Advanced Linear and Deep Learning based Channel Estimation Techniques in Doubly Dispersive Environments

Abdul Karim Gizzini

Doctoral Thesis

Doctoral School : EM2PSI (n° 405)

Domain : STIC

Speciality : Telecommunications

Establishment : CY Cergy Paris University

Research Lab : ETIS - UMR 8051

Thesis presented and defended by :

Abdul Karim GIZZINI

Thesis Topic :

Advanced Linear and Deep Learning based Channel Estimation Techniques in Doubly Dispersive Environments

Thesis publicly defended on 03/12/2021

Jury members :

AYMERIC HISTACE	Professor, ENSEA, France	Thesis director
MARWA CHAFII	Associate Professor, NYU Abu Dhabi, UAE	Thesis supervisor
C. FAOUZI BADER	Professor, ISEP, France	Reviewer
MARCO DI RENZO	CNRS Research Director, CentraleSupélec-UPSaclay, France	Reviewer
MICHEL TERRÉ	Professor, CNAM, France	Examiner (President)
E. VERONICA BELMEGA	Associate Professor, ENSEA, France	Examiner
AHMAD NIMR	Research Group Leader, TUD, Germany	Invited member

Remerciements

I express my sincere appreciation to those who have contributed to this thesis and supported me in one way or the other during this amazing journey for without any of them, this research work would not have been possible.

First and foremost, I wish to place on records my sincere thanks to my supervisor Dr. Marwa CHAFII for providing me an opportunity to complete my PhD thesis. I appreciate her contributions of time and ideas to make my work productive and stimulating. Her valuable suggestions, comments and guidance encourage me to learn more day by day. Her deep insights helped me at various stages of my research. Big thanks once again go to her for without her this work would have never seen the light as it is today.

My sincere gratitude is reserved for Prof. Aymeric HISTACE for his great support. I really appreciate his willingness to help me and provide all the required resources in order to accomplish my work with success.

I would also like to take this opportunity to thank Dr. Ahmad NIMR for sharing his ideas and helpful suggestions for my work. The useful discussion and comments that he suggests widens my knowledge in various fields of the subject throughout the course of my thesis. Thank you for your valuable contributions for this thesis.

I am also indebted to my wife for her love and support that has motivated me to work harder and to strive towards my goal.

Finally, words cannot express the feelings I have for my mom for her unconditional support both in my study and my career as well. Mom you have been a constant source of strength and inspiration to me especially in the moment when there was no one to answer my queries. Your prayer for me was what sustained me this far and put me through from many difficulties that I am facing for all these years.

I also thank all the departmental staff for helping me during these years of my study both academically and officially.

Abstract

Wireless communications revolution plays a significant role in facilitating several mobile applications like unmanned aerial vehicles, high-speed railway, and vehicular communications. Particularly, the concept of connected vehicles brings a new level of connectivity to vehicles. Along with novel on board computing and sensing technologies, vehicular networks serve as a key enabler of intelligent transportation systems and smart cities. This new generation of networks have a profound impact on the society, making every day traveling safer, greener, and more efficient and comfortable. However, in vehicular environments, the propagation medium between the network nodes is highly time-varying leading to considerable reliability challenges. In fact, transmitted signals propagate through multiple paths, each with a different delay, attenuation, in addition to Doppler shift effect resulting from the motion of vehicles and the surrounding environment. Ensuring communication reliability by the means of accurate channel estimation in such environments is very important. Therefore, the accuracy of the channel estimation influences the system performance, since a precisely estimated channel response influences the follow-up equalization, demodulation, and decoding operations at the receiver. In literature, there exists an extensive work on conventional channel estimation for vehicular communications. However, these conventional estimators rely on many assumptions that limit their performance in highly dynamic time-varying channels. Moreover, linear conventional estimators are impractical solutions in real case scenarios as they rely on statistical models and require high implementation complexity. Although there exists simple linear estimation with affordable complexity, they lack robustness in highly dynamic environments. Therefore, investigating estimators with a good trade-off complexity vs. performance is a significant task. As a prevailing approach to AI, deep learning (DL) develops efficient methods to analyze data by finding patterns and learning underlying structures and represents an effective data driven approach to problems encountered in various scientific fields. The main reason behind integrating DL in wireless communications is to find solutions to communication problems where analytical solutions are intractable or highly complex. DL has a strong ability to overcome this challenge via low-complexity and robust solutions that improve the performance of wireless systems. Additionally, the GPU-based distributed processing enables the DL employment in real-time applications. As a result, DL can be leveraged to exploit the data generated in vehicular networks. In this context, this thesis aims to investigate

how to adapt such tools to account for the characteristics of high mobility vehicular networks. We show that integrating optimized DL architectures brings low-complexity solutions for vehicular channel estimation either by improving the performance compared to the simplified linear channel estimators, or by approaching the performance of complex robust model-based estimators with feasible implementation. Therefore, unlike conventional estimators, DL-based estimators provide a good trade-off between the computational complexity and the system performance. Moreover, the generalization ability gives robustness to the system when deployed in highly dynamic environments.

Résumé

La révolution des communications sans fil joue un rôle important dans la facilitation de plusieurs applications mobiles telles que les drones, les trains à grande vitesse et les communications entre véhicules. En particulier, le concept de véhicules connectés apporte une bonne connectivité aux véhicules. En plus des nouvelles technologies de calcul et de détection embarquées, les réseaux de véhicules servent de catalyseur clé aux systèmes de transport intelligents et aux villes intelligentes. Cette nouvelle génération de réseaux a un impact profond sur la société, rendant chaque jour les déplacements plus sûrs, plus écologiques, plus efficaces et plus confortables. Cependant, dans les environnements véhiculaires, le milieu de propagation entre les nœuds du réseau varie fortement dans le temps, ce qui pose des problèmes de fiabilité. En fait, les signaux transmis se propagent sur plusieurs chemins, chacun avec un retard et une atténuation différente, en plus de l'effet de décalage Doppler résultant du mouvement des véhicules et de l'environnement. Il est très important de garantir la fiabilité de la communication au moyen d'une estimation précise des canaux dans de tels environnements. Par conséquent, la précision de l'estimation de canal influence les performances du système, car une réponse de canal estimée avec précision influence les opérations d'égalisation, de démodulation et de décodage au niveau du récepteur. Dans la littérature, il existe plusieurs travaux sur les méthodes classiques d'estimation du canal pour les communications véhiculaires. Cependant, ces estimateurs conventionnels reposent sur de nombreuses hypothèses qui limitent leurs performances dans les canaux hautement dynamiques variant dans le temps. De plus, les estimateurs linéaires conventionnels sont des solutions peu pratiques dans des scénarios de cas réels car ils reposent sur des modèles statistiques qui nécessitent une complexité élevée. Bien qu'il existe des estimateurs linéaires simples avec une complexité abordable, ils manquent de robustesse dans les environnements très dynamiques. Par conséquent, étudier des estimateurs avec un bon compromis complexité/performance est une problématique importante à investir. En tant qu'approche dominante de l'IA, l'apprentissage profond développe des méthodes efficaces pour analyser des données en apprenant efficacement les structures pour plusieurs problèmes rencontrés dans divers domaines scientifiques. La principale raison de l'intégration de l'apprentissage profond dans les communications sans fil est de trouver des solutions aux problèmes de communication lorsque les solutions analytiques sont insolubles ou très complexes. L'apprentissage profond a une forte capacité à relever ce défi grâce à des solutions peu complexes et robustes qui améliorent les

performances des systèmes sans fil. De plus, le traitement distribué basé sur GPU permet l'utilisation de l'apprentissage profond dans des applications à temps réel. En conséquence, l'apprentissage automatique s'exploite pour les différentes données générées dans les réseaux véhiculaires. Dans ce contexte, cette thèse vise à étudier comment adapter de tels outils pour tenir compte des caractéristiques des réseaux de véhicules à haute mobilité. Nous montrons que l'intégration d'architectures optimisées d'apprentissage profond apporte des solutions de faible complexité pour l'estimation de canaux des réseaux véhiculaires soit en améliorant les performances par rapport aux estimateurs de canaux linéaires, soit en approchant les performances d'estimateurs robustes tout en réduisant la complexité d'implémentation. Par conséquent, contrairement aux estimateurs conventionnels, les estimateurs basés sur l'apprentissage profond offrent un bon compromis entre la complexité de calcul et les performances du système. De plus, la capacité de généralisation rend le système plus robuste surtout quand il est déployé dans des environnements hautement dynamiques.

Contents

List of Figures

List of Tables

Abbreviations

DSRC	dedicated short-range communications
C-ITS	cooperative intelligent transport system
RSU	road side unit
TDL	tapped delay line
ITS	Intelligent Transportation Systems
IEEE	Institute of Electrical and Electronics Engineers
WAVE	Wireless Access in Vehicular Environment
V2V	vehicle-to-vehicle
V2I	vehicle-to-infrastructure
CCH	control channel
SCH	service channels
STS	short training symbols
LTS	long training symbols
SoA	state-of-the-art
DPA	data-pilot aided
STA	spectral temporal averaging
CDP	constructed data pilots
TRFI	time domain reliable test frequency domain interpolation
MMSE-VP	minimum mean square error using virtual pilots
iCDP	Improved CDP
SBS	symbol-by-symbol
FBF	frame-by-frame
SR-CNN	super resolution CNN
DN-CNN	denoising CNN
RBF	radial basis function
CNN	convolutional neural network
TS-ChannelNet	Temporal spectral ChannelNet
WSSUS	wide-sense stationary uncorrelated scattering
TDR	transmission data rate

LSTM	long short-term memory
ALS	accurate LS
SLS	simple LS
ChannelNet	channel network
ADD-TT	average decision-directed with time truncation
WI	weighted interpolation
SR-ConvLSTM	super resolution convolutional long short-term memory
RS	reliable subcarriers
URS	unreliable subcarriers
AE-DNN	auto-encoder deep neural network
T-DFT	truncated discrete Fourier transform
TA-TDFT	temporal averaging T-DFT
TA	time averaging
PDP	power delay profile
AWGN	additive white Gaussian noise
BER	bit error rate
CP	cyclic prefix
DFT	discrete Fourier transform
DL	deep learning
DNN	deep neural network
FFT	fast Fourier transform
IEEE	institute of electrical and electronics engineers
LMMSE	linear minimum mean square error
LOS	line-of-sight
LS	least squares
LTV	linear time-variant
MMSE	minimum mean squared error
MSE	mean squared error
NLOS	non-line-of-sight
NMSE	normalized mean-squared error
OFDM	orthogonal frequency division multiplexing
PDF	probability distribution function
SNR	signal-to-noise ratio

Mathematical Notations

Notations

$\mathbb{C}$	Complex number field
$\mathbb{R}$	Real number field
$\boldsymbol{A}$, $\boldsymbol{a}$	Uppercase (Lowercase) and bold face for matrix (vector)
a, A	Scalar value
$\mathcal{A}$	Calligraphic letter for set
$\lvert\mathcal{A}\rvert$	Number of elements in the set $\mathcal{A}$
$\boldsymbol{A}[n, m]$, $\boldsymbol{a}[n]$	The (n, m)-th element (n-th element) of a matrix (vector). The result is zero if the index is out of the range.
$\boldsymbol{A}[n, :], \boldsymbol{A}[:, m]$	The n-th column vector and the m-th row vector, receptively
$x[n]$	The discrete-time signal value at the sample n
$x(t)$	Continuous-time signal value at the instance t
$\boldsymbol{I}_N$	Identity matrix of size $N \times N$
$\delta[n]$	Discrete Dirac pulse, $\delta[0] = 1$ and $\delta[n] = 0, n \neq 0$

Functions and operators

$\mathrm{vec}\{\boldsymbol{A}\}$	Vectorization operation
$\mathrm{diag}\{\boldsymbol{A}\}$	Diagonal elements of $\boldsymbol{A}$
$\{\cdot\}^H$	Matrix or vector Hermitian transpose $\boldsymbol{A}^H = \{\boldsymbol{A}^*\}^T$
$\{\cdot\}^{-1}$	Inverse of a square matrix
$\mathrm{trace}\{\cdot\}$	Trace of a square matrix $\boldsymbol{A} \in \mathbb{C}^{N\times N}$, $\mathrm{trace}\{\boldsymbol{A}\} = \sum_{n=0}^{N-1} \boldsymbol{A}[n,n]$
$\|\cdot\|$	Second order norm of a vector, $\|\boldsymbol{a}\| = \boldsymbol{a}^H\boldsymbol{a}$
$\mathrm{E}[\cdot]$	Expectation function
$\mathrm{Re}\{\cdot\}$	Real part of a complex-valued input
$\mathrm{Im}\{\cdot\}$	Imaginary part a complex-valued input
$\odot$	Element-wise product of matrices or vectors
$\oslash$	Element-wise division Element-wise product of matrices or vectors
$*$	Linear convolution, $y[n] = h[n] * x[n] = \sum_l h[l]x[n-l]$
$\circledast$	Circular convolution, $\boldsymbol{y}[n] = \boldsymbol{h}[n] \circledast \boldsymbol{x}[n] = \sum_{l=0}^{N-1} \boldsymbol{h}[l]\boldsymbol{x}[\langle n-l\rangle_N]$, $\tilde{\boldsymbol{y}} = \tilde{\boldsymbol{h}} \odot \tilde{\boldsymbol{x}}$, $\boldsymbol{y} = \boldsymbol{C}^{(h)}\boldsymbol{x}$

Introduction

Wireless Communication is the fastest growing technology in the communication field. It can be defined as a method of transmitting information from one point to another without using any connection like wires, cables or any physical medium. Generally, in a communication system, information is transmitted from transmitter to receiver that are placed over a limited distance, or with the help of wireless communication, where the transmitter and the receiver can be placed anywhere between a few meters (Bluetooth communications) to a few thousand kilometers (satellite communication).

The primary and important benefit of wireless communication is mobility that offers the freedom to move around while still being connected to the network. There exist several applications where network terminals are said to be in motion. Unmanned aerial vehicles (UAV) also referred to as drones, are playing recently an increasingly important role in military, public, and civilian applications. Therefore, UAVs have become a topic of central research interest in the wireless communication community. Another important application is the internet of things and its related fields [Chafii et al., 2018b, Chafii et al., 2018a]. Moreover, vehicular communications applications like autonomous driving and vehicles platooning systems are considered as a trendy research topics that contribute in designing and managing smart cities. In such applications, the mobility feature brings several challenges that highly impact the communication reliability, like fast and frequent handovers, carrier frequency offset, inter-carrier interference, high penetration loss, and fast time-varying wireless channel, where the channel suffers from multi-path fading besides large Doppler spread.

In this context, estimating and tracking the wireless channel variations is an important task, since a precisely estimated channel response is critical for the follow-up equalization, demodulation, and decoding operations applied at the receiver [Bomfin et al., 2021a, Bomfin et al., 2021b]. Therefore, the accuracy of the channel estimation influences the system performance. Moreover, channel state information is highly relevant for several applications. For instance, it is employed in wireless physical layer security for secrete key generation [Hyadi et al., 2016, Hamamreh et al., 2019, Wu et al., 2018], indoor localization [Njima et al., 2020, Njima et al., 2021b, Njima et al., 2021a, Njima et al., 2021c, de Almeida et al., 2021], and MIMO based systems [Ehsanfar et al., 2020, Ehsanfar et al., 2019].

Deep learning (DL) has recently drawn attentions for its great success in improving several physical layer applications of communications, especially, channel estimation. DL-based algorithms can provide a good generalization ability and trade-off between the system performance and the required computational complexity. In this context, this dissertation addresses mainly DL-based channel estimation in high mobility vehicular communications.

Thesis Outline

The organization of this dissertation and chapters content are presented as follows.

Chapter 1 presents the fundamentals used in the dissertation. First, a background on wireless communications and time-varying channel with its different statistical properties are recalled. Next, the model of the OFDM transmission over a multipath channel is formalized. Finally, vehicular communications concept is introduced as well as the motivation behind using deep learning (DL) techniques in the channel estimation procedure.

