldsymbol{X}_{test}, \boldsymbol{Y}_{test} := \boldsymbol{Y}\}. \tag{5.12}$$

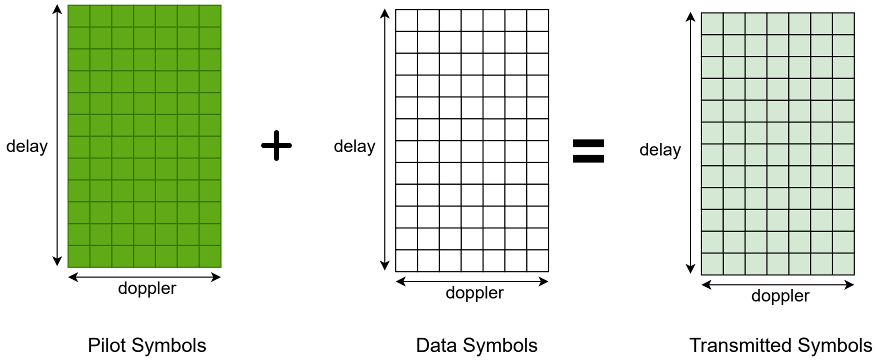


Figure 5.7: Superimposed Pilot in the delay-Doppler Domain

Note that the overhead calculation of the superimposed pilot is not straightforward. If we were to directly calculate pilot overhead based on the delay-Doppler domain, the pilot overhead would likely be 0% since data symbols spread over the entire grid in the delay-Doppler domain. However, we may calculate overhead in terms of power required by the superimposed pilots. Given the duality between time-frequency and delay-Doppler, we may still calculate the overhead by using (5.9) on the time-frequency domain.

5.3.2 Neural Network Structure for Equalization – Reservoir Computing

Reservoir computing (RC) is a memory-enabled neural network framework. A vanilla version of RC comprises a nonlinear activation function, an input layer, a recurrent layer, and an output layer. The computational power of RC is realized by mapping the recurrent states to desired signals through learning the output layers. Furthermore, due to the fixed reservoir dynamics, RC training is only conducted on the readout layers, which effectively avoids gradient vanishing/explosion issues by avoiding back-propagation through time.

A discrete-time realization of RC can be characterized by its internal state transition and output mapping. The state transition is formulated as follows,

$$\boldsymbol{s}(t+1) = \sigma\left(\boldsymbol{W}_{tran}\begin{bmatrix}\boldsymbol{s}(t)\\ \boldsymbol{y}(t)\end{bmatrix}\right) \tag{5.13}$$

where t stands for the time index, σ is a nonlinear function, $\boldsymbol{s}(t)$ is a vector representing the internal reservoir state, $\boldsymbol{y}(t)$ is the input vector, and $\boldsymbol{W}_{tran}$ stands for the reservoir weight matrix which is often chosen with a spectral radius smaller than 1 in order to asymptotically achieve a “similarity” to the desired dynamic model[41]. The output equation is simply treated as

$$\boldsymbol{z}(t) = \boldsymbol{W}_{out}\begin{bmatrix}\boldsymbol{s}(t)\\ \boldsymbol{y}(t)\end{bmatrix}, \tag{5.14}$$

where $\boldsymbol{W}_{out}$ is the output weight matrix and $\boldsymbol{z}(t)$ stands for the output. We can stack the features in an extended state matrix

$$\bar{\boldsymbol{S}} = \left[\begin{bmatrix}\boldsymbol{s}(0)\\ \boldsymbol{y}(0)\end{bmatrix}, \begin{bmatrix}\boldsymbol{s}(1)\\ \boldsymbol{y}(1)\end{bmatrix}, \cdots, \begin{bmatrix}\boldsymbol{s}(t)\\ \boldsymbol{y}(t)\end{bmatrix},\right]^T. \tag{5.15}$$

The loss function for the output layer learning is then given by,

$$\min_{\boldsymbol{W}} \|\mathcal{R}C_{\boldsymbol{W}}(\boldsymbol{Y}_{train}) - \boldsymbol{X}_{train}\|_F \tag{5.16}$$

where the input and output dimensions of $\mathcal{R}C_{\boldsymbol{W}}(\cdot)$ are configured as N, and the sequence length is set as M. Once $\boldsymbol{Y}_{train}$ is fed into RC, an internal extended state matrix $\bar{\boldsymbol{S}}$ can be obtained ready for output layer learning as suggested by (5.14).

5.3.3 Learning Algorithms

We introduce two different ways to learn the output weights of RC in the following two subsections corresponding to the two different pilot patterns.

Interleaved Pilot

Here we define the loss function as

$$\min_{\boldsymbol{W}} \|\boldsymbol{X}_{train} - \boldsymbol{\Omega} \odot (\bar{\boldsymbol{S}}_{train}\boldsymbol{W}_{out})\|_F^2.$$

A closed-form solution of the above problem can be obtained via rephrasing the objective function as,

$$\min_{\boldsymbol{W}} \sum_{n=1}^{N} \|((\boldsymbol{X}_{train})_{:,n} - \text{diag}(\boldsymbol{\Omega}_{:,n})\bar{\boldsymbol{S}}_{train}(\boldsymbol{W}_{out})_{:,n}])\|_2^2$$

By solving the above separable least squares, the closed-form solution of $\boldsymbol{W}_{out}$ is given by each of its column vectors as follows,

$$(\boldsymbol{W}_{out})_{:,n} = \left[\text{diag}\left(\boldsymbol{\Omega}_{:,n}\right)\bar{\boldsymbol{S}}_{train}\right]^{+}(\boldsymbol{X}_{train})_{:,n}\,. \tag{5.17}$$

The transmitted data symbols are estimated by applying the output weights through,

$$\hat{\boldsymbol{X}}_{test} = \mathcal{Q}\left(\bar{\boldsymbol{S}}_{test}\boldsymbol{W}_{out}\right) \odot \boldsymbol{\Omega}. \tag{5.18}$$

where $\mathcal{Q}$ is the optimal quantizer for the constellation of $\mathcal{C}$. Note that $\bar{\boldsymbol{S}}_{train} = \bar{\boldsymbol{S}}_{test}$ since $\boldsymbol{Y}_{train} = \boldsymbol{Y}_{test}$.

Superimposed Pilot

The objective of superimposed pilot based training is given as follows,

$$\min_{\boldsymbol{W}} \|\boldsymbol{X}_{train} - \bar{\boldsymbol{S}}_{train}\boldsymbol{W}_{out}\|_F^2,$$

We can directly obtain the coefficients of the readout weight as follows,

$$\boldsymbol{W}_{out} = \bar{\boldsymbol{S}}_{train}^{+}\boldsymbol{X}_{train}. \tag{5.19}$$

Then the transmitted data symbols are obtained by

$$\hat{\boldsymbol{X}}_{test} = \mathcal{Q}(\bar{\boldsymbol{S}}_{test}\boldsymbol{W}_{out}). \tag{5.20}$$

Note that quantization from RC output to the estimated symbols essentially eliminates the contribution from the aided interference term in (5.10). The learning and testing stages for the above two different pilot patterns are summarized jointly in Algorithm 5.

Algorithm 5 RC for OTFS Equalization

```
Input: $\{\boldsymbol{Y}_{train}, \boldsymbol{X}_{train}\}$, $\boldsymbol{Y}_{test}$
Output: $\boldsymbol{W}_{out}$, $\hat{\boldsymbol{X}}_{test}$
  for $l \in [0, L_{forget}]$ do
    Input sequence $[\boldsymbol{Y}, \boldsymbol{O}_l]$ into RC and obtain a delayed extended state matrix $\bar{\boldsymbol{S}}_l := \bar{\boldsymbol{S}}(:, l:)$, where $\bar{\boldsymbol{S}}$ is the extended state matrix as defined in (5.15), and $\boldsymbol{O}_l$ is a zero matrix with $l$ columns.
    Calculate a corresponding output layer $\boldsymbol{W}_{out}^{l}$ by using (5.17) for interleaved pilots and (5.19) for superimposed pilots.
    Cache the LOSS$_l$ value of the learned output weight.
  end for
  Sort the optimal $l^\star = \arg\min_p \text{LOSS}_p$, and select $\boldsymbol{W}_{out} = \boldsymbol{W}_{tout}^{l^\star}$
  Obtain $\hat{\boldsymbol{X}}_{test}$ by using (5.18) for interleaved pilot and (5.20) for superimposed pilot.
```

5.4 Extensions and Enhancements

5.4.1 Learning in MIMO-OTFS

We assume there are N_t and N_r antennas at transmitter and receiver, respectively. The end-to-end relation between transmitted and received symbols is given by,

$$\begin{aligned} y_{n_r}[k,l] &= \frac{1}{NM}\sum_{n_t=0}^{N_t-1}\sum_{k'=0}^{N-1}\sum_{l'=0}^{M-1}\bar{x}_{n_t}\left[k',l'\right]h_{n_r,n_t}\left[k-k',l-l'\right] \\ k &= 0,\cdots,N-1; l=0,\cdots,M-1 \end{aligned}, \tag{5.21}$$

where $h_{n_r,n_t}[k,l]$ denotes the channel between the n_r-th received antenna and the n_t-th transmitted antenna represented in the delay-Doppler domain, and $\bar{x}_{n_t}$ is the periodized representation of the transmitted symbols at the n_tth antenna in the delay-Doppler domain.

