Untrained Neural Network based Symbol Detection for OTFS Systems

Hao Chang

A thesis submitted in fulfilment of the requirements for the degree of Master of Philosophy

Supervisors:
Dr. Wibowo Hardjawana
Prof. Branka Vucetic

Faculty of Engineering
School of Electrical and Computer Engineering
The University of Sydney

2023

To my mom and grandparents.

Statement of Originality

To the best of my knowledge, this thesis does not include materials that have been submitted for any degree or other purposes except where due acknowledgment has been made in the text.

I certify that the intellectual content of this thesis is the product of my own work and that all the assistance received in preparing this thesis and sources have been acknowledged.

Hao Chang

18 October 2023

Authorship Attribution Statement

I, Hao Chang, declare that the contents of this thesis are my own work. I am the lead author for the conference papers covered in this thesis as a Master of Philosophy (MPhil) candidate in the School of Electrical and Computer Engineering at the University of Sydney. The main contributions of this thesis are presented in Chapters 4 and 5.

• Chapter 4 presents an untrained neural network denoiser used to replace the MMSE one in Bayesian-based parallel interference cancellation. This work is presented in [C1].

• Chapter 5 presents a graph-based untrained neural network that embeds wireless interference knowledge into the neural network design.

In addition to the statements above, in cases where I am not the corresponding author of a published item, permission to include the published material has been granted by the corresponding author.

Hao Chang

18 October 2023

As a supervisor for the candidature upon which this thesis is based, I can confirm that the authorship attribution statements above are correct.

Wibowo Hardjawana

18 October 2023

Abstract

Future mobile systems will support various high-mobility scenarios (e.g., unmanned aerial vehicles and high-speed trains) with strict mobility requirements. However, the current orthogonal frequency division multiplexing (OFDM) is unsuitable for these scenarios because Doppler shifts caused by high-mobility reflectors lead to high inter-carrier interference (ICI). A new modulation scheme, orthogonal time frequency space (OTFS), was proposed to overcome this challenge. OTFS multiplexes the transmitted symbols in the delay-Doppler (DD) domain, then spreads them into the time-frequency (TF) domain. The DD domain multiplexing captures the delay and Doppler characteristics of the channel paths due to the high-mobility reflectors that cause ICI. The receiver needs to know numerous channel responses representing all wireless paths for each symbol as a matrix. The signal processing of a large matrix poses a significant challenge in OTFS symbol detector design.

Currently, state-of-the-art OTFS detectors consist of classical and training-based deep neural network (DNN) detectors. Classical OTFS detectors typically utilize matrix inversion operations to denoise the received signal before or within their ICI cancellation algorithms, resulting in high computational complexity. Training-based DNN OTFS detectors outperform classical detectors regarding symbol error rate (SER) performance. These detectors employ datasets to train the DNN to learn the symbol detection process before implementation. However, these detectors rely on enormous computation resources and the fidelity of datasets for the

training phase, and both are expensive. An alternative approach, an untrained DNN known as deep image prior (DIP), has recently been proposed as a denoiser for image processing. DIP requires no training datasets and employs an encoder-decoder DNN architecture to denoise the image. The appealing aspect of DIP in OTFS detector design is its potential to deliver a similar SER performance as training-based DNN. However, DIP consists of tens of layers and up to millions of trainable parameters; this results in high latency, making it unsuitable for real-time symbol detection.

In this thesis, we first propose an untrained DNN denoiser to iteratively denoise the received signal. We develop a new real-time, decoder-only DNN architecture for DIP called D-DIP denoiser. We use a decoder-only DNN to ensure low latency by using significantly fewer trainable parameters and layers than DIP's encoder-decoder architecture. We then combine the D-DIP denoiser with Bayesian-based parallel interference cancellation (BPIC) for symbol detection, resulting in the D-DIP-BPIC OTFS detector. The iterative BPIC algorithm employs PIC to subtract interference from the received signal using symbol estimates from the previous iteration. The Bayesian inference is then employed to obtain new symbol estimates based on the PIC outputs. Our simulation results demonstrate that the proposed D-DIP-BPIC OTFS detector outperforms state-of-the-art OTFS detectors regarding SER performance and computational complexity.

To further enhance the performance of the D-DIP denoiser, we introduce a novel approach by incorporating a graph representation of wireless interference knowledge into the D-DIP denoiser. Specifically, we devise a graph that captures the wireless interference relationships among the multiplexed transmitted symbols observed by the receiver. We construct an adjacency matrix based on this graph representation. To integrate this knowledge into the D-DIP denoiser, we embed the adjacency matrix and an additional layer after D-DIP to further improve its performance. We name the combination of graph wireless interference represen-

tation and D-DIP as GDIP denoiser. Simulation results demonstrate that the proposed GDIP denoiser requires significantly fewer iterations than the D-DIP one to denoise the received signal. The combination of GDIP and BPIC shows an excellent SER performance under various OTFS configurations.

Acknowledgements

First and foremost, I want to thank Dr. Wibowo Hardjawana for his consistent encouragement and guidance throughout my research journey. His expertise and mentorship have provided me with the necessary understanding of proper research practices. I am grateful for his numerous ideas and assistance in paper writing. Likewise, I sincerely thank Prof. Branka Vucetic for her professional advice and insightful suggestions.

I would also like to acknowledge the support and contributions of my friends and teammates, Alva Kosasih and Xinwei Qu. During my MPhil candidature, Alva provided valuable suggestions that have significantly enhanced my research. Xinwei has been instrumental in assisting me with mathematical models. I want to express my gratitude to Dr. Floriana Badalotti for her careful proofreading of my thesis. I am grateful to The University of Sydney for allowing me to study at one of the top universities in the world.

Finally, I express my deepest gratitude to my mom and grandparents for allowing me to study abroad and achieve certain milestones.

List of Publications

Conference Papers

[C1] H. Chang, A. Kosasih, W. Hardjawana, X. Qu, and B. Vucetic, "Untrained neural network based Bayesian detector for OTFS modulation systems," in *Proc. IEEE INFOCOM Wkshps*, 2023, pp. 1–6.

Contents

Statement of Originality ii

Authorship Attribution Statement iii

Abstract v

Acknowledgements viii

List of Publications ix

List of Figures xiii

List of Tables xv

List of Acronyms xv

1 Introduction 1

1.1 Research Problems and Contributions 4

1.2 Thesis Outline 6

2 Background 8

2.1 Linear Detectors 9

2.2 Iterative Detectors 10
2.2.1 Approximate Message Passing 11
2.2.2 Unitary Approximate Message Passing 12
2.2.3 Bayesian-based Parallel Interference Cancellation 13
2.2.4 Expectation Propagation 15
2.3 Training-based DNN Detector 16
2.4 Untrained Neural Network 18
2.4.1 Deep Image Prior 18
2.4.2 Graph Convolutional Generator 22

3 OTFS Systems **24**
3.1 OTFS System Model 25
3.1.1 OTFS Transmitter 25
3.1.2 OTFS Wireless Channel 26
3.1.3 OTFS Receiver 26
3.2 OTFS with Ideal Waveform 27
3.3 OTFS with Rectangular Waveform 29
3.4 OTFS Channel Condition Analysis 31

4 D-DIP-BPIC **34**
4.1 D-DIP Denoiser 35
4.1.1 Decoder DNN Architecture 35
4.1.2 Stopping Criteria and Convergence Analysis 37
4.2 Complexity Analysis 38
4.3 Simulation Results 39

5 GDIP-BPIC **43**

5.1 GDIP Denoiser 44

5.1.1 Graph Representation of Wireless Interference 44

5.1.2 Graph Embedding 44

5.2 Simulation Results 46

6 Conclusion **53**

6.1 Summary of the thesis 53

6.2 Future work 54

List of Figures

2.1 General wireless communication system 9

2.2 Images for denoising 19

2.3 Iterative denoising in image representation 21

2.4 Iterative denoising in PSNR representation 22

2.5 An undirected graph signal example 23

3.1 OTFS modulation system model 24

3.2 CDF of channel condition number 33

4.1 D-DIP-BPIC model 35

4.2 D-DIP iterative process 36

4.3 CDF of the iterations needed for D-DIP to satisfy stopping criteria 37

4.4 SER performance of the ideal waveform 41

4.5 SER performance of the rectangular waveform 42

4.6 SER performance of the rectangular waveform with fractional Doppler 42

5.1 Directed wireless interference graph 45

5.2 GDIP-BPIC model 46

5.3 SER performance under untrained environment 48

5.4 SER performance of GDIP for different modulation order 49

5.5 SER performance comparison of state-of-the-art OTFS detectors for different modulation order 49

5.6 Channel estimation error 51

5.7 OTFS vs OFDM 51

List of Tables

4.1 Computational complexity comparison 39

List of Acronyms

AMP approximate message passing.

AWGN additive white Gaussian noise.

BPIC Bayesian-based parallel interference cancellation.

BSE Bayesian symbol estimation.

BSO Bayesian symbol observation.

CDF cumulative distribution function.

CNN convolutional neural network.

DD delay-Doppler.

DIP deep image prior.

DNN deep neural network.

DSC decision statistics combining.

EP expectation propagation.

FC fully connected.

GCG graph convolutional generator.

GNN graph-based neural network.

ICI inter-carrier interference.

IDI inter-Doppler interference.

ISFFT inverse symplectic finite Fourier transform.

ISI inter-symbol interference.

MIMO multiple-input multiple-output.

MMSE minimum-mean-square-error.

MP message passing.

MRC maximum ratio combining.

MRI magnetic resonance imaging.

MSE mean-square-error.

OFDM orthogonal frequency division multiplexing.

OTFS orthogonal time frequency space.

PIC parallel interference cancellation.

PSNR peak signal to noise ratio.

QAM quadrature amplitude modulation.

SER symbol error rate.

SISO single-input and single-output.

SNR signal to noise ratio.

SVD singular value decomposition.

TF time-frequency.

UAMP unitary approximate message passing.

VB variational Bayes.

ZF zero-forcing.

Chapter 1

Introduction

Future wireless communication systems aim to accommodate diverse emerging applications (e.g., unmanned aerial vehicles, high-speed trains, and low-earth-orbit satellites) in high-mobility environments. However, communicating in such environments poses challenges due to the inherent time-varying nature of the wireless channel. Conventional orthogonal frequency division multiplexing (OFDM) modulation can achieve high spectral efficiency under time-invariant channels [1]. However, OFDM suffers from significant performance degradation under time-varying channels, where high Doppler shifts exist in high-mobility environments [2]. This degradation is due to the high inter-carrier interference (ICI) caused by high Doppler shifts. Consequently, OFDM is unsuitable to be employed in high-mobility scenarios. Promisingly, the orthogonal time frequency space (OTFS) [3] has been recently proposed for high-mobility scenarios, offering superior performance to the conventional OFDM modulation system. OTFS processes the transmitted symbols in the delay-Doppler (DD) and time-frequency (TF) domains, requiring tracking the channel's delay and Doppler characteristics of high-mobility reflectors that cause ICI. The OTFS channel responses parameterize the time-varying channel's effects by capturing the latter's delay and Doppler parameters

[1]. However, this results in a large OTFS channel matrix size, which poses a significant challenge in the design of OTFS detectors. Various OTFS detectors have been investigated in the literature, and they can be categorized as 1) classical and 2) Training-based deep neural network (DNN).

Classical OTFS detectors can be further divided into linear and iterative detectors. Low complexity linear detectors zero-forcing (ZF) and minimum-mean-square-error (MMSE) proposed in [4] only have a logarithmic complexity compared to conventional ZF and MMSE that have high computational complexity matrix inversion operation. However, they still cannot achieve an acceptable symbol error rate (SER) performance in OTFS systems. Iterative detectors typically provide a better SER performance than linear ones due to their interference approximation or interference cancellation algorithm. The message passing (MP) [5] and approximate message passing (AMP) [6] achieve superior SER performance to linear detectors by using Gaussian functions to approximate the interference. A variational Bayes (VB) OTFS detector proposed in [7] has shown the ability to improve the convergence of the MP OTFS detector. Unfortunately, they suffer from performance degradation in rich scattering environments (i.e., multiple mobile reflectors exit). The unitary approximate message passing (UAMP) [6] detector was proposed to address this issue by performing singular value decomposition (SVD) to apply the unitary transformation to the received signal and channel matrix before executing AMP. Additionally, a similar performance in terms of reliability and complexity to the UAMP detector has also been achieved by iterative Bayesian-based parallel interference cancellation (BPIC) one in [8]. It combined an MMSE denoiser, the Bayesian concept, and PIC to perform iterative symbol detection. Unfortunately, their performance is still sub-optimal compared to the expectation propagation (EP) [9] OTFS detector. EP can achieve Bayes-optimal performance using the Bayesian concept and multivariate Gaussian distributions to approximate the mean and variance of posterior detected symbols from the ob-

served received signals iteratively. However, the excellent performance achieved by UAMP, MMSE-BPIC, and EP comes at the cost of high complexity due to matrix inversion or SVD operations. The signal processing of these complex operations is costly in OTFS systems, as the OTFS channel matrix is large.

In addition to those iterative detectors, training-based DNN approaches are widely used in symbol detector design. These detectors utilize datasets to train the trainable parameters within the DNN, enabling them to learn the symbol detection process before implementation. Recently proposed training-based DNN OTFS detectors, e.g., symbol DNN [10] and 2-D convolutional neural network (CNN) [11] achieve a good SER performance by employing a large number of trainable parameters in the DNN design. Meanwhile, GAMP-NET[12], BPICNet[13], and EP-NET [14] have shown excellent performance in terms of SER and convergence by embedding only several trainable parameters into the corresponding GAMP, BPIC, and EP algorithms. However, training-based detectors typically require large datasets to train before implementation. There are two major disadvantages to the training-based DNN approach: 1) dependency on the availability of extensive computation resources that necessitate substantial energy or CO2 consumptions and high cost of the training phase [15]; and 2) the fidelity of synthetic training data, artificially generated due to high cost of acquiring real datasets, in the realistic environment [16]. For example, a high-fidelity training dataset implies the distribution functions for all possible velocity of mobile reflectors is known beforehand, which is impossible.

Promisingly, the recently proposed untrained DNN approach named deep image prior (DIP) [17] avoids the need for training datasets. The encoder-decoder architecture used in the original DIP shows excellent performance in image denoising, but the use of up to millions of trainable parameters results in high latency. Recently, the authors in [18] have shown that the decoder-only DIP performs similarly to an encoder-decoder DIP architecture when applied to magnetic resonance

imaging (MRI). The computational complexity and latency of a decoder-only DIP are significantly lower than the original encoder-decoder DIP, thus enhancing its potential use as a real-time OTFS detector. To date, the DIP concept has not been used in OTFS detector design.

To improve the performance of DNN, graph-based neural network (GNN) is proposed to process the signal by exploring their graph representation in the neural network design. The training-based GNN has shown an excellent SER performance in wireless communications, e.g., massive multiple-input multiple-output (MIMO) detection [19], multi-user MIMO detection [20], [21], and OTFS detection [22]. However, training those GNN will result in high computational complexity (e.g., need extensive computation resources to model the interference between different users in massive MIMO). Recently, an untrained GNN proposed in [23] uses a combination of signal's graph representation and CNN, referred to as graph convolutional generator (GCG). This approach has shown promising denoising performance in signal processing, avoiding the complex training process. Until now, no research has been conducted on untrained GNN in OTFS detector design.

1.1 Research Problems and Contributions

The main objective of this thesis is to develop untrained DNN based OTFS detectors that can eliminate the training process for the DNN and achieve high performance with low computational complexity. While existing OTFS detectors have demonstrated good SER performance, they suffer from high computational complexity when OTFS frame size increases. Additionally, some detectors perform poorly under ill-conditioned OTFS channels (i.e., high interference exists).

The first research problem we investigated in this thesis is how to eliminate high computational complex matrix inversion operation within the state-of-the-art

OTFS detector (i.e., BPIC [8]). We propose an untrained DNN denoiser to replace the MMSE one within the BPIC. We want to improve BPIC because the MMSE denoiser's performance is limited in terms of SER performance and computational complexity.

The main contribution of this architecture is that it is the first to combine untrained DNN (i.e., DIP), Bayesian inference, and PIC concept in OTFS detector design. Specifically, we propose an untrained DNN denoiser based on the DIP, decoder DNN structure, and stopping criteria, referred to as the D-DIP denoiser. We use stopping criteria to control the iterative denoising process of D-DIP. We then combine the D-DIP denoiser and BPIC for iterative symbol detection, leading to the D-DIP-BPIC OTFS detector. The SER performance and computational complexity of this architecture are presented in [C1], considering the rectangular (i.g., high interference exists) and ideal waveforms (i.g., low interference exists). The simulation results show that the D-DIP-BPIC offers a close to EP and BPIC-Net SER performance with much lower computational complexity under various OTFS configurations.