Chapter 2 introduces a detailed illustration of the physical layer transceiver system model. In addition, the signal model description and the vehicular channel models employed along the dissertation are provided.

In Chapter 3, conventional state-of-the-art (SoA) channel estimators that were proposed for vehicular communications are presented and discussed. After that, a performance evaluation is carried on, where the conventional estimators are compared in terms of bit error rate (BER), normalized mean-squared error (NMSE), and computational complexity. Finally, their limitations are discussed, and our strategies for solving the conventional estimators limitations are briefly introduced.

Chapter 4 describes our first strategy for improving the channel estimation without the use of DL techniques. In this context, two conventional estimators namely truncated discrete Fourier transform (T-DFT) and temporal averaging T-DFT (TA-TDFT) are proposed. The basic idea behind these estimators is to estimate only the channel dominant taps while considering the rest taps as an additional noise. Moreover, analytical analysis shows that applying time averaging to the T-DFT estimated channel is able to further improves the overall performance. Analytical and simulation results implemented using different vehicular channel models reveal the performance superiority of the proposed estimators compared to the conventional estimators while recording a significant decrease in the computational complexity.

In chapter 5, the proposed DL-based symbol-by-symbol (SBS) channel estimators for doubly-selective wireless channels are provided and discussed, where deep neural network (DNN) and long short-term memory (LSTM) networks are utilized. Employing DL networks in the channel estimation reveals a significant computational complexity decrease while preserving a good performance. The proposed estimators are compared with the

recently proposed SoA DL-based estimators, and shown to outperform them in different vehicular channel models.

Chapter 6 focuses on the proposed DL-based frame-by-frame (FBF) estimators, where convolutional neural network (CNN) is utilized. The proposed DL-based FBF estimators improve the estimation accuracy through reformulating a robust and accurate initial estimation based on weighted interpolation (WI). After that, further performance improvement is achieved by employing optimized CNN architectures. Extensive numerical experiments demonstrate that the developed estimators significantly outperform the recently proposed DL-based FBF estimators in different vehicular scenarios, while substantially reducing the overall computational complexity.

This thesis started in January 2019 and defended in December 2021, and it is supported by the CY Initiative of Excellence through the ASIA Chair of Excellence Grant (PIA/ANR-16-IDEX-0008).

Thesis Contributions

In this dissertation, we mainly investigate improving channel estimation methods using DL and signal processing techniques, specifically, in high mobility vehicular communications. However, we note that the proposed estimators in this thesis can be easily adapted to any other application by just re-configuring the simulation parameters according to the standards managing each application. Accordingly, the contributions of this thesis can be summarized as follows:

- **DFT-based SBS Channel Estimation**
  - Propose a low-complexity T-DFT channel estimator that adapts the discrete Fourier transform (DFT) based channel estimation to the vehicular communications standard by estimating only the channel impulse response of dominant taps using the predefined pilots.
  - Show that applying an appropriate temporal averaging to T-DFT results in an improved performance in different vehicular channels conditions.
  - Derive analytically the NMSE expressions of T-DFT and TA-TDFT estimators, and demonstrate how the noise power is degraded throughout the received frame.
  - Show that the proposed T-DFT and TA-TDFT estimators outperform conventional estimators with less computational complexities. In addition to their robustness superiority compared to conventional linear minimum mean square error (LMMSE) estimator.
- **DL-based SBS Channel Estimation**

- Propose two DNN-based SBS channel estimators for vehicular communications. The proposed estimators are based on applying a DNN based processing on top of conventional spectral temporal averaging (STA) and time domain reliable test frequency domain interpolation (TRFI) vehicular channel estimators.
- Propose an LSTM-based SBS estimator for vehicular environment, where we show that LSTM is more robust and efficient in tracking the vehicular channel especially in high mobility scenarios. Simulation results show that LSTM-based estimation outperforms DNN-based estimation. This can be explained by the high ability of LSTM in learning the channel time correlations, compared with a simple DNN architecture.
- Perform intensive experiments on several DNN and LSTM architectures using the grid search algorithm, in order to select the most suitable hyper parameters in terms of both performance and complexity. The proposed DNN and LSTM based estimators are able to significantly outperform the recently proposed DL-based estimator, while achieving a considerable computational complexity decrease.

▷ **DL-based FBF Channel Estimation**

- Propose a hybrid, adaptive, low-complexity, and robust WI channel estimators that use modified pilot allocation schemes within the transmitted frame and adapt the employed scheme according to the mobility condition.
- Derive analytically the expression of the employed interpolation matrix for the proposed pilot allocation schemes.
- Integrate an optimized super resolution CNN (SR-CNN) and denoising CNN (DN-CNN) networks on top of the WI estimators to enhance the performance.
- Show that the proposed WI estimators outperform the recently proposed DL-based FBF estimators in terms of latency and transmission data rates, while recording substantial reduction in computational complexity.

Chapter 1

Fundamentals and Background

Contents

This chapter gives a summary of some fundamentals about the time-varying wireless channel, where the modeling and statistical tools used to characterize it are firstly covered. Second, the orthogonal frequency division multiplexing (OFDM) transmission and its benefits in a multipath environment are presented. After that, vehicular communications are presented as a case study, where its main concept is introduced. Finally, the motivation of using DL as a promising solution for wireless communications challenges, especially channel estimation, is discussed.

1.1 Wireless Communications Channel

In wireless communication, the transmitted signal propagates through a multi-path noisy environment, thus, it takes different paths that arrive at the receiver at different timing

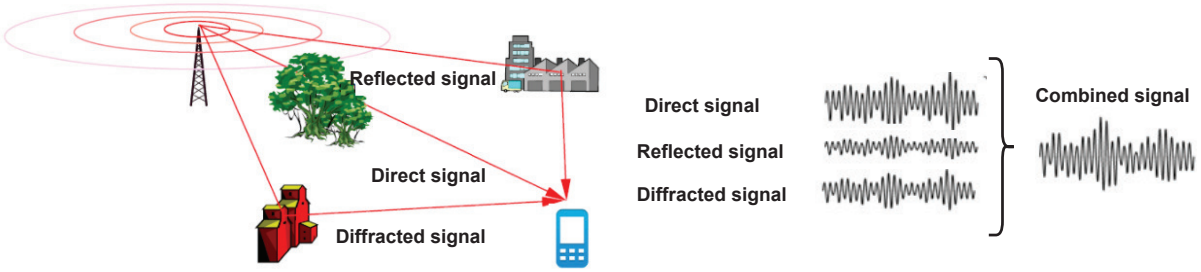
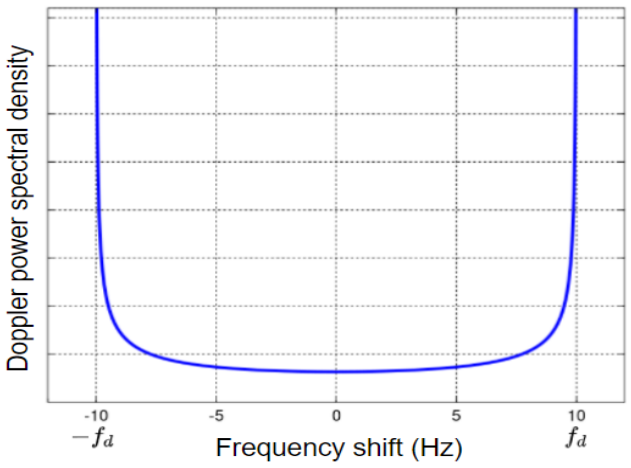


Figure 1.1: Signal propagation in wireless environment.

with different signal strength as shown in Fig. 1.1. As a result, the signal coming into the receiver is the superimposition of all the components that are constructively or destructively combined. This combination of all the incoming signals makes the received signal different from the original transmitted signal. This phenomenon is called fading.

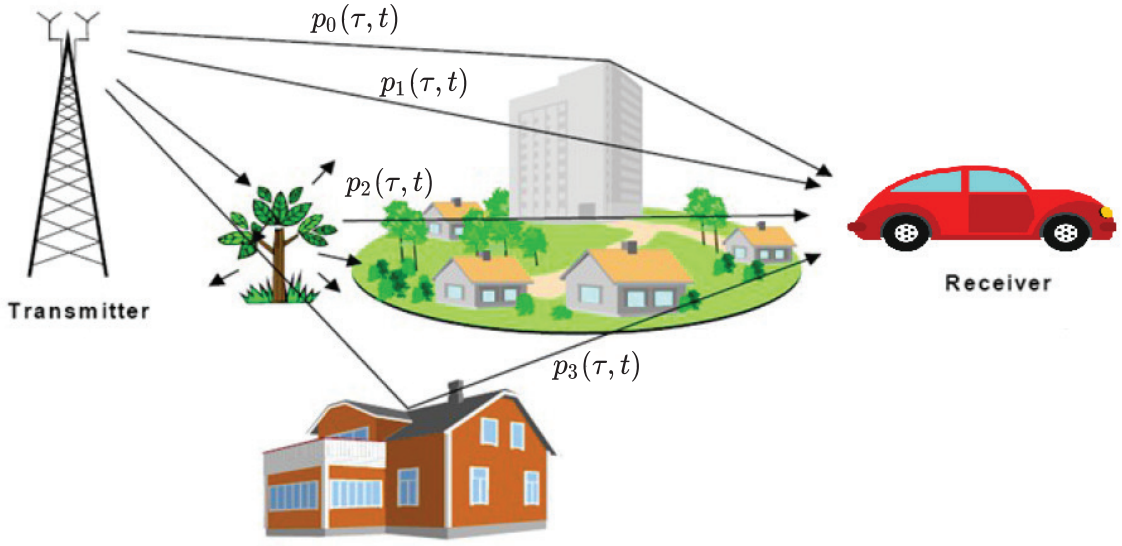
1.1.1 Major Components of Fading

There are many factors generating fading effect, where the most important channel fading factors are: (*i*) Multi-path propagation and (*ii*) Doppler Shift. Multi-path propagation is caused by multiple receptions of the same signal. The signal travels along different paths, and therefore is received at slightly different times, leading to a frequency-selective channel. Moreover, relative motion between the transmitter and receiver (or surrounding objects) causes random frequency modulation, since each multi-path component has a different Doppler shift. Hence, the channel becomes doubly selective, i.e. varies in time and frequency.



$$f_d = \frac{v f_c}{c}$$

Figure 1.2: Doppler spectrum in doubly selective wireless channel.



$$p_l(\tau, t) = \alpha_l(t)\delta(\tau - \tau_l(t))$$

Figure 1.3: Transmitted signal propagation in doubly selective wireless channel.

The impact of Doppler shift is illustrated and calculated as shown in Fig. 1.2. As Doppler spectrum shows how much the carrier frequency f_c gets spread by Doppler effect indicated by f_d that is determined by the velocity v of an object. We note that f_d and v are proportionally related, meaning if v gets higher, f_d gets larger, then, the Doppler spectrum width gets wider, and vise versa.

1.1.2 Doubly Selective Channel Characterization

The doubly selective wireless channel can be modeled as a linear time-variant (LTV) channel that can be characterized by four functions [Bello, 1963] as follows:

1.1.2.1 Delay-Time Response

In order to explain what is the channel delay-time response, let us consider the following case shown in Fig. 1.3.

A vehicle in motion is communicating with the base station and there are many buildings between them. One may intuitively guess that the vehicle would receive multiple copies of the transmitted signal from the base station because the transmitted signal will reach the vehicle via multiple path as we discussed. Let us assume that we have four different paths as shown in Fig. 1.3. There are two major time-variant characteristics of these multi-paths: (i) time delay and (ii) attenuation. The mathematical representation of the time-variant channel impulse response can be expressed as follows:

$$\boldsymbol{h}(\tau, t) = \sum_{l=0}^{L-1} \alpha_l(t)\delta(\tau - \tau_l(t)). \tag{1.1}$$

where $\alpha_l(t)$ and $\tau_l(t)$ denote the channel propagation gain known as attenuation and delay of the l-th propagation path as function of time respectively.

According to the central limit theorem can be applied [Dodge, 2008], the time-variant impulse response can be modeled as a complex-valued Gaussian random process in the t variable, and each channel multi-path component $p_l(\tau, t)$ is associated with specific statistical distribution. When the channel multi-paths are non-line-of-sight (NLOS), the channel power profile shows a Rayleigh distribution. Moreover, the probability distribution function (PDF) of the Rayleigh distribution channel is expressed as follows:

$$P_{\text{Ray}}(x) = \frac{x}{\sigma_h^2} e^{\frac{-r^2}{2\sigma_h^2}}, \tag{1.2}$$

where x is a positive real value and $\sigma_h^2 = E\{|h(t)|^2\}$. Moreover, the PDF of the phase of a Rayleigh process noted $\Phi_{\text{Ray}}(\theta)$ can be expressed as:

$$\Phi_{\text{Ray}}(\theta) = \frac{1}{2\pi} \quad \forall\theta \in [-\pi, \pi]. \tag{1.3}$$

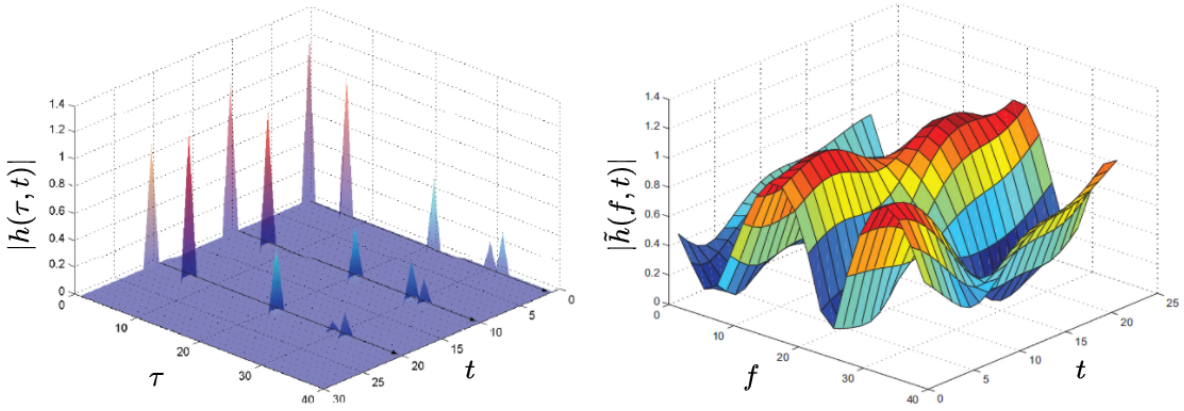
On the other hand, when a strong dominant component denoted as line-of-sight (LOS) path is presented, then the corresponding probability density function that follows the Rician distribution is noted $P_{\text{Ric}}(x)$, and given as [Rice, 1948]:

$$P_{\text{Ric}}(x) = \frac{x}{\sigma_h^2} e^{\frac{-r^2+\rho^2}{2\sigma_h^2}} J_0(\frac{x\rho}{\sigma_h^2}). \tag{1.4}$$

where $J_0(.)$ is the modified Bessel function of the first kind with order zero, ρ^2 is the variance of the line of sight path, and σ_h^2 the variance of the scattered NLOS components. We define the Rice factor $C_R = \frac{\rho^2}{\sigma_h^2}$ as the ratio between the power of the LOS received multi-path component and all the NLOS multi-path components. The probability density function of the phase Φ_{Ric} is given by:

$$\Phi_{\text{Ric}}(\theta) = \frac{e^{-C_R}}{2\pi}\left(1 + \sqrt{\frac{\pi\rho^2}{\sigma_h^2}}\cos(\theta - \theta_\rho) e^{\frac{\rho^2\cos^2(\theta-\theta_\rho)}{\sigma_h^2}}(1 + erf(\frac{\rho\cos(\theta-\theta_\rho)}{\sigma_h^2}))\right). \tag{1.5}$$

where erf(.) is the error function and θ_ρ is the phase of the LOS path. For $x \geq 0$, the error function is defined as $erf(x) = \frac{2}{\pi}\int_0^x e^{-t^2}\mathrm{d}t$, θ is defined as $-\pi \leq \theta \leq \pi$. We remark in (1.4) and (1.5) that if $\rho^2 \to 0$, we naturally get the Rayleigh model that has been previously described.