In the MIMO scenario, we consider the pilot pattern at any single Tx antennas of the MIMO system follows the same design as the SISO scenario; at the same time, it is reused by different Tx antennas. This pilot structure allows the neural network to perceive the interference pattern of the spatial channel at the receiver side. By contrast, conventional channel estimation methods often employ non-overlapping pilots to avoid antenna interference. To process the RC based learning, we stack the transmitted symbols as a three-mode matrix $\boldsymbol{\mathcal{X}} \in \mathcal{C}^{M\times N\times N_t}$. Accordingly, the training dataset is generalized as follows,

$$\left\{\boldsymbol{X}_{train} := \begin{bmatrix}\boldsymbol{\Omega}\\ \boldsymbol{\Omega}\\ \vdots\\ \boldsymbol{\Omega}\end{bmatrix} \odot \begin{bmatrix}\boldsymbol{\mathcal{X}}_{:,:,1}\\ \boldsymbol{\mathcal{X}}_{:,:,2}\\ \vdots\\ \boldsymbol{\mathcal{X}}_{:,:,N_t}\end{bmatrix}, \boldsymbol{Y}_{train} := \begin{bmatrix}\boldsymbol{\mathcal{Y}}_{:,:,1}\\ \boldsymbol{\mathcal{Y}}_{:,:,2}\\ \vdots\\ \boldsymbol{\mathcal{Y}}_{:,:,N_t}\end{bmatrix}\right\}.$$

Table 5.2: Notation Appearing in the learning system

Notation	Definitions
$\boldsymbol{X}_{train} \in \mathbb{C}^{MN_t \times N}$	Transmitted pilot symbols
$\boldsymbol{X}_{test} \in \mathbb{C}^{MN_t \times N}$	Transmitted data symbols
$\boldsymbol{X} \in \mathbb{C}^{MN_t \times N}$	Transmitted symbols which are assembled in $\boldsymbol{X}_{train}$ and $\boldsymbol{X}_{test}$
$\boldsymbol{\mathcal{X}} \in \mathbb{C}^{M \times N \times N_t}$	Transmitted symbols in a 3-mode form
$\boldsymbol{Y}_{train} \in \mathbb{C}^{MN_r \times N}$	Received pilot symbols
$\boldsymbol{Y}_{test} \in \mathbb{C}^{MN_r \times N}$	Received data symbols
$\boldsymbol{Y} \in \mathbb{C}^{MN_r \times N}$	Received symbols which contain $\boldsymbol{Y}_{train}$ and $\boldsymbol{Y}_{test}$
$\boldsymbol{\mathcal{Y}} \in \mathbb{C}^{M \times N \times N_r}$	Received symbols in a 3-mode form

The testing dataset is given by,

$$\left\{ \boldsymbol{X}_{test} := \begin{bmatrix} \bar{\boldsymbol{\Omega}} \\ \bar{\boldsymbol{\Omega}} \\ \vdots \\ \bar{\boldsymbol{\Omega}} \end{bmatrix} \odot \begin{bmatrix} \boldsymbol{\mathcal{X}}_{:,:,1} \\ \boldsymbol{\mathcal{X}}_{:,:,2} \\ \vdots \\ \boldsymbol{\mathcal{X}}_{:,:,N_t} \end{bmatrix}, \boldsymbol{Y}_{test} := \begin{bmatrix} \boldsymbol{\mathcal{Y}}_{:,:,1} \\ \boldsymbol{\mathcal{Y}}_{:,:,2} \\ \vdots \\ \boldsymbol{\mathcal{Y}}_{:,:,N_t} \end{bmatrix} \right\},$$

Similarly, the training and testing dataset for the superimposed pilot in the MIMO system are,

$$\left\{ \boldsymbol{X}_{train} := \begin{bmatrix} \mathrm{SFFT}(\boldsymbol{\Omega}) \\ \mathrm{SFFT}(\boldsymbol{\Omega}) \\ \vdots \\ \mathrm{SFFT}(\boldsymbol{\Omega}) \end{bmatrix}, \boldsymbol{Y}_{train} := \begin{bmatrix} \mathrm{SFFT}(\mathrm{ISFFT}(\boldsymbol{\mathcal{Y}}_{:,:,1}) \odot \boldsymbol{\Omega}) \\ \mathrm{SFFT}(\mathrm{ISFFT}(\boldsymbol{\mathcal{Y}}_{:,:,2}) \odot \boldsymbol{\Omega}) \\ \vdots \\ \mathrm{SFFT}(\mathrm{ISFFT}(\boldsymbol{\mathcal{Y}}_{:,:,N_t}) \odot \boldsymbol{\Omega}) \end{bmatrix} \right\}. \tag{5.22}$$

The training and testing process follows the same algorithm flow defined in Algorithm 5. The RC input dimension is thereby set as $N_r \times N$ (The number of antennas times the axis-length along the Doppler domain).

5.4.2 Learning using Multiple RCs

The aforementioned training framework utilizes one RC in the entire delay-Doppler domain. We can also consider applying multiple RCs in one OTFS frame, where an individual RC is assigned to learn a local channel feature. For instance, when a blockwise pilot pattern is employed for SISO-OTFS, we can split the training dataset (5.7) into N subsets, where the nth subset is defined as

$$\{\boldsymbol{x}_{train}^{(n)} := \boldsymbol{\Omega}_{:,n} \odot \mathrm{FFT}(\boldsymbol{X})_{:,n}, \boldsymbol{y}_{train}^{(n)} := \mathrm{FFT}(\boldsymbol{Y})_{:,n}\}, \tag{5.23}$$

where FFT operates along the row of the matrix $\boldsymbol{X}$. Accordingly, the kth testing dataset is,

$$\{\boldsymbol{x}_{test}^{(n)} := \bar{\boldsymbol{\Omega}}_{:,n} \odot \mathrm{FFT}(\boldsymbol{X})_{:,n}, \boldsymbol{y}_{test}^{(n)} := \mathrm{FFT}(\boldsymbol{Y})_{:,n}\}. \tag{5.24}$$

Therefore, the input dimension of each RC is reduced to 1. We then have N RCs to be individually learned over one OTFS frame. In general, we can group any number of consecutive columns of $\boldsymbol{X}$ as a subset. Accordingly, the number of RCs is reduced to K, where K stands for the number of subsets. Similar to the analysis in [96], we can easily show that the complexity of using K RCs is proportional to $\mathcal{O}(N_n N_t NM + min\{(\eta MN/K)^2, N_n^2\}KN_t)$ FLOPS, where η is the percentage of pilot overhead and N_n is the size of the extended state. When $K = N$, the second term in the complexity formula becomes $min\{(\eta M)^2, N_n^2\}NN_t$. When $K = 1$, it is $min\{(\eta M)^2, N_n^2\}N_t$. Therefore, the resulting complexity in these two extreme cases indicates that using multiple RCs tends to be more costly than using one RC in the computation, though the training set size is reduced in the former case. In the next section, we observe that changing the number of RCs can impact the bit error rate performance, offering a tradeoff between the equalization complexity and performance. Meanwhile, the complexity of message passing (MP) algorithm [77] is $\mathcal{O}(MNN_t(M\eta NN_t))$,

Table 5.3: Simulation Parameters

Parameters	**Values**
Channel Model	3GPP 5G NR CDL
Carrier frequency	4 GHz
Delay Profile in Channel Model	CDL-C
Delay Spread	10 ns
Maximum Doppler Shift	555 Hz
Carrier Spacing	15 KHz
Sampling Rate	15.36 MHz
Pilot Overhead for SISO	4.69%
M	1024
N	14

where $M\eta N$ represents the number of non-zero taps of the channel impulse kernel in the delay-Doppler domain. This analysis reveals that MP spends more computational resources than our introduced approach.

5.5 Numerical Experiments

In our simulation, we implement OTFS using the standalone approach, as shown in the bottom part of Fig. 5.3. Regarding OTFS parameters, we choose M as 14 to be compatible with the sub-frame length of OFDM as defined in 4G and 5G systems. The number of sub-carriers is set at 1024. The channel model is selected as 3GPP 5G NR clustered delay line (CDL) channel, where the parameter settings are listed in Table 5.3. The CP length for OTFS is set at the maximum channel length generated by the channel model. The channel Doppler shift is 555Hz which represents mobility at highway speeds as high as 90 mph. All curves shown below are obtained by running the algorithms over a transmission link having 30 different time-and-frequency dispersive channel realizations. The training overhead for SISO is set at 4.7% by default, and for MIMO at 18.7%. The x-axis is the signal to noise ratio in dB defined as the received signal power normalized by the background noise. The

RC configuration is based on the paper [96], where a windowed buffer is added in the input layer with size 20. The size of the recurrent internal state in RC for the superimposed pilot is set as 8. For the interleaved pilot learned by multiple RCs, the internal state for each RC is set as 8 for SISO and 12 for MIMO. Finally, the link reliability metric utilized below is uncoded bit error rate.

The comparison between OFDM and OTFS systems is shown in Fig. 5.8. The evaluation methods by order of the legend are respectively:

1. RC using Interleaved pilot in OTFS (denoted as Interleaved-RC-OTFS), where the number of RCs in one OTFS frame is configured as 7, and it is selected as the blockwise pattern in Fig. 5.6.