The second research problem we investigate is how to embed domain knowledge (i.g., wireless channel information) in the untrained DNN. We propose embedding the graph representation of wireless interference into D-DIP to improve it further, referred to as GDIP denoiser.

The main contribution of this architecture is that it is the first to combine graph representation of wireless interference and untrained DNN in the OTFS detector design. We first design a graph based on the wireless interference among the transmitted symbols seen at the receiver side. We then construct the adjacency matrix based on the designed graph. Finally, we embed the adjacency matrix and additional layer after D-DIP. The simulation results show that the GDIP denoiser significantly reduces the number of iterations needed to denoise the received signal

compared to the D-DIP one. Additionally, after being combined with BPIC, the proposed GDIP-BPIC is the only detectors that achieve a near EP SER performance under various OTFS configurations, while GDIP-BPIC has a much lower complexity.

1.2 Thesis Outline

Chapter 1 explains the research motivation and research problems. A brief overview of state-of-the-art OTFS detectors is also presented.

Chapter 2 investigates the basic concept of symbol detection and existing state-of-the-art detectors, consisting of linear, iterative, and training-based DNN detectors. Then an investigation on untrained DNN approaches, i.e., deep image prior and untrained graph convolutional network, is conducted.

Chapter 3 explains the OTFS system model, consisting of an OTFS transmitter, wireless channel, and OTFS receiver. We analyze the OTFS systems with different waveforms (i.e., rectangular and ideal waveforms) and different Doppler (i.e., integer and fractional Doppler). We then analyze the corresponding channel conditions.

Chapter 4 presents an untrained DNN OTFS detector consisting of the D-DIP denoiser and BPIC. We analyze the convergence of the D-DIP denoiser and compare the SER performance of the proposed detector with existing state-of-the-art OTFS detectors.

Chapter 5 discusses the approach of embedding graph interference knowledge into the D-DIP denoiser. We present a graph representation of the wireless interference and create a new denoiser architecture by embedding the graph into the denoiser design.

Chapter 6 summarizes the thesis and provides some insights into future work.

Chapter 2

Background

Symbol detection plays an essential role in communication systems, determining the quality of the estimated transmitted symbols. In this chapter, we investigate several types of detectors for wireless communication systems.

A general wireless transmission process is shown in Fig. 2.1, consisting of a transmitter, wireless channel, and receiver. The received signal $\mathbf{y}_c$ is obtained by passing the transmitted symbols $\mathbf{x}_c$ through the wireless channel matrix $\mathbf{H}_c$, expressed as

$$\mathbf{y}_c = \mathbf{H}_c\mathbf{x}_c + \mathbf{n}_c, \tag{2.1}$$

where $\mathbf{x}_c \in \mathbb{C}^{N_t}$ represents the complex-valued vector of transmitted symbols (e.g., quadrature amplitude modulation (QAM) symbols) with size N_t. $\mathbf{y}_c, \mathbf{n}_c \in \mathbb{C}^{N_r}$ and $\mathbf{H} \in \mathbb{C}^{N_r \times N_t}$ represent the complex-valued received signal, additive white Gaussian noise (AWGN) with a zero mean and covariance matrix $\sigma^2\mathbf{I}_{N_r}$, and channel matrix respectively. $\mathbf{I}_{N_r}$ is an N_r dimensional identical matrix, and N_t and N_r represent the number of transmitted and received symbols, respectively. For convenience,

Figure 2.1: General wireless communication system

we transfer the complex-valued model in (2.1) to a real-valued model as

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}, \tag{2.2}$$

where $\mathbf{x} = \left[\Re(\mathbf{x}_c)^T \ \Im(\mathbf{x}_c)^T\right]^T \in \mathbb{R}^{2N_t}$, $\mathbf{y} = \left[\Re(\mathbf{y}_c)^T \ \Im(\mathbf{y}_c)^T\right]^T \in \mathbb{R}^{2N_r}$, $\mathbf{n} = \left[\Re(\mathbf{n}_c)^T \ \Im(\mathbf{n}_c)^T\right]^T \in \mathbb{R}^{2N_r}$, $\mathbf{H} = \begin{bmatrix} \Re(\mathbf{H}_c) & -\Im(\mathbf{H}_c) \\ \Im(\mathbf{H}_c) & \Re(\mathbf{H}_c) \end{bmatrix} \in \mathbb{R}^{2N_r \times 2N_t}$, $\Re(\cdot)$ and $\Im(\cdot)$ takes the real and imaginary parts, respectively. We use this real-valued model for the rest of this chapter.

Symbol detection is the process of estimating the transmitted symbols $\mathbf{x}$ from the received signal $\mathbf{y}$ with the knowledge of channel information $\mathbf{H}$. In this chapter, we investigate several types of detectors, categorized as 1) linear, 2) iterative, and 3) training-based DNN detectors.

2.1 Linear Detectors

Classical linear detectors such as ZF and MMSE [4] are widely used. For ZF, the transmitted symbol $\hat{\mathbf{x}}$ is estimated by a matrix inversion operation

$$\hat{\mathbf{x}} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{y}, \tag{2.3}$$

where $(\cdot)^T$ represents the transpose of a matrix. Compared to ZF, MMSE considers the noise in the symbol estimations process, expressed as

$$\hat{\mathbf{x}} = (\mathbf{H}^T\mathbf{H} + \sigma^2\mathbf{I})^{-1}\mathbf{H}^T\mathbf{y}. \tag{2.4}$$

Typically, MMSE outperforms ZF because the noise impact is considered in the inversion operation. However, the matrix inversion operation within ZF and MMSE is costly for a large channel matrix. This will result in high computational complexity when applied in OTFS systems, as the OTFS channel matrix size is relatively large. Thus, they are not suitable to be implemented as OTFS detectors.

2.2 Iterative Detectors

Compared to linear detectors, iterative ones such as AMP [21], UAMP [24], MMSE-BPIC [25], and EP [20] can achieve a better SER performance as they use the Gaussian function to approximate the posterior probability distribution of the transmitted symbols conditioned on the received signal. Typically, the process within those detectors can be divided into symbol observation and symbol estimation models. In the symbol observation model, the mean and variance of the transmitted symbols are conducted based on the information of the received signal, channel, and noise. In the symbol estimation model, the transmitted symbols' mean and variance are obtained by the Gaussian function (also known as Bayesian estimation) and feedback to the observation model for the next iteration. The iterative process will stop when the maximum number of iterations is reached.

2.2.1 Approximate Message Passing

AMP is a low-complexity algorithm proposed to solve sparse linear inverse problems in compressed sensing [21]. Its low complexity is because only the matrix-vector multiplication is involved in its iterative process. It is also widely used in wireless communication applications, e.g., MIMO detection [21] and OTFS detection [6].

We use the AMP algorithm proposed in [26] and [21]. The symbol observation model is expressed as

$$V_k^{(t)} = \sum_{q=1}^{N_t} |h_{kq}|^2 v_q^{(t)}, \tag{2.5}$$

$$Z_k^{(t)} = \sum_{q=1}^{N_t} h_{kq}\hat{x}_q^{(t)} - \frac{V_k^{(t)}(y_k - Z_k^{(t-1)})}{\sigma^2 + V_k^{(t-1)}}, \tag{2.6}$$

$$\Sigma_q^{(t)} = \sum_{k=1}^{N_r} (|h_{kq}|^2(\sigma^2 + V_k^{(t-1)}))^{-1}, \tag{2.7}$$

$$r_q^{(t)} = \hat{x}_q^{(t)} + \Sigma_q^{(t)} \sum_{k=1}^{N_r} \frac{h_{kq}(y_k - Z_k^{(t)})}{\sigma^2 + V_k^{(t)}}, \tag{2.8}$$

where h_{kq} represents (k, q)-th entry of channel matrix $\mathbf{H}$, and y_k represents k-th symbol in the received signal $\mathbf{y}$, where $k = 1, \cdots, N_r, q = 1, \cdots, N_t$. $\hat{x}_q^{(t)}$ and $v_q^{(t)}$ are the estimated mean and variance of the q-th transmitted symbol, respectively. When $t = 1$, we set $\hat{x}_q^{(0)} = 0$, $v_q^{(0)} = 1$, and $Z_k^{(0)} = y_k$, as shown in [26] and [21]. The symbol's mean $r_q^{(t)}$ in (2.8) and variance $\Sigma_q^{(t)}$ in (2.7) are then forwarded to the estimation model to perform the posterior mean and variance estimation [21]

$$\hat{x}_q^{(t+1)} = \mathbb{E}\left[x_q \middle| r_q^{(t)}, \Sigma_q^{(t)}\right], \tag{2.9}$$

$$v_q^{(t+1)} = \mathbf{Var}\left[x_q \middle| r_q^{(t)}, \Sigma_q^{(t)}\right], \tag{2.10}$$

This process will repeat for $t = 1, \cdots, T_{\text{AMP}}$, where T_{AMP} is the maximum number

of AMP iterations. Finally, the estimated symbol is outputted in vector form, as $\hat{\mathbf{x}}^{(T_{\mathrm{AMP}})} = [\hat{x}_1^{(T_{\mathrm{AMP}})}, \cdots, \hat{x}_q^{(T_{\mathrm{AMP}})}, \cdots, \hat{x}_{N_t}^{(T_{\mathrm{AMP}})}]^T$.

This form of AMP is named Bayes-optimal AMP, which can achieve Bayes-optimal performance in i.i.d (sub)-Gaussian [26]. However, the AMP may file to converge when the channel matrix $\mathbf{H}$ is ill-conditioned (e.g., large channel condition number, non-zero mean) [6], [24], [27]. Therefore, AMP may suffer from performance degradation when applied in OTFS systems as the channel matrix in OTFS systems is probably ill-conditioned [28].

2.2.2 Unitary Approximate Message Passing

The UAMP proposed in [24] solves the problem of AMP by applying the unitary transformation to model (2.2) instead of directly working on it. The model after unitary transformation is obtained by SVD operation on channel matrix $\mathbf{H}$ (i.e., $\mathbf{H} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}$), expressed as in [24]

$$\mathbf{r} = \mathbf{\Phi}\mathbf{x} + \mathbf{w}, \tag{2.11}$$

where $\mathbf{r} = \mathbf{U}^H\mathbf{y}, \mathbf{\Phi} = \mathbf{U}^H\mathbf{H} = \mathbf{\Lambda}\mathbf{V}$. $\mathbf{U}, \mathbf{V}$ are unitary matrix, $\mathbf{\Lambda}$ is a rectangular diagonal matrix. The UAMP algorithm is implemented by replacing the channel matrix $\mathbf{H}$ and received signal $\mathbf{y}$ with $\mathbf{\Phi}$ and $\mathbf{r}$, respectively, before executing the AMP algorithm.

The UAMP performs better than AMP benefits from the SVD operation for the unitary transformation. However, the computational complexity of the SVD operation is high when applied in OTFS systems, as the OTFS channel matrix is large.

2.2.3 Bayesian-based Parallel Interference Cancellation

BPIC combines the Bayesian inference and parallel interference cancellation (PIC) concept. PIC is used to cancel the interference from the received signal iteratively, and Bayesian inference is used to estimate the transmitted symbols. Typically, BPIC is used together with MMSE denoiser for symbol estimation [8], [25]. BPIC executes MMSE before BPIC iterations, similar to UAMP (i.e., SVD is executed before AMP iteration).

BPIC consists of three modules: Bayesian symbol observation (BSO), Bayesian symbol estimation (BSE), and decision statistics combining (DSC). BSO is a matched filter-based PIC scheme that is used to estimate the transmitted symbols in iteration t based on the estimated symbols $\hat{\mathbf{x}}^{(t-1)}$ from the previous iteration. The BSO module is shown as

$$\mu_q^{(t)} = \hat{x}_q^{(t-1)} + \frac{\mathbf{h}_q^T \left(\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}^{(t-1)}\right)}{\|\mathbf{h}_q\|^2}, \tag{2.12}$$

where $\mu_q^{(t)}$ is the soft estimate of q-th symbol x_q in iteration t, and $\mathbf{h}_q$ is the q-th column of matrix $\mathbf{H}$. $\hat{\mathbf{x}}^{(t-1)} = [\hat{x}_1^{(t-1)}, \cdots, \hat{x}_q^{(t-1)}, \cdots, \hat{x}_{2N_t}^{(t-1)}]^T$ is the vector of the estimated symbol. The variance $\Sigma_q^{(t)}$ of the q-th symbol estimate is derived in[25] as

$$\Sigma_q^{(t)} = \frac{1}{(\mathbf{h}_q^T \mathbf{h}_q)^2} \left(\sum_{\substack{j=1 \\ j \neq q}}^{N_t} (\mathbf{h}_q^T \mathbf{h}_q)^2 v_j^{(t-1)} + (\mathbf{h}_q^T \mathbf{h}_q)\sigma^2 \right), \tag{2.13}$$

where $v_j^{(t-1)}$ is the j-th element in a vector of symbol estimates variance $\mathbf{v}^{(t-1)}$ in iteration $t-1$, and $\mathbf{v}^{(t-1)} = [v_1^{(t-1)}, \cdots, v_q^{(t-1)}, \cdots, v_{N_t}^{(t-1)}]^T$. We set $\mathbf{v}^{(0)} = 0$ because we have no prior knowledge of the variance at the beginning. Then the estimated symbol $\boldsymbol{\mu}^{(t)} = [\mu_1^{(t)}, \cdots, \mu_q^{(t)}, \cdots, \mu_{N_t}^{(t)}]^T$and variance $\boldsymbol{\Sigma}^{(t)} = [\Sigma_1^{(t)}, \cdots, \Sigma_q^{(t)}, \cdots, \Sigma_{N_t}^{(t)}]^T$ are forwarded to the BSE module.

In the BSE module, we compute the Bayesian symbol estimates and the variance of the q-th symbol obtained from the BSO module. given as

$$\hat{x}_q^{(t)} = \mathbb{E}\left[x_q \middle| \mu_q^{(t)}, \Sigma_q^{(t)}\right] = \sum_{a\in\Omega} a\hat{p}^{(t)}(x_q = a|\mathbf{y}) \tag{2.14}$$

$$v_q^{(t)} = \mathbb{E}\left[\left|x_q - \mathbb{E}\left[x_q \middle| \mu_q^{(t)}, \Sigma_q^{(t)}\right]\right|^2\right], \tag{2.15}$$

where $\hat{p}^{(t)}\left(x_q|\mathbf{y}\right) = \mathcal{N}(x_q : \mu_q^{(t)}, \Sigma_q^{(t)})$ is obtained from the BSO module, and it is normalized so that $\sum_{a\in\Omega}\hat{p}^{(t)}\left(x_q = a|\mathbf{y}\right) = 1$. The outputs of the BSE module, $\hat{x}_q^{(t)}$and $v_q^{(t)}$, are then sent to the following DSC module.

The DSC module performs a linear combination of the symbol estimates in two consecutive iterations, shown as

$$\hat{x}_q^{(t)} = \left(1 - \rho_q^{(t)}\right)\hat{x}_q^{(t-1)} + \rho_q^{(t)}\hat{x}_q^{(t)} \tag{2.16}$$

$$v_q^{(t)} = \left(1 - \rho_q^{(t)}\right)v_q^{(t-1)} + \rho_q^{(t)}v_q^{(t)}. \tag{2.17}$$

The weighting coefficient is determined by maximizing the signal-to-interference-plus-noise-ratio variance, given as

$$\rho_q^{(t)} = \frac{\epsilon_q^{(t-1)}}{\epsilon_q^{(t)} + \epsilon_q^{(t-1)}}, \tag{2.18}$$

where $\epsilon_q^{(t)}$ is defined as the instantaneous square error of the q-th symbol estimate, computed by using the maximum ratio combining (MRC) filter, shown as

$$\epsilon_q^{(t)} = \left\| \frac{\mathbf{h}_q^T}{\|\mathbf{h}_q\|^2}\left(\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}^{(t)}\right)\right\|^2. \tag{2.19}$$

The weighted symbol estimates $\hat{\mathbf{x}}^{(t)}$, and their variance $\mathbf{v}^{(t)}$ are then returned to the BSO module to continue the iteration. After T_{BPIC} iterations, $\hat{\mathbf{x}}^{(T_{\text{BPIC}})}$ is taken

as a vector of symbol estimates.

MMSE denoiser is used to yield good initial symbol estimates as

$$\hat{\mathbf{x}}^{(0)} = (\mathbf{H}^T\mathbf{H} + \sigma^2\mathbf{I})^{-1}\mathbf{H}^T\mathbf{y} \tag{2.20}$$

This is because the BPIC performance is highly related to the symbol estimate in the first iteration [25]. However, this also brings extra complexity due to the matrix inversion inside the MMSE denoiser, which is costly when applied in OTFS systems.