(a) Channel impulse response $\boldsymbol{h}(\tau, t)$ **(b)** Channel frequency response $\tilde{\boldsymbol{h}}(f, t)$

Figure 1.4: Channel Impulse and frequency responses

1.1.2.2 Frequency-Time Response

The channel frequency-time response known as the time-variant transfer function is obtained by Fourier transforming the impulse response with respect to the delay variable τ as shown below:

$$\tilde{\boldsymbol{h}}(f,t) = F_{\tau}\{\boldsymbol{h}(\tau,t)\} = \int_{-\infty}^{\infty} \boldsymbol{h}(t,\tau)e^{-j2\pi f\tau}d\tau = \sum_{l=0}^{L-1} \alpha_l(t)e^{-j2\pi f\tau_l(t)}. \tag{1.6}$$

Figure 1.4 shows the time variant channel impulse and frequency responses.

1.1.2.3 Frequency-Doppler Response

The frequency-Doppler response known as Doppler-spread function is obtained by Fourier transforming the time-variant transfer function with respect to the time variable t as shown in 1.7, such that:

$$\ddot{\boldsymbol{h}}(f,v) = F_t\{\tilde{\boldsymbol{h}}(f,t)\} = \int_{-\infty}^{\infty} \tilde{\boldsymbol{h}}(f,t)e^{-j2\pi vt}dt. \tag{1.7}$$

1.1.2.4 Delay-Doppler Response

The delay-Doppler spread function is obtained by Fourier transforming the time-variant impulse response with respect to the time variable t as shown in Fig. (1.8), or by taking the inverse Fourier-transform of the output Doppler-spread function with respect to the frequency. The delay-Doppler spread function describes the complex gain of the channel in a specific delay and Doppler shift interval.

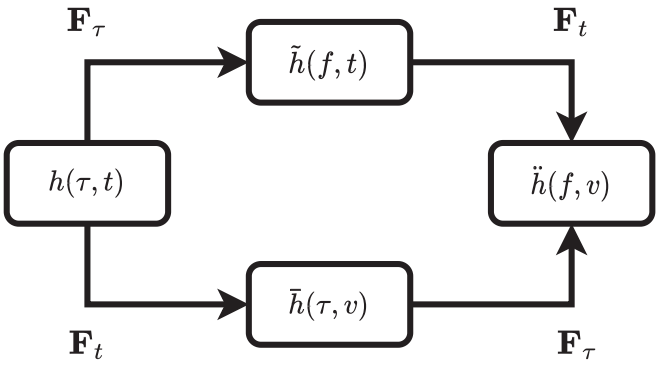


Figure 1.5: Relationships between LTV channel responses.

$$\bar{\boldsymbol{h}}(\tau,v) = F_t\{\boldsymbol{h}(\tau,t)\} = \int_{-\infty}^{\infty} \tilde{\boldsymbol{h}}(\tau,t)e^{-j2\pi vt}dt \tag{1.8}$$

Figure 1.5 describes the relationships between different time variant LTV channel responses.

1.1.3 LTV Channel Correlation Functions

Assuming that the wireless channel follows the wide-sense stationary uncorrelated scattering (WSSUS) model, therefore we have:

- ▷ Each path $h_l(t)$ in (1.1) is a zero mean Gaussian complex process, $E\{h_l(t)\} = 0, \forall t$, and then the mean of each path is independent of the time variations. Furthermore, the time correlation function $r_{h_l}(t_1,t_2) = E\{h_l(t_1)h_l^*(t_2)\}$ can only be written with the difference $\Delta(t) = (t_1 - t_2)$, such that:

$$r_{h_l}(t_1,t_2) = r_{h_l}(\Delta_t). \tag{1.9}$$

- ▷ Uncorrelated scattering (US) means that the paths are uncorrelated, so for $l_1 \neq l_2$ we have:

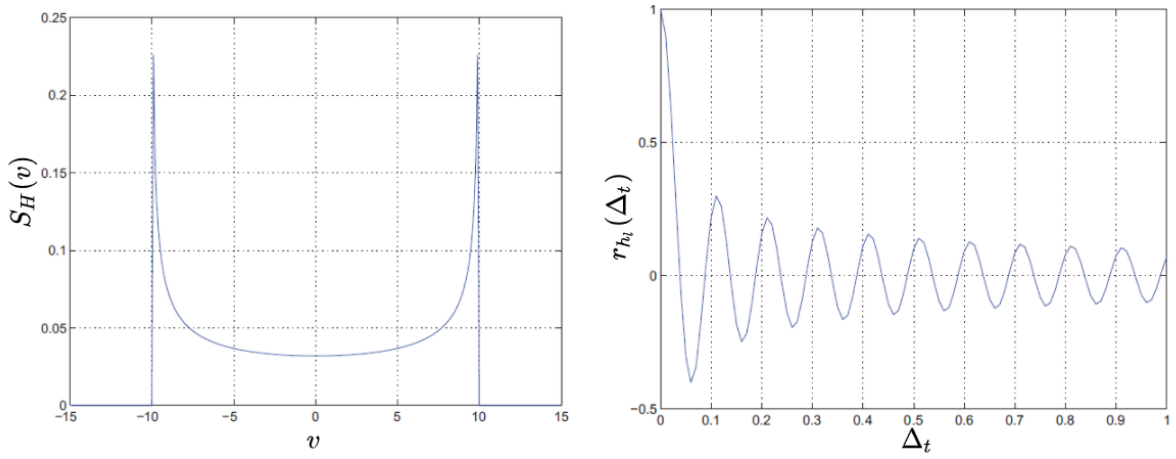
$$E\{h_{l_1}(t)h_{l_2}^*(t)\} = 0. \tag{1.10}$$

Following the above assumptions, we can define the following functions for the time-variant WSSUS channel model.

1.1.3.1 Time Correlation Function

The time-frequency correlation function, noted P_H and defined by $P_H(f_1,f_2,t_1,t_2) = E\{\tilde{\boldsymbol{h}}(f_1,t_1)\tilde{\boldsymbol{h}}^*(f_2,t_2)\}$ has the following property:

$$P_H(f_1,f_2,t_1,t_2) = P_H(\Delta_f,\Delta_t). \tag{1.11}$$



(a) Channel Doppler spectrum $S_H(v)$ **(b)** Channel time correlation $r_{h_l}(\Delta_t)$

Figure 1.6: Channel Doppler Spectrum and time correlation function for $f_d = 10$ Hz, $\sigma_h^2 = 1$.

where $\Delta_f = (f_1 - f_2)$. Following the Jakes model derivation provided in [G Vega et al., 2019], the time correlation function can be calculated by considering $\Delta_f = 0$, and expressed as $P_H(\Delta_f = 0, \Delta_t) = r_{h_l}(\Delta_t)$, such that:

$$r_{h_l}(\Delta_t) = \sigma_h^2 J_0(2\pi f_d \Delta_t). \tag{1.12}$$

where J_0 is the Bessel function of the first kind with order zero, and f_d is called maximum Doppler frequency, and represents the time variation of the mobile channel.

1.1.3.2 Doppler Spectrum

The channel Doppler spectrum is another useful function describing the channel. Considering a Rayleigh distributed channel response, it is shown that the channel Doppler spectrum follows the "U" shaped Jakes' model [G Vega et al., 2019]. The channel Doppler spectrum denoted as $S_H(\Delta_f, v)$ is defined as the Fourier transform of the time correlation function $r_{h_l}(\Delta_t)$. Hence, we note $S_H(\Delta_f = 0, v) = S_H(v)$, such that:

$$S_H(v) = F_{\Delta_t}\{r_{h_l}(\Delta_t)\} = \begin{cases} \frac{\sigma_h^2}{\pi f_d \sqrt{1-(\frac{v}{f_d})^2}}, & |v| \leq f_d. \\ 0, & elsewhere, \end{cases} \tag{1.13}$$

We note that, if the channel is time invariant ($r_{h_l}(\Delta_t) = 1$), i.e. there is no time variations in the channel, then $S_H(v) = \delta(v)$ and there is no spectral broadening observed in the transmission of a pure frequency tone. Moreover, the range of v where $S_H(v)$ is non zero is called the Doppler spread f_d of the channel. Since $S_H(v)$ and $r_{h_l}(\Delta_t)$ are related by the Fourier transform, the reciprocal of f_d is a measure of the channel coherence time T_c, such that:

$$T_c = \sqrt{\frac{9}{16\pi}}\frac{1}{f_d} = \frac{0.423}{f_d}. \tag{1.14}$$

In this context, we can say that a slowly changing channel has a large coherence time, or equivalently a small Doppler spread.

1.1.3.3 Power Delay Profile

As the channel is supposed to follow a WSSUS channel model, the paths are uncorrelated, then the channel intensity profile as a function of time delay τ and difference Δ_t denoted as channel power delay profile (PDP) $\Gamma_H(\tau, \Delta_t)$. We note that, $\Gamma_H(\tau)$ is measured in practice by taking the spatial average of the channel impulse response $|\boldsymbol{h}(\tau, t)|^2$ over a local area [Acosta-Marum and Ingram, 2006]. Moreover, the range of τ values where $\Gamma_H(\tau)$ is non zero is called the multipath spread of the channel denoted by $\tau_{\max}$.

1.1.3.4 Frequency Correlation Function

We here give the relation between the frequency correlation function and the channel PDP. We restrict the correlation function (1.11) to its frequency part, i.e. the function $P_H(\Delta_f, \Delta_t)$ for $\Delta_t = 0$, that we note $P_H(\Delta_f)$. The frequency correlation function and the channel PDP are linked by a Fourier transform relation as:

$$P_H(\Delta_f) = F_\tau\{\Gamma_H(\tau)\}. \tag{1.15}$$

Since $P_H(\Delta_f)$ is an autocorrelation function in the frequency variable, it provides a measure of the frequency coherence of the channel. Hence, as a result of the Fourier transform relationship, the reciprocal of the multipath spread is the coherence bandwidth of the channel denoted as:

$$B_C = \frac{1}{\beta\ \tau_{max}}. \tag{1.16}$$

β is a variable factor related to the definition of the coherence bandwidth. For example, if we define the coherence bandwidth as the range of frequencies over which the frequency correlation is above 0.5, then, we have $\beta = 5$, and we have 50% coherence bandwidth. We note that, if B_C is small in comparison of the bandwidth of the transmitted signal, the channel is said to be frequency selective, in this case the transmitted signal is severely distorted by the channel. On the other hand, if B_C is large in comparison with the bandwidth of the transmitted signal, then the channel is said to be frequency flat.

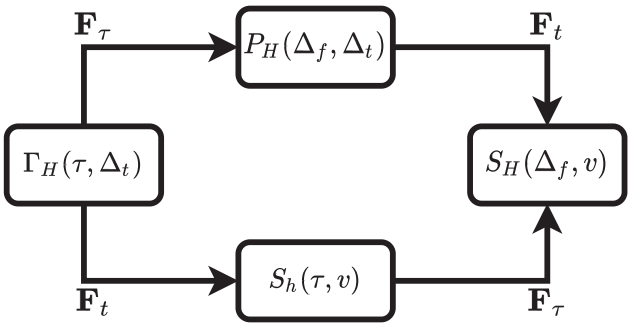


Figure 1.7: Relationships between the correlation functions of a WSSUS time-variant channel.

1.1.3.5 Scattering Function

The fourth statistical function that is used to describe the channel is the scattering function, noted S_h, and function of the variables v and τ . Determining $S_h(\tau, v)$ can be easily derived from the previously defined functions with the following relation:

$$S_h(\Delta_f) = F_{\Delta_t}\{\Gamma_H(\tau)\} \tag{1.17}$$

We note that the shape of $S_h(\tau, v = 0)$ determines the length and gain of the channel multi-paths. Moreover, As the channel varies rapidly, it becomes very difficult to obtain a dynamic knowledge of its perfect knowledge characteristics. However, studying the time variant channel responses and correlation functions especially by approximations and models measurements, may help in managing the transmission operations. Moreover, analyzing the doubly selective channel characteristics and its variation speed is crucial to establish a robust and reliable wireless communications, where the doubly selective channel estimation task plays a vital role in determining the overall system performance. The statistical functions and the links between them are provided in Fig. 1.7.

1.2 OFDM Theory

OFDM, is a key technology in current high data rate wireless communications, since it has shown success in combating multi-path fading impact. As a definition, OFDM is a multicarrier transmission technique that divides the available band into K subbands, corresponding to subcarriers with frequency separation Δf. OFDM can efficiently utilize the available spectrum by spacing subcarriers closer together and making them orthogonal to one another, and thus, preventing guard interval allocation and interference problems. Therefore, the allocated spectrum of each subcarrier has a null at the center frequency of each of the other subcarriers.

In order to illustrate the OFDM transmission principle, let us assume that we have K orthogonal pulses $\boldsymbol{u}_k(t)$, with $k = \{0, 1, \ldots, K-1\}$. In the n-th symbol interval, K

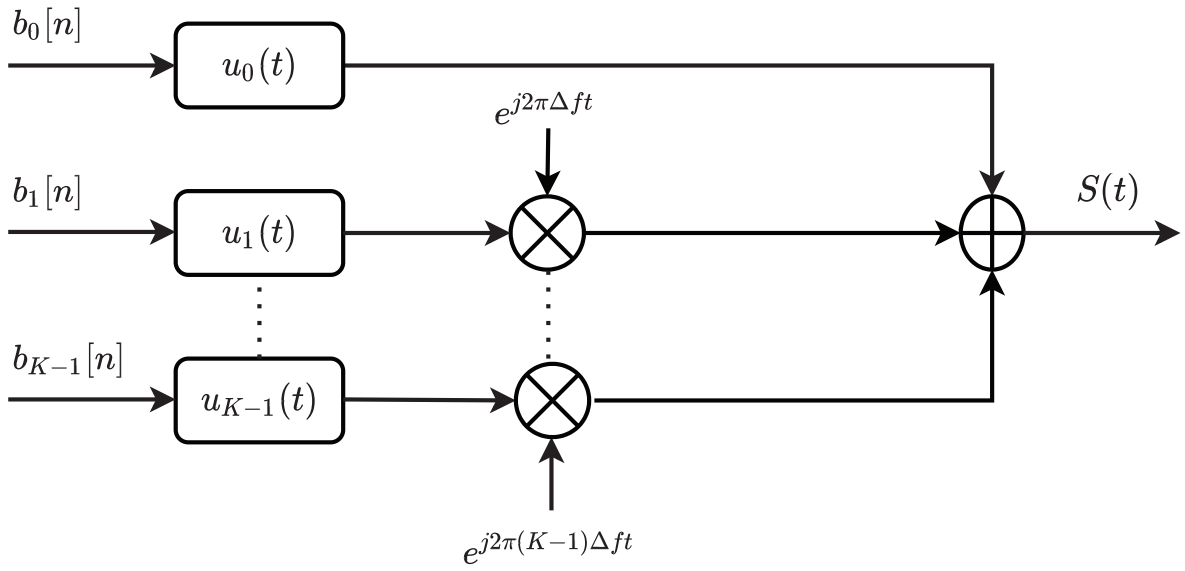


Figure 1.8: OFDM transmitter block diagram.

symbols $\boldsymbol{b}_k[n]$ are transmitted simultaneously. Thus, the transmitted signal component corresponding to the n-th symbol interval is:

$$\boldsymbol{s}_n(t) = \sum_{k=0}^{K-1} \boldsymbol{b}_k[n]\boldsymbol{u}_k(t - nT_s)e^{j2\pi k\Delta ft}. \tag{1.18}$$

The overall transmitted signal, where all signals components corresponding to various symbol intervals are added, can be expressed as follows:

$$\boldsymbol{s}(t) = \sum_{n=-\infty}^{+\infty} \boldsymbol{s}_n(t) = \sum_{n=-\infty}^{+\infty} \sum_{k=0}^{K-1} \boldsymbol{b}_k[n]\boldsymbol{u}_k(t - nT_s)e^{j2\pi k\Delta ft}. \tag{1.19}$$

A block diagram of the OFDM transmitter is shown in Fig. 1.8. In order to achieve orthogonality, OFDM uses sinusoidal pulses spectrally centered about different carrier frequencies. For simplicity, we will assume that the pulses $\boldsymbol{u}_k(t)$ are rectangular pulses of duration $1/\Delta f$ and amplitude $\sqrt{\Delta f}$, given by:

$$\boldsymbol{u}_k(t) = \begin{cases} \sqrt{\Delta f}, & 1 \leq t < \frac{1}{\Delta f}. \\ 0, & \frac{1}{\Delta f} \leq t < T. \end{cases} \tag{1.20}$$

We note that, other pulses can be used such us raised cosine [Chen and Wyglinski, 2010]. As it is shown in Fig. 1.9, pulses do not overlap, as long as $T \geq 1/\Delta f$, or equivalently until $T\Delta f \geq 1$. This fundamental property has to be fulfilled in order to ensure orthogonality of pulses in the time-domain. Moreover, if the above stated condition is not satisfied, there is not any set of orthogonal pulses and therefore the interference between successive received OFDM symbols defined as inter symbol interference (ISI) cannot be avoided in general.