2. Superimposed pilot enabled RC using OTFS (denoted as Superimposed-RC-OTFS);

3. Decision feedback equalization in OTFS using the same amount of training overhead as RC-OTFS (denoted as DFE-OTFS);

4. LMMSE channel estimation with linear interpolation and LMMSE equalization in time-frequency domain (denoted as LMMSE-OTFS);

5. Message-passing based equalization in OTFS using estimated CSI which is acquired using the same pilot overhead as other methods. However, the obtained CSI is imperfect in the low SNR regime, which degrades the BER. We show the performance of this approach by increasing the received SNR for pilot symbols to 17 dB;

6. Message-passing based equalization in OTFS using estimated CSI, where the CSI is obtained by using pilot symbols with SNR 30 dB;

7. Multiple-RC in OFDM, where 7 RCs are applied in one OFDM subframe. The pilot overhead is set the same as RC-OTFS (denoted as RC-OFDM);

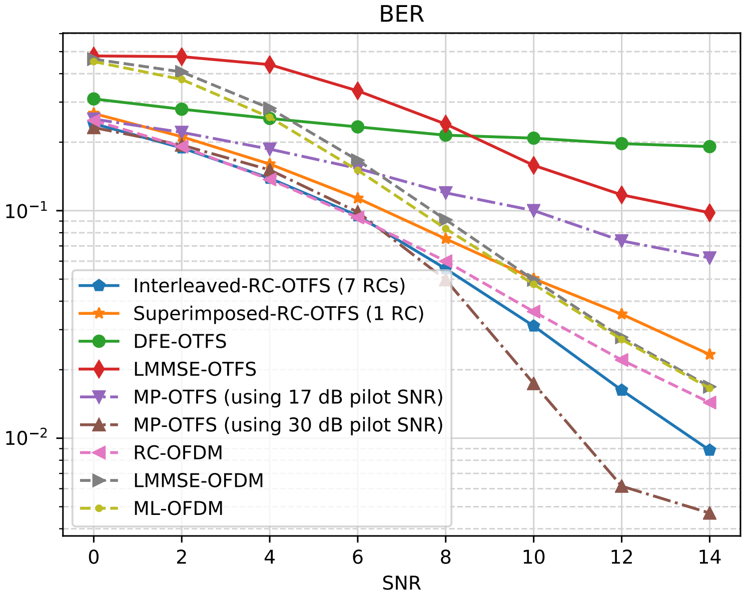


Figure 5.8: Comparison of BER performance in SISO-OTFS and SISO-OFDM systems with different equalization algorithms. Note that the pilot SNR in MP-OTFS is increased to 17 dB and 30 dB to show its BER drop.

8. LMMSE channel estimation with linear interpolation and LMMSE equalization in the time-frequency domain using the same pilot overhead as RC-OTFS (denoted as LMMSE-OFDM);

9. Maximum likelihood estimation in OFDM with LMMSE channel estimation and interpolation (denoted as ML-OFDM).

We observe that RC with OTFS outperforms other approaches in the low SNR regime except for RC-OFDM. The advantage enjoyed by RC-OFDM is because each RC handles the local channel change within each two OFDM symbols. To obtain more insight on how the number of RCs utilized in the system can impact performance, we plot the result of using different numbers of RCs in Fig. 5.9. The figure reveals that there is a tradeoff between the number of RCs and the BER performance. In addition, the configuration of each RC in OFDM

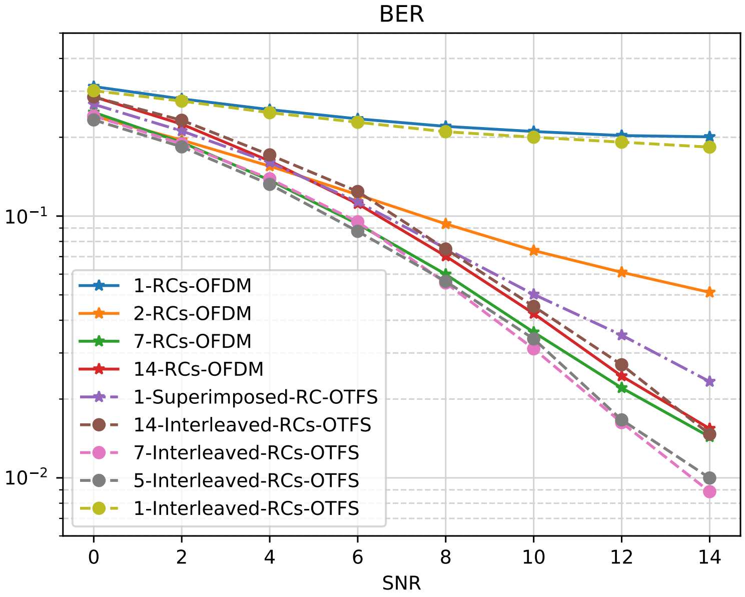


Figure 5.9: Comparison of BER performance in SISO-OTFS and SISO-OFDM systems with different numbers of RC.

is the same as the one in OTFS. Moreover, using one RC for the superimposed pilot can significantly improve upon using one for the interleaved pilot. Applying more RCs improves the performance at the cost of increased complexity. As a result, using the OTFS waveform shows incremental gain over OFDM in the evaluation. More significantly, the OFDM system cannot accommodate a superimposed pilot similar to OTFS and achieve a comparable BER performance to OTFS using a single RC. These observations demonstrate the benefits of modulating information symbols on the delay-Doppler domain.

Fig. 5.10 shows a BER comparison between superimposed pilots and interleaved pilots by changing the overhead. Note that the pilot overhead for the superimposed one is calculated in the time-frequency domain. However, the superimposed pilot-based approach also can be treated as zero overhead in the delay-Doppler domain given a BER threshold. The value of the pilot overhead ranges from 4.7% to 39.0%. The scenario under interleaved pilot utilizes 2 RCs in the learning. We can observe that the superimposed pilot becomes inefficient as

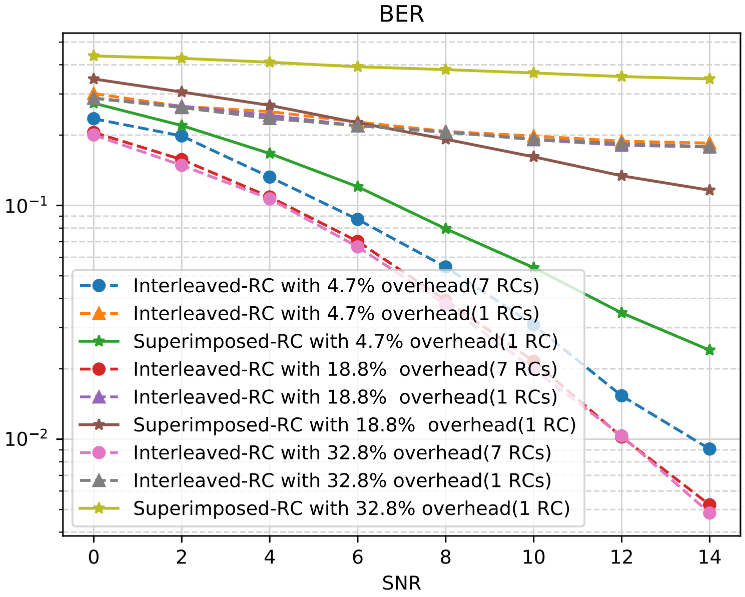


Figure 5.10: Comparison of pilot overhead for the two different pilot patterns, where the overhead of interleaved RC is calculated in the delay-Doppler domain, while the overhead in Superimposed-RC is calculated in the time-frequency domain.

“overhead” increases due to more interference in data symbols added. It is also important to note that the learning approach for interleaved pilots can eliminate the interference between pilot and data symbols even though no guard interval is added.

Fig. 5.11 depicts the BER performance of RC and DFE as we change the value of velocity (Doppler shift). We can observe that the processing of RC is more robust than DFE in the low SNR regime. The poor performance of DFE as SNR increases is because of the imposed limited training constraints (only 4% pilot symbols are utilized via a block pilot pattern on the delay-Doppler domain). To the best of our knowledge, it is still an open problem to efficiently implement DFE in the 2D delay-Doppler domain with a limited length of training symbols. Fig.5.12 shows the BER performance using RC for MIMO-OTFS through blockwise interleaved pilot patterns. We observe that utilizing more RCs need not lead to a monotonic performance improvement. Fig. 5.13 shows the BER performance under

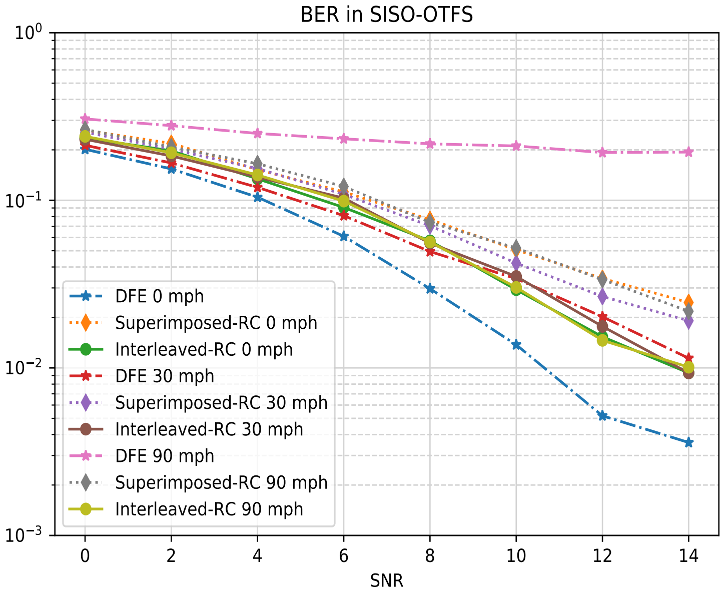


Figure 5.11: BER in SISO-OTFS under different mobile station (MS) velocities using DFE and RC.

different MS velocities in OTFS and OFDM systems using the same method. The MIMO case also demonstrates the advantage of our approach in the OTFS system. Note that the comparison between the RC-based approach and the conventional OFDM system is studied in [96]. Meanwhile, to the best of our knowledge, model-based approaches in MIMO-OTFS systems have to tackle more interference than the SISO case in the low SRN regime, hence perform far behind the RC learning-based approaches.