2.2.4 Expectation Propagation

EP algorithm can achieve a Bayes-optimal performance compared to AMP, UAMP, and MMSE-BPIC. We consider the EP in [20], where the symbol observation process can be expressed as

$$\boldsymbol{\Sigma}^{(\boldsymbol{t})} = (\sigma^{-2}\mathbf{H}^H\mathbf{H} + \boldsymbol{\lambda}^{(t-1)})^{-1}, \tag{2.21}$$

$$\boldsymbol{\mu}^{(\boldsymbol{t})} = \boldsymbol{\Sigma}^{(\boldsymbol{t})}(\sigma^{-2}\mathbf{H}^H\mathbf{y} + \boldsymbol{\gamma}^{(t-1)}), \tag{2.22}$$

$$\mathbf{v}^{(t)} = (diag(\boldsymbol{\Sigma}^{(\boldsymbol{t})})^{-1} - \boldsymbol{\lambda}^{(t-1)}), \tag{2.23}$$

$$\mathbf{x}^{(t)} = \mathbf{v}^{(t)}(\boldsymbol{\mu}^{(\boldsymbol{t})}diag(\boldsymbol{\Sigma}^{(\boldsymbol{t})})^{-1} - \boldsymbol{\gamma}^{(t-1)}), \tag{2.24}$$

where $\boldsymbol{\lambda}^{(t)}, \boldsymbol{\gamma}^{(t)}$ are tunable parameters. When $t = 1$, they are set as $\boldsymbol{\lambda}^{(0)} = \mathbf{1}, \boldsymbol{\gamma}^{(0)} = \mathbf{0}$. Similar to AMP, the mean $\mathbf{x}^{(t)}$ in (2.24) and variance $\mathbf{v}^{(t)}$ in (2.23) are then forwarded to the symbol estimation model, shown as

$$\hat{\mathbf{x}}^{(t)} = \mathbb{E}\left[\mathbf{x}\middle|\mathbf{x}^{(t)}, \mathbf{v}^{(t)}\right], \tag{2.25}$$

$$\hat{\mathbf{v}}^{(t)} = \mathbf{Var}\left[\mathbf{x}\middle|\mathbf{x}^{(t)}, \mathbf{v}^{(t)}\right]. \tag{2.26}$$

We then calculate $\boldsymbol{\lambda}^{(t)}, \boldsymbol{\gamma}^{(t)}$ according to the estimated mean and variance in (2.25) and (2.26), respectively.

$$\boldsymbol{\lambda}^{(\boldsymbol{t})} = (\hat{\mathbf{v}}^{(t)})^{-1} - (\mathbf{v}^{(t)})^{-1}, \tag{2.27}$$

$$\boldsymbol{\gamma}^{(\boldsymbol{t})} = \hat{\mathbf{x}}^{(t)}(\hat{\mathbf{v}}^{(t)})^{-1} - \mathbf{x}^{(t)}.(\mathbf{v}^{(t)})^{-1} \tag{2.28}$$

Finally, we weight $\boldsymbol{\lambda}^{(t)}, \boldsymbol{\gamma}^{(t)}$ based on the previous and current iteration with a damping factor η

$$\boldsymbol{\lambda}^{(\boldsymbol{t})} = (1-\eta)\boldsymbol{\lambda}^{(\boldsymbol{t})} + \eta\boldsymbol{\lambda}^{(\boldsymbol{t-1})}, \tag{2.29}$$

$$\boldsymbol{\gamma}^{(\boldsymbol{t})} = (1-\eta)\boldsymbol{\gamma}^{(\boldsymbol{t})} + \eta\boldsymbol{\gamma}^{(\boldsymbol{t-1})}. \tag{2.30}$$

The iterative process will repeat until the maximum number of EP iterations (i.e., $t = 1, \cdots, T_{\mathrm{EP}}$). The final symbol estimation is shown as $\hat{\mathbf{x}}^{(T_{\mathrm{EP}})}$

However, the matrix inversion operation in (2.21) is done in every EP iteration, which brings a high computational complexity when applied to OTFS systems. Thus, low complexity and near EP performance detector design is attractive.

2.3 Training-based DNN Detector

Deep neural network (DNN) has been a hot spot in recent years. It can learn a specific process using datasets to train the neural network, optimizing the neural network parameters. A well-trained DNN can be used to predict the output for a given input via the learned process. DNN-based detectors are widely used in wireless communications, such as BPICNet [13], which can achieve a close to EP SER performance with lower computational complexity. BPICNet embeds learnable neurons into the BSO and DSC models for every BPIC iteration. The

equations (2.12),(2.13), and (2.19) can be rewritten for BPICNet as

$$\mu_q^{(t)}(\theta_1^{(t)}) = \hat{x}_q^{(t-1)} + \theta_1^{(t)} \frac{\mathbf{h}_q^T \left(\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}^{(t-1)}\right)}{\|\mathbf{h}_q\|^2}, \tag{2.31}$$

$$\Sigma_q^{(t)}(\theta_2^{(t)}) = \frac{\theta_2^{(t)}}{(\mathbf{h}_q^T \mathbf{h}_q)^2} \left(\sum_{\substack{j=1 \\ j \neq q}}^{Nt} (\mathbf{h}_q^T \mathbf{h}_q)^2 v_j^{(t-1)} + (\mathbf{h}_q^T \mathbf{h}_q)\sigma^2 \right), \tag{2.32}$$

$$\epsilon_q^{(t)}(\theta_3^{(t)}) = \theta_3^{(t)} \left\| \mathbf{h}_q^T \left(\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}^{(t)}\right) \right\|^2, \tag{2.33}$$

where $\theta_1^{(t)}$, $\theta_2^{(t)}$, and $\theta_3^{(t)}$ for $t = 1, \cdots, T_{\text{BPICNet}}$ are trainable neurons that are adjusted by minimizing the loss function via backpropagation, shown as

$$\mathcal{L} = \frac{1}{256} \sum_{w=1}^{256} \sum_{q=1}^{Nt} (x_{q,w} - \hat{x}_{q,w}^{(T_{\text{BPICNet}})})^2 \tag{2.34}$$

where w is the index of the training sample in each batch and $\hat{x}_{q,w}^{(T_{\text{BPICNet}})}$ refers to the final BPICNet symbol estimate of $x_{q,w}$ obtained from the last layer.

Although the SER performance achieved by BPICNet is close to EP, there are two main weaknesses for this trained DNN-based detector. First, it requires a large amount of training data in the training process (i.e., 40 batches of 256 samples in each epoch, resulting in approximately five million samples for 500 epochs). Thus, the training process takes a long time before implementation. Second, it might suffer from performance degradation when testing for unseen data (i.e., training for some channel configurations but testing for different configurations, which often happens as the mobility of mobile reflectors varies).

2.4 Untrained Neural Network

Compared to training-based neural networks (e.g., DNN, GNN), untrained one avoids the need for training data and the training process, which makes it attractive in OTFS detector design. Firstly, we consider an untrained DNN approach named deep image prior [17]. We then consider an graph convolutional generator (GCG) proposed in [23] that combines the signal's graph representation and untrained CNN.

2.4.1 Deep Image Prior

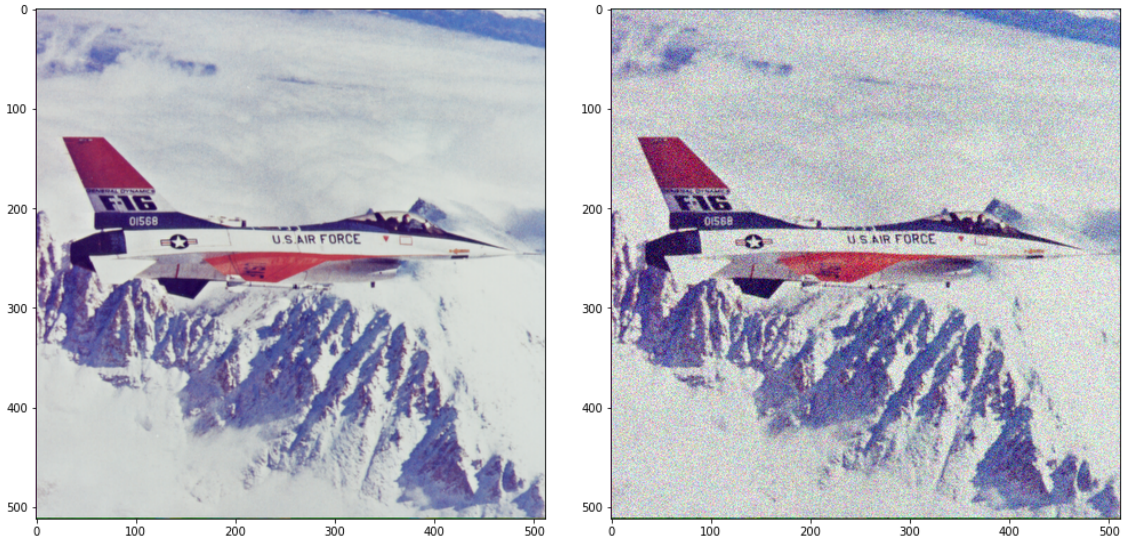
DIP is an untrained DNN requiring no training datasets [17]. It was initially proposed to solve image inverse problems as a denoiser. Recently, DIP has been applied successfully in many areas (e.g., magnetic resonance imaging [18], positron emission tomography [29], and compressed sensing [30]). DIP only needs a noisy image and a well-constructed DNN for image denoising. This avoids the need for labeled data that might be hard to obtain in some cases (e.g., medical images [18]). Typically, in image denoising, the system model is different from (2.2), where $\mathbf{H}$ is set to $\mathbf{I}$. Thus, we consider an image denoising model $\mathbf{x}_0 = \mathbf{x} + \mathbf{n}$, where $\mathbf{x}$ is the ground truth clean image, $\mathbf{x}_0$ is the noisy observation, and $\mathbf{n}$ is the noise. The denoising process estimates the clean image $\mathbf{x}$ from the noise observation $\mathbf{x}_0$.

DIP uses a L-layer DNN as prior knowledge to solve the denoising problem by assuming the DNN output is the estimated clean image, expressed as

$$\hat{\mathbf{x}}^{(i)} = f_L^{(i)}(f_{L-1}^{(i)}(\cdots(f_1^{(i)}(\mathbf{z}_1^{(i)})))), \tag{2.35}$$

$$f_l^{(i)}(\mathbf{z}_l^{(i)}) = ReLU(\mathbf{W}_l^{(i)}\mathbf{z}_l^{(i)} + \mathbf{b}_l^{(i)}), \tag{2.36}$$

where $\hat{\mathbf{x}}^{(i)}$ is the DNN output at iteration i that is assumed as the estimated



(a) Ground truth clean image (b) Noisy observation

Figure 2.2: Images for denoising

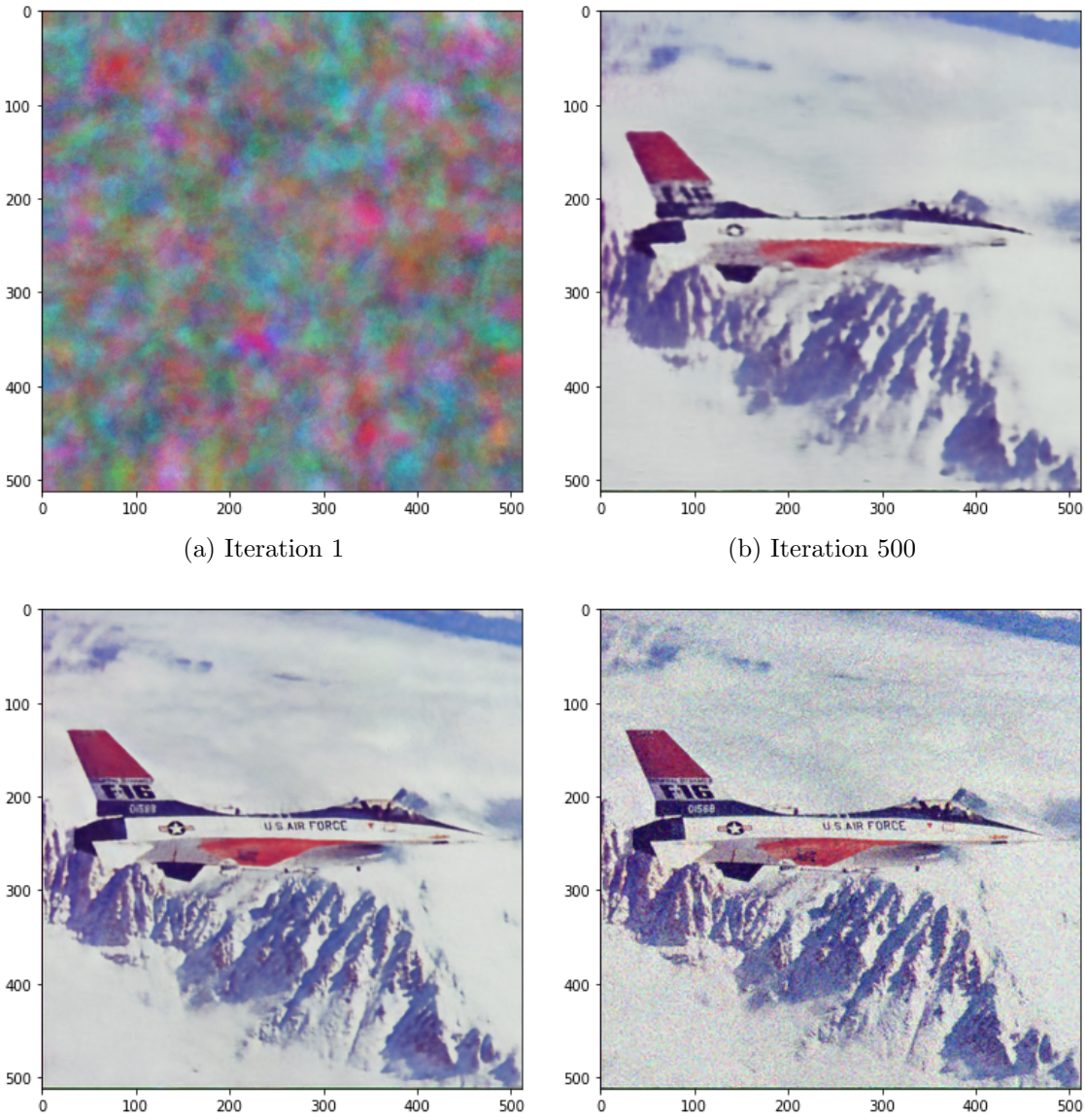
clean image, and $f_l^{(i)}(\mathbf{z}_l^{(i)})$ and $\mathbf{z}_l^{(i)}$ represent the output and input of layer l at iteration i, respectively. $\mathbf{z}_1^{(i)} \in \mathbb{R}^{c_1 \times 1}$ is a random input vector, and it is fixed during the iteration. $\mathbf{W}_l^{(i)} \in \mathbb{R}^{c_l \times c_{l-1}}$ represents the weight matrix between layer l and $l-1$ at iteration i. $\mathbf{b}_l^{(i)} \in \mathbb{R}^{c_l \times 1}$ is the bias vector in layer l at iteration i. c_l is the number of neurons in layer l. *ReLU* is the activation function. We use the mean-square-error (MSE) loss function to calculate the loss, shown as

$$\mathcal{L}^{(i)} = \frac{1}{c_L} \|\hat{\mathbf{x}}^{(i)} - \mathbf{x}_0\|^2, \tag{2.37}$$

we then calculate the gradients and update the DNN weights and bias via optimization techniques (e.g., Adam optimizer [31]).

Here, we give an example of image denoising, as shown in Fig.2.2. Fig. 2.2a is the clean image $\mathbf{x}$, and Fig. 2.2b is the noisy observation $\mathbf{x}_0$ after adding noise. Fig. 2.3 shows the iterative denoising performance in image representation. Fig. 2.3a shows that at iteration 1, the DIP output $\hat{\mathbf{x}}^{(1)}$ is messy due to the random input to

DIP. As the iteration goes on, DIP is learning the images and the basic structure of the images is shown in Fig. 2.3b. Moreover, Fig. 2.3c shows that the DIP output $\hat{\mathbf{x}}^{(2500)}$ is close to the ground truth clean image $\mathbf{x}$ at around 2500 iterations. However, Fig. 2.3d shows that the DIP output is similar to noisy observation $\mathbf{x}_0$. Also, Fig. 2.4 shows the peak signal to noise ratio (PSNR) measurement of the DIP iterative denoising process: the value of PSNR determines the difference between two images (i.e., a large PSNR means a slight difference, and vice versa for a small PSNR). We use two curves to represent the PSNR of $\hat{\mathbf{x}}^{(i)}$ to $\mathbf{x}$ and $\mathbf{x}_0$, respectively. Fig. 2.4 demonstrates that DIP has the ability to learn the clean image $\mathbf{x}$ and noise $\mathbf{n}$ but learns the clean image faster than the noise. However, the DIP output will eventually overfit to the noise, as shown at the point of 10000 iterations. Thus, a proper stopping criteria for DIP is needed, as the fixed number of iterations used in the DIP [17] is hard to generalize. That is because the iterations required for denoising different images might differ [32]. Note that the image denoising model can be extended to (2.2) for received signal denoising in wireless communication. We will explain the details in Chapter 4.