Thus, in order to reduce ISI, we have to ensure that consecutive pulses are spaced enough to overcome the maximum delay spread of the channel. On the other hand, large $T\Delta f$ reduces spectral efficiency of a system, which is inversely proportional to the effective time-bandwidth product.

Since, most of practical implementations use discrete-time processing, lets define the sampled transmitted signal $\boldsymbol{s}[m] = \boldsymbol{s}(mTs)$ using sampling frequency $B = K\Delta f$, such that:

$$\boldsymbol{s}[m] = \frac{1}{K} \sum_{n=-\infty}^{+\infty} \sum_{k=0}^{K-1} \boldsymbol{a}_k[n] e^{\frac{j2\pi km}{K}}. \tag{1.21}$$

Here m and N denote the discrete time variable and the sampling period respectively. Moreover, since OFDM is initially proposed to overcome the channel frequency selectivity problem, we will also assume that the transmitted signal passes through a frequency selective channel, modeled by a finite-length impulse response $\boldsymbol{h}(\tau)$ of length $\tau_{\max}$. Hence, by sampling of $\boldsymbol{h}(\tau)$, we get the discrete time impulse response $\boldsymbol{h}[l]$ with length L. Therefore, the received signal, ignoring any noise effects for simplicity, is depicted in Fig. 1.10 and can be represented by following linear convolution:

$$\boldsymbol{r}[m] = \boldsymbol{h} * \boldsymbol{s}[m] = \sum_{l=0}^{L-1} \boldsymbol{h}[l]\boldsymbol{s}[m-l]. \tag{1.22}$$

According to Fig. 1.10, we have the rectangular pulses of duration K and guard interval $(N-K)$ between consecutive pulses are linearly convoluted with channel impulse response $h[l]$ of duration L. As long as $(N-K) \geq (L-1)$, pulses at the output of the channel do not overlap and can be effectively demodulated by receiver, therefore omitting the impact of ISI. As a result, we conclude that ISI can be avoided if the guard interval between the consecutive OFDM symbols $(N-K)$ is at least as long as the channel impulse response $(L-1)$.

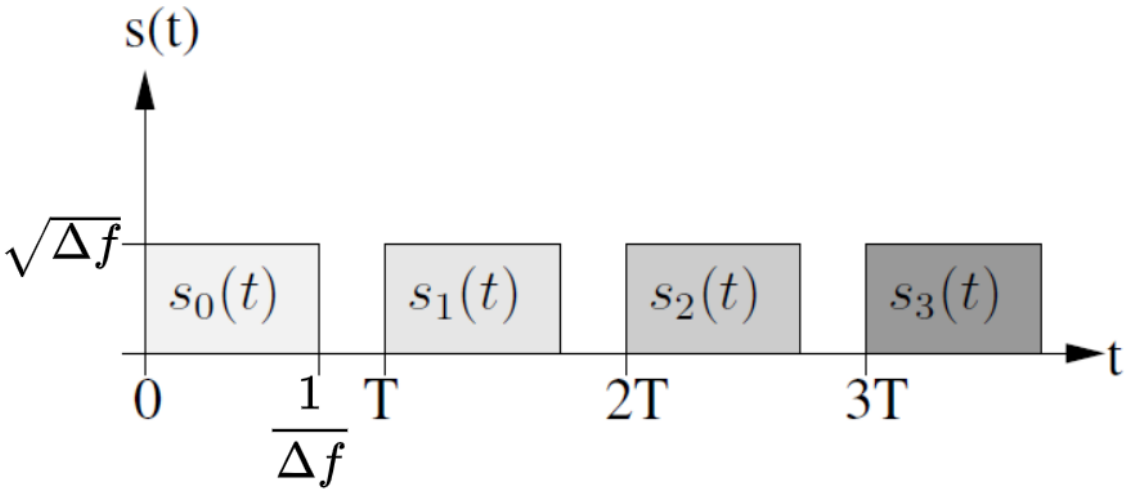


Figure 1.9: OFDM transmitted signal.

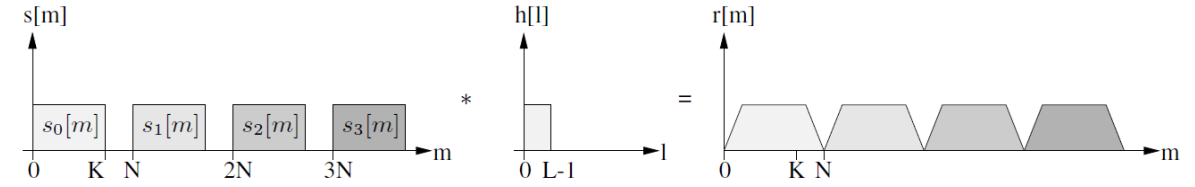


Figure 1.10: Received OFDM signal after passing through a frequency selective channel without appending CP.

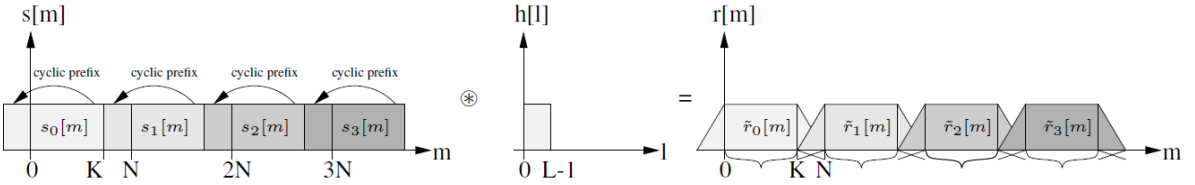


Figure 1.11: Received OFDM signal after passing through a frequency selective channel with appending CP.

At the receiver, demodulation can be efficiently implemented with fast Fourier transform (FFT). However, convolution in time domain is equivalent to multiplication in the frequency domain only if the signals are of infinite length or if at least one of the signals is periodic over the range of the convolution [Lostanlen, 2017]. Thus FFT of the above stated linear convolution, does not correspond to product of channel coefficients with the transmitted symbols. However, it is possible to convert linear convolution into circular, by simply appending each OFDM symbol with replica of the last $(N - K)$ samples of the OFDM symbol, denoted as cyclic prefix (CP). By doing so, the cyclically-extended OFDM symbol now appears periodic when it is convoluted with the channel. Therefore, the received OFDM symbols are given by cyclic-convolution as shown in Fig. 1.11, where the cyclic-convolution corresponds to following multiplicative input-output relation, where each subcarrier is affected by a single channel coefficient, such that:

$$\boldsymbol{r}_k[n] = \boldsymbol{H}_k \boldsymbol{s}_k[n] + \boldsymbol{z}_k[n]. \tag{1.23}$$

We have seen the potential of OFDM transmission in alleviating the channel frequency selective fading impact by omitting the ISI error. It is worth mentioning that, splitting the total band into K subcarriers means that these subcarriers are affected by also K relatively flat fading channels, hence, this fact greatly simplifies the equalization performed by receiver, where the transmitted signal can be re-obtained from received signal using simple zero forcing equalization, in case the channel is accurately estimated, especially when the wireless channel is doubly selective. Finally, we note that the disadvantages of OFDM systems lies in the transmitted signal high peak-to-average power ratio [Hosseinzadeh Aghdam and Sharifi, 2019], due to employing large number of subcarriers, resulting in significant energy loss.

1.3 Vehicular Communications

In this dissertation we will mainly investigate improving channel estimation methods using DL and signal processing techniques, specifically, in high mobility vehicular communications. However, we note that the proposed estimators in this thesis can be easily adapted to any other applications by just re-configuring the simulation parameters according to the new standards managing each application.

1.3.1 Background

About one million traffic accidents occur annually in the USA. In 2003 alone, these accidents accounted for 230 billion in damaged property, 2,889,000 nonfatal injuries, and 42,643 deaths [National Center for Statistics and Analysis, 2004]. Evidently, most of those accidents are preventable by implementing comprehensive wireless communication mechanism to exchange vital safety and emergency information between moving vehicles. In this context, the development and standardization of vehicular communications is motivated by the desire to disseminate road safety information among vehicles to prevent accidents and improve the road safety.

All data collected from the sensors on a vehicle can be displayed to the driver or broadcasted to neighbouring vehicles depending on certain requirements [Djenouri, 2008]. Many more applications are proposed for vehicular networks besides road safety like car-to-home communication, travel and tourism information distribution, autonomous driving, multimedia and game applications, and Internet connectivity.

1.3.2 IEEE Standards and Network Infrastructure

As shown in Fig. 1.12, the architecture of vehicular communications network falls in three main categories:

- **vehicle-to-vehicle (V2V) communications**: In this category, the vehicles communicate among each other with no infrastructure support. Any valuable information collected from sensors on a vehicle, or communicated to a vehicle, can be sent to neighbouring vehicles.
- **vehicle-to-infrastructure (V2I) communication**: In this category, the vehicles can use cellular gateways and wireless local area network access points to connect to the Internet and facilitate vehicular applications.
- **Hybrid V2I communication**: Vehicles can use infrastructure to communicate with each other and share the information received from infrastructure with other vehicles. This architecture includes V2V communication and provides greater flexibility in content sharing.

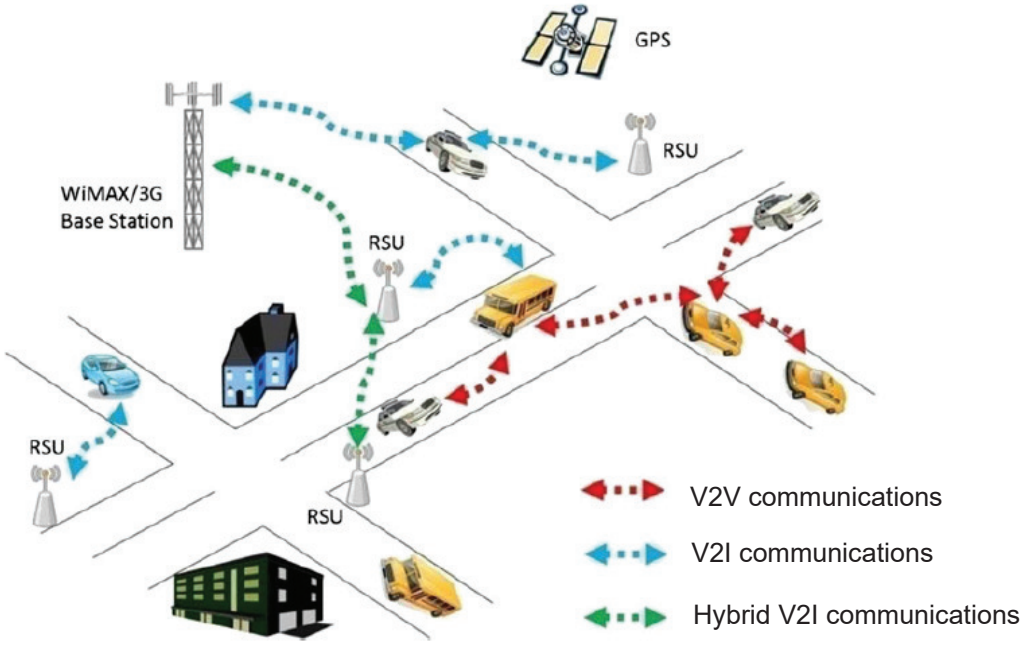


Figure 1.12: Vehicular networks architecture.

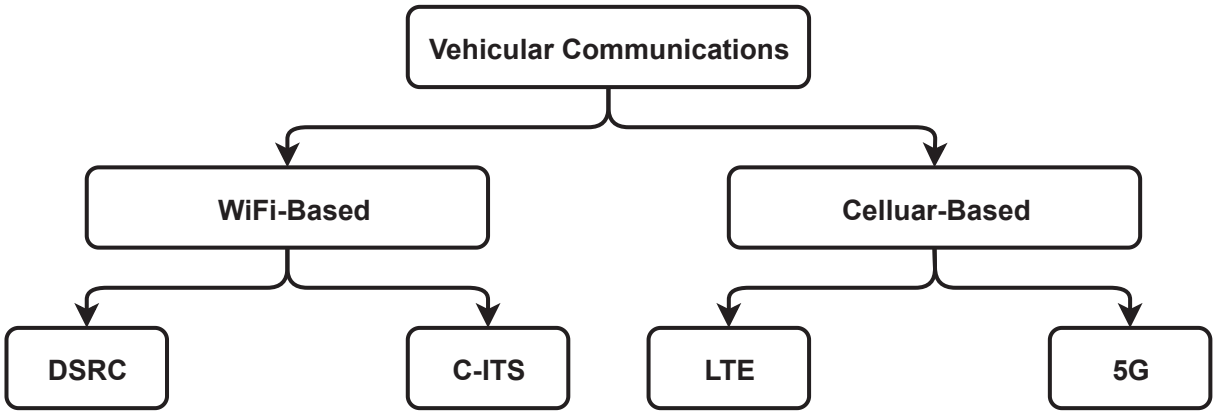


Figure 1.13: Vehicular communications technologies.

Historically, vehicular communications standardization started based on the WiFi technology, where two standards sets have been developed in parallel, in both the U.S. (dedicated short-range communications (DSRC) [Kenney, 2011]) and Europe (cooperative intelligent transport system (C-ITS) [Sjoberg et al., 2017]). This is mainly because the activities were supported by different research and development programs and promoted by different stakeholders, leading to different sets of standards. On the other hand, cellular systems like LTE [Sempere-García et al., 2021] and 5G [Chen et al., 2017] have been used recently to support vehicular communications due to existence of the cellular infrastructure. Table 1.1 show different features of the vehicular communications technologies. We note that, in this dissertation, we will focus on the DSRC standard in our system model.

DSRC is a suite of standards proposed to enable WiFi-based vehicular

Table 1.1: Vehicular communication technologies specifications.

Feature	**WiFi-based**	**Cellular-based**
Bandwidth	Unlicensed band	Licensed band
Waveform	OFDM	SC-FDMA
Channel coding	Convolutional	Turbo
Range	Short radio range	extended communications
Symbol duration (μs)	8	71
Data rate (Mbps)	$3-7$	150
Multimedia & cloud services support	No	Yes
Synchronization	Asynchronous	Synchronous

communications. DSRC enables fast exchanging of safety messages, combined with knowledge about other moving vehicles that may not be visible to drivers in a timely manner, therefore extending the safety concepts beyond the desired expectations [US Department of Transportation, 2006].