5.6 Conclusion

In this paper, we introduced a neural network-based framework for OTFS equalization. Our results demonstrate that using a reservoir computing-based neural network structure coupled with specifically designed pilot patterns (training dataset) can significantly reduce inter-symbol interference. More significantly, the training of the equalization neural network

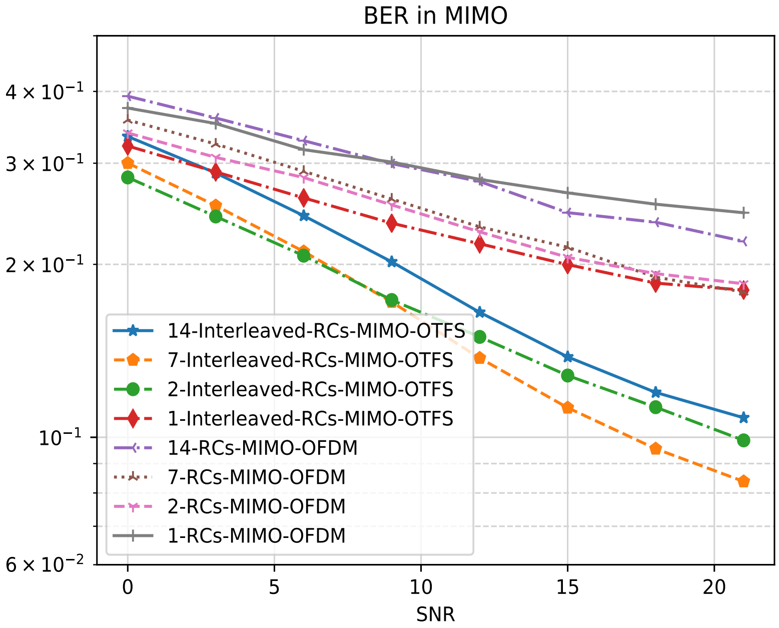


Figure 5.12: BER of RC in MIMO-OTFS and MIMO-OFDM with different numbers of RC.

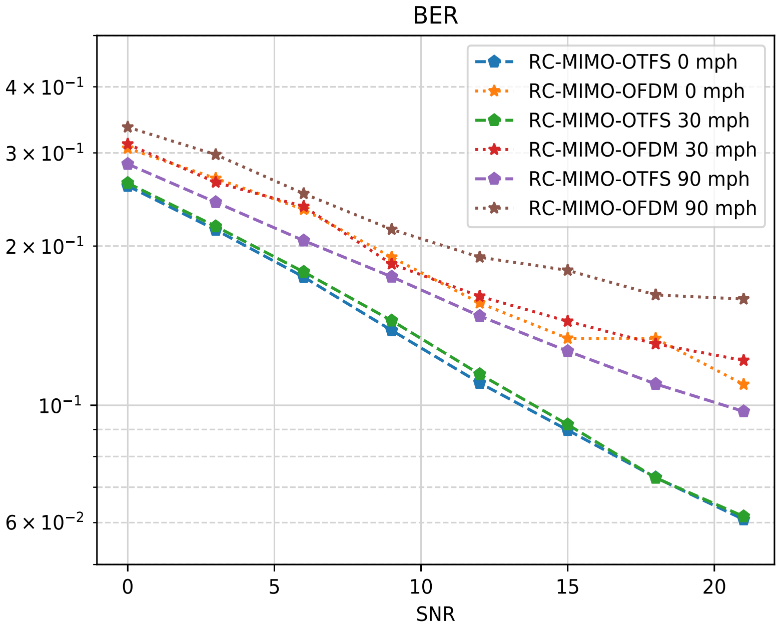


Figure 5.13: BER of RC in MIMO-OTFS and MIMO-OFDM with different MS velocities.

utilizes observations from a single OTFS frame which is competitive with the control plane employed in current cellular systems. The success of the resulting “learning through one OTFS frame” is also because the offered channel is an invariant feature of OTFS modulation, which transforms doubly selective wireless channels to nearly constant convolution kernels in the delay-Doppler domain. Our numerical results show that the neural network-based equalization method outperforms traditional methods in various Doppler spread settings. In addition, we can observe the performance gain of OTFS by applying the same neural network structure to an OFDM system with the same pilot control overhead. In conclusion, we believe an OTFS system employed with our implemented equalizer can provide either better transmission reliability or lower processing complexity to state-of-the-art equalization approaches for OFDM systems in high Doppler environments.

Chapter 6

Summary

This dissertation introduces various neural network structures to handle the symbol detection/equalization task in wireless communications systems. The major challenge is to learn the neural network coefficients with the same training overhead as defined in current physical layer standards. The limited training constraint makes the symbol detection task unique compared with conventional NN applications, such as image and speech recognition. To this end, this dissertation considers using reservoir computing as the basic neural network building block due to its appealing generalization performance. In addition, different inductive biases are incorporated into the neural network structures to handle the air interference under different MIMO configurations. According to the dissertation's presentation order, the introduced inductive biases include shifting-invariant of the digital constellation, time-frequency relation of OFDM modulation, multi-mode features of full-dimensional antennas, and delay-Doppler representation from OTFS modulation. The numerical experiments show that using these inductive priors improves the symbol detection/equalization performance in the low SNR regime under a variety of MIMO settings, including point-to-point and multi-user MIMO. Meanwhile, the evaluations contain two air-interface technologies, OFDM and OTFS, where OTFM is the dominant technology for modern cellular systems, while OTFS is developed for high mobility environments. In conclusion, the introduced NN-based approaches can solve model mismatch issues, thus surpass conventional model-based methods.

Appendices

Appendix A

Formulas for Chapter 2

A.1 Equivalent Real-value Signal Model

Let $g(\cdot)$ represent the entire linear mapping in (2.1),

$$\tilde{\boldsymbol{Y}} = g(\tilde{\boldsymbol{X}}) + \tilde{\boldsymbol{N}}. \tag{A.1}$$

The real-valued form of the above observation equation can be written as the following composite function

$$\begin{bmatrix} \mathrm{Re}(\boldsymbol{Y}) \\ \mathrm{Im}(\boldsymbol{Y}) \end{bmatrix} = \sum_{\ell=0}^{L-1} g^{(\ell)} \left(f^{(\ell)} \left(\begin{bmatrix} \mathrm{Re}(\boldsymbol{X}) \\ \mathrm{Im}(\boldsymbol{X}) \end{bmatrix} \right) \right) + \begin{bmatrix} \mathrm{Re}(\boldsymbol{N}) \\ \mathrm{Im}(\boldsymbol{N}) \end{bmatrix}$$

where $f^{(\ell)}(\cdot)$ is defined as

$$\begin{bmatrix} \mathrm{Re}(\boldsymbol{X}\boldsymbol{F}\boldsymbol{J}_\ell)^T \\ \mathrm{Im}(\boldsymbol{X}\boldsymbol{F}\boldsymbol{J}_\ell)^T \end{bmatrix} = \begin{bmatrix} \mathrm{Re}(\boldsymbol{F}\boldsymbol{J}_\ell)^T, -\mathrm{Im}(\boldsymbol{F}\boldsymbol{J}_\ell)^T \\ \mathrm{Im}(\boldsymbol{F}\boldsymbol{J}_\ell)^T, \mathrm{Re}(\boldsymbol{F}\boldsymbol{J}_\ell)^T \end{bmatrix} \begin{bmatrix} \mathrm{Re}(\boldsymbol{X})^T \\ \mathrm{Im}(\boldsymbol{X})^T \end{bmatrix}$$

and $g^{(\ell)}(\cdot)$ is defined as

$$\begin{bmatrix} \mathrm{Re}(\boldsymbol{Y}) \\ \mathrm{Im}(\boldsymbol{Y}) \end{bmatrix} = \begin{bmatrix} \mathrm{Re}(\boldsymbol{H}[\ell]), -\mathrm{Im}(\boldsymbol{H}[\ell]) \\ \mathrm{Im}(\boldsymbol{H}[\ell]), \mathrm{Re}(\boldsymbol{H}[\ell]) \end{bmatrix} \begin{bmatrix} \mathrm{Re}(\boldsymbol{X}\boldsymbol{F}\boldsymbol{J}_\ell) \\ \mathrm{Im}(\boldsymbol{X}\boldsymbol{F}\boldsymbol{J}_\ell) \end{bmatrix}.$$

A.2 Channel Estimation

Channel estimation is conducted in the frequency domain. Multiplying the discrete Fourier transform (DFT) matrix $\boldsymbol{F}^H$ on both sides of (2.1), we have

$$\boldsymbol{Y}\boldsymbol{F}^H = \sum_{\ell=0}^{L-1} \boldsymbol{H}[\ell]\boldsymbol{X}\,\mathrm{diag}(\boldsymbol{l}_\ell) + \boldsymbol{N}\boldsymbol{F}^H \tag{A.2}$$

where $\boldsymbol{l}_\ell$ is the vector of eigenvalues of $\boldsymbol{J}_\ell$, since the eigenvectors of a circulant matrix is the Fourier matrix, and the diag() operator converts its vector argument to a diagonal matrix. Accordingly, the j^{th} column of $\boldsymbol{X}$ can be written as

$$\left(\boldsymbol{Y}\boldsymbol{F}^H\right)_{:,j} = \sum_{\ell=0}^{L-1} l_\ell(j)\boldsymbol{H}[\ell]\boldsymbol{x}_j + (\boldsymbol{N}\boldsymbol{F}^H)_{:,j} \tag{A.3}$$

where $\sum_{\ell=0}^{L-1} l_\ell(j)\boldsymbol{H}[\ell]$ stands for the equivalent channel on the j^{th} sub-carrier which can be estimated via the LMMSE method using the Q symbols from the training set in (2.8).