(a) Iteration 1

(b) Iteration 500

(c) Iteration 2500

(d) Iteration 10000

Figure 2.3: Iterative denoising in image representation

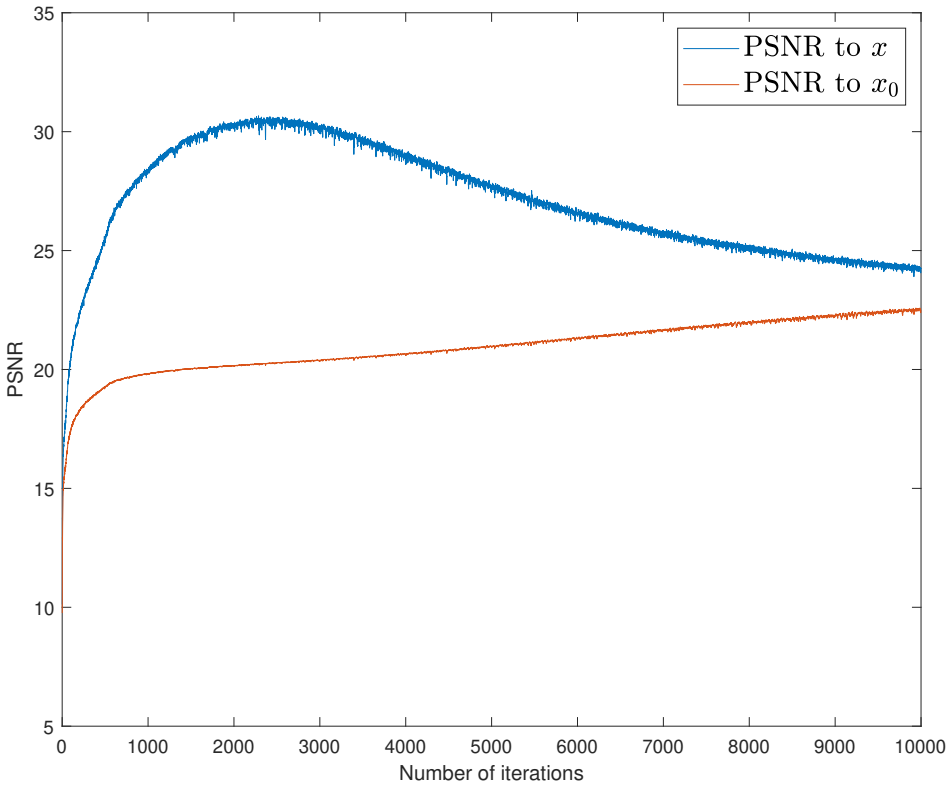


Figure 2.4: Iterative denoising in PSNR representation

2.4.2 Graph Convolutional Generator

Different from DIP, which is used to denoise images in fixed grids (e.g., 512x512 sized images in Fig. 2.2). The untrained graph neural network proposed in [23] can denoise un-euclidean data (i.e., a graph signal), referred to graph convolution generator (GCG). A graph signal can be represented as $\mathcal{G}(v, \epsilon)$, where v and ϵ represent sets of nodes and edges in graph $\mathcal{G}$. For example, we consider an undirected graph with four nodes, as shown in Fig. 2.5. Node i, j are connected if $e_{ij} \neq 0$, then the adjacency matrix of the graph can be represented as $\mathbf{A}$, where each entry $A_{i,j} = e_{j,i}$. Note that, for an undirected graph, the adjacency matrix is symmetric. The neural network in the GCG can be seen as embedding graph

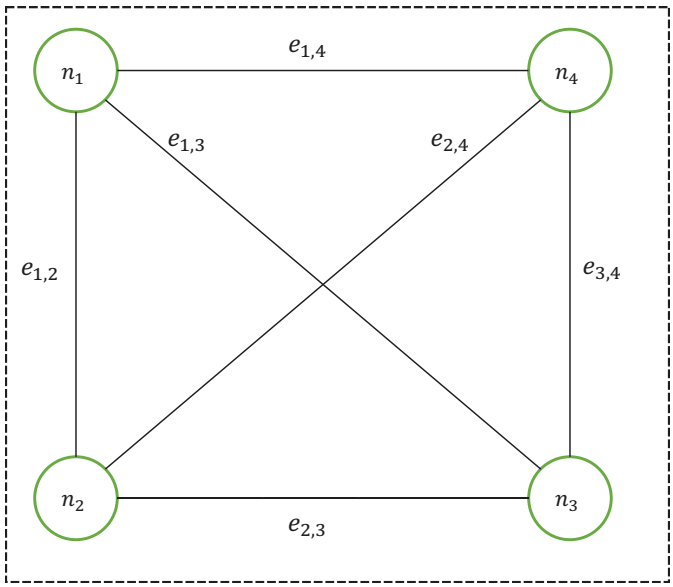


Figure 2.5: An undirected graph signal example

information into the neural network layers. Thus, we can rewrite (2.36) as

$$f_l^{(i)}(\mathbf{z}_l^{(i)}) = \begin{cases} ReLU(\mathbf{W}_l^{(i)}(\mathbf{A}'\mathbf{z}_l^{(i)}) + \mathbf{b}_l^{(i)})) & l = 1, \cdots, L-1 \\ \mathbf{W}_l^{(i)}(\mathbf{A}'\mathbf{z}_l^{(i)}) + \mathbf{b}_l^{(i)}) & l = L, \end{cases} \tag{2.38}$$

where $\mathbf{A}'$ is a polynomial of the adjacency matrix, as shown in [23]. $\mathbf{A}' \coloneqq \sum_{m=0}^{M-1} h_m \mathbf{A}^m$, where M determines the diffuse signals across $(M-1)$-hop neighborhoods, and h_m is the corresponding coefficient.

Chapter 3

OTFS Systems

The interference approximation and cancellation within those iterative detectors are beneficial for OTFS systems, as the OTFS has the ability to model the interference efficiently. In this chapter, we investigate a single-input and single-output (SISO) OTFS system used in [5] and [6], as illustrated in Fig. 3.1. First, we explain the OTFS system from the transmitter, wireless channel, and receiver perspective. We then analyze the OTFS input-output relationship under ideal and rectangular waveforms with integer and fractional Doppler. Finally, we analyze the channel conditions for OTFS systems with different waveforms and Doppler.

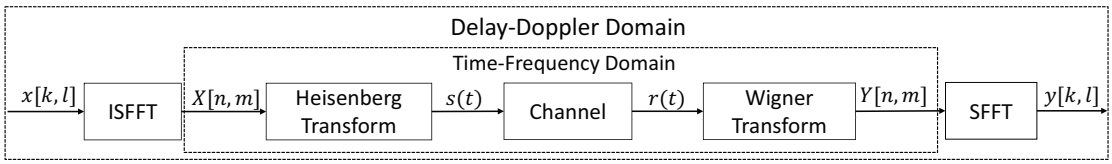


Figure 3.1: OTFS modulation system model

3.1 OTFS System Model

3.1.1 OTFS Transmitter

On the transmitter side, we allocate O-ary quadrature amplitude modulation (O-QAM) symbols into an $N \times M$ grid in the DD domain, shown as $x[k,l]$, where N and M represent the number of subcarriers and time slots used, respectively. $k = 0, \cdots, N-1$, $l = 0, \cdots, M-1$, are the indices of discretized delay and Doppler shifts, respectively. As illustrated in Fig. 3.1, the DD domain symbols are transformed into the TF domain by using the inverse symplectic finite Fourier transform (ISFFT) [5], shown as

$$X[n,m] = \frac{1}{NM} \sum_{k=0}^{N-1} \sum_{l=0}^{M-1} x[k,l] e^{j2\pi(\frac{nk}{N} - \frac{ml}{M})}, \tag{3.1}$$

where $n = 0, \cdots, N-1$, $m = 0, \cdots, M-1$.

The Heisenberg transform is applied to generate the time domain transmitted signal with a transmit waveform $g_{tx}(t)$

$$s(t) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} X[n,m] g_{tx}(t - nT) e^{j2\pi m\Delta f(t-nT)}, \tag{3.2}$$

where Δf is the subcarrier spacing and $T = \frac{1}{\Delta f}$. The transmitted signal then passes through the wireless channel.

3.1.2 OTFS Wireless Channel

The OTFS wireless channel is a time-varying multi-path channel, represented by the impulse responses in the DD domain

$$h(\tau, v) = \sum_{i=1}^{P} h_i \delta(\tau - \tau_i)\delta(v - v_i), \tag{3.3}$$

where $\delta(\cdot)$ is the Dirac delta function, $h_i \sim \mathcal{N}(0, 1/P)$ denotes the i-th path gain, and P is the total number of propagation paths. Each path represents a channel between a moving reflector/transmitter and a receiver with different delay τ_i and Doppler v_i characteristics. The delay and Doppler shifts are given as

$$\tau_i = l_i \frac{1}{M\Delta f}, \tag{3.4}$$

$$v_i = \frac{k_i + \kappa_i}{NT}, \tag{3.5}$$

where l_i and k_i denote the indices of the delay index and integer Doppler index for the i-th path, respectively. For each path, $l_i \in [0, l_{max}]$ and $k_i \in [-k_{max}, k_{max}]$ are randomly selected integers. κ_i represents the fractional Doppler associated with the i-th path that is drawn from a uniform distribution $(-0.5, 0.5]$ because the Doppler index may not perfectly align with the grids as illustrated in [5].

3.1.3 OTFS Receiver

On the receiver side, the time domain received signal $s(t)$ is obtained by passing $s(t)$ through a time-varying multi-path channel expressed as

$$r(t) = \int\int h(\tau, v)s(t - \tau)e^{j2\pi v(t-\tau)}d\tau dv = \sum_{i=1}^{P} h_i s(t - \tau_i)e^{j2\pi v_i(t-\tau_i)}, \tag{3.6}$$

where $h(\tau, v)$ is the channel impulse response. The time domain received signal $r(t)$ is transformed to TF domain with a receive waveform $g_{rx}(t)$

$$Y(t,f) = \int g_{rx}^{*}(t'-t) r(t') e^{-j2\pi f(t'-t)} dt', \tag{3.7}$$

we then can get a discrete TF domain received signal $Y[n,m]$ by sampling $Y(t,f)$ at $t = nT$ and $f = m\Delta f$, shown in [33] as

$$\begin{aligned} Y[n,m] = Y(t,f)|_{t=nT, f=m\Delta f} = \sum_{n'=0}^{N-1} \sum_{m'=0}^{M-1} X[n',m'] \\ \times \sum_{i=1}^{P} h_i e^{j2\pi m' \Delta f(-\tau_i)} e^{j2\pi v_i(-\tau_i)} e^{j2\pi v_i(nT)} \\ \times A_{g_{rx},g_{tx}}((n-n')T - \tau_i, (m-m')\Delta f - v_i), \end{aligned} \tag{3.8}$$

where $A_{g_{rx},g_{tx}}((n-n')T - \tau_i, (m-m')\Delta f - v_i)$ is the ambiguity function between $g_{rx}(t)$ and $g_{tx}(t)$, which is used to model the interference generated by the pulses.

Finally, the DD domain received signal $y[k,l]$ is obtained by applying SFFT to $Y[n,m]$ given by

$$y[k,l] = \frac{1}{NM} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} Y[n,m] e^{-j2\pi(\frac{nk}{N} - \frac{ml}{M})}, \tag{3.9}$$

3.2 OTFS with Ideal Waveform

In this section, we investigate the ideal waveform in the OTFS system, which can help us find its upper bound despite its not applicable in the real world [5]. Under ideal waveform, $g_{tx}(t)$ and $g_{rx}(t)$ satisfy the bi-orthogonal property, resulting in $A_{g_{rx},g_{tx}}((n-n')T - \tau_i, (m-m')\Delta f - v_i) = 1$, for $n = n', m = m'$ and zeros otherwise [33] in (3.8). The OTFS input-output relationship in the DD domain

can then be expressed as [6]

$$y[k,l] = \sum_{i=1}^{P} \sum_{\gamma=-N_i}^{N_i} h_i \frac{1 - e^{-j2\pi(-\gamma-\kappa_i)}}{N - Ne^{-j2\pi\frac{-\gamma-\kappa_i}{N}}} e^{-j2\pi\frac{l_i(k_i+\kappa_i)}{NM}} \times x[[k-k_i+\gamma]_N, [l-l_i]_M] + w[k,l], \tag{3.10}$$

where $N_i < N$ is an integer, $w[k,l]$ is the Gaussian noise in the DD domain that follows $\mathcal{N}(0,\sigma_c^2)$, σ_c^2 is the noise variance. The received signal $y[k,l]$ is a linear combination of $S = \Sigma_{i=1}^{P} 2N_i + 1$ transmitted signal. The transmitted signals $x[[k-k_i+\gamma]_N, [l-l_i]_M]$, for $\gamma = -N_i, \cdots, N_i$ and $\gamma \neq 0$ are treated as interference to the signal $x[[k-k_i]_N, [l-l_i]_M]$, referred to as inter-Doppler interference (IDI) [5]. The matrix-vector form of (3.10) is shown as

$$\mathbf{y}_{\mathrm{DD}} = \mathbf{H}_{\mathrm{ideal}}\mathbf{x}_{\mathrm{DD}} + \mathbf{w}_{\mathrm{DD}}, \tag{3.11}$$

where $\mathbf{y}_{\mathrm{DD}}, \mathbf{x}_{\mathrm{DD}}, \mathbf{w}_{\mathrm{DD}} \in \mathbb{C}^{NM\times 1}$ represent the received signal, transmitted signal, and noise in the DD domain, respectively. The $(k+Nl)$-th element of $\mathbf{y}_{\mathrm{DD}}$ is $y_{k+Nl} = y[k,l]$ for $k = 0, \cdots, N-1, l = 0, \cdots, M-1$, and similar for $\mathbf{x}_{\mathrm{DD}}$ and $\mathbf{w}_{\mathrm{DD}}$. $\mathbf{H}_{\mathrm{ideal}} \in \mathbb{C}^{NM\times NM}$ is the effective channel matrix with the ideal waveform in the DD domain, and it can be expressed as [6]

$$\mathbf{H}_{\mathrm{ideal}} = \sum_{i=1}^{P} \sum_{\gamma=-N_i}^{N_i} \mathbf{I}_N(-[\gamma-k_i]_N) \otimes [\mathbf{I}_M(l_i)h_i \times \left(\frac{1 - e^{-j2\pi(-\gamma-\kappa_i)}}{N - Ne^{-j2\pi\frac{-\gamma-\kappa_i}{N}}}\right) e^{-j2\pi\frac{l_i(k_i+\kappa_i)}{NM}}], \tag{3.12}$$

where $\mathbf{I}_N(-[\gamma-k_i]_N)$ is an $N \times N$ matrix that circularly shifts the rows of the identity matrix by $-[-\gamma-k_i]_N$, and similar for $\mathbf{I}_M(l_i)$.