DSRC was initially proposed by the Federal Communication Commission (FCC), which, in 1999, reserved 75 MHz of bandwidth in the 5.9 GHz frequency (5.850–5.925 GHz) known as the Intelligent Transportation Systems (ITS) band to support both V2V and V2I communications. The ITS band is composed of seven 10 MHz channels as shown in Fig. 1.14, the central channel is called the control channel (CCH) and it is restricted to safety communications only and it carries the most critical alarms and beacons. It is the unique channel shared between all the DSRC devices, and consequently, constitutes a joining point between the nodes. The two channels at the edge of the spectrum are reserved for prospective applications and particular employment, such as advanced accident prevention and public safety uses. The remaining channels are service channels (SCH) for the residual applications and regular communications. Pairs of these adjacent channels can also be blended in a 20 MHz channel. Nevertheless, the necessity to have 10 MHz channels is associated with the use of parallel applications, and as a consequence, the partial reduction of channel congestion. Besides, physical tests prove that 10 MHz channels are more suitable both for delays and the Doppler effect ascertained in vehicular environments. We note that the channel time is divided into synchronization intervals with a fixed length of 100 ms, consisting of intervals of equal length alternating between the CCH and the SCH. During the CCH interval, all vehicular devices must tune in to the CCH frequency for safety-related and system control data exchange. During the SCH interval, vehicles optionally switch to one of the SCH frequencies. At the beginning of each interval, a guard time of 4 ms is set to take into account the radio switching delay and timing inaccuracies in the devices. Coordination between channels uses Coordinated Universal Time (UTC) for a global time reference provided by a global satellite navigation

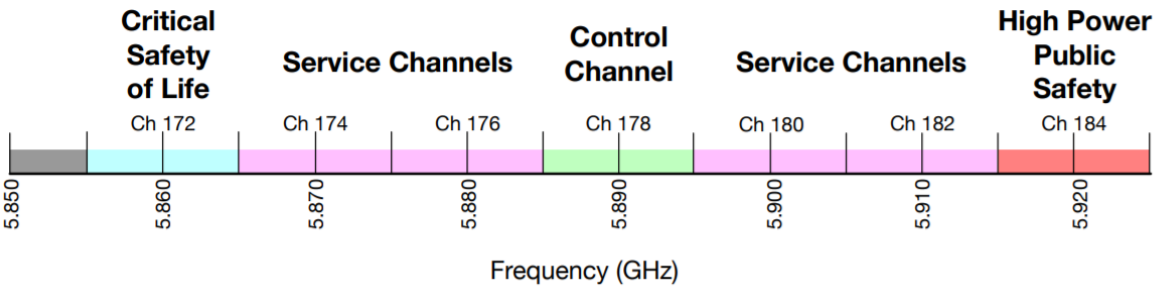


Figure 1.14: DSRC spectrum.

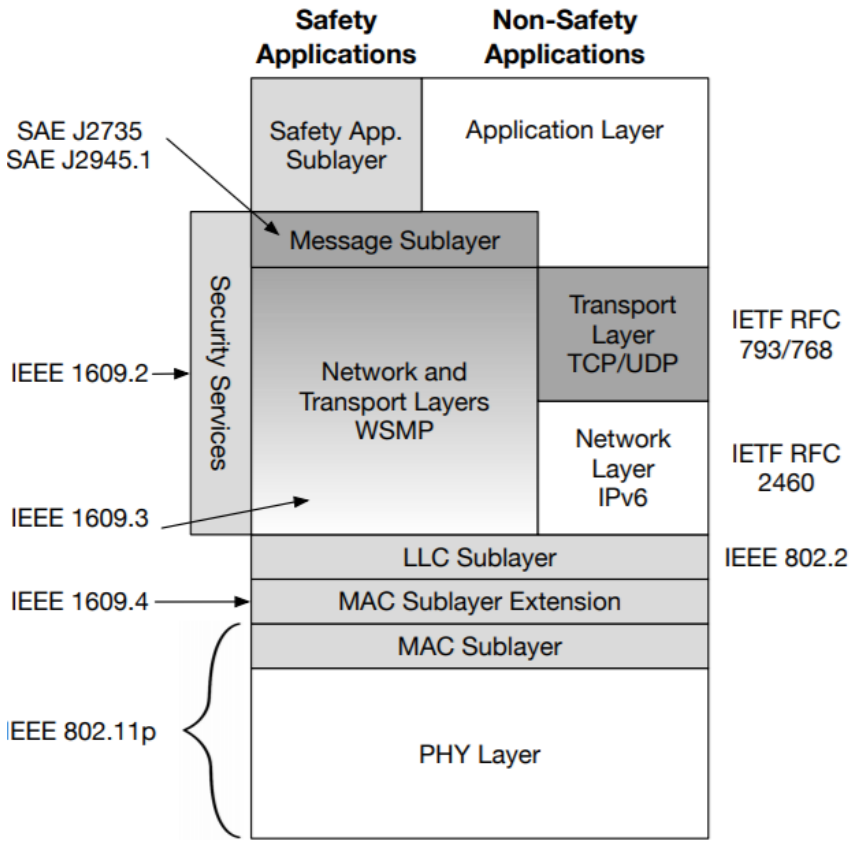


Figure 1.15: DSRC architecture [Kenney, 2011].

system.

Concerning the DSRC architecture presented in Fig 1.15, the institute of electrical and electronics engineers (IEEE) defines a series of standards for Wireless Access in Vehicular Environment (WAVE) that is considered as the most promising technology for vehicular networks [Ahmed et al., 2013]. This family of IEEE 1609 standards defines the architecture and the set of services and interfaces that enable secure wireless communication and physical access for high speed (up to 27 MB/s), short range (up to 1000 m), and low latency wireless communication in the vehicular environment [Abdelgader and Wu, 2014].

The WAVE IEEE 1609 standard series includes IEEE 1609.1 (application layer), IEEE 1609.2 (security layer), IEEE 1609.3 (network layer), IEEE 1609.4 (upper MAC Layer), and IEEE 802.11p (lower MAC and physical layers). The SAE J2735 standard defines message sets, data-frames, and data-elements used by applications to exchange data. Moreover, the SAE J2735 includes the following message categories: general, safety, geolocation, traveler information, and electronic payment.

1.3.3 Characteristics and Challenges

The network nodes in vehicular communication are self-organized and can self-manage information in a distributed fashion without a centralized authority or a server dictating the communication [Brendha and Prakash, 2017]. It means that nodes can act as servers and/or clients at the same time and exchange information with each other. Moreover, vehicular communications network has unique attractive features as follows [Nekovee, 2005]:

- ▷ **Higher transmission power and storage**: The network nodes (vehicles) are usually equipped with higher power and storage units.
- ▷ **Higher computational capability**: Operating vehicles can afford higher computing, communication and sensing capabilities.
- ▷ **Predictable mobility**: The movement of the network nodes can be predicted because they move on a road network. Roadway information is often available from positioning systems and map-based technologies such as GPS. Given the average speed, current speed, and road trajectory, the future position of a vehicle can be predicted.

On the other hand, vehicular communications suffer from several challenges that directly impact the quality of services provided by the network. These challenges [Blum et al., 2004] can be summarized as:

- ▷ **Large scale network**: vehicular network can in principle extend over the entire road network according to the environment, thus, it includes many participants.
- ▷ **High mobility**: The environment in which vehicular networks operate is extremely dynamic, and includes extreme configurations: in highways, relative speed of up to 200 km/h may occur.
- ▷ **High dynamic topology**: Due to high speed of movement between vehicles, the topology of vehicular networks is always changing. For example, assume that the wireless transmission range of each vehicle is 250 m, so that there is a link between two cars if the distance between them is less than 250 m. In the worst case, if two cars with the speed of 60 mph (25 m/sec) are driving in opposite directions, the link will last only for at most 10 sec.

- ▷ **Frequently disconnected networks**: The connectivity of the vehicular networks could also be changed frequently. Especially when the vehicle density is low, it has higher probability that the network is disconnected.
- ▷ **Various communications environments**: Vehicular networks are usually operated in two typical communications environments. In highway traffic scenarios, the environment is relatively simple and straightforward (one-dimensional movement), while in city conditions it becomes much more complex. The streets in a city are often separated by buildings, trees and other obstacles. Therefore, there is not always a direct line sight of communications in the direction of the intended data communication.

As a result of the multi-path fading and high mobility conditions in vehicular environments, both V2V and V2I communications suffer from a harsh signal propagation due to the following two reasons: (*i*) One or both communications nodes i.e. the transmitter and the receiver are in motion. As a result, the vehicular channel variation raises as the vehicle velocity increases, leading to a short channel coherence time. (*ii*) There are fixed and mobile scatterers that introduce significant channel multi-path components. Therefore, the vehicular channel becomes doubly selective in time and frequency, therefore its estimation is challenging.

In literature, there exist two channel estimation categories: (*i*) SBS estimation, where the channel is estimated for each received symbol separately. (ii) FBF estimation, where the receiver should wait for the whole frame reception before starting the channel estimation process. Moreover, the previous, current and future pilots are employed in the channel estimation for each received symbol. However, accurate conventional channel estimation requires transmitting a huge training symbols, thus the actual transmitted data rate becomes limited. On the other hand, DL techniques are considered to be a robust and efficient tools to employ in doubly selective channel estimation, while preserving low overall computational complexity. In this context, the thesis main objective lies in how we can improve the vehicular channel estimation accuracy in high mobility scenarios by taking into consideration the advantages of several DL techniques.

1.4 Deep Learning Based Channel Estimation

Recently, DL techniques have been integrated into wireless communications physical layer applications [Wang et al., 2017, O'Shea and Hoydis, 2017] including channel estimation [Yang et al., 2019, Ma et al., 2018, Ye et al., 2018], due to its great success in improving the overall system performance, especially when integrated with low-complexity conventional estimators. DL techniques are characterized by robustness, low-complexity, and good generalization ability making the integration of DL into communications systems beneficial. Motivated by these advantages, DL algorithms have been integrated

in vehicular channel estimation in two different manners: (*i*) DNN and LSTM networks with different architectures and configurations are integrated with SBS estimators, where the channel is estimated for each received data symbol directly. (*ii*) CNNs are integrated in the FBF estimators. The higher performance accuracy can be achieved by utilizing FBF estimators mainly in high mobility vehicular communications, since the channel estimation of each symbol takes advantage from the knowledge of previous, current, and future allocated pilots within the frame. Unlike, SBS estimators, where only the previous and current pilots are exploited in the channel estimation for each received symbol.

1.4.1 DL Techniques Overview

In this section, we will focus mainly on the DL techniques that will be used in this dissertation, where the mathematical representation of each technique is provided and discussed.

1.4.1.1 DNN

Neural networks are among the most popular machine learning algorithms [Schmidhuber, 2015a]. Initially, neural networks are inspired by the neural architecture of a human brain, and like in a human brain, the basic building block is called a neuron. Its functionality is similar to a human neuron, i.e. it takes in some inputs and fires an output. In purely mathematical terms, a neuron represents a placeholder for a mathematical function, and its only job is to provide an output by applying the function on the inputs provided. Neurons are stacked together to form a layer. The neural network consists at least of one layer, and when multiple layers are used, the neural network is called DNN.

Consider a DNN that consists of L layers, including one input layer, $L-2$ hidden layers, and one output layer as shown in Fig. 1.16. The l-th hidden layer of the network consists of J neurons where $2 \leq l \leq L-1$, and $1 \leq j \leq J$. The DNN inputs $\boldsymbol{i}$ and outputs $\boldsymbol{o}$ are expressed as $\boldsymbol{i} = [\boldsymbol{i}_1, \boldsymbol{i}_2, ..., \boldsymbol{i}_N] \in \mathbb{R}^{N\times 1}$ and $\boldsymbol{o} = [\boldsymbol{o}_1, \boldsymbol{o}_2, ..., \boldsymbol{o}_M] \in \mathbb{R}^{M\times 1}$, where N and M denote the number of DNN inputs and outputs respectively. $\boldsymbol{W}_l \in \mathbb{R}^{J_{l-1}\times J_l}$, and $\boldsymbol{b}_l \in \mathbb{R}^{J_l\times 1}$ are used to denote the weight matrix and the bias vector of the l-th hidden layer respectively.

Each neuron $n_{(l,j)}$ performs a nonlinear transform of a weighted summation of output values of the preceding layer. This nonlinear transformation is represented by the activation function $f_{(l,j)}$ on the neuron's input vector $\boldsymbol{i}_{(l)} \in \mathbb{R}^{J_{l-1}\times 1}$ using its weight vector $\boldsymbol{\omega}_{(l,j)} \in \mathbb{R}^{J_{l-1}\times 1}$, and bias $b_{(l,j)}$ respectively. The neuron's output $o_{(l,j)}$ is:

$$o_{(l,j)} = f_{(l,j)}\Big(b_{(l,j)} + \boldsymbol{\omega}_{(l,j)}^T \boldsymbol{i}_{(l)}\Big). \tag{1.24}$$

The DNN over all output of the l-th hidden layer is represented by the vector form:

$$\boldsymbol{o}_{(l)} = \boldsymbol{f}_{(l)}\Big(\boldsymbol{b}_{(l)} + \boldsymbol{W}_{(l)}\boldsymbol{i}_{(l)}\Big),\ \boldsymbol{i}_{(l+1)} = \boldsymbol{o}_{(l)}, \tag{1.25}$$

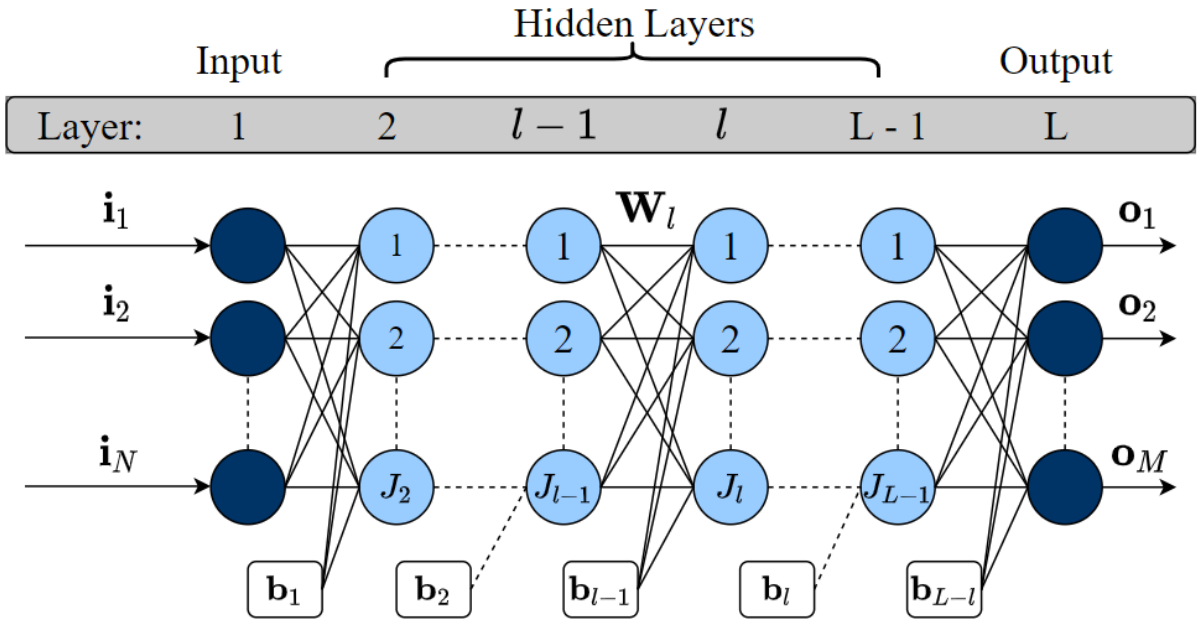


Figure 1.16: DNN architecture.

where $\boldsymbol{f}_{(l)}$ is a vector that results from the stacking of the n_l activation functions.