Appendix B

Proofs for Chapter 3

We now consider using alternative least squares to solve the problem (3.10). When $\boldsymbol{W}_{tout}$ is given, (3.10) can be rewritten as

$$\begin{aligned} \min_{\boldsymbol{w}_{fout}(n)} \quad & \|\tilde{\boldsymbol{F}}(n)\boldsymbol{S}\boldsymbol{W}_{tout}\text{diag}(\boldsymbol{w}_{fout}(n)) - \boldsymbol{Z}(n)\|_F^2 \\ s.t. \quad & \text{diag}(|\boldsymbol{w}_{fout}(n)|) = \boldsymbol{I}, \quad \forall n = 0, \cdots, (N_{sc} - 1), \end{aligned} .$$

where the definition of $\tilde{\boldsymbol{F}}(n)$, $\boldsymbol{Z}(n)$ and $\boldsymbol{S}$ are given in Sec. 3.3.2. By spelling out the objective, the optimization problem is equivalent to

$$\begin{aligned} \min_{\boldsymbol{w}_{fout}(n)} \; & - \text{Real}(\text{Tr}(\bar{\boldsymbol{Z}}(n)\text{diag}(\boldsymbol{w}_{fout}(n))\boldsymbol{Z}(n)^H)) \\ s.t. \quad & \text{diag}(|\boldsymbol{w}_{fout}(n)|) = \boldsymbol{I}, \quad \forall n = 0, \cdots, (N_{sc} - 1), \end{aligned}$$

where $\bar{\boldsymbol{Z}}(n) \triangleq \tilde{\boldsymbol{F}}(n)\boldsymbol{S}\boldsymbol{W}_{tout}$. In addition,

$$\text{Tr}(\bar{\boldsymbol{Z}}(n)\text{diag}(\boldsymbol{w}_{fout}(n))\boldsymbol{Z}(n)^H) = \sum_j w_{fout,j}(n)\text{Tr}(\bar{\boldsymbol{z}}^j(n)\boldsymbol{z}^j(n)^H)$$

where $\bar{\boldsymbol{z}}^j(n)$ is the jth column of $\bar{\boldsymbol{Z}}(n)$, $\boldsymbol{z}^j(n)$ is the jth column of $\boldsymbol{Z}(n)$ and $w_{fout,j}(n)$ is the j the entry of $\boldsymbol{w}_{fout}(n)$. In order to minimize the objective function value, we can select

$$\angle w_{fout,j}(n) = -\angle(\boldsymbol{z}^j(n)^H\bar{\boldsymbol{z}}^j(n)) \tag{B.1}$$

where $\angle$ represents the angle of a complex number. When $\{\boldsymbol{w}_{fout}(n)\}_{n=0}^{N_{sc}-1}$ is fixed in (3.10), $\boldsymbol{W}_{tout}$ is learned by

$$\min_{\boldsymbol{W}_{tout}} \|\tilde{\boldsymbol{F}}\boldsymbol{S}\boldsymbol{W}_{tout} - \hat{\boldsymbol{Z}}\|_F^2 \tag{B.2}$$

where $\tilde{\boldsymbol{F}} \triangleq \operatorname{diag}(\boldsymbol{F}, \cdots, \boldsymbol{F}) \in \mathbb{C}^{QN_{sc} \times QN_{sc}}$ in which $\boldsymbol{F}$ is the Fourier matrix.

$$\hat{\boldsymbol{Z}} \triangleq [\hat{\boldsymbol{Z}}_0^T, \cdots, \hat{\boldsymbol{Z}}_{Q-1}^T]^T,$$

and

$$\hat{\boldsymbol{Z}}_q \triangleq [\hat{\boldsymbol{z}}_q(0), \hat{\boldsymbol{z}}_q(1), \cdots, \boldsymbol{z}_q(N_{sc}-1)]^T, \tag{B.3}$$

in which $\hat{\boldsymbol{z}}_q(n) \triangleq \boldsymbol{z}_q(n)\operatorname{diag}(\boldsymbol{w}_{fout}^*(n))$. Therefore, the learning rule of $\boldsymbol{W}_{tout}$ is

$$\boldsymbol{W}_{tout} = \boldsymbol{S}^+ \tilde{\boldsymbol{F}}^H \hat{\boldsymbol{Z}}. \tag{B.4}$$

Appendix C

Derivations for Chapter 4

C.1 Proof of Theorem 1

Lemma C.1. *Given two full-rank matrices* $\boldsymbol{X} \in \mathbb{C}^{M_1 \times M_2}$ *and* $\boldsymbol{G} \times \mathbb{C}^{H \times M_2}$*, where* $H \geq M_1$*, the following least squares problem has a unique solution if and only if* $rank(\boldsymbol{X}) = rank(\boldsymbol{W}^\star)$*, where* $\boldsymbol{W}^\star$ *is the optimum.*

$$\min_{\boldsymbol{W}} \|\boldsymbol{X} - \boldsymbol{W}\boldsymbol{G}\|_F \tag{C.1}$$

Proof. Based on the assumptions, we have $rank(\boldsymbol{X}) = \min\{M_1, M_2\}$ and $rank(\boldsymbol{G}) = \min\{H, M_2\}$. In general, (C.1) has a unique solution if and only if $rank(\boldsymbol{G}) = H$ [61].

$rank(\boldsymbol{G}) = H$ implies $M_2 \geq H$. Accordingly, the solution is given by $\boldsymbol{X}\boldsymbol{G}^+$, where $\boldsymbol{G}^+$ represents the Moor-Penrose inverse of matrix $\boldsymbol{G}$. Since $\boldsymbol{X}$ is with full row rank, $\boldsymbol{G}^+$ is with full column rank and $H \geq M_1$, we have $rank(\boldsymbol{X}\boldsymbol{G}^+) = rank(\boldsymbol{X})$. Thus, we have $rank(\boldsymbol{W}^\star) = rank(\boldsymbol{X}\boldsymbol{G}^+) = rank(\boldsymbol{X})$.

On the contrary, suppose $rank(\boldsymbol{W}^\star) = rank(\boldsymbol{X})$ holds. We have $H \leq M_2$, otherwise there exists $\boldsymbol{W}^\star + \boldsymbol{H}$ which is also a minimum of (C.1), where $\boldsymbol{H}$ is a non-zero solution of $\boldsymbol{H}\boldsymbol{G} = \boldsymbol{0}$. Since $\boldsymbol{G}$ is assumed with full rank, we have $rank(\boldsymbol{G}) = H$. □

Lemma C.2. *Given the same assumption as Lemma C.1, the sufficient and necessary* *condition for the uniqueness of (C.1) can be expressed as*

$$M_2 \geq H \tag{C.2}$$

Proof. Since the necessary and sufficient condition for the uniqueness is $rank(\boldsymbol{G}) = H$. Therefore, the uniqueness condition can be alternatively written as (C.1) is $M_2 \geq H$.

□

Before proceeding on the proof of Theorem 1, we introduce the following concepts to characterize rank properties of tensor [19].

Definition C.3. The mode-n rank of a tensor $\boldsymbol{\mathcal{G}}$ is the mode-n unfolding of $\boldsymbol{G}$, i.e., $\boldsymbol{G}_{(n)}$.

Definition C.4. A N-order tensor $\boldsymbol{\mathcal{G}}$ is with rank-$(M_1, M_2, \cdots, M_N)$ when its mode-1 rank, mode-2 rank to mode-N rank are equal to M_1, M_2, and M_N, respectively.

Lemma C.5. *Given $\boldsymbol{\mathcal{X}} \in \mathbb{C}^{M_1 \times M_2 \times \cdots \times M_N}$ and $\boldsymbol{\mathcal{G}} \in \mathbb{C}^{H_1 \times H_2 \times \cdots \times H_N}$, which are with rank-$(M_1, M_2, \cdots, M_N)$ and rank-$(H_1, H_2, \cdots, H_N)$ respective, and $H_n \geq M_n$, $N > 2$, the minimization of*

$$\min_{\boldsymbol{W}_1, \cdots, \boldsymbol{W}_N} \|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{G}} \times_1 \mathbf{W}_1 \times_2 \mathbf{W}_2 \cdots \times_N \mathbf{W}_N\|_F \tag{C.3}$$

is unique if

$$rank(\boldsymbol{X}_{(n)}) = rank(\boldsymbol{W}_n^\star) \tag{C.4}$$

where $\boldsymbol{W}_1^\star$, $\cdots$, $\boldsymbol{W}_N^\star$ is the optimum.

Proof. We prove this theorem by mathematical induction. According to Lemma 1, the uniqueness holds for $\min_{\boldsymbol{W}_1} \|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{G}} \times_1 \boldsymbol{W}_1\|_F$. Then, we assume it holds for N order tensor $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{G}}$ with $N-1$ factor matrices.