When only the integer Doppler exists (i.e., $\kappa_i = 0, \forall i$). IDI disappears as the

Doppler frequency exactly matches the corresponding Doppler taps. By replacing $N_i = 0$, the received signal in (3.10) can be rewritten as [5]

$$y[k,l] = \sum_{i=1}^{P} h_i e^{-j2\pi \frac{l_i k_i}{NM}} x[[k-k_i]_N, [l-l_i]_M] + w[k,l], \tag{3.13}$$

and correspondingly, the $\mathbf{H}_{\text{ideal}}$ in integer Doppler cases can be simplified as [6]

$$\mathbf{H}_{\text{ideal}} = \sum_{i=1}^{P} \mathbf{I}_N(k_i) \otimes \left(\mathbf{I}_M(l_i) h_i e^{-j2\pi \frac{l_i k_i}{NM}} \right). \tag{3.14}$$

3.3 OTFS with Rectangular Waveform

We analyze a more practical rectangular waveform commonly used in the literature, as the ideal waveform can not be used in practice. Under the rectangular waveform, the bi-orthogonal property does not exist. Thus, interference generated in (3.8) degrades the OTFS system performance. Here, we assume the rectangular pulse has an amplitude of $\frac{1}{\sqrt{T}}$ for $t \in [0, T]$, and 0 otherwise. The OTFS input-output relationship in the DD domain is expressed as [5],

$$\begin{aligned} y[k,l] = \sum_{i=1}^{P} \sum_{q=-N_i}^{N_i} & h_i e^{j2\pi(\frac{l-l_i}{M})(\frac{k_i+\kappa_i}{N})} \alpha_i(k,l,q) \\ & \times x[[k-k_i+q]_N, [l-l_i]_M] + w[k,l], \end{aligned} \tag{3.15}$$

where

$$\alpha_i(k,l,q) = \begin{cases} \frac{1}{N}(\beta_i(q)-1) e^{-j2\pi \frac{[k-k_i+q]_N}{N}}, & 0 \le l < l_i \\ \frac{1}{N}\beta_i(q), & l_i \le l < M, \end{cases} \tag{3.16}$$

and

$$\beta_i(q) = \frac{e^{-j2\pi(-q-\kappa_i)} - 1}{e^{-j\frac{2\pi}{N}(-q-\kappa_i)} - 1}. \tag{3.17}$$

Note the rectangular pulses do not satisfy the biorthogonality conditions. Thus, there will be inter-symbol interference (ISI) and inter-carrier interference (ICI) due to the ambiguity function and IDI due to the fractional Doppler. That differs from the ideal waveform, where ISI and ICI do not exist. We then write the matrix-vector form of (3.15) as

$$\mathbf{y}_{\mathrm{DD}} = \mathbf{H}_{\mathrm{rect}}\mathbf{x}_{\mathrm{DD}} + \mathbf{w}_{\mathrm{DD}}, \tag{3.18}$$

where $\mathbf{y}_{\mathrm{DD}}, \mathbf{x}_{\mathrm{DD}}, \mathbf{w}_{\mathrm{DD}} \in \mathbb{C}^{NM\times 1}$ represent the received signal, transmitted signal, and noise in the DD domain, respectively. $\mathbf{H}_{\mathrm{rect}}$ is the effective channel matrix with the rectangular waveform in the DD domain, and it can be expressed as [34]

$$\mathbf{H}_{\mathrm{rect}} = \sum_{i=1}^{P}\sum_{q=-N_i}^{N_i} \mathbf{I}_N(-[q-k_i]_N) \otimes \left(\mathbf{\Lambda}\mathbf{I}_M(l_i) h_i e^{-j2\pi\frac{l_i(k_i+\kappa_i)}{MN}}\right) \cdot \mathbf{\Delta}^{-[q-k_i]_N}, \tag{3.19}$$

where $\mathbf{\Lambda} \in \mathbb{C}^{M\times M}$ is a diagonal matrix and its l-th diagonal element $\mathbf{\Lambda}_{ll}$ is shown as

$$\mathbf{\Lambda}_{ll} = \begin{cases} e^{-j2\pi\frac{l(k_i+\kappa_i)}{MN}}(\beta_i(q)-1)/N, & 0 \le l < l_i, \\ e^{-j2\pi\frac{l(k_i+\kappa_i)}{MN}}\beta_i(q)/N, & l_i \le l < M. \end{cases} \tag{3.20}$$

$\mathbf{\Delta}^{-[q-k_i]_N} \in \mathbb{C}^{MN\times MN}$ is an $MN \times MN$ block diagonal matrix by shifting the block in matrix Δ by $-[q-k_i]_N$, where $\mathbf{\Delta} = diag[\mathbf{\Delta}_0, \cdots, \mathbf{\Delta}_{N-1}]$, $\mathbf{\Delta}_n = \mathbf{\Psi}\boldsymbol{I}_M(l_i)$, where $\mathbf{\Psi} \in \mathbb{C}^{M\times M}$ is a diagonal matrix with l-th diagonal element Ψ_{ll} expressed as

$$\Psi_{ll} = \begin{cases} e^{-j\frac{2\pi l}{N}}, & 0 \le l < l_i \\ 1, & l_i \le l < M. \end{cases} \tag{3.21}$$

When only integer Doppler exists (i.e., $\kappa_i = 0, \forall i$), we can rewrite (3.15) as [5]

$$y[k,l] = \sum_{i=1}^{P} h_i e^{j2\pi(\frac{l-l_i}{M})\frac{k_i}{N}} \alpha_i(k,l) x[[k-k_i]_N, [l-l_i]_M] + w[k,l], \tag{3.22}$$

where

$$\alpha_i(k,l) = \begin{cases} \frac{N-1}{N} e^{-j2\pi \frac{[k-k_i]_N}{N}}, & 0 \le l < l_i \\ 1, & l_i \le l < M. \end{cases} \tag{3.23}$$

Therefore, The $\mathbf{H}_{\text{rect}}$ can be simplified as

$$\mathbf{H}_{\text{rect}} = \sum_{i=1}^{P} \mathbf{I}_N(k_i) \otimes \left(\mathbf{\Lambda} \mathbf{I}_M(l_i) h_i e^{-j2\pi \frac{l_i k_i}{MN}} \right) \cdot \mathbf{\Delta}^{k_i}, \tag{3.24}$$

and

$$\mathbf{\Lambda}_{ll} = \begin{cases} e^{-j2\pi \frac{lk_i}{MN}} (N-1)/N, & 0 \le l < l_i \\ e^{-j2\pi \frac{lk_i}{MN}} N, & l_i \le l < M. \end{cases} \tag{3.25}$$

For convenience, we transfer the complex-valued model in (3.11) and (3.18) into the real-valued model and omit the subscript, following the same transformation in (2.2). Thus we have

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n} \tag{3.26}$$

where $\mathbf{x} = \left[\Re(\mathbf{x}_{\text{DD}})^T \ \Im(\mathbf{x}_{\text{DD}})^T \right]^T \in \mathbb{R}^{2NM}$, $\mathbf{y} = \left[\Re(\mathbf{y}_{\text{DD}})^T \ \Im(\mathbf{y}_{\text{DD}})^T \right]^T \in \mathbb{R}^{2NM}$, $\mathbf{n} = \left[\Re(\mathbf{w}_{\text{DD}})^T \ \Im(\mathbf{w}_{\text{DD}})^T \right]^T \in \mathbb{R}^{2NM}$, $\mathbf{H} = \begin{bmatrix} \Re(\mathbf{H}_{\text{ideal}}) & -\Im(\mathbf{H}_{\text{ideal}}) \\ \Im(\mathbf{H}_{\text{ideal}}) & \Re(\mathbf{H}_{\text{ideal}}) \end{bmatrix} \in \mathbb{R}^{2NM \times 2NM}$, or $\mathbf{H} = \begin{bmatrix} \Re(\mathbf{H}_{\text{rect}}) & -\Im(\mathbf{H}_{\text{rect}}) \\ \Im(\mathbf{H}_{\text{rect}}) & \Re(\mathbf{H}_{\text{rect}}) \end{bmatrix} \in \mathbb{R}^{2NM \times 2NM}$ depending on different waveforms used, we assume $\mathbf{H}$ is perfectly known at the receiver, and we use the model (3.26) for the rest of the thesis.

3.4 OTFS Channel Condition Analysis

OTFS system's performance is highly related to the channel conditions, as symbol detection in an ill-conditioned channel is challenging [20], [35]. Typically, the

channel condition number, defined as the ratio of maximum to minimum singular values of the channel matrix, is used to evaluate the channel condition. A large condition number implies an ill-conditioned channel. Therefore, a small channel condition number is often desired, as it impacts the robustness of matrix inversion and decomposition in the symbol detection process [35].

Here, we evaluate the channel condition number of the OTFS channel matrix for different waveforms and Doppler cases. We generate 5000 channel realizations for the OTFS setting of $N = M = 8, l_{max} = 7, k_{max} = 2, P = 8, N_i = 1$. Fig. 3.2 shows the cumulative distribution function (CDF) for four different cases: 1) the ideal waveform with integer Doppler, 2) the ideal waveform with fractional Doppler, 3) the rectangular waveform with integer Doppler, 4) the rectangular waveform with fractional Doppler. Fig. 3.2 demonstrates that as the interference in the OTFS channel increases, so does the channel condition number, resulting in an ill-conditioned OTFS channel matrix. Specifically, fractional Doppler will result in a large condition number compared to integer one. The condition number in the rectangular waveform is larger than that in the ideal one, as ISI and ICI are considered additional interference in the former case.

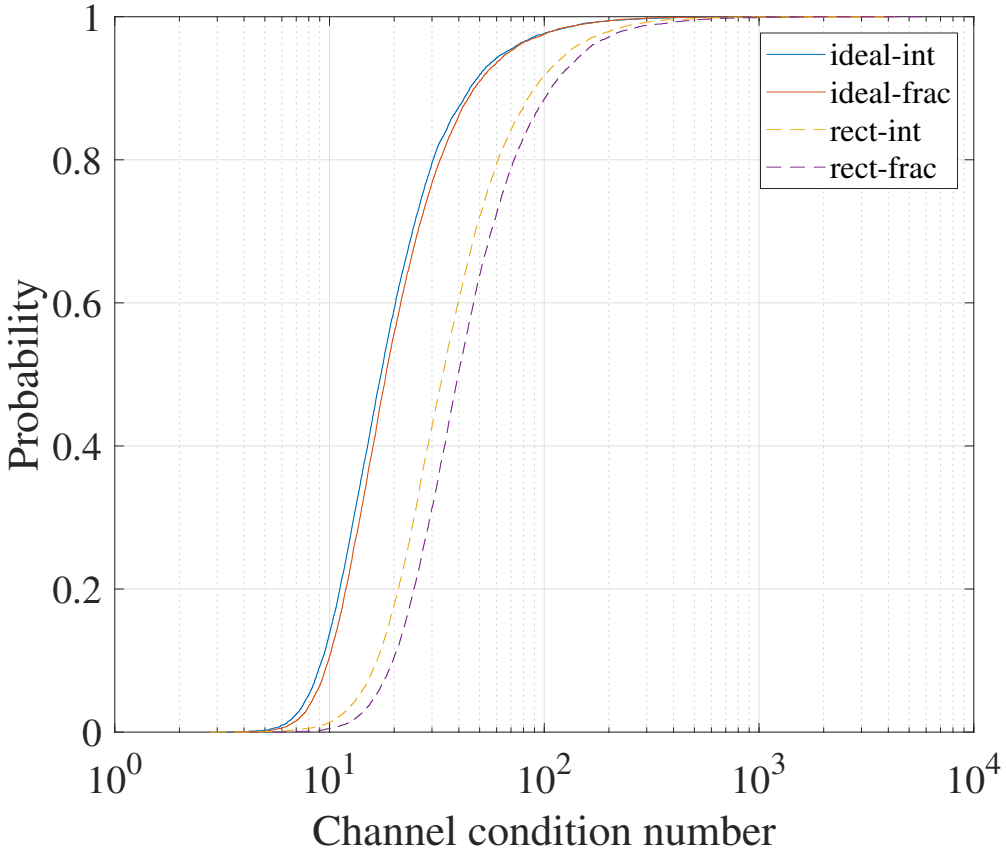


Figure 3.2: CDF of channel condition number

Chapter 4

D-DIP-BPIC

In Chapter 3, we analyzed OTFS systems with rectangular and ideal waveform, and their corresponding channel conditions. The channel in the rectangular waveform is typically ill-conditioned, which will influence the symbol detection quality on the receiver side. To achieve excellent SER performance under ill-conditioned OTFS channels (i.e., under rectangular waveform), we propose an untrained DNN based OTFS detector. In this chapter, we explain the D-DIP-BPIC OTFS detector, consisting of the D-DIP denoiser and BPIC, as shown in Fig.4.1. First, we present the decoder DNN architecture of the D-DIP denoiser and analyze the stopping criteria and its convergence. We then investigate the computational complexity and SER performance of the D-DIP-BPIC OTFS detector.

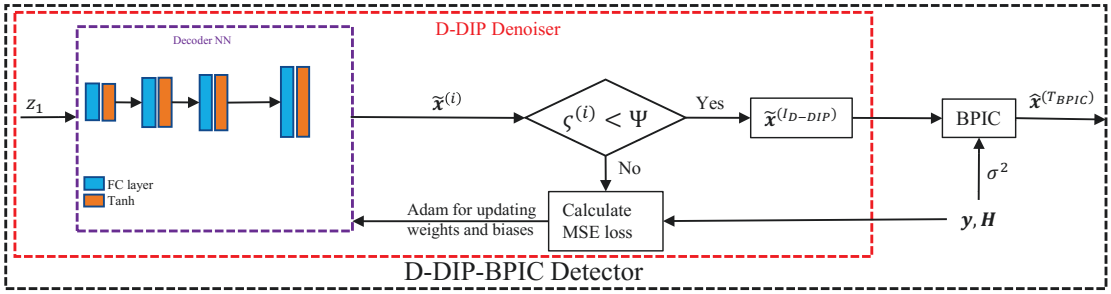


Figure 4.1: D-DIP-BPIC model

4.1 D-DIP Denoiser

4.1.1 Decoder DNN Architecture

The NN architecture used in D-DIP is shown in the decoder NN block in Fig. 4.1, consisting of $L = 4$ fully connected (FC) layers, and the Tanh activation function is used after each FC layer. The NN output is defined as

$$\tilde{\mathbf{x}}^{(i)} = \alpha f_L^{(i)}(f_{L-1}^{(i)}(\cdots(f_1^{(i)}(\mathbf{z}_1^{(i)})))), \tag{4.1}$$

where $\tilde{\mathbf{x}}^{(i)}$ is the NN output in iteration i. The NN first layer input $\mathbf{z}_1^{(i)} \in \mathbb{R}^{4\times 1}$ is a random vector drawn from a normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{1})$ and it is fixed during the iterative process. α is a constant used to control the output range of the NN. We use $f_l^{(i)}(\mathbf{z}_l^{(i)})$ to represent the NN output of layer l at iteration i, where $\mathbf{z}_l^{(i)}$ is the input of layer l, which is also the output of layer $l-1$ in iteration i (i.e., $\mathbf{z}_l^{(i)} = f_{l-1}^{(i)}(\mathbf{z}_{l-1}^{(i)})$). The layer l output is defined as

$$f_l^{(i)}(\mathbf{z}_l^{(i)}) = \text{Tanh}(\mathbf{W}_l^{(i)}\mathbf{z}_l^{(i)} + \mathbf{b}_l^{(i)}) \tag{4.2}$$

where $\mathbf{W}_l^{(i)} \in \mathbb{R}^{c_l \times c_{l-1}}$ represents the weight matrix between layer l and $l-1$ at iteration i. $\mathbf{b}_l^{(i)} \in \mathbb{R}^{c_l \times 1}$ is the bias vector in layer l at iteration i. Each entry

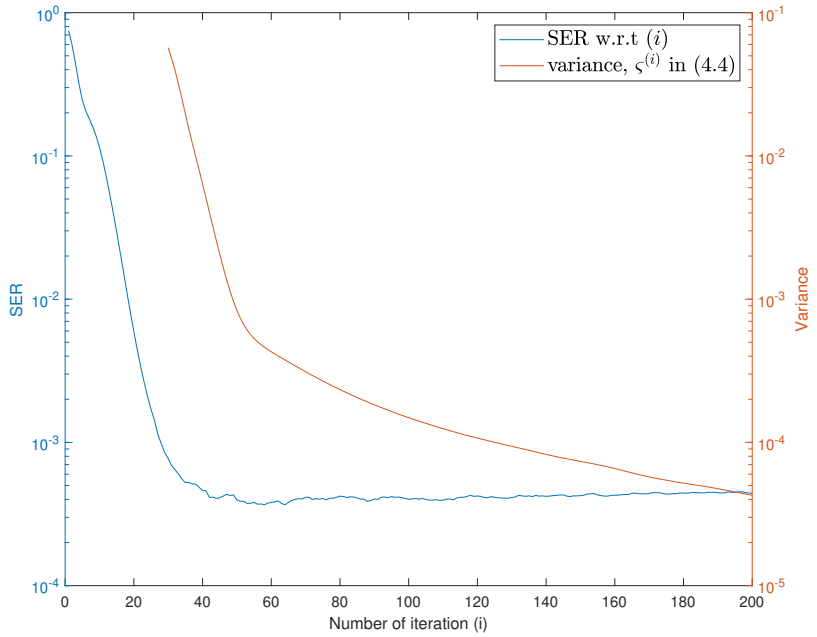


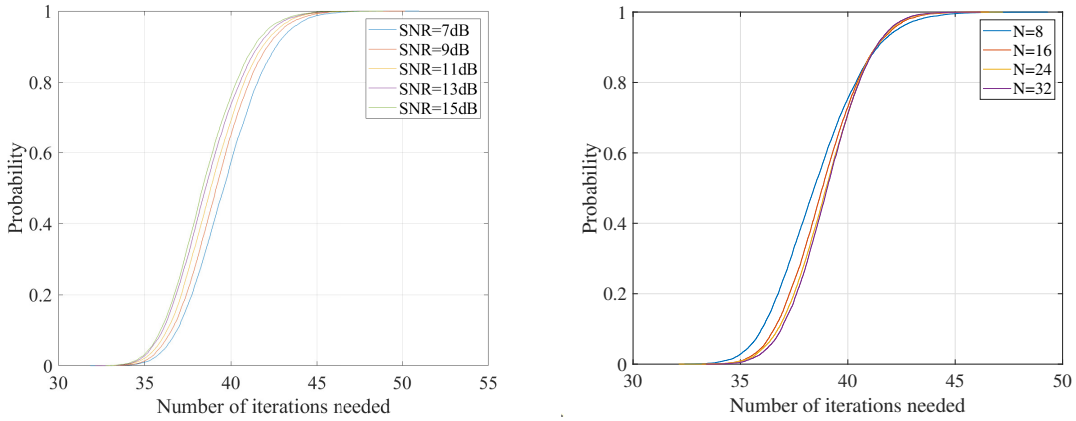
Figure 4.2: D-DIP iterative process

of $\mathbf{W}_l^{(0)}$ and $\mathbf{b}_l^{(0)}$ are initialized following a uniform distribution with a range of $(\frac{-1}{\sqrt{c_l}}, \frac{1}{\sqrt{c_l}})$ [36], where c_l is the output size of layer l. $c_1 = 8, c_2 = 16, c_3 = 32, c_4 = 2NM$.