Once the DNN architecture has been chosen, the parameter $\theta = (\boldsymbol{W}, \boldsymbol{B})$ that represents the total DNN weights and biases have to be estimated through the learning procedure applied during the DNN training phase. As well known, θ estimation is obtained by minimizing a loss function $\text{Loss}(\theta)$ (1.26). A loss function measures how far apart the predicted DNN outputs ($\boldsymbol{o}_{(L)}^{(\text{P})}$) from the true outputs ($\boldsymbol{o}_{(L)}^{(\text{T})}$). Therefore, DNN training phase carried over N_{train} training samples can be described in two steps: (i) calculate the loss, and (ii) update θ. This process will be repeated until convergence, so that the loss becomes very small.

$$\text{Loss}(\theta) = \arg\min_{\theta}(\boldsymbol{o}_{(L)}^{(\text{P})} - \boldsymbol{o}_{(L)}^{(\text{T})}). \tag{1.26}$$

Various optimization algorithms can be used to minimize $\text{Loss}(\theta)$ by iteratively updating the parameter θ, i.e., stochastic gradient descent [Schmidhuber, 2015b], root mean square prop [ichi Amari, 1993], adaptive moment estimation (ADAM) [De et al., 2018]. DNN optimizers updates θ according to the magnitude of the loss derivative with respect to it as follows:

$$\theta_{\text{new}} = \theta - \rho \frac{\partial \text{Loss}(\theta)}{\partial \theta}, \tag{1.27}$$

where ρ represents the learning rate of the DNN, which controls how quickly θ is updated. Smaller learning rates require more training, given the smaller changes made to θ in each update, whereas larger learning rates result in rapid changes and require less training. The final step after DNN training, is to test the trained DNN on new data so that its performance is evaluated. A detailed comprehensive analysis of DNN different principles is presented in [Ruder, 2017].

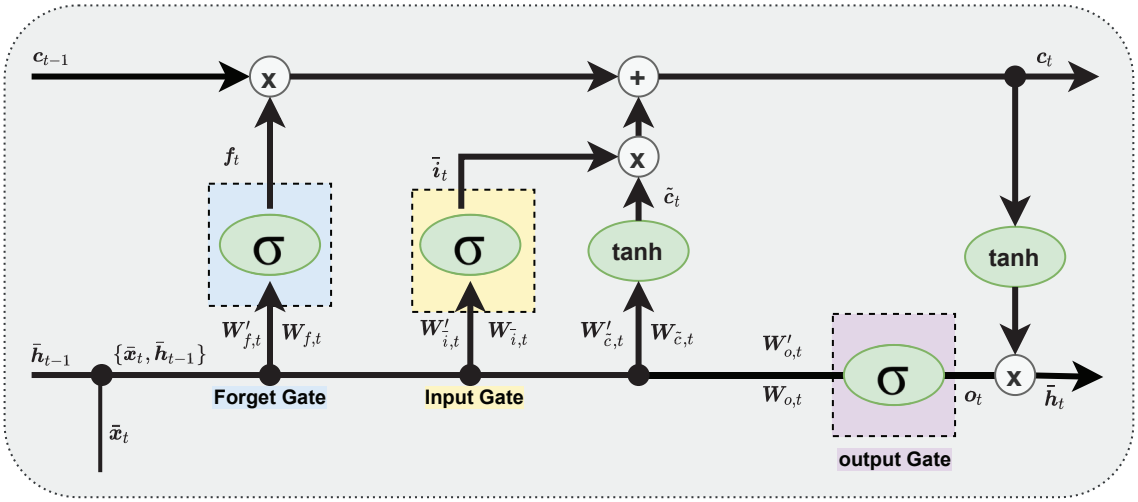


Figure 1.17: LSTM unit architecture.

1.4.1.2 LSTM

Another well known DL tools are the LSTM networks, that are basically proposed to deal with sequential data where the order of the data matters and there exists a correlation between the previous and the future data. In this context, LSTM networks are defined with a special architecture that can learn the data correlation over time, thus giving the LSTM network the ability to predict the future data based on the previous observations.

LSTM unit contains computational blocks known as gates which are responsible for controlling and tracking the information flow over time. The LSTM network mechanism can be explained in four major steps as follow:

1.4.1.2.1 Forget the irrelevant information

In general, the LSTM unit classify the input data into relevant and irrelevant information. The first processing step is to eliminate the irrelevant information that are not important for the future data prediction. This can be performed through the forget gate that decide which information the LSTM unit should keep, and which information it should delete. The forget gate processing is defined as below

$$\boldsymbol{f}_t = \sigma(\boldsymbol{W}_{f,t}\bar{\boldsymbol{x}}_t + \boldsymbol{W}'_{f,t}\bar{\boldsymbol{h}}_{t-1} + \bar{\boldsymbol{b}}_{f,t}), \tag{1.28}$$

where $\bar{\sigma}$ is the sigmoid function, $\boldsymbol{W}_{f,t} \in \mathbb{R}^{P\times K_{in}}$, $\boldsymbol{W}'_{f,t} \in \mathbb{R}^{P\times P}$ and $\bar{\boldsymbol{b}}_{f,t} \in \mathbb{R}^{P\times 1}$ are the forget gate weights and biases at time t, $\bar{\boldsymbol{x}}_t \in \mathbb{R}^{K_{in}\times 1}$ and $\bar{\boldsymbol{h}}_{t-1}$ denote the LSTM unit input vector of size K_{in}, and the previous hidden state of size P respectively.

1.4.1.2.2 Store the relevant new information

After classifying the relevant information, the LSTM unit applies some computations on the selected information through the input gate

$$\bar{\boldsymbol{i}}_t = \sigma(\boldsymbol{W}_{\bar{i},t}\bar{\boldsymbol{x}}_t + \boldsymbol{W}'_{\bar{i},t}\bar{\boldsymbol{h}}_{t-1} + \bar{\boldsymbol{b}}_{\bar{i},t}), \tag{1.29}$$

$$\tilde{\boldsymbol{c}}_t = \tanh(\boldsymbol{W}_{\tilde{c},t}\bar{\boldsymbol{x}}_t + \boldsymbol{W}'_{\tilde{c},t}\bar{\boldsymbol{h}}_{t-1} + \bar{\boldsymbol{b}}_{\tilde{c},t}). \tag{1.30}$$

1.4.1.2.3 Update the new cell state

Now, the LSTM unit should update the current cell state $\boldsymbol{c}_t$ based on the two previous steps such that

$$\boldsymbol{c}_t = \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1} + \bar{\boldsymbol{i}}_t \odot \tilde{\boldsymbol{c}}_t. \tag{1.31}$$

where $\odot$ denotes the Hadamard product.

1.4.1.2.4 Generate the LSTM unit output

The final processing step is to update the hidden state and generate the output by the output gate. The output is considered as a cell state filtered version and can be computed such that

$$\boldsymbol{o}_t = \sigma(\boldsymbol{W}_{o,t}\bar{\boldsymbol{x}}_t + \boldsymbol{W}'_{o,t}\bar{\boldsymbol{h}}_{t-1} + \bar{\boldsymbol{b}}_{o,t}), \tag{1.32}$$

$$\bar{\boldsymbol{h}}_t = \boldsymbol{o}_t \odot \tanh \boldsymbol{c}_t. \tag{1.33}$$

We note that, there exists in literature several LSTM architecture variants, where the interactions between the LSTM unit gates are modified. The authors in [Greff et al., 2017] performed a nice comparison of popular LSTM architecture variants.

1.4.1.3 CNN

CNN is a type of deep learning model for processing data that has a grid pattern, such as images [Albawi et al., 2017]. Thus, CNN has generally become the state of the art for many visual applications such as image classification, due to its great ability in extracting patterns from the input image. CNN can be seen as a set of several layers stacked together in order to accomplish the required task. These layers include:

- ▷ **Input layer**: It represents the 2D or 3D input image. For simplicity, let us consider a 2D image input to the l -th CNN layer denoted by $\boldsymbol{X}_l \in \mathbb{R}^{h_l \times w_l}$, where h_l and w_l are the height and the width of the $\boldsymbol{X}_l$ input image.

▷ **Convolutional layer**: is a specialized type of linear operation used for feature extraction, where predefined filters called kernels scan the input matrix to fill the output matrix denoted as feature map as illustrated in Fig. 1.18. We note that different kernels can be considered as different feature extractors.
Two key hyper parameters that define the CNN convolutional layer are size and number of kernels denoted by f_l and n_l respectively. The typical kernel size is 3×3, but sometimes 5×5 or 7×7. The latter is arbitrary, and determines the depth of output feature maps. These parameters can be tuned according to the application type. Moreover, the process of training a CNN model with regard to the convolution layer is to identify the kernels values that work best for a given task based on a given training dataset. Kernels are the only parameters automatically learned during the training process in the convolution layer. Mathematically speaking, for a given input image $\boldsymbol{X}_l$ and kernel $\boldsymbol{K_l} \in \mathbb{R}^{f_l \times f_l \times 1}$, here we consider one kernel for simplicity, then, the generated feature map $\boldsymbol{Y}_l \in \mathbb{R}^{(h_l - f + 1) \times (w_l - f + 1)}$ can be expressed as follows:

$$\boldsymbol{Y}_l(x, y) = \sum_{i=1}^{h_l} \sum_{j=1}^{w_l} \boldsymbol{K}_l(i, j) \boldsymbol{X}_l(x + i - 1, y + j - 1). \tag{1.34}$$

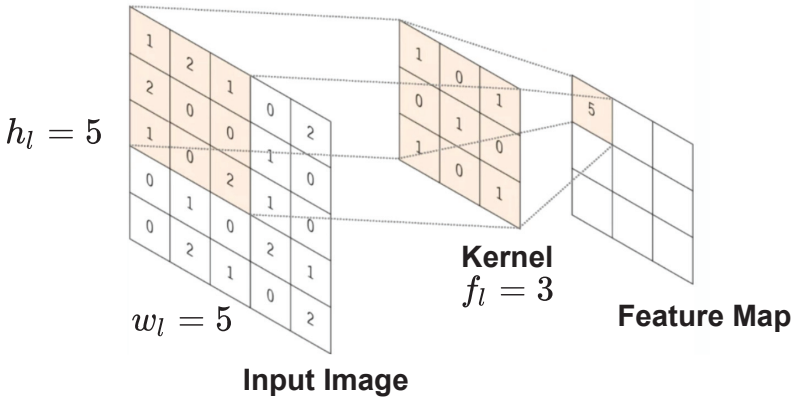


Figure 1.18: CNN convolutional layer example.

▷ **Activation layer**: The outputs of a linear operation such as convolution are then passed through a nonlinear activation function. The role of the activation function is to introduce non-linear processing to the CNN architecture since the input-output CNN pairs relation might be non-linear. There are several non-linear activation functions such as sigmoid or hyperbolic tangent (tanh) function, but the most common function used presently is the rectified linear unit (ReLU).

▷ **Pooling layer**: This layer is employed to reduce the number of parameters when the images are too large. Pooling operation is also called sub-sampling or down-sampling which reduces the dimensionality of each feature map but retains important information. It is of note that there is no learnable parameter in any of the pooling layers. The most popular form of pooling operation is max pooling, which extracts patches from the input feature maps, outputs the maximum value

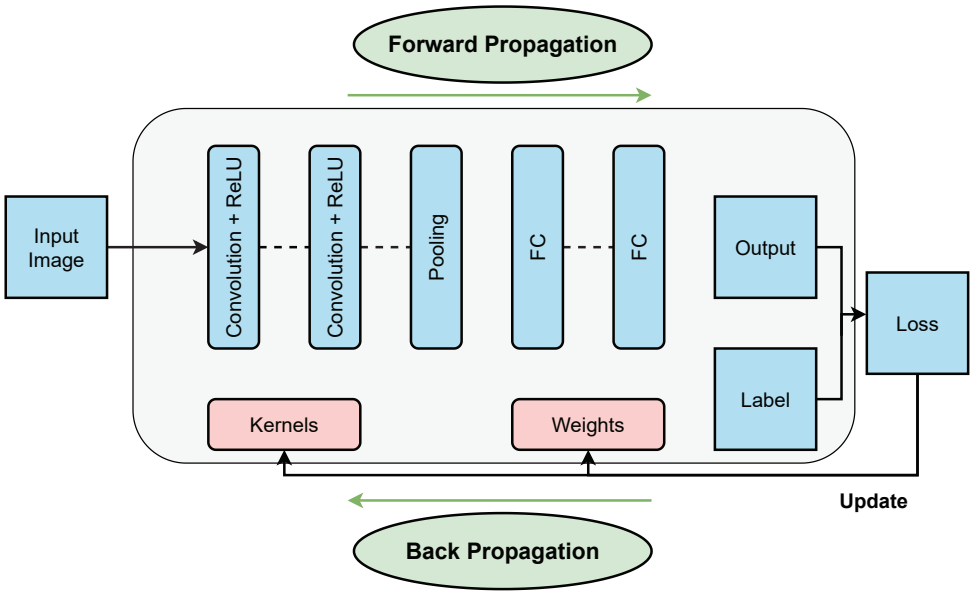


Figure 1.19: CNN classical architecture.

in each patch, and discards all the other values. However, there exists other pooling operations such as global average pooling [Sun et al., 2017].

- ▷ **Fully connected layer**: This layer forms the last block of the CNN architecture, and it is employed mainly in classification problems. It is a simple feed forward neural network layer, consisting of one or more hidden layers, its role is to transform the 2D CNN layer output into a 1D vector. In classification problems, the final outputs of the CNN network represent the probabilities for each class, where the final fully connected layer typically has the same number of output nodes as the number of classes.
- ▷ **Batch normalization**: It is used to make the CNN output more stable by normalizing the output of each layer. Moreover, batch normalisation layer reduces over fitting and speeds up the CNN training.
- ▷ **Output layer**: This layer is configured according to the studied problem, for example, in classification problems the CNN output layer is a fully connected layer with softmax activation function, while in regression problems, the CNN output does not use any activation function.

Figure 1.19 illustrates the classical CNN architecture. As we can note in the figure, the only trainable parameters within the CNN network are the kernels and the fully connected layer weights. We note that, similarly to all other DL techniques, CNN network updates its trainable parameters by minimizing the CNN loss function that measures how far are the inputs from the outputs, this operation is called . After that, the CNN kernels and weights are updated in the back propagation operation [Qi, 2016]. Finally, the performance of the trained CNN model is evaluated in the testing phase where new unobserved images are fed to the trained CNN model.

Chapter 2

IEEE 802.11p System Model

Contents

This chapter provides the system model employed in this thesis, where the physical layer characteristics including the considered communication standard and transceiver design are presented. After that, the signal model followed by the vehicular channel models that are used in the simulation results of Chapters 3 , 4, 5, and 6 are discussed.

2.1 Background

2.1.1 Physical Layer Parameters

As discussed in Chapter 1, vehicular communications are managed by the IEEE 802.11p standard which is an approved upgrade of the IEEE 802.11a standard that adds wireless access in vehicular environments. This upgrade includes data exchange between high-speed vehicles (V2V) and between the vehicles and the roadside infrastructure (V2I) in the licensed intelligent transportation systems band [Weidong et al., 2009]. IEEE 802.11p

supports sending data at different data rates employing different modulation orders. Moreover, it employs OFDM transmission scheme with $K = 64$ total subcarriers. $K_{\text{on}} = 52$ active subcarriers are used, and they are divided into $K_{\text{d}} = 48$ data subcarriers and $K_{\text{p}} = 4$ pilot subcarriers. The remaining $K_{\text{n}} = 12$ subcarriers are used as a guard band. Compared to the IEEE 802.11a standard, the channel bandwidth is halved, resulting in a 10 MHz bandwidth instead of 20 MHz in 802.11a. Accordingly, all OFDM duration parameters used in the regular 802.11a standard are doubled. In particular, the CP size increases, which reduces the ISI caused by multi-path propagation. Table 2.1 lists the IEEE 802.11p physical layer main specifications. A detailed discussion of the IEEE 802.p standard and all its features is presented in [Abdelgader and Wu, 2014].

Table 2.1: IEEE 802.11p physical layer specifications.