Now, we consider the case with N factor matrices,

$$
\begin{aligned}
&\min_{\boldsymbol{W}_1,\cdots,\boldsymbol{W}_N} \|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{G}} \times_1 \mathbf{W}_1 \times_2 \mathbf{W}_2 \cdots \times_N \mathbf{W}_N\|_F \\
&= \min_{\boldsymbol{W}_N}\{\min_{\boldsymbol{W}_1,\cdots,\boldsymbol{W}_{N-1}} \|\boldsymbol{X}_{(N)} - \mathbf{W}_N \boldsymbol{G}_{(N)} (\mathbf{W}_1 \otimes \cdots \otimes \mathbf{W}_{N-1})^T\|_F\}
\end{aligned}
$$

We denote $\boldsymbol{W}_N^\star$ as one of the optima of the above problem. Since $\boldsymbol{W}_N^\star$ is with full row rank based on the assumption, we have tensor $\boldsymbol{\mathcal{G}} \times_N \boldsymbol{W}_N^\star$ with rank-$(H_1, H_2, \cdots, M_N)$. This is because $\boldsymbol{G}_{(n)}$ is with full row rank. Therefore, $(\boldsymbol{W}_1^\star, \cdots, \boldsymbol{W}_{N-1}^\star)$ as an optimum of $\min_{\boldsymbol{W}_1,\cdots,\boldsymbol{W}_{N-1}} \|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{G}} \times_1 \mathbf{W}_1 \times_2 \mathbf{W}_2 \cdots \times_N \mathbf{W}_N^\star\|_F$ is unique based on the inductive assumption. We then assert $\boldsymbol{W}_N^\star$ is unique. Otherwise it contracts to Lemma 1.

□

Remarks:

- The proof does not conduct the mathematical induction on the tensor mode. This is because the uniqueness does not hold for matrix decomposition, i.e., $\min_{\boldsymbol{W}_1,\boldsymbol{W}_2} \|\boldsymbol{X} - \boldsymbol{W}_1 \boldsymbol{G} \boldsymbol{W}_2^T\|_F$.

- This theorem is only a sufficient condition for the uniqueness of (C.3).

Theorem C.6. *Under the same condition as Lemma 3, using ALS to solve (C.3) by initializing the factor matrices with full rank, the achieved solution is unique when*

$$\sum_{i \neq n} M_i \geq H_n.$$

Proof. If we solve the optimization problem (C.3) through ALS as well as initializing factor matrices as full rank, updated factor matrices are with full rank at each iteration if we assume

$$\sum_{i \neq n} M_i \geq H_n$$

due to Lemma 2. Here, the prerequisite of Lemma 2 is met because in the updating rule for $\boldsymbol{W}_n$, $rank(\boldsymbol{G}_{(n)}(\mathbf{W}_1 \otimes \cdots \mathbf{W}_{n-1} \otimes \mathbf{W}_{n+1} \otimes \cdots \otimes \mathbf{W}_N)^T) = H_n$ is guaranteed by the full rank initialization assumption on factor matrices. Therefore, when ALS terminated at an optimum of (C.3), we can assert this optimum is the unique by using Lemma 3. □

With minor revisions on the statement of the above Theorem, we can arrive at Theorem 1.

C.2 Low Complexity Factor Matrix Calculation in ALS

At each step of using alternating least squares to solve the factor matrices $\boldsymbol{W}_{out-1}$ and $\boldsymbol{W}_{out-1}$ in (4.11), suppose we directly solve the following sub-problem to obtain $\boldsymbol{W}_{out-1}$,

$$\boldsymbol{W}_{out-1} = \arg\min_{\boldsymbol{W}_{out-1}} \|\boldsymbol{Z}_{(1)} - \boldsymbol{W}_{out-1}\boldsymbol{G}_{(1)}(\boldsymbol{W}_{out-2} \otimes \boldsymbol{I}_{N_T} \otimes \boldsymbol{I}_{N_K})^T\|_F.$$

The resulting memory costs on $\boldsymbol{W}_{out-2} \otimes \boldsymbol{I}_{N_T} \otimes \boldsymbol{I}_{N_K}$ are very large. To cope with the bottleneck from memory units, we alternatively solve $\boldsymbol{W}_{out-1}$ via the following approach,

$$\begin{aligned}
\boldsymbol{W}_{out-1} &= \arg\min_{\boldsymbol{W}_{out-1}} \|\boldsymbol{\mathcal{Z}} - \boldsymbol{\mathcal{G}} \times_1 \boldsymbol{W}_{out-1} \times_2 \boldsymbol{W}_{out-2}\| \\
&\overset{(a)}{=} \arg\min_{\boldsymbol{W}_{out-1}} \|\boldsymbol{\mathcal{Z}} - \sum_k \boldsymbol{\mathcal{G}}^{(k)} \times_1 \boldsymbol{W}_{out-1}^{(k)} \times_2 \boldsymbol{W}_{out-2}^{(k)}\| \\
&= \arg\min_{\boldsymbol{W}_{out-1}} \|\boldsymbol{Z}_{(1)} - \sum_k \boldsymbol{W}_{out-1}^{(k)} (\boldsymbol{\mathcal{G}}^{(k)} \times_2 \boldsymbol{W}_{out-2}^{(k)})_{(1)}\| \\
&= \arg\min_{\boldsymbol{W}_{out-1}} \|\boldsymbol{Z}_{(1)} - \boldsymbol{W}_{out-1}[(\boldsymbol{\mathcal{G}}^{(1)} \times_2 \boldsymbol{W}_{out-2}^{(1)})_{(1)}, \cdots, (\boldsymbol{\mathcal{G}}^{(K)} \times_2 \boldsymbol{W}_{out-2}^{(K)})_{(1)}]^T\|
\end{aligned}$$

where (a) comes from the partition definition of Tucker decomposition in (4.4) and $K := N! + 1$. As the above calculation suggests, we can first calculate mode-2 product between each partitioned core tensor and factor matrix, then concatenate them as a big matrix to calculate a pseudoinverse to reach a least squares solution of $\boldsymbol{W}_{out-1}$. Similar tricks can be apply to calculate $\boldsymbol{W}_{out-2}, \cdots, \boldsymbol{W}_{out-N}$ in general multi-mode RC.

Appendix D

Summary of Complexity Comparison

Here, the appendix summarizes the complexity of the RC-based neural networks. The calculation is based on the number of FLOPS utilized in the training and testing of sequences with MIMO multi-carrier modulation. The resulting FLOPS shown in Table D.1 are obtained by combining the forward path and output weight learning. In Table D.1, the notation definitions are: N_{in} represents the input dimension of RC-based NNs; N_{output} is the output dimension; N_K is the length of training sequence; K is the number of RCs in Deep-RC or the number of RCs utilized for the training in one sub-frame; N_c is the number of sub-carriers in multi-carrier modulation; N_{out-n} stands for the dimension of output at the nth mode of multi-mode RC; N_{in-n} represents the dimension of input at the nth mode of multi-mode RC; and L is the length of training and testing sequence together.

Table D.1: Complexity Comparison of Introduced Neural Networks

Approaches	Complexity in FLOPS
Shifting RC	$\mathcal{O}(N_{out}(N_{in}+N_s)L+(N_{in}+N_s)N_K^2+N_cN_{in})$
Shallow Time RC	$\mathcal{O}(N_{out}(N_{in}+N_s)L+(N_{in}+N_s)N_K^2)$
Deep Time RC	$\mathcal{O}(K(N_{out}(N_{in}+N_s)L+(N_{in}+N_s)N_K^2))$
Shallow Time-Frequency RC	$\mathcal{O}(N_{out}(N_{in}+N_s)L+(N_{in}+N_s)N_K^2+N_cN_{in})$
Deep Time-Frequency RC	$\mathcal{O}(K(N_{out}(N_{in}+N_s)L+(N_{in}+N_s)N_K^2+N_cN_{in}))$
Multi-Mode RC	$\mathcal{O}(\sum_n N_{out-n}(\prod_n N_{in-n}N!+N_s^N)L+\sum_n N_{f-n}^2\prod_n N_{out-n}N_K)$
Multiple RCs	$\mathcal{O}(N_{out}(N_{in}+N_s)L+(N_{in}+N_s)N_K^2K)$

Bibliography

[1] Study on 3D channel model for LTE. (TR 36.873), 2018.

[2] Multiplexing and channel coding. (TS 38.212), 2020.

[3] Physical channels and modulation in nr, 2020.

[4] Physical channels and modulation in nr. (TS 38.211), 2020.

[5] Multiplexing and channel coding. (TS 36.212), 2020.

[6] Physical channels and modulation. (TS 36.211), 2020.

[7] Naofal Al-Dhahir and Ali H Sayed. The finite-length multi-input multi-output mmse-dfe. *IEEE Transactions on Signal processing*, 48(10):2921–2936, 2000.

[8] A. Alkhateeb, S. Alex, P. Varkey, Y. Li, Q. Qu, and D. Tujkovic. Deep learning coordinated beamforming for highly-mobile millimeter wave systems. *IEEE Access*, 6: 37328–37348, 2018. ISSN 2169-3536.

[9] Ahmed Alkhateeb, Iz Beltagy, and Sam Alex. Machine learning for reliable mmwave systems: Blockage prediction and proactive handoff. In *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 1055–1059. IEEE, 2018.

[10] Zeyuan Allen-Zhu and Yuanzhi Li. What can ResNet learn efficiently, going beyond kernels? In *NeuIPS*, volume 32, pages 9017–9028, 2019.

[11] Jeffrey G Andrews and Teresa H Meng. Optimum power control for successive interference cancellation with imperfect channel estimation. *IEEE Trans. Wireless Commun.*, 2(2):375–383, 2003.

[12] Fayçal Ait Aoudia and Jakob Hoydis. End-to-end learning of communications systems without a channel model. In *2018 52nd Asilomar Conference on Signals, Systems, and Computers*, pages 298–303. IEEE, 2018.