We then use the MSE loss function to calculate the difference between the denoised signal $\tilde{\mathbf{x}}^{(i)}$ obtained from (4.1) and the received signal $\mathbf{y}$ shown as

$$\mathcal{L}^{(i)} = \frac{1}{2NM} \|\mathbf{H}\tilde{\mathbf{x}}^{(i)} - \mathbf{y}\|^2. \tag{4.3}$$

Adam optimizer [31] is then used to calculate the gradient and update weights $\mathbf{W}_l^{(i)}$ and biases $\mathbf{b}_l^{(i)}$ by using the calculated error in (4.3).



(a) CDF for different SNR (b) CDF for different OTFS size

Figure 4.3: CDF of the iterations needed for D-DIP to satisfy stopping criteria

4.1.2 Stopping Criteria and Convergence Analysis

As discussed in Chapter 2, the DIP output will eventually overfit to noise without early stopping. Here, we use the stopping criteria in [37] to control the iterative process of D-DIP by using the computed variance of the NN output in (4.1). This is given as

$$\varsigma^{(i)} = \frac{1}{D}\sum_{j=i-D}^{i} \|\tilde{\mathbf{x}}^{(j)} - \frac{1}{D}\sum_{j'=i-D}^{i} \tilde{\mathbf{x}}^{(j')}\|^2, i \geq D, \tag{4.4}$$

where $\varsigma^{(i)}$ is the variance value at iteration i. D is the window size that determines how many outputs are used for calculating the variance. When $i < D$, the variance calculation is inactive. We control the iterative process by comparing $\varsigma^{(i)}$ with a threshold Ψ. If $\varsigma^{(i)} < \Psi$, the iterative process of D-DIP will stop. When the stopping criteria is satisfied, the D-DIP output $\tilde{\mathbf{x}}^{(I_{\text{D-DIP}})}$ will be used as initial symbol estimates in BPIC (i.e., $\hat{\mathbf{x}}^{(0)} = \tilde{\mathbf{x}}^{(I_{\text{D-DIP}})}$, in (2.12)), where $I_{\text{D-DIP}}$ denotes the last D-DIP iteration. Otherwise, the iterative process of D-DIP is repeated, as shown in the D-DIP denoiser block in Fig. 4.1.

Fig. 4.2 gives an example of the SER performance of iterative D-DIP denoising and the corresponding variance value $\varsigma^{(i)}$ with an OTFS configuration $N = M = 8, l_{max} = 7, k_{max} = 2, P = 8$, ideal waveform, and integer Doppler. 4-QAM modulation is considered, and the signal to noise ratio (SNR) is set to $15dB$. The label 'SER w.r.t (i)' shows the SER computed based on the output of (4.1) for a different iteration number (i). The SER curve decreases as the number of iterations increases. However, the SER increases after it reaches the valley. This is referred to as the DIP overfitting problem, as discussed in Chapter 2. The corresponding variance curve is also shown in Fig. 4.2, starting from iteration 30 (i.e., $D = 30$). We control the iterative process by comparing $\varsigma^{(i)}$ with a threshold Ψ. If $\varsigma^{(i)} < \Psi$, the iterative process of D-DIP will stop. For example, if we set $\Psi = 0.001$, the D-DIP will stop at around 50 iterations, which shows almost the lowest SER value. Moreover, Fig. 4.3a shows the CDF of $I_{\mathrm{D-DIP}}$ (i.e., the number of D-DIP iterations needed to satisfy the stopping scheme in (4.4) for $D = 30, \Psi = 0.001$). The figure shows that the number of iterations required for D-DIP to converge, $I_{\mathrm{D-DIP}}$, is not sensitive to the noise level (i.e., SNR values). Additionally, Fig. 4.3b shows the CDF of $I_{\mathrm{D-DIP}}$ for different OTFS sizes (i.e., $N = 8, 16, 24, 32$). The figure demonstrates that $I_{\mathrm{D-DIP}}$ is not sensitive to the OTFS frame size. Therefore, the stopping criteria can be generalized to various scenarios (e.g., different SNR values and OTFS frame size), and the computational complexity of D-DIP is independent of the iteration number $I_{\mathrm{D-DIP}}$.

4.2 Complexity Analysis

In this section, we analyze the computational complexity of the proposed D-DIP-BPIC detector. As for the complexity of D-DIP, the computational complexity of fully-connected layers is matrix-vector multiplications with a cost of $\mathcal{O}(M^2N^2I_{\mathrm{D-DIP}})$, where $I_{\mathrm{D-DIP}}$ denotes the number of iterations needed for D-

OTFS Detector	Computational Complexity order
MP[5]	$\mathcal{O}(MNPOT_{\mathrm{MP}})$
VB[7]	$\mathcal{O}(MNPOT_{\mathrm{VB}})$
MMSE[4]	$\mathcal{O}(M^3N^3)$
D-DIP	$\mathcal{O}(M^2N^2I_{\mathrm{D-DIP}})$
AMP[6]	$\mathcal{O}(M^2N^2I_{\mathrm{AMP}})$
UAMP[6]	$\mathcal{O}(M^3N^3 + M^2N^2T_{\mathrm{UAMP}})$
MMSE-BPIC[8]	$\mathcal{O}(M^3N^3 + M^2N^2T_{\mathrm{BPIC}})$
EP[9]	$\mathcal{O}(M^3N^3T_{\mathrm{EP}})$
BPICNet[13]	$\mathcal{O}(M^3N^3 + (MN + M^2N^2)T_{\mathrm{BPICNet}})$
D-DIP-BPIC	$\mathcal{O}(M^2N^2I_{\mathrm{D-DIP}} + M^2N^2T_{\mathrm{BPIC}})$

Table 4.1: Computational complexity comparison

DIP. The computational complexity for different detection algorithms is shown in Table 4.1, where $T_{MP}, T_{VB}, T_{AMP}, T_{UAMP}, T_{BPIC}, T_{EP}, T_{BPICNet}$ represent the iterations needed for the MP, VB, AMP, UAMP, BPIC, EP, and BPICNet detectors, respectively. N, M, P, and O represent the number of subcarriers, timeslots, paths, and modulation order, respectively. Our proposed D-DIP-BPIC detector has a higher complexity than those low complex ones (e.g., MP, VB, and AMP). It has a lower complexity order compared to the state-of-the-art high complexity OTFS detectors (e.g., MMSE-BPIC, UAMP, EP, and BPICNet). On the other hand, the complexity of high computational complexity order detectors increases significantly as the OTFS size increases compared to the D-DIP-BPIC.

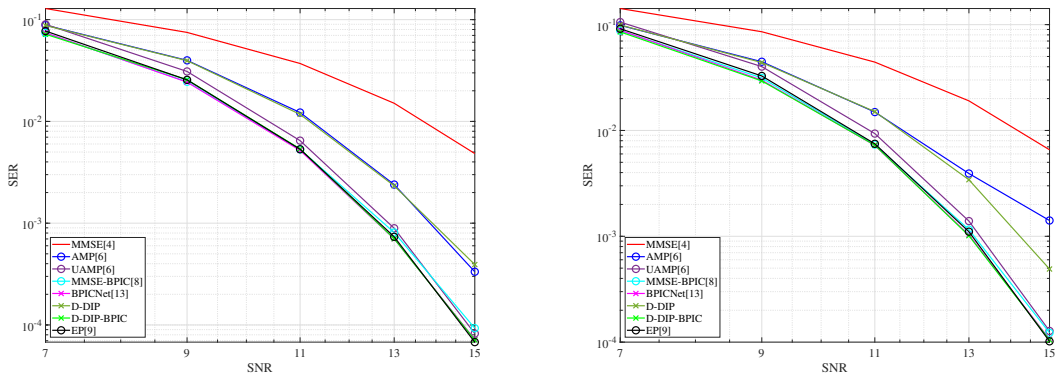
4.3 Simulation Results

In this section, we evaluate the performance of our proposed detector by comparing its SER performance with those in MMSE[4], AMP [6], UAMP [6], MMSE-BPIC [8], BPICNet [13], and EP [9]. For the simulations, we set $N = M = 8, P = 8, l_{max} = 7, k_{max} = 2$, $\Delta f = 15$kHz. The carrier frequency is set to $f_c = 10$GHz.

We set $N_i = 1$ for the fractional Doppler case. Note that the OTFS settings we use in this thesis differ from those in [C1], as we want to unify them for the entire thesis. The 4-QAM modulation is employed for the simulations, and we set $\alpha = 1/\sqrt{2}$, which corresponds to the normalized power of constellations to normalize the NN output. We use the Adam optimizer with a learning rate of 0.01 to optimize the weights and biases in the NN. The stopping criteria parameter for (4.4), D, is set to 30, and the threshold Ψ is set to 0.001. The number of iterations for the AMP, UAMP, BPIC, EP, and BPICNet is set to 10 to ensure convergence. For the training setting of BPICNet, we use the same configuration as above. 500 epochs are used during the training process. In each epoch, 40 batches of 256 samples were generated. The values of SNR are uniformly distributed in a certain range. Different waveforms (i.e., ideal and rectangular) and Doppler cases (i.e., integer or fractional Doppler) are trained for BPICNet separately.

Fig. 4.4 shows the SER performance of different OTFS detectors for the ideal waveform. In Fig. 4.4a, DIP performs the same as AMP. However, the latter suffers from performance degradation when fractional Doppler is considered, as shown in Fig. 4.4b. D-DIP-BPIC achieves a close to EP and BPICNet SER performance, slightly outperforming MMSE-BPIC and UAMP for both cases in the ideal waveform.

Fig. 4.5 shows the SER performance of different OTFS detectors for the rectangular waveform where all the detectors' SER performance degrades compared to the SER performance in the ideal waveform, as high interference exists (i.e., ill-conditioned channel). However, the outperformance of D-DIP-BPIC over MMSE-BPIC and UAMP is widened to around 0.5 dB compared to the ideal waveform, and so does for EP and BPICNet. The AMP outperforms D-DIP under the integer case as it has the ability to cancel the ICI and ISI. However, this outperformance disappears in the fractional Doppler cases as IDI occurs.

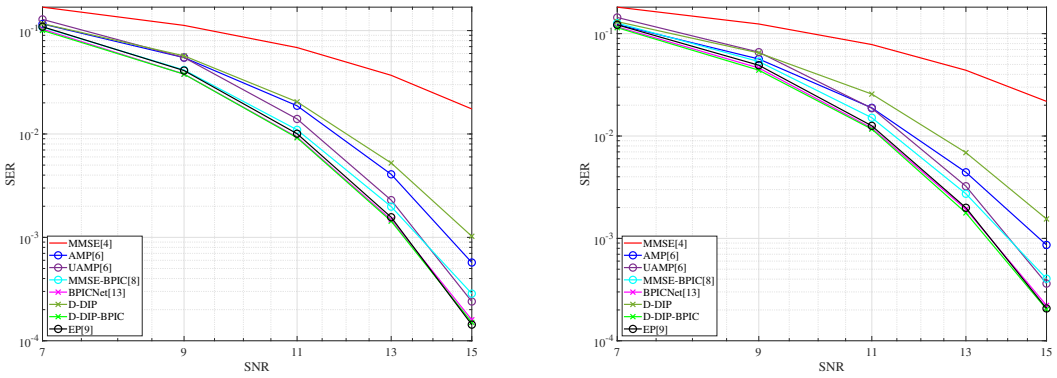


(a) Integer Doppler (b) Fractional Doppler

Figure 4.4: SER performance of the ideal waveform

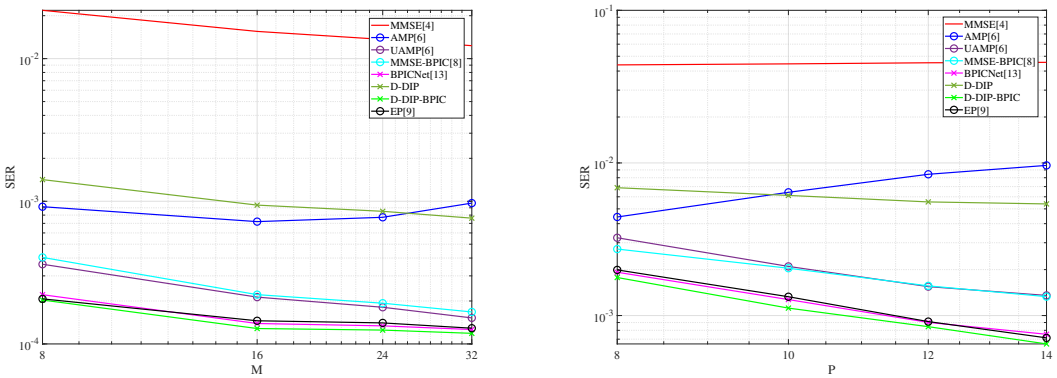
Fig. 4.6a evaluates the scalability of our proposed D-DIP-BPIC detector. As we increase the number of time slots $M = 8, 16, 24, 32$ and fix the number of sub-carriers $N = 8$, D-DIP-BPIC achieves the best SER performance among all those detectors. Fig. 4.6b shows that when the number of paths (e.g., mobile reflectors) increases, the proposed D-DIP-BPIC detector achieves the best SER performance compared to others. While AMP suffer from performance degradation due to the ill-conditioned channel, as discussed in Chapter 2.2.1.

In conclusion, the proposed D-DIP-BPIC offers close to EP and BPICNet SER performance with much lower complexity under various OTFS configurations. D-DIP-BPIC is shown to be robust to the rectangular waveform. This is beneficial, as the rectangular waveform is more practical in real environments. In contrast, UAMP and MMSE-BPIC suffer from performance degradation under this waveform compared to the ideal one. Additionally, D-DIP-BPIC is robust to the OTFS frame size and scattering environment (i.e., large number mobile reflectors exist).



(a) Integer Doppler (b) Fractional Doppler

Figure 4.5: SER performance of the rectangular waveform



(a) Varying OTFS frame size (b) Varying number of paths

Figure 4.6: SER performance of the rectangular waveform with fractional Doppler

Chapter 5

GDIP-BPIC

In the previous chapter, the proposed D-DIP-BPIC detector demonstrated excellent SER performance with low computational complexity order under ill-conditioned OTFS channels discussed in Chapter 3. However, this performance is at the cost of iterations needed for D-DIP to denoise the received signal. Thus, reducing the number of iterations for the denoising process is preferable. In this chapter, we propose to embed the graph concept into the D-DIP denoiser design. Specifically, we first explain the graph representation of wireless interference. We then present how to embed the graph representation into the D-DIP denoiser neural network layers. In the simulations, we consider 16-QAM and 64-QAM to show the robustness of the proposed detector under high modulation order. The reason is that most existing detectors are designed for low modulation orders, and they perform poorly under high ones [38]. We then investigate the impact of channel estimation error on the proposed OTFS detector. Finally, we compare the SER performance of the proposed detector in OTFS with it in the OFDM system.