Parameter	**IEEE 802.11p**
Bandwidth	10 MHz
CP duration	1.6 μs
CP-OFDM Symbol duration	8 μs
short training symbols (STS) duration	1.6 μs
long training symbols (LTS) duration	6.4 μs
Overall STS duration	16 μs
Overall LTS duration	16 μs
Total subcarriers	64
Pilot subcarriers	4
Data subcarriers	48
Null subcarriers	12
$\mathcal{K}_{\text{p}}$	$\{\pm 7, \pm 21\}$
$\mathcal{K}_{\text{n}}$	$\{0, \pm 27, \pm 28, \pm 29, \pm 30, \pm 31, -32\}$
$\mathcal{K}_{\text{d}}$	$\mathcal{K}_{on} \setminus \mathcal{K}_{\text{p}} \cup \mathcal{K}_{\text{n}}$
Subcarrier spacing	156.25 KHz

2.1.2 Baseband Transceiver

As shown in Fig. 2.1, the first operation on the transmitter side is the binary bits generation. The number of generated bits $N_{\mathbf{b}}$ depends on coding rate, the utilized modulation scheme, as well as on the number of transmitted OFDM data symbols, defined as:

$$N_b = K_d I R N_c, \tag{2.1}$$

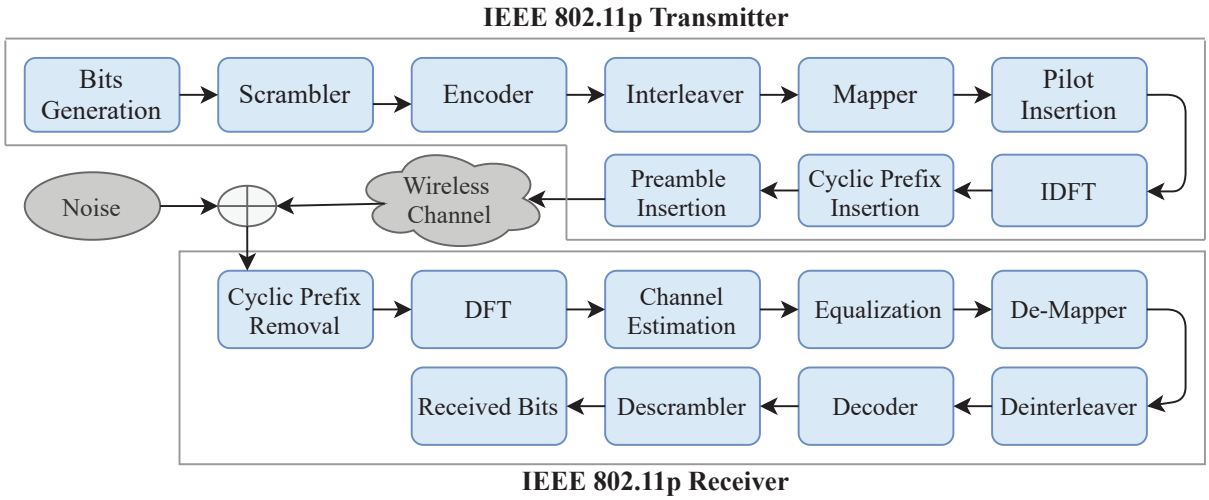


Figure 2.1: IEEE802.11p transmitter-receiver block diagram.

where, I denotes the number of transmitted OFDM symbols per frame, R is the coding rate, and N_c refers to number of bits corresponding to one constellation point.

Generated bits are scrambled in order to randomize the bits pattern, which may contain long streams of 1s or 0s. Bits scrambling [Carona et al., 2010] facilitates the work of a timing recovery circuit and eliminates the dependence of signal's power spectrum upon the actual transmitted data. The scrambled bits are then passed to a convolutional encoder [Grami, 2016] with code rate $R = 1/2$, which introduces some redundancy into the bits stream. This redundancy is used for error correction that allows the receiver to combat the effects of the channel, hence reliable communications can be achieved. On the other hand, IEEE 802.11p standard defines rates 3/4 and 2/3 that can be achieved by puncturing. Puncturing is a procedure that derives codes of different rate from the initial code $R = 1/2$, by periodically deleting certain coded bits in the coded sequence of the initial code. The bits are deleted according to the given puncture vectors. Puncture vector is a pattern of 1s and 0s, with 0 indicating stolen bits, which is deleted from the initial code. As a next step, bits interleaving [Shi et al., 2004] is used to cope with the channel noise such as burst errors or fading. The interleaver rearranges the encoded input bits such that consecutive bits are split among different blocks. This can be done using a permutation process that ensures that adjacent bits are modulated onto non-adjacent subcarriers and thus allows better error correction at the receiver.

After that, the interleaved bits are mapped according to the used modulation technique [Sobolewski, 2003]. IEEE 802.11p standard defines four modulation techniques: BPSK, QPSK, 16QAM and 64QAM. Moreover, Gray code is used, where the mapped symbol corresponding to neighboring symbols differs by exactly one bit. Bits mapping operation is followed by constructing the OFDM symbols to be transmitted. The data symbols and pilots are mapped to the active subcarriers and passed to the IDFT block to generate the time-domain OFDM symbols and followed by inserting the CP. Finally, the

Table 2.2: IEEE 802.11p modulation orders and data rates.

Modulation orders	**BPSK**		**QPSK**		**16QAM**		**64QAM**	
Coding rate	$\frac{1}{2}$	$\frac{3}{4}$	$\frac{1}{2}$	$\frac{3}{4}$	$\frac{1}{2}$	$\frac{3}{4}$	$\frac{2}{3}$	$\frac{3}{4}$
Data rate (Mbps)	3	4.5	6	9	12	18	24	27
Data bits per OFDM symbol	24	36	48	72	96	144	192	216

IEEE 802.11p packet is formed by concatenating the constructed CP-OFDM symbols, and the predefined preamble symbols in one frame.

At the receiver side, the CP is removed followed by DFT. Then, initial channel estimation is applied using the preamble and the pilot subcarriers. After that, the received OFDM data symbols are forwarded to the equalizer. The equalized data are de-mapped to obtain the encoded bits. The process of de-mapping can be as simple as hard demapping, or more sophisticated as soft demapping. Hard de-mapping [Dany et al., 2003] is based on the minimum Euclidean distance between the received symbols and all the valid symbols of the constellation. However, in order to decrease the probability of residual decoding errors, it is advantageous to use soft de-mapping [Hassan et al., 2018] instead of hard de-mapping. Afterwards, deinterleaving, decoding using the Viterbi algorithm [Yamada, 1990], and descrambling are performed to obtain the detected bits. Table. 2.2 shows different modulation orders and data rates defined by the IEEE 802.11p standard.

2.1.3 Frame Structure

After preparing the time-domain OFDM data symbols to be transmitted, the complete IEEE 802.11p frame is constructed. It consists mainly of three parts: (*i*) preamble, (*ii*) signal field, and (*iii*) OFDM data symbols. The preamble includes ten STS t_1 to t_{10}, each of duration 1.6 μs. The STS are used at the receiver for signal detection, diversity selection, coarse frequency offset estimation and timing synchronization. The following two 6.4 μs LTS are prepended with a CP of duration 3.2 μs, and used for channel estimation at the receiver. After that, the signal field (SF) is used to specify rate and length information that are required for decoding the received OFDM data symbols. It consists of one CP-OFDM symbol that is BPSK modulated at 6 Mbps and is encoded at a $R = 1/2$ rate. We note that, the signal field carries 24 bits divided into four sub fields described in Table. 2.3.

Finally, the OFDM data symbols that carry the actually transmitted data bits are inserted. Fig. 2.2 illustrates the IEEE 802.11p frame structure in the time domain.

Table 2.3: Signal field specifications.

Sub field	Bits	Function
Rate	4	Define the employed modulation and coding rate
Length	12	Indicate the number of data octets requested in the transmission.
Reserved	2	Reserved for future use.
Tail	6	Return the convolutional encoder to the zero state.

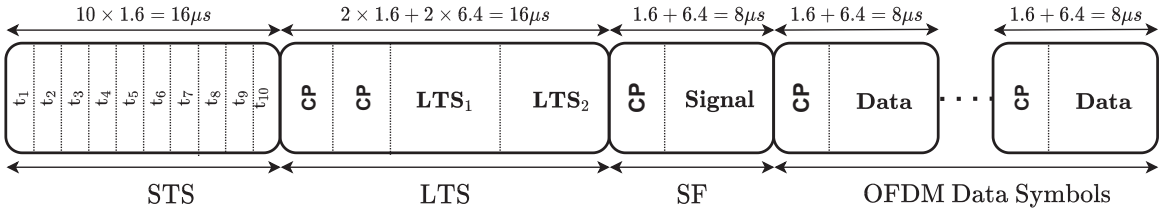


Figure 2.2: IEEE 802.11p transmitted frame structure in time domain.

2.2 Signal Model

In this section, the signal model used in this thesis is defined. Since our focus is on channel estimation, we assume perfect synchronization at the receiver, and we ignore the signal field for simplicity. Therefore, we focus on a frame that consists of two LTS at the beginning followed by I OFDM data symbols as shown in Fig. 2.3.

2.2.1 Transmitted Signal

The i-th transmitted frequency-domain OFDM symbol $\tilde{\boldsymbol{x}}_i[k]$, is partitioned as:

$$\tilde{\boldsymbol{x}}_i[k] = \begin{cases} \tilde{\boldsymbol{x}}_{\mathrm{d}_i}[k], & k \in \mathcal{K}_{\mathrm{d}}. \\ \tilde{\boldsymbol{x}}_{\mathrm{p}_i}[k], & k \in \mathcal{K}_{\mathrm{p}}. \\ 0, & k \in \mathcal{K}_{\mathrm{n}}, \end{cases} \tag{2.2}$$

where $0 \leq k \leq K-1$. $\tilde{\boldsymbol{x}}_{\mathrm{d}_i}[k]$ and $\tilde{\boldsymbol{x}}_{\mathrm{p}_i}[k]$ denote the modulated data symbols and the predefined pilot symbols allocated at $\mathcal{K}_{\mathrm{d}}$ and $\mathcal{K}_{\mathrm{p}}$, respectively. The other guard band subcarriers are allocated at $\mathcal{K}_{\mathrm{n}}$. $\boldsymbol{x}_i[k]$ is converted to the time domain by applying the inverse discrete Fourier transform, such that:

$$\boldsymbol{x}_i[n] = \frac{1}{\sqrt{K}} \sum_{k=0}^{K-1} \tilde{\boldsymbol{x}}_i[k] e^{j2\pi \frac{nk}{K}}. \tag{2.3}$$

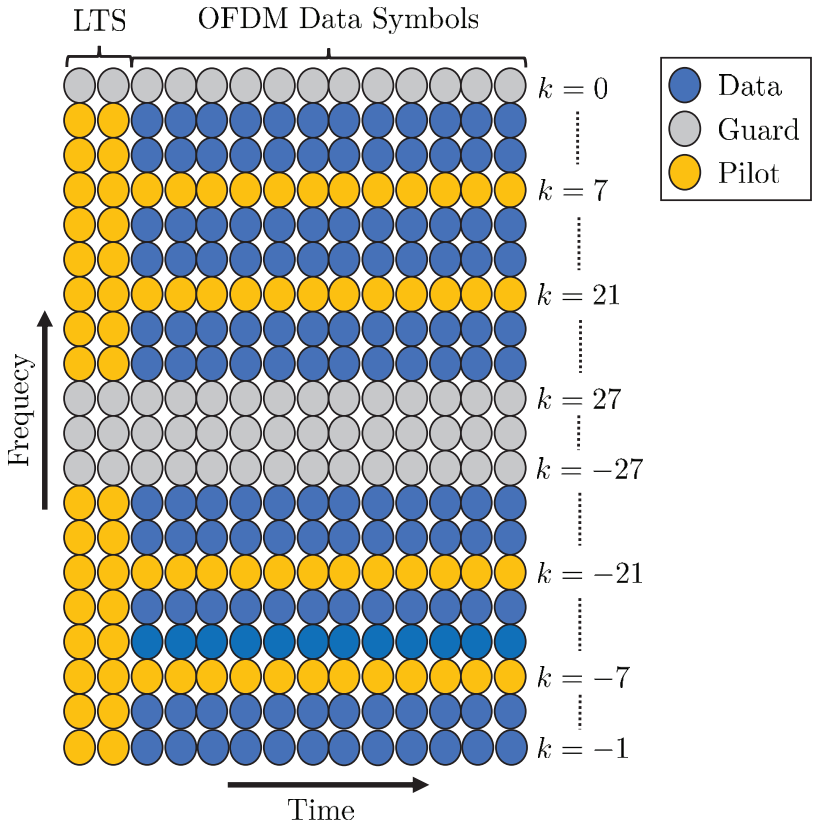


Figure 2.3: IEEE 802.11p transmitted frame structure in frequency domain.

2.2.2 Received Signal

After passing through the doubly-selective vehicular channel, the received OFDM symbol $\boldsymbol{y}_i[n]$ can be expressed as:

$$\begin{aligned} \boldsymbol{y}_i[n] &= \sum_{l=0}^{L-1} \boldsymbol{h}_i[l,n]\boldsymbol{x}_i[n-l] + \boldsymbol{v}_i[n] \\ &= \frac{1}{\sqrt{K}} \sum_{k=0}^{K-1} \tilde{\boldsymbol{h}}_i[k,n]\tilde{\boldsymbol{x}}_i[k]e^{j2\pi\frac{nk}{K}} + \boldsymbol{v}_i[n], \end{aligned} \tag{2.4}$$

where $\boldsymbol{h}_i[l,n]$ denotes the delay-time response of the discrete LTV channel of L taps at the i-th OFDM symbol, whereas $\tilde{\boldsymbol{h}}_i[k,n] = \sum_{l=0}^{L-1} \boldsymbol{h}_i[l,n]e^{-j2\pi\frac{lk}{K}}$ is the frequency-time response. Moreover, $\boldsymbol{v}_i$ denotes the additive white Gaussian noise (AWGN) of variance σ^2. The i-th received frequency-domain OFDM symbol is obtained from (2.4) by means of DFT, and thus:

$$\tilde{\boldsymbol{y}}_i[k] = \frac{1}{K} \sum_{q=0}^{K-1} \tilde{\boldsymbol{x}}_i[q] \sum_{n=0}^{K-1} \tilde{\boldsymbol{h}}_i[q,n]e^{-j2\pi\frac{n(k-q)}{K}} + \tilde{\boldsymbol{v}}_i[k]. \tag{2.5}$$

The time selectivity of the channel depends on the mobility. In very low mobility, where $f_\mathrm{d} \approx 0$, $\tilde{\boldsymbol{h}}_i[q,n] = \tilde{\boldsymbol{h}}[q]$ is constant during the whole frame. For moderate to high mobility,

the channel variation within the duration of one OFDM symbol is negligible, and therefore, $\tilde{\boldsymbol{h}}_i[q,n] = \tilde{\boldsymbol{h}}_i[q]$. At very high mobility, the channel becomes variant within a single OFDM symbol. In this case, $\tilde{\boldsymbol{h}}_i[q,n] = \tilde{\boldsymbol{h}}_i[q] + \tilde{\boldsymbol{\epsilon}}_i[q,n]$, where

$$\tilde{\boldsymbol{h}}_i[q] = \frac{1}{K}\sum_{n=0}^{K-1}\tilde{\boldsymbol{h}}_i[q,n], \text{ and } \tilde{\boldsymbol{\epsilon}}_i[q,n] = \tilde{\boldsymbol{h}}_i[q,n] - \tilde{\boldsymbol{h}}_i[q]. \tag{2.6}$$

Replacing this in (2.5), we get

$$\tilde{\boldsymbol{y}}_i[k] = \tilde{\boldsymbol{h}}_i[k]\tilde{\boldsymbol{x}}_i[k] + \tilde{\boldsymbol{e}}_{i,\mathrm{d}}[k] + \tilde{\boldsymbol{v}}_i[k], \ k \in \mathcal{K}_{on}. \tag{2.7}$$

The term $\tilde{\boldsymbol{e}}_{\mathrm{d}i}[k]$ represents the Doppler interference given by:

$$\begin{aligned}\tilde{\boldsymbol{e}}_{\mathrm{d}i}[k] &= \frac{1}{K}\sum_{\substack{q=0\\q\neq k}}^{K-1}\sum_{n=0}^{K-1}\tilde{\boldsymbol{h}}_i[q,n]e^{-j2\pi\frac{n(k-q)}{K}}\tilde{\boldsymbol{x}}_i[q] \\ &= \frac{1}{K}\sum_{\substack{q\in\mathcal{K}_{on}\\q\neq k}}\sum_{l=0}^{L-1}\bar{\boldsymbol{h}}_i[l,k-q]e^{-j2\pi\frac{lq}{K}}\tilde{\boldsymbol{x}}_i[q].\end{aligned} \tag{2.8}$$

The Doppler interference destroys the orthogonality of the subcarriers within the received OFDM symbol, resulting in a significant degradation in the overall system performance [Matz and Hlawatsch, 2011]. Assuming the subcarriers are uncorrelated with power E_q, i.e. $\mathrm{E}\left[\tilde{\boldsymbol{x}}_i[q]\tilde{\boldsymbol{x}}_i^*[q']\right] = E_q\delta[q-q']$ and using (2.14) then:

$$\begin{aligned}\mathrm{E}\left[\tilde{\boldsymbol{e}}_{i,\mathrm{d}}[k]\tilde{\boldsymbol{e}}_{i,\mathrm{d}}^*[k']\right] &= \sum_{l=0}^{L-1}\sum_{\substack{q\in\mathcal{K}_{on}\\q\neq k}}E_q\rho[l,k-q]\delta[k-k'] \\ &= \sigma_d^2[k]\delta[k-k'].\end{aligned} \tag{2.9}$$

Thus, the Doppler interference is assumed uncorrelated. However, variance $\sigma_d^2[k] = \mathrm{E}\left[|\tilde{\boldsymbol{e}}_{\mathrm{d}i}[k]|^2\right]$ depends on the subcarrier index. Noting that:

$$\tilde{\boldsymbol{h}}_i[k] = \frac{1}{K}\sum_{l=0}^{L-1}\bar{\boldsymbol{h}}_i[l,0]e^{-j2\pi\frac{kl}{K}}, \tag{2.10}$$

then, the channel gain and Doppler interference are uncorrelated, i.e. $\mathrm{E}\left[\tilde{\boldsymbol{h}}_i[k]\tilde{\boldsymbol{e}}_{i,\mathrm{d}}^*[k]\right] = 0$. Moreover, $\tilde{\boldsymbol{h}}_i[k]$ can be estimated from L uncorrelated tapes defined by $\bar{\boldsymbol{h}}_i[l,0]$.