[13] Fayçal Ait Aoudia and Jakob Hoydis. Model-free training of end-to-end communication systems. *arXiv preprint arXiv:1812.05929*, 2018.

[14] Ezio Biglieri, Patchava Raviteja, and Yi Hong. Error performance of orthogonal time frequency space (otfs) modulation. In *2019 IEEE Intl Conf. Commun. Wks (ICC Wks)*, pages 1–6, 2019.

[15] Emil Björnson, Erik G Larsson, and Thomas L Marzetta. Massive MIMO: Ten myths and one critical question. *IEEE Commun. Mag.*, 54(2):114–123, 2016.

[16] Mingzhe Chen, Ursula Challita, Walid Saad, Changchuan Yin, and Mérouane Debbah. Machine learning for wireless networks with artificial intelligence: A tutorial on neural networks. *arXiv preprint arXiv:1710.02913*, 2017.

[17] Qian Chen, Shunqing Zhang, Shugong Xu, and Shan Cao. Efficient mimo detection with imperfect channel knowledge-a deep learning approach. *arXiv preprint arXiv:1903.07831*, 2019.

[18] L. Cheng, Y. Wu, J. Zhang, and L. Liu. Subspace identification for DOA estimation in massive/full-dimension MIMO systems: Bad data mitigation and automatic source enumeration. *IEEE Trans. Signal Process.*, 63(22):5897–5909, 2015. doi: 10.1109/TSP.2015.2458788.

[19] Lieven De Lathauwer. Decompositions of a higher-order tensor in block terms—Part II: Definitions and uniqueness. *SIAM J. on Matrix Anal. and Appl.*, 30(3):1033–1066, 2008.

[20] Lieven De Lathauwer and Dimitri Nion. Decompositions of a higher-order tensor in block terms—Part III: Alternating least squares algorithms. *SIAM J. on Matrix Anal. and Appl.*, 30(3):1067–1083, 2008.

[21] Alexandra Duel-Hallen. Equalizers for multiple input/multiple output channels and pam systems with cyclostationary input sequences. *IEEE Journal on Selected areas in Communications*, 10(3):630–639, 1992.

[22] S. Dörner, S. Cammerer, J. Hoydis, and S. t. Brink. Deep learning based communication over the air. *IEEE Journal of Selected Topics in Signal Processing*, 12(1):132–143, Feb 2018.

[23] Salah El Hihi and Yoshua Bengio. Hierarchical recurrent neural networks for long-term dependencies. In *Advances in neural information processing systems*, pages 493–499, 1996.

[24] N. Farsad and A. Goldsmith. Neural network detection of data sequences in communication systems. *IEEE Transactions on Signal Processing*, 66(21):5663–5678, Nov 2018.

[25] Nariman Farsad, Milind Rao, and Andrea Goldsmith. Deep learning for joint source-channel coding of text. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2326–2330. IEEE, 2018.

[26] FG-NET-2030. Network 2030: A blueprint of technology, applications and market drivers towards the year 2030 and beyond.

[27] A. Fish, S. Gurevich, R. Hadani, A. M. Sayeed, and O. Schwartz. Delay-doppler channel estimation in almost linear complexity. *IEEE Transactions on Information Theory*, 59(11):7632–7644, 2013. doi: 10.1109/TIT.2013.2273931.

[28] Yoav Freund. Boosting a weak learning algorithm by majority. *Information and computation*, 121(2):256–285, 1995.

[29] Claudio Gallicchio and Alessio Micheli. Echo state property of deep reservoir computing networks. *Cognitive Computation*, 9(3):337–350, 2017.

[30] Claudio Gallicchio, Alessio Micheli, and Luca Pedrelli. Deep reservoir computing: a critical experimental analysis. *Neurocomputing*, 268:87–99, 2017.

[31] Xiang Gao, Ove Edfors, Fredrik Rusek, and Fredrik Tufvesson. Massive MIMO performance evaluation based on measured propagation data. *IEEE Trans. Wireless Commun.*, 14(7):3899–3911, 2015.

[32] Arash Ghasemmehdi and Erik Agrell. Faster recursions in sphere decoding. *IEEE Trans. Inf. Theory*, 57(6):3530–3536, 2011.

[33] Alexander N Gorban and Ivan Yu Tyukin. Blessing of dimensionality: mathematical foundations of the statistical physics of data. *Philosophical Trans. of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 376(2118):20170237, 2018.

[34] A. Graves, A. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6645–6649, May 2013.

[35] Ronny Hadani and Anton Monk. OTFS: A new generation of modulation addressing the challenges of 5G. *arXiv preprint arXiv:1802.02623*, 2018.

[36] Ronny Hadani, Shlomo Rakib, Michail Tsatsanis, Anton Monk, Andrea J Goldsmith, Andreas F Molisch, and R Calderbank. Orthogonal time frequency space modulation. In *2017 IEEE Wireless Commun. Netw. Conf. (WCNC)*, pages 1–6, 2017.

[37] Moritz Hardt, Ben Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International Conference on Machine Learning*, pages 1225–1234. PMLR, 2016.

[38] Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Brian Kingsbury, and Tara Sainath. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal Processing Magazine*, 29:82–97, November 2012.

[39] Georg Holzmann and Helmut Hauser. Echo state networks with filter neurons and a delay&sum readout. *Neural Networks*, 23(2):244–256, 2010.

[40] Ken ichi Funahashi and Yuichi Nakamura. Approximation of dynamical systems by continuous time recurrent neural networks. *Neural Networks*, 6(6):801 – 806, 1993. ISSN 0893-6080. doi: https://doi.org/10.1016/S0893-6080(05)80125-X. URL http://www.sciencedirect.com/science/article/pii/S089360800580125X.

[41] Herbert Jaeger. The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. *Bonn, Germany: German National Research Center for Inf. Technol. GMD Technical Report*, 148(34):13, 2001.

[42] Herbert Jaeger. Discovering multiscale dynamical features with hierarchical echo state networks. Technical report, Jacobs University Bremen, 2007.

[43] C. Jeon, R. Ghods, A. Maleki, and C. Studer. Optimality of large mimo detection via approximate message passing. In *2015 IEEE International Symposium on Information Theory (ISIT)*, pages 1227–1231, 2015.

[44] C. Jiang, H. Zhang, Y. Ren, Z. Han, K. Chen, and L. Hanzo. Machine learning

paradigms for next-generation wireless networks. *IEEE Wireless Communications*, 24 (2):98–105, April 2017.

[45] Peiwen Jiang, Tianqi Wang, Bin Han, Xuanxuan Gao, Jing Zhang, Chao-Kai Wen, Shi Jin, and Geoffrey Ye Li. Artificial intelligence-aided OFDM receiver: Design and experimental results. *arXiv preprint arXiv:1812.06638*, 2018.

[46] Jingon Joung, Chin Keong Ho, Koichi Adachi, and Sumei Sun. A survey on power-amplifier-centric techniques for spectrum-and energy-efficient wireless communications. *IEEE Communications Surveys & Tutorials*, 17(1):315–333, 2014.

[47] M. Khani, M. Alizadeh, J. Hoydis, and P. Fleming. Adaptive neural signal detection for massive MIMO. *IEEE Trans. Wireless Commun.*, 19(8):5635–5648, 2020.

[48] Hyeji Kim, Yihan Jiang, Sreeram Kannan, Sewoong Oh, and Pramod Viswanath. Deepcode: Feedback codes via deep learning. In *Advances in Neural Information Processing Systems*, pages 9436–9446, 2018.

[49] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.

[50] Christos Komninakis, Christina Fragouli, Ali H Sayed, and Richard D Wesel. Multi-input multi-output fading channel tracking and equalization using kalman estimation. *IEEE Transactions on Signal Processing*, 50(5):1065–1076, 2002.

[51] Jean Kossaifi, Yannis Panagakis, Anima Anandkumar, and Maja Pantic. Tensorly: Tensor learning in python. *J. of Machine Learning Research*, 20(26), 2019.

[52] V Lebedev, Y Ganin, M Rakhuba, I Oseledets, and V Lempitsky. Speeding-up convolutional neural networks using fine-tuned cp-decomposition. In *ICLR*, 2015.

[53] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553): 436–444, 2015.

[54] L. Liu, R. Chen, S. Geirhofer, K. Sayana, Z. Shi, and Y. Zhou. Downlink MIMO in LTE-advanced: SU-MIMO vs. MU-MIMO. *IEEE Commun. Mag.*, 50(2):140–147, 2012.

[55] Lingjia Liu, Runhua Chen, Stefan Geirhofer, Krishna Sayana, Zhihua Shi, and Yongxing Zhou. Downlink mimo in lte-advanced: Su-mimo vs. mu-mimo. *IEEE Communications Magazine*, 50(2):140–147, 2012. doi: 10.1109/MCOM.2012.6146493.

[56] Mantas Lukoševičius. A practical guide to applying echo state networks. In *Neural networks: Tricks of the trade*, pages 659–686. Springer, 2012.

[57] Mantas Lukoševičius and Herbert Jaeger. Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3):127–149, 2009.

[58] Thomas L Marzetta. Massive MIMO: an introduction. *Bell Labs Tech. J.*, 20:11–22, 2015.

[59] Thomas L Marzetta et al. Noncooperative cellular wireless with unlimited numbers of base station antennas. *IEEE Trans. Wireless Commun.*, 9(11):3590, 2010.

[60] Juha Meinilä, Pekka Kyösti, Tommi Jämsä, and Lassi Hentilä. Winner II channel models. *Radio Technologies and Concepts for IMT-Advanced*, pages 39–92, 2009.