5.1 GDIP Denoiser

5.1.1 Graph Representation of Wireless Interference

The graph representation of wireless interference is shown as a directed graph in Fig. 5.1, where nodes $x_1, \cdots, x_{2NM}$ represent the transmitted symbols seen at the receiver, and $\mathbf{x} = [x_1, \cdots, x_{2NM}]$ in (3.26). We refer to $x_1, \cdots, x_{2NM}$ as the features of the transmitted symbols and $e_{i,j}$ as an edge representing the interference between x_i and x_j. $e_{i,j} \neq e_{j,i}$ as the interference from x_i to x_j is different from the interference from x_j to x_i. The value of edges is defined as $e_{i,j} = \frac{\mathbf{h}_i^T \mathbf{h}_j}{\|\mathbf{h}_j\|^2}$, where $h_{i,j}$, $\mathbf{h}_j$ represent the (i,j)-th entry and j-th column of channel matrix $\mathbf{H}$, respectively. Also, each node has a self-loop when updating its feature, as shown in Fig. 5.1. Thus, the adjacency matrix of the graph is expressed as $\mathbf{F} \in \mathbb{R}^{2NM \times 2NM}$, where the (i,j)-th entry of $\mathbf{F}$ is $e_{i,j}$.

5.1.2 Graph Embedding

The interference graph knowledge is embedded after the decoder NN in Fig. 5.2. We use the decoder NN (i.e., layer from 1 to L) to obtain the initial node features and the graph embedding process (i.e., layer $L+1$) to measure the interference between transmitted symbols. As a consequence, (4.2) can be expanded to

$$f_l^{(i)}(\mathbf{z}_l^{(i)}) = \begin{cases} \text{Tanh}(\mathbf{W}_l^{(i)} \mathbf{z}_l^{(i)} + \mathbf{b}_l^{(i)}) & l = 1, 2, 3, 4, \\ \text{Tanh}(\mathbf{W}_l^{(i)} (\mathbf{F} \mathbf{z}_l^{(i)}) + \mathbf{b}_l^{(i)})) & l = 5. \end{cases} \tag{5.1}$$

Note that the output size for each NN layer is $c_1, \cdots, c_5 = 8, 16, 32, 2NM, 2NM$, respectively. From (5.1), we can see that the wireless interference graph embedding process adds the 5-th layer in GDIP NN design, which makes GDIP learn the signal using the NN weights (i.e., $\mathbf{W}_l^{(i)}$), biases (i.e., $\mathbf{b}_l^{(i)}$), and the neighboring

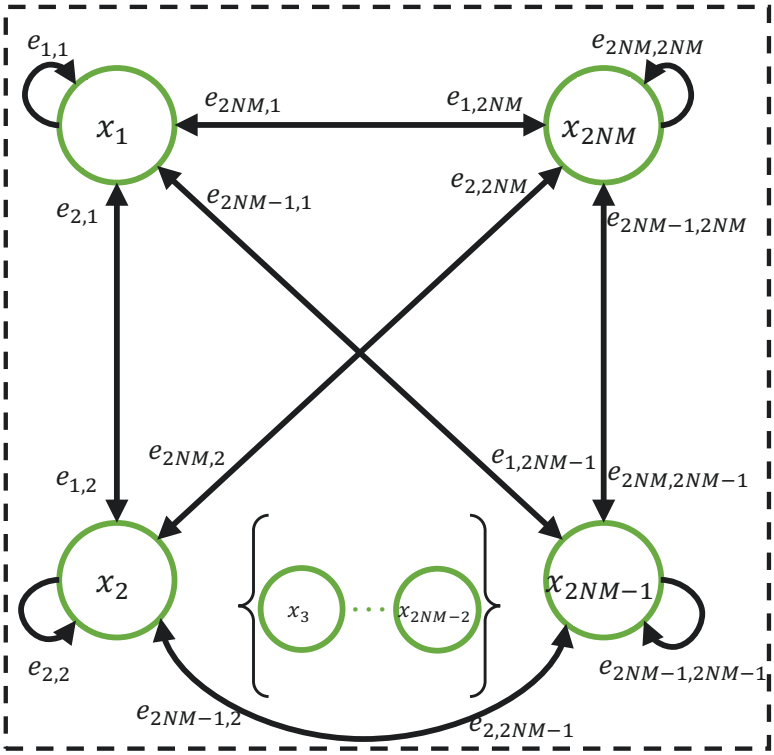


Figure 5.1: Directed wireless interference graph

information from the adjacency matrix (i.e., **F**). In contrast, D-DIP learns the signal by fitting only the weights and biases of the NN (i.e., layers 1 to 4), as discussed in Chapter 4.1.1. Therefore, GDIP will learn faster than D-DIP in signal denoising.

Here, the graph design differs from the GCG in [23]. Specifically, we use a directed graph to represent the wireless interference relationship, i.e., $e_{i,j} \neq e_{j,i}$ in the adjacency matrix. That is because OTFS has different wireless channel responses for different subcarriers and Doppler shifts. In [23], an undirected graph is used that implies $e_{i,j} = e_{j,i}$, which is not relevant to our OTFS scenarios. Secondly, we use a L layer decoder NN to get the initial node features for the graph in the last layer, followed by the adjacency matrix to update those features to yield the final symbol estimates (i.e., denoised signal). In contrast, the initial node features are

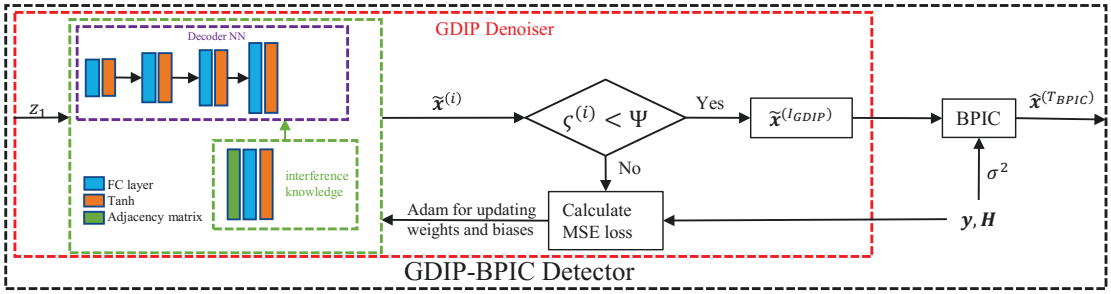


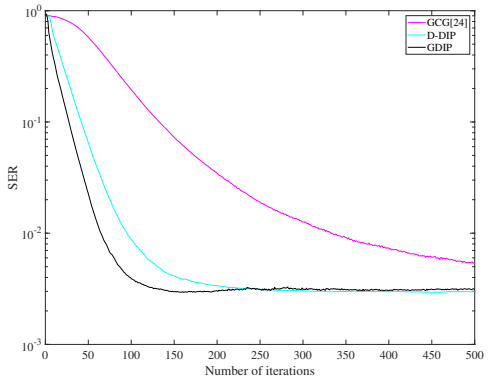
Figure 5.2: GDIP-BPIC model

set randomly in [23], which may result in a longer feature learning process.

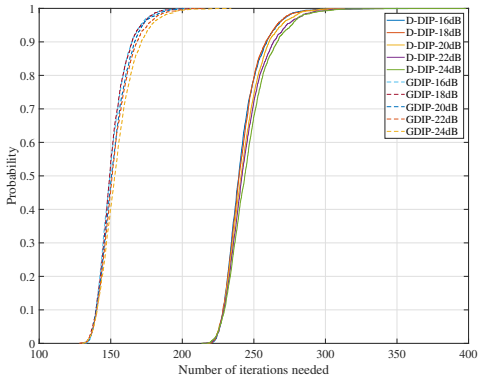
We follow the same way to combine GDIP with BPIC, as discussed in Chapter 4. We use the GDIP output $\tilde{\mathbf{x}}^{(I_{\text{GDIP}})}$ as the initial symbol estimates of BPIC, leading to the GDIP-BPIC detector.

5.2 Simulation Results

In this section, we first compare the proposed D-DIP and GDIP denoiser with state-of-the-art untrained graph neural network (i.e., GCG [23]). We then compare the proposed GDIP-BPIC detector with the state-of-the-art OTFS detectors, MMSE[4], AMP[6], UAMP[6], MMSE-BPIC [8], BPICNet[13] and EP [9]. Finally, we consider the impact of channel estimation error on the proposed detector and its performance in the OFDM system. Unless specified, we set $N = 8, M = 8, l_{max} = 7, k_{max} = 2, P = 8$,and use 16-QAM and rectangular waveform and fractional Doppler for all the simulations. We set $N_i = 1$ in the fractional Doppler case. For both D-DIP and GDIP, the learning rate is 0.01, and $\alpha = 3/\sqrt{10}$ for 16-QAM modulation. For the stopping criteria, we set the threshold $\Psi = 0.001$ and the window size $D = 120, 200$ for GDIP and D-DIP, respectively. The number of iterations for AMP and UAMP is set to 30 to ensure



(a) SER vs Iteration



(b) CDF

convergence, and they are set to 10 for BPIC, EP, and BPICNet. For the training of BPICNet, we follow the same settings as Chapter 4 with 16-QAM modulation instead.

We first compare the SER performance of GDIP, D-DIP, and GCG [23]. Fig. **??** demonstrates that the proposed GDIP denoiser needs significantly fewer iterations to denoise the received signal than D-DIP.... The reason is that the adjacency matrix $\mathbf{F}$ in (5.1) incorporates prior information (i.e., graph interference knowledge) in the signal denoising process [23]. Also, it outperforms GCG in performing OTFS symbol detection. Fig. 5.3b shows the cumulative density function (CDF) of the number of iterations needed for D-DIP and GDIP to satisfy the stopping criteria in (4.4). It shows that the stopping criteria is not sensitive to SNR. Thus, we can apply the same parameters in the stopping criteria for various SNR values. Also, it shows that the proposed GDIP denoiser needs significantly fewer iterations to denoise the received signal than D-DIP, which results in a lower computational complexity for the GDIP-BPIC detector than D-DIP-BPIC one. D-DIP-BPIC is less complex than those high complexity detectors (e.g., EP, MMSE-BPIC, UAMP, BPICNet), as discussed in Chapter 4.2. Therefore, the proposed GDIP-BPIC is the most computationally advantageous detector com-

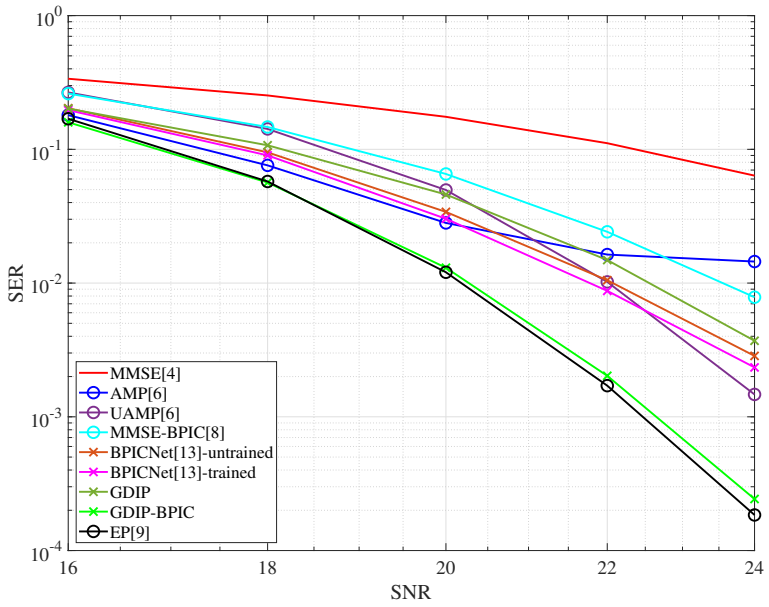
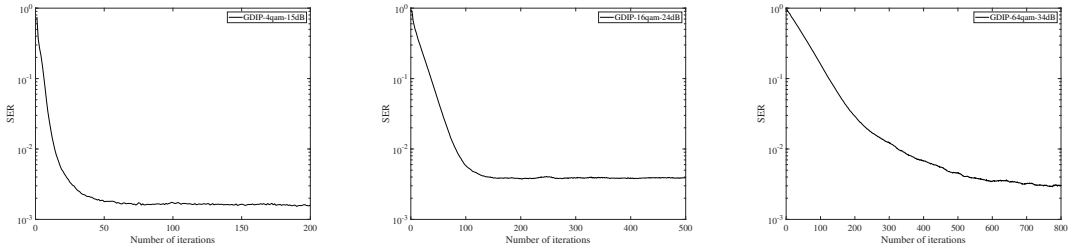


Figure 5.3: SER performance under untrained environment

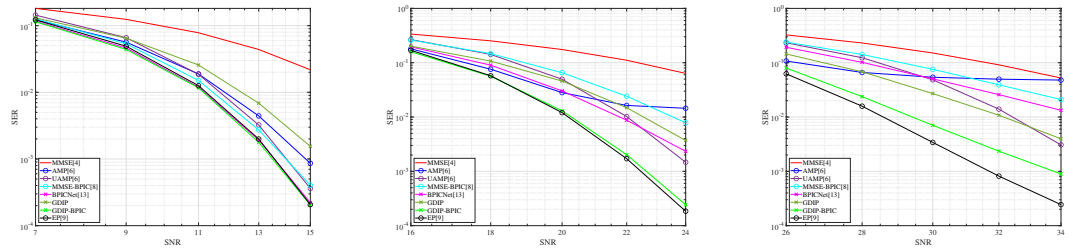
pared to state-of-the-art OTFS ones. Here, the low complexity detectors, e.g., MP, AMP, and VB, are not considered, as their SER performance is much worse than our proposed detectors.

The SER performance comparison of the proposed GDIP and GDIP-BPIC to state-of-the-art OTFS detectors is shown in Fig. 5.3, where "BPICNet-trained" is trained under $l_{max} = 7, k_{max} = 2, P = 8$, and "BPICNet-untrained" is trained under $l_{max} = 4, k_{max} = 2, P = 4$. The Fig. 5.3 shows that GDIP outperforms MMSE-BPIC and AMP by 1dB, and after being combined with BPIC, the proposed GDIP-BPIC outperforms UAMP and BPICNet-trained/BPICNet-untrained by around 2dB. Also, its SER performance is close to EP, which has a much higher computational complexity. Moreover, BPICNet-trained outperforms BPICNet-untrained by around 0.5dB. This is referred to as the training dataset fidelity issue, which is the performance degradation when the training environment



(a) 4-QAM SNR=15dB (b) 16-QAM SNR=24dB (c) 64-QAM SNR=34dB

Figure 5.4: SER performance of GDIP for different modulation order



(a) 4-QAM (b) 16-QAM (c) 64-QAM

Figure 5.5: SER performance comparison of state-of-the-art OTFS detectors for different modulation order

and testing environment are mismatched. This often happens in real wireless environments, as the path delay, number of mobile reflectors, and their mobility are unknown beforehand.

Moreover, we investigate the impact of modulation order on the proposed GDIP denoiser. Fig. 5.2 shows the SER performance of the GDIP denoiser for different modulation orders with different SNR values. Here, we make a fair comparison at the same SER level by considering different SNR values for each modulation order. For 4-QAM 15dB, Fig. 5.4a shows that GDIP needs around 60 iterations to converge. While for 16-QAM 24dB and 64-QAM 34dB, GDIP requires approximately 150 and 800 iterations to denoise the received signal, as shown in Fig. 5.4b and Fig. 5.4c, respectively. From those results, we can see that as modu-

lation order increases, the number of iterations needed for the GDIP denoiser to denoise the received signal increases. The reason is that the NN needs a longer learning process to learn complex signals. We then compare the SER performance of GDIP-BPIC with other detectors for three different modulation orders (i.e., 4-QAM, 16-QAM, and 64-QAM). T_{AMP} and T_{UAMP} are set to 10, 30, and 50 for 4-QAM, 16-QAM and 64-QAM, respectively. $T_{\text{BPIC}} = T_{\text{EP}} = 10$, and BPICNet is trained individually for all those modulation orders. For GDIP, the windows size $D = 30, 120, 700$ for 4-QAM, 16-QAM, and 64-QAM, respectively. Fig. 5.5 shows the SER performance comparison of state-of-the-art OTFS detectors under different modulation orders. In Fig. 5.5a, we can see GDIP-BPIC has a comparable SER performance with BPICNet and EP and outperforms UAMP and MMSE-BPIC by around 0.5dB for 4-QAM. As for 16-QAM in Fig. 5.5b, GDIP-BPIC outperforms UAMP, MMSE-BPIC, and BPICNet over 2dB and remains comparable to EP. However, this outperformance is widened to 3dB for 64-QAM, as shown in Fig. 5.4c. Thus, we can see that GDIP-BPIC and EP are robust to modulation orders compared to UAMP, MMSE-BPIC, and BPICNet. However, EP has significantly higher computational complexity than ours.