[61] Carl D Meyer. *Matrix analysis and applied linear algebra*, volume 71. SIAM, 2000.

[62] Erik G Miller, Nicholas E Matsakis, and Paul A Viola. Learning from one example through shared densities on transforms. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, volume 1, pages 464–471. IEEE, 2000.

[63] S. S. Mosleh, L. Liu, C. Sahin, Y. R. Zheng, and Y. Yi. Brain-inspired wireless communications: Where reservoir computing meets MIMO-OFDM. *IEEE Trans. Neural Netw. Learn. Syst.*, 29(10):4694–4708, 2018. doi: 10.1109/TNNLS.2017.2766162.

[64] Somayeh Mosleh, Lingjia Liu, Cenk Sahin, Yahong Rosa Zheng, and Yang Yi. Brain-inspired wireless communications: Where reservoir computing meets MIMO-OFDM. *IEEE Trans. Neural Netw. Learn. Syst.*, 29(10):4694–4708, Oct 2018.

[65] Somayeh Susanna Mosleh, Lingjia Liu, Cenk Sahin, Yahong Rosa Zheng, and Yang Yi. Brain-inspired wireless communications: Where reservoir computing meets mimo-ofdm. *IEEE Transactions on Neural Networks and Learning Systems*, 29(10):4694–4708, 2018. doi: 10.1109/TNNLS.2017.2766162.

[66] KR Murali and A Chockalingam. On otfs modulation for high-doppler fading channels. In *2018 Information Theory and Applications Workshop (ITA)*, pages 1–10. IEEE, 2018.

[67] Eliya Nachmani, Elad Marciano, Loren Lugosch, Warren J Gross, David Burshtein, and Yair Be'ery. Deep learning methods for improved decoding of linear codes. *IEEE Journal of Selected Topics in Signal Processing*, 12(1):119–131, 2018.

[68] Alexander Novikov, Dmitrii Podoprikhin, Anton Osokin, and Dmitry P Vetrov. Tensorizing neural networks. In *NeurIPS*, pages 442–450, 2015.

[69] Ivan V Oseledets. Tensor-train decomposition. *SIAM J. on Scientific Comput.*, 33(5): 2295–2317, 2011.

[70] Timothy J O'Shea, Tugba Erpek, and T Charles Clancy. Deep learning based MIMO communications. *arXiv preprint arXiv:1707.07980*, 2017.

[71] T. O'Shea and J. Hoydis. An introduction to deep learning for the physical layer. *IEEE Transactions on Cognitive Communications and Networking*, 3(4):563–575, Dec 2017.

[72] Christos H Papadimitriou. On the complexity of integer programming. *Journal of the ACM (JACM)*, 28(4):765–768, 1981.

[73] Stefan Parkvall, Erik Dahlman, Anders Furuskar, and Mattias Frenne. NR: The new 5G radio access technology. *IEEE Communications Standards Magazine*, 1(4):24–30, 2017.

[74] Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. How to construct deep recurrent neural networks. *arXiv preprint arXiv:1312.6026*, 2013.

[75] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *Int. Conf. on Machine Learning (ICML)*, pages 1310–1318, 2013.

[76] Arogyaswami J Paulraj, Dhananjay A Gore, Rohit U Nabar, and Helmut Bolcskei. An overview of mimo communications-a key to gigabit wireless. *Proceedings of the IEEE*, 92(2):198–218, 2004.

[77] Patchava Raviteja, Khoa T Phan, Yi Hong, and Emanuele Viterbo. Interference cancellation and iterative detection for orthogonal time frequency space modulation. *IEEE Trans. Wireless Commun.*, 17(10):6501–6515, 2018.

[78] Patchava Raviteja, Khoa T Phan, and Yi Hong. Embedded pilot-aided channel estimation for otfs in delay–doppler channels. *IEEE Transactions on Vehicular Technology*, 68(5):4906–4917, 2019.

[79] Luca Rugini, Paolo Banelli, and Geert Leus. Simple equalization of time-varying channels for ofdm. *IEEE communications letters*, 9(7):619–621, 2005.

[80] N. Samuel, T. Diskin, and A. Wiesel. Deep mimo detection. In *2017 IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pages 1–5, July 2017.

[81] Neev Samuel, Tzvi Diskin, and Ami Wiesel. Learning to detect. *IEEE Trans. Signal Process.*, 67(10):2554–2564, 2019.

[82] Ulf D Schiller and Jochen J Steil. Analyzing the weight dynamics of recurrent learning algorithms. *Neurocomputing*, 63:5–23, 2005.

[83] Philip Schniter. Low-complexity equalization of ofdm in doubly selective channels. *IEEE Transactions on Signal processing*, 52(4):1002–1011, 2004.

[84] R. Shafin, L. Liu, J. Ashdown, J. Matyjas, M. Medley, B. Wysocki, and Y. Yi. Realizing green symbol detection via reservoir computing: An energy-efficiency perspective. In *2018 IEEE Intl Conf on Commun (ICC)*, pages 1–6, May 2018.

[85] R. Shafin, L. Liu, V. Chandrasekhar, H. Chen, J. Reed, and J. C. Zhang. Artificial intelligence-enabled cellular networks: A critical path to beyond-5G and 6G. *IEEE Trans. Wireless Commun.*, 27(2):212–217, 2020.

[86] Rubayet Shafin, Lingjia Liu, Vikram Chandrasekhar, Hao Chen, Jeffrey Reed, and Jianzhong Charlie Zhang. Artificial intelligence-enabled cellular networks: A critical path to beyond-5g and 6g. *IEEE Wireless Communications*, 27(2):212–217, 2020. doi: 10.1109/MWC.001.1900323.

[87] Haoran Sun, Xiangyi Chen, Qingjiang Shi, Mingyi Hong, Xiao Fu, and Nicholas D

Sidiropoulos. Learning to optimize: Training deep neural networks for interference management. *IEEE Transactions on Signal Processing*, 66(20):5438–5453, 2018.

[88] Tharaj Thaj and Emanuele Viterbo. Low complexity iterative rake decision feedback equalizer for zero-padded otfs systems. *IEEE Transactions on Vehicular Technology*, 2020.

[89] Shashank Tiwari, Suvra Sekhar Das, and Vivek Rangamgari. Low complexity lmmse receiver for otfs. *IEEE Commun. Lett.*, 23(12):2205–2209, 2019. doi: 10.1109/LCOMM.2019.2945564.

[90] Kien T Truong and Robert W Heath. Effects of channel aging in massive MIMO systems. *J. of Commun. and Netw.*, 15(4):338–351, 2013.

[91] Joao Vieira, Steffen Malkowsky, Karl Nieman, Zachary Miers, Nikhil Kundargi, Liang Liu, Ian Wong, Viktor Öwall, Ove Edfors, and Fredrik Tufvesson. A flexible 100-antenna testbed for massive mimo. In *2014 IEEE Globecom Wkshps*, pages 287–293, 2014.

[92] Zitong Yang, Yaodong Yu, Chong You, Jacob Steinhardt, and Yi Ma. Rethinking bias-variance trade-off for generalization of neural networks. *arXiv preprint arXiv:2002.11328*, 2020.

[93] H. Ye, G. Y. Li, and B. Juang. Power of deep learning for channel estimation and signal detection in OFDM systems. *IEEE Wireless Commun. Lett.*, 7(1):114–117, 2018.

[94] Zhongyuan Zhao. Deep-waveform: A learned OFDM receiver based on deep complex convolutional networks. *arXiv preprint arXiv:1810.07181*, 2018.

[95] Lizhong Zheng and David N. C. Tse. Diversity and multiplexing: A fundamental

tradeoff in multiple-antenna channels. *IEEE Transactions on information theory*, 49 (5):1073–1096, 2003.

[96] Zhou Zhou, Shashank Jere, Lizhong Zheng, and Lingjia Liu. Learning with knowledge of structure: A neural network-based approach for MIMO-OFDM detection. In *2020 IEEE Asilomar Conf on Signals, Systems, and Computers*.

[97] Zhou Zhou, Lingjia Liu, and Jianzhong Zhang. FD-MIMO via pilot-data superposition: Tensor-based doa estimation and system performance. *IEEE J. Sel. Topics Signal Process.*, 13(5):931–946, 2019.

[98] Zhou Zhou, Shashank Jere, Lizhong Zheng, and Lingjia Liu. Learning for integer-constrained optimization through neural networks with limited training. In *NeurIPS Wkshps on Learning Meets Combinatorial Algorithms*, Dec. 2020.

[99] Zhou Zhou, Lingjia Liu, Vikram Chandrasekhar, Jianzhong Zhang, and Yang Yi. Deep reservoir computing meets 5g MIMO-OFDM systems in symbol detection. In *AAAI Conference on Artificial Intelligence*, volume 34, pages 1266–1273, 2020.

[100] Zhou Zhou, Lingjia Liu, and Hao-Hsuan Chang. Learning for detection: MIMO-OFDM symbol detection through downlink pilots. *IEEE Trans. Wireless Commun.*, 19(6):3712–3726, 2020.

[101] Zhou Zhou, Lingjia Liu, Shashank Jere, Jianzhong Zhang, and Yang Yi. Rcnet: Incorporating structural information into deep rnn for online mimo-ofdm symbol detection with limited training. *IEEE Tran. on Wireless Commun.*, 20(6):3524 – 3537, 2021.

[102] Zhou Zhou, Lingjia Liu, Jiarui Xu, and Robert Calderbank. Learning to equalize OTFS. *arXiv preprint arXiv: 2107.08236*, 2021.