The impact of channel estimation error is investigated in Fig. 5.6. Here, we consider a channel estimation error model in [5] as

$$\hat{h}_i = h_i + n_e, \tag{5.2}$$

where $n_e \sim \mathcal{CN}(0, \sigma_e^2)$. Fig. 5.6 demonstrates that the gap between GDIP-BPIC and EP decreases as the error variance σ_e^2 increases. $\sigma_e^2 = 0$ represents perfect channel state information. Therefore, the proposed GDIP-BPIC OTFS detector is more robust to channel estimation error than EP. However, EP has significantly higher computational complexity than ours.

Finally, we compare the proposed GDIP-BPIC's performance under OFDM and

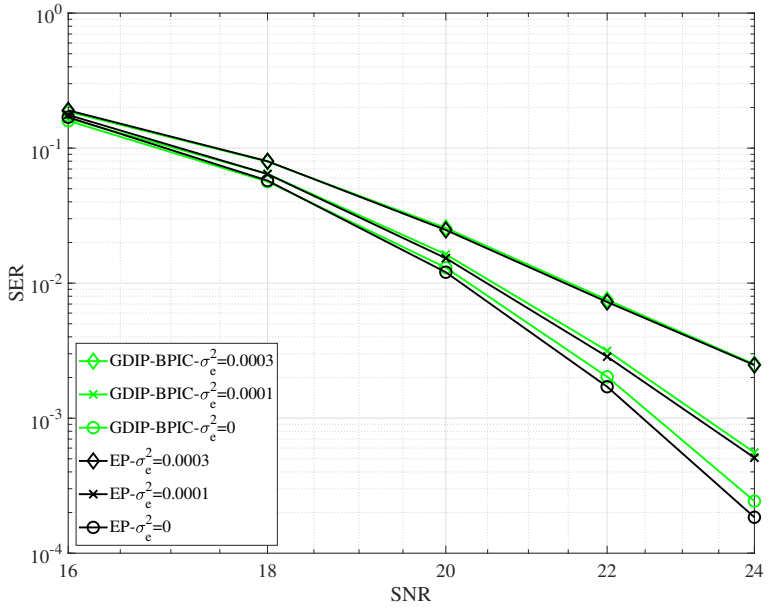


Figure 5.6: Channel estimation error

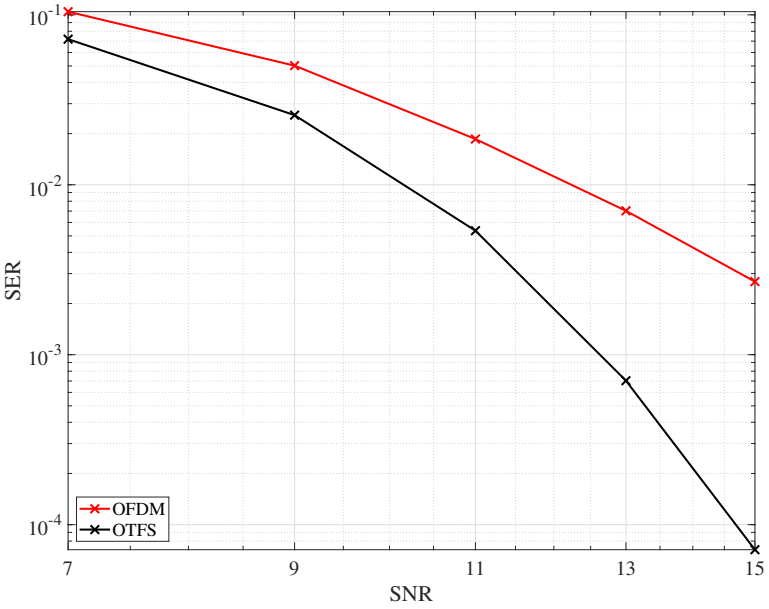


Figure 5.7: OTFS vs OFDM

OTFS systems. Here, we use the OFDM system model in [5], $\mathbf{y} = \mathbf{H}_{\text{ofdm}}\mathbf{x} + \mathbf{z}$ where, $\mathbf{x} \in \mathcal{C}^{N\times 1}\, \mathbf{y} \in \mathcal{C}^{N\times 1}, \mathbf{z} \in \mathcal{C}^{N\times 1}$ is the transmitted, received OFDM symbol and noise. $\mathbf{H}_{\text{ofdm}} = \mathbf{W}\mathbf{H}_t\mathbf{W}^H$, and $\mathbf{W}$ is N-point FFT matrix, $\mathbf{H}_t$ is the time domain channel matrix, where the (p, q)-th entry is expressed as,

$$H_t[p,q] = \sum_{i=1}^{P} h_i \delta\left[\left[p - q - \frac{\tau_i N}{T}\right]_N\right] e^{j\frac{2\pi(q-1)v_i}{N}}, p, q = 1, \cdots, N \tag{5.3}$$

Fig. 5.7 shows the SER performance of the proposed GDIP-BPIC detector under 4-QAM with ideal waveform and integer Doppler. We can see that OTFS outperforms OFDM by over 3dB due to the effective tracking of Doppler shift in the DD domain by using OTFS.

In conclusion, we use graph embedding of wireless interference to reduce the number of iterations needed for D-DIP denoiser. The proposed GDIP-BPIC OTFS detector outperfoms MMSE-BPIC, UAMP and BPICNet for both 16 and 64-QAM. Training-based NN detector, e.g., BPICNet suffers from performance degradation in untrained environment. However, our proposed GDIP-BPIC is independent with traing datasets, thus, is beneficial for dynamic wireless environment.

Chapter 6

Conclusion

In this thesis, we proposed a denoiser design based on untrained DNN to iteratively denoise the received signal. We combined the denoiser with BPIC to perform iterative symbol detection in OTFS systems. This combination shows excellent SER performance and low complexity compared to state-of-the-art OTFS detectors.

6.1 Summary of the thesis

In Chapter 2, we investigated several types of detectors, including linear, iterative, and training-based DNN detectors. We then showed the fundamental concept of DIP for the image denoising process. Additionally, we showed how graph representation can be embedded into the DIP neural network layer design.

In Chapter 3, we developed a SISO OTFS model. We analyzed the input-output relationship in different waveforms and Doppler and the corresponding interference. We then used the condition number to evaluate the OTFS channel condition under different waveforms and Doppler. These analyses helped us in the OTFS symbol detector design and the analysis of state-of-the-art OTFS detectors.

In Chapter 4, we proposed a D-DIP denoiser to replace the high computational complex MMSE one within the BPIC. The D-DIP denoiser combines the concept of DIP, decoder-only DNN, and stopping criteria to iteratively denoise the received signal. The simulation results showed that the proposed D-DIP-BPIC detector offers comparable SER performance to EP and BPICNet with much lower computational complexity under various OTFS configurations.

In Chapter 5, we proposed to embed the graph representation of wireless interference into the D-DIP denoiser to improve its denoising performance further. The simulation results showed that the GDIP denoiser needs significantly fewer iterations to denoise the received signal than the D-DIP one. After being combined with BPIC, the GDIP-BPIC outperforms others except EP under different OTFS configurations, while GDIP-BPIC has a much lower computational complexity.

6.2 Future work

In this thesis, we proposed D-DIP and GDIP denoisers to iteratively denoise the received signal to improve the BPIC's symbol estimation performance. However, the number of iterations needed for denoiser is a problem for high modulation order (e.g., 64-QAM). The performance gap between the proposed OTFS detectors and EP increases when the modulation order increases. Therefore, we will focus on improving the proposed detectors' performance under high modulation order in the future.

The proposed detector's performance under high IDI environments will be investigated in the future, as our current OTFS model has a small IDI due to the small OTFS frame size. Moreover, in the future, we will investigate the proposed detectors' performance under more practical channel models (e.g., the 3GPP 3D-urban macro (UMa) channel model). Furthermore, we will investigate the sparse

property of the OTFS channel matrix in the untrained neural network design in the future.

Bibliography

[1] P. Raviteja, K. T. Phan, and Y. Hong, "Embedded pilot-aided channel estimation for otfs in delay–doppler channels," *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 4906–4917, 2019.

[2] T. Jiang, H. H. Chen, H. C. Wu, and Y. Yi, "Channel modeling and inter-carrier interference analysis for V2V communication systems in frequency-dispersive channels," *Mob. Netw. Appl.*, vol. 15, no. 1, pp. 4–12, May 2010.

[3] R. Hadani, S. Rakib, M. Tsatsanis, A. Monk, A. J. Goldsmith, A. F. Molisch, and R. Calderbank, "Orthogonal time frequency space modulation," in *Proc. IEEE Wireless Commun. and Netw. Conf. (WCNC)*, USA, Mar. 2017, pp. 1–6.

[4] G. D. Surabhi and A. Chockalingam, "Low-complexity linear equalization for otfs modulation," *IEEE Commun. Lett.*, vol. 24, no. 2, pp. 330–334, 2020.

[5] P. Raviteja, K. T. Phan, Y. Hong, and E. Viterbo, "Interference cancellation and iterative detection for orthogonal time frequency space modulation," *IEEE Trans. Wirel. Commun.*, vol. 17, no. 10, pp. 6501–6515, Aug. 2018.

[6] Z. Yuan, F. Liu, W. Yuan, Q. Guo, Z. Wang, and J. Yuan, "Iterative detection for orthogonal time frequency space modulation with unitary approximate message passing," *IEEE Trans. Wireless Commun.*, vol. 21, no. 2, pp. 714–725, 2022.

[7] W. Yuan, Z. Wei, J. Yuan, and D. W. K. Ng, "A simple variational bayes detector for orthogonal time frequency space (otfs) modulation," *IEEE Trans. Veh. Technol.*, vol. 69, no. 7, pp. 7976–7980, 2020.

[8] X. Qu, A. Kosasih, W. Hardjawana, V. Onasis, and B. Vucetic, "Bayesian-based symbol detector for orthogonal time frequency space modulation systems," in *Proc. IEEE Int. Symp. Pers. Indoor Mob. Radio Commun. (PIMRC)*, Sep. 2021, pp. 1154–1159.

[9] H. Li, Y. Dong, C. Gong, Z. Zhang, X. Wang, and X. Dai, "Low complexity receiver via expectation propagation for otfs modulation," *IEEE Wireless Commun. Lett.*, vol. 25, no. 10, pp. 3180–3184, 2021.

[10] A. Naikoti and A. Chockalingam, "Low-complexity delay-doppler symbol dnn for otfs signal detection," in *Proc. IEEE Veh. Technol. Conf. (VTC2021-Spring)*, Sep. 2021, pp. 1–6.

[11] Y. K. Enku, B. Bai, F. Wan, C. U. Guyo, I. N. Tiba, C. Zhang, and S. Li, "Two-dimensional convolutional neural network-based signal detection for OTFS systems," *IEEE Wireless Commun. Lett.*, vol. 10, no. 11, pp. 2514–2518, 2021.

[12] X. Zhang, L. Xiao, S. Li, Q. Yuan, L. Xiang, and T. Jiang, "Gaussian amp aided model-driven learning for otfs system," *IEEE Commun. Lett.*, vol. 26, no. 12, pp. 2949–2953, 2022.

[13] A. Kosasih, X. Qu, W. Hardjawana, C. Yue, and B. Vucetic, "Bayesian neural network detector for an orthogonal time frequency space modulation," *IEEE Wireless Commun. Lett.*, vol. 11, no. 12, pp. 2570–2574, 2022.

[14] S. Li, C. Ding, L. Xiao, X. Zhang, G. Liu, and T. Jiang, "Expectation propagation aided model driven learning for otfs signal detection," *IEEE Trans. Veh. Technol.*, pp. 1–6, 2023.

[15] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for deep learning in NLP," in *Annu. Meet. Assoc. Comput. Linguist. (ACL)*, Jul. 2019, pp. 3645–3650.

[16] A. Linden, *Is synthetic data the future of AI?* gartner.com. `https://www.gartner.com/en/newsroom/press-releases/2022-06-22-is-synthetic-data-the-future-of-ai` (accessed Jan. 8, 2023).

[17] V. Lempitsky, A. Vedaldi, and D. Ulyanov, "Deep image prior," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, USA, Jun. 2018, pp. 9446–9454.

[18] M. Z. Darestani and R. Heckel, "Accelerated MRI with un-trained neural networks," *IEEE Trans. Comput. Imag.*, vol. 7, pp. 724–733, 2021.

[19] A. Scotti, N. N. Moghadam, D. Liu, K. Gafvert, and J. Huang, *Graph neural networks for massive mimo detection*, 2020. arXiv: `2007.05703 [eess.SP]`.

[20] A. Kosasih, V. Onasis, V. Miloslavskaya, W. Hardjawana, V. Andrean, and B. Vucetic, "Graph neural network aided mu-mimo detectors," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 9, pp. 2540–2555, 2022.

[21] H. He, A. Kosasih, X. Yu, J. Zhang, S. Song, W. Hardjawana, and K. B. Letaief, "Gnn-enhanced approximate message passing for massive/ultra-massive mimo detection," in *Proc. IEEE Wireless Commun. and Netw. Conf. (WCNC)*, 2023, pp. 1–6.

[22] X. Zhang, S. Zhang, L. Xiao, S. Li, and T. Jiang, "Graph neural network assisted efficient signal detection for otfs systems," *IEEE Communications Letters*, pp. 1–1, 2023. DOI: `10.1109/LCOMM.2023.3286800`.

[23] S. Rey, S. Segarra, R. Heckel, and A. G. Marques, "Untrained graph neural networks for denoising," *IEEE Trans. Signal Process.*, vol. 70, pp. 5708–5723, 2022.

[24] Z. Yuan, Q. Guo, and M. Luo, "Approximate message passing with unitary transformation for robust bilinear recovery," *IEEE Trans. Signal Process.*, vol. 69, pp. 617–630, 2021.

[25] A. Kosasih, V. Miloslavskaya, W. Hardjawana, C. She, C. K. Wen, and B. Vucetic, "A Bayesian receiver with improved complexity-reliability trade-off in massive MIMO systems," *IEEE Trans. Commun.*, vol. 69, no. 9, pp. 6251–6266, Sep. 2021.

[26] Q. Zou and H. Yang, *A concise tutorial on approximate message passing*, 2022. arXiv: `2201.07487 [cs.IT]`.

[27] S. Rangan, P. Schniter, A. K. Fletcher, and S. Sarkar, "On the convergence of approximate message passing with arbitrary matrices," *IEEE Trans. Inf. Theory*, vol. 65, no. 9, pp. 5339–5351, 2019.

[28] H. Qu, G. Liu, L. Zhang, S. Wen, and M. A. Imran, "Low-complexity symbol detection and interference cancellation for otfs system," *IEEE Trans. Commun.*, vol. 69, no. 3, pp. 1524–1537, 2021.

[29] Q. Shan, J. Wang, and D. Liu, "Deep image prior based pet reconstruction from partial data," *IEEE Trans. Radiat. Plasma Med. Sci.*, pp. 1–1, 2023.

[30] D. Van Veen, A. Jalal, M. Soltanolkotabi, E. Price, S. Vishwanath, and A. G. Dimakis, "Compressed sensing with deep image prior and learned regularization," *arXiv preprint arXiv:1806.06438*, 2018.

[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, USA, 2015, pp. 1–15.

[32] Z. Shi, P. Mettes, S. Maji, and C. G. Snoek, "On measuring and controlling the spectral bias of the deep image prior," *International Journal of Computer Vision*, vol. 130, no. 4, pp. 885–908, 2022.

[33] Q. Wang, Y. Liang, Z. Zhang, and P. Fan, "2d off-grid decomposition and sbl combination for otfs channel estimation," *IEEE Trans. Wirel. Commun.*, vol. 22, no. 5, pp. 3084–3098, 2023.

[34] F. Liu, Z. Yuan, Q. Guo, Z. Wang, and P. Sun, "Message passing-based structured sparse signal recovery for estimation of otfs channels with fractional doppler shifts," *IEEE Trans. Wirel. Commun.*, vol. 20, no. 12, pp. 7773–7785, 2021.

[35] J. Tong, Q. Guo, S. Tong, J. Xi, and Y. Yu, "Condition number-constrained matrix approximation with applications to signal estimation in communication systems," *IEEE Signal Process. Lett.*, vol. 21, no. 8, pp. 990–993, 2014.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Chile, Dec. 2015, pp. 1026–1034.

[37] H. Wang, T. Li, Z. Zhuang, T. Chen, H. Liang, and J. Sun, "Early stopping for deep image prior," *arXiv:2112.06074*, 2022.

[38] M. A. Albreem, A. H. Alhabbash, S. Shahabuddin, and M. Juntti, "Deep learning for massive mimo uplink detectors," *IEEE Commun. Surveys Tuts*, vol. 24, no. 1, pp. 741–766, 2022.