2.2.3 Channel Model Characteristics

Note that $\tilde{\boldsymbol{h}}_i[q,n]$ is time-variant at the scale of the OFDM symbol duration and within the symbol itself. Accordingly:

$$\tilde{\boldsymbol{h}}_i[q,n] = \sum_{l=0}^{L-1}e^{-j2\pi\frac{lq}{K}}\int_{\nu=-\nu_d}^{\nu=\nu_d}\bar{h}(l,\nu)e^{j2\pi\nu n_i}e^{j2\pi\nu n}d\nu, \tag{2.11}$$

where $\bar{h}(l,\nu)$ is the channel delay-Doppler response, ν denotes the normalized Doppler frequency, $n_i = i(K+K_{\text{cp}})+K_{\text{cp}}$. and $\nu_d = \frac{f_d}{F_s}$ is the maximum Doppler frequency. Let:

$$\begin{aligned}\bar{\boldsymbol{h}}_i[l,v] &= \frac{1}{K}\sum_{q=0}^{K-1}\sum_{n=0}^{K-1}\tilde{\boldsymbol{h}}_i[q,n]e^{-j2\pi\frac{nv}{K}}e^{j2\pi\frac{ql}{K}} \\ &= \int_{\nu=-\nu_d}^{\nu=\nu_d}\bar{h}(l,\nu)e^{j2\pi\nu n_i}\sum_{n=0}^{K-1}e^{-j2\pi(\nu-\frac{v}{K})n}d\nu,\end{aligned} \tag{2.12}$$

be the discrete delay-Doppler response at the i-th OFDM symbol. Assuming $\bar{h}(l,\nu)$ is uncorrelated in both domains, such that $\mathrm{E}\left[\bar{h}(l,\nu)\bar{h}^*(l',\nu')=S_h(l,\nu)\delta[l-l']\delta(\nu-\nu')\right]$, where $S_h(l,\nu)$ is the delay-Doppler spectrum, then:

$$\begin{aligned}&\mathrm{E}\left[\bar{\boldsymbol{h}}_i[l,v]\bar{\boldsymbol{h}}_i^*[l,v']\right] = \\ &\int_{\nu=-\nu_d}^{\nu=\nu_d}S_h(l,\nu)\sum_{n=0}^{K-1}\sum_{n'=0}^{K-1}e^{-j2\pi\nu(n-n')}e^{-j2\pi\frac{n'v'-nv}{K}}d\nu.\end{aligned} \tag{2.13}$$

This correlation is independent of the index i, and it can be approximated as:

$$\mathrm{E}\left[\bar{\boldsymbol{h}}_i[l,v]\bar{\boldsymbol{h}}_i^*[l,v']\right] \approx K^2\rho[l,v]\delta[v-v'],\ \rho[l,v]=S_h(l,\frac{v}{N}). \tag{2.14}$$

2.3 IEEE 802.11p Vehicular Channel Models

As explained in Section 1.1, the wireless channel in vehicular environments is considered a time-varying channel including multi-path propagation and large Doppler shift [Mecklenbrauker et al., 2011]. Various studies that analyze the statistical characteristics of the wireless channel in vehicular environments are presented in [Wang et al., 2009], [Karedal et al., 2009], and [Viriyasitavat et al., 2015]. In this thesis, we consider the tapped delay line (TDL) vehicular channel models proposed in [Acosta-Marum and Ingram, 2007a]. These TDL models include six vehicular channel models for different vehicular environments. They are obtained by a measurement campaign that was implemented in metropolitan Atlanta. Table 2.4 provides the different characteristics of these vehicular channel models that are summarized below:

- ▷ **VTV Expressway (VTV-EX)**: the measurements here are performed between two vehicles entering the highway at the same time, then they are accelerated to reach 104 km/h. In this scenario there is no wall separating the two highway sides.
- ▷ **VTV Urban Canyon (VTV-UC)**: this scenario has been measured in Edge wood Avenue in Downtown Atlanta, where urban canyon characteristics exist. The vehicles move at 32-48 km/hr velocity range in a dense traffic environment.
- ▷ **VTV Expressway Same Direction with Wall (VTV-SDWW)**: The communication between the two vehicles is established on a highway having center wall between its lanes. The separation distance between both vehicles is 300–400 m, and vehicles speed was 104 km/hr.

Table 2.4: Vehicular channel models characteristics following Jake's Doppler spectrum.

Channel model	**Channel taps**	**Vehicle velocity [kmph]**	**Max Doppler shift [Hz]**	**Average path gains [dB]**	**Path delays [ns]**
RTV-SS	12	32-48	500	[0, 0, -9.3, -9.3, -14, -14, -18, -18, -19.4, -24.9, -27.5, -29.8]	[0, 1, 100, 101, 200, 201, 300,301, 400, 500, 600, 700]
RTV-EX	12	104	700	[0, 0, 0, -9.3, -9.3, -9.3, -20.3, -20.3, -21.3, -21.3, -28.8, -28.8]	[0, 1, 2, 100, 101, 102, 200, 201, 300, 301, 400, 401]
RTV-UC	12	32-48	300	[0, 0, 0, -11.5, -11.5, -11.5, -19, -19, -25.6, -25.6, -28.1, -28.1]	[0, 1, 2, 100, 101, 102, 200, 201, 300, 301, 500, 501]
VTV-EX	11	104	1200	[0, 0, 0, -6.3, -6.3, -25.1, -25.1, -25.1, -22.7, -22.7, -22.7]	[0, 1, 2, 100, 101, 200, 201, 202, 300, 301, 302]
VTV-UC	12	32-48	500	[0, 0, -10, -10, -10, -17.8, -17.8, -17.8, -21.1, -21.1, -26.3, -26.3]	[0, 1, 100, 101, 102, 200, 201, 202, 300, 301, 400, 401]
VTV-SDWW	12	104	1150	[0, 0, -11.2, -11.2, -19, -21.9, -25.3, -25.3, -24.4, -28, -26.1, -26.1]	[0, 1, 100, 101, 200, 300, 400, 401, 500, 600, 700, 701]

- ▷ **RTV Suburban Street (RTV-SS)**: The road side unit (RSU) is placed at roads intersection in a sub urban environment. The vehicle is far away from the RSU by 100 m and moving at 32-48 km/hr velocity range.
- ▷ **RTV Expressway (RTV-EX)**: In this scenario, the RSU is placed on a highway, and the vehicle moves towards the RSU at a speed of 104 km/hr.
- ▷ **RTV Urban Canyon (RTV-UC)**: the measurements here are performed in a dense traffic environment. The RSU transmitting antenna is mounted on a pole

near the urban intersection of Peach tree Street and Peach tree Circle. The vehicle moves at 32-48 km/hr speed and 100 m far away from the RSU.

Further investigations and discussion concerning the detailed channel models measurement setups are shown in [Acosta-Marum, 2007]. Among the presented vehicular channel models, we choose two of them to be used in the thesis simulations, since they satisfy different mobility conditions that are sufficient to validate the performance of the studied vehicular channel estimators. The defined simulation scenarios are as follows:

- ▷ Low mobility vehicular scenario, where VTV-UC channel model is employed using V = 45 Kmph which is equivalent to $f_d = 250$ Hz as a maximum Doppler shift.
- ▷ High mobility vehicular scenario, where VTV-SDWW channel model with V = 100 Kmph and $f_d = 500$ Hz is employed.
- ▷ Very high mobility scenario, where VTV-SDWW channel model is considered with V = 200 Kmph and $f_d = 1000$ Hz. This scenario is employed in order to further evaluate the robustness of the proposed and studied vehicular channel estimators.

Chapter 3

Conventional SoA Channel Estimators

Contents

The IEEE 802.11p frame structure allocates two preamble symbols that are used for basic least squares (LS) estimation, and four pilot subcarriers per OFDM symbols in the payload frame which are used for channel variation tracking over time. This basic LS estimation at the beginning of the frame is simple, but the estimated channel becomes outdated for the equalization of the successive transmitted symbols as a result of high channel variation

in vehicular environments. Hence, accurately tracking the channel variation over time is a crucial need, due to its contribution in improving the overall system performance.

Conventional channel estimators that have been proposed for vehicular communications assume that the allocated pilot subcarriers are not sufficient for accurately tracking the vehicular channel, since they are not spaced closely enough to capture the variation of the channel in the frequency domain. Therefore, conventional estimators are mainly based on the demapped data subcarriers, besides pilot subcarriers to update the channel estimate for each received symbol. This procedure is referred to as data-pilot aided (DPA) channel estimation. We note that the DPA estimation suffers from a considerable performance degradation especially in high mobility scenarios since Doppler error affects the data subcarriers demapping accuracy. As a result, the data subcarriers could be demapped to the wrong constellation and it is impractical to be employed in high mobility scenarios.

Moreover, the channel estimation can be implemented according to two methods: (i) symbol-by-symbol (SBS) estimators, where the channel is estimated for each received symbol separately. (ii) frame-by-frame (FBF) estimators, where the receiver waits for the whole orthogonal frequency division multiplexing (OFDM) frame reception, then, the channel for each received OFDM symbol is estimated in an 2D manner, i.e. employing all the received pilots within the received frame where the previous, current and future pilots are employed in the channel estimation for each received symbol. Then, higher performance accuracy can be achieved by employing FBF estimators, since the channel estimation of each symbol takes advantage from the knowledge of previous, current, and future allocated pilots within the frame. Unlike, SBS estimators, where only the previous and current pilots are employed in the channel estimation for each received symbol. However, in the FBF estimation, the receiver should wait for the complete frame reception before starting the channel estimation, thus penalizes the latency.

In this chapter, we shed light on the conventional state-of-the-art (SoA) SBS estimators that are presented in Section 3.1. After that, the performance evaluation and computational complexity analysis of the studied SBS estimators are presented in Section 3.2 and Section 3.3 respectively. Finally, the limitations of the conventional SBS estimators and our proposed strategies to overcome these limitations are provided in Section 3.4.

3.1 Symbol by Symbol Estimators

There exists an extensive work in the literature concerning the conventional SBS estimators, where most of them are based on the DPA estimation. Hereafter, the steps that are applied in each SBS estimator are presented.

3.1.1 DPA

The DPA estimator [Fernandez et al., 2012] is based on the basic LS estimation and the demapped data subcarriers of the previous received OFDM symbol to estimate the channel for the current OFDM symbol according to the following steps:

1. **LS estimation**: that is implemented using the two long training symbols (LTS) received preambles denoted as $\tilde{\boldsymbol{y}}_1^{(p)}[k]$, and $\tilde{\boldsymbol{y}}_2^{(p)}[k]$, and the predefined preamble sequence $\tilde{\boldsymbol{p}}[k]$ such that:

$$\hat{\tilde{\boldsymbol{h}}}_{\mathrm{LS}}[k] = \frac{\tilde{\boldsymbol{y}}_1^{(p)}[k] + \tilde{\boldsymbol{y}}_2^{(p)}[k]}{2\tilde{\boldsymbol{p}}[k]}. \tag{3.1}$$

2. **Equalization**: the i-th received OFDM symbol is equalized by the previously DPA estimated channel, such that:

$$\tilde{\boldsymbol{y}}_{\mathrm{eq}_i}[k] = \frac{\tilde{\boldsymbol{y}}_i[k]}{\hat{\tilde{\boldsymbol{h}}}_{\mathrm{DPA}_{i-1}}[k]},\ \hat{\tilde{\boldsymbol{h}}}_{\mathrm{DPA}_0}[k] = \hat{\tilde{\boldsymbol{h}}}_{\mathrm{LS}}[k]. \tag{3.2}$$

3. **Demapping**: $\tilde{\boldsymbol{y}}_{\mathrm{eq}_i}[k]$ is demapped to the nearest constellation point to obtain $\tilde{\boldsymbol{d}}_i[k]$.

4. **Final DPA estimation**: DPA updates the final estimated channel for the i-th received OFDM symbol by:

$$\hat{\tilde{\boldsymbol{h}}}_{\mathrm{DPA}_i}[k] = \frac{\tilde{\boldsymbol{y}}_i[k]}{\tilde{\boldsymbol{d}}_i[k]}. \tag{3.3}$$

DPA scheme is considered as the initial estimation process utilized by most vehicular channel conventional estimators. However, DPA estimation has two main limitations. First, it is based on the basic $\hat{\tilde{\boldsymbol{h}}}_{\mathrm{LS}}$ estimation suffering from noise enhancement. Second, the demapping step in DPA results in a significant demapping error mainly in low signal-to-noise ratio (SNR) region, and this error is enlarged in high mobility scenarios employing high modulation orders. Moreover, since the DPA estimated channels are updated iteratively over the received frame, the demapping error propagates through the frame leading to significant performance degradation.

3.1.2 STA

The spectral temporal averaging (STA) estimator [Fernandez et al., 2012] has been proposed to further improve the DPA estimation by applying two additional steps on top of DPA estimation. The first step is by performing frequency domain averaging over the DPA estimated channel including the current and the neighboring subcarriers as follows:

$$\hat{\tilde{\boldsymbol{h}}}_{\mathrm{FD}_i}[k] = \sum_{\lambda=-\beta}^{\lambda=\beta} \omega_\lambda \hat{\tilde{\boldsymbol{h}}}_{\mathrm{DPA}_i}[k+\lambda],\ \omega_\lambda = \frac{1}{2\beta+1}. \tag{3.4}$$

After that, the final STA channel estimate is updated using time averaging between the previously STA estimated channel and the frequency averaged channel in (3.4), such that

$$\hat{\tilde{\boldsymbol{h}}}_{\mathrm{STA}_i}[k] = (1 - \frac{1}{\alpha})\hat{\tilde{\boldsymbol{h}}}_{\mathrm{STA}_{i-1}}[k] + \frac{1}{\alpha}\hat{\tilde{\boldsymbol{h}}}_{\mathrm{FD}_i}[k]. \tag{3.5}$$

STA estimator performs well in low SNR region. However, it suffers from a considerable error floor in high SNR region because of the large DPA demapping error. It is worth mentioning that in [Fernandez et al., 2012], the values of the frequency and time averaging coefficients are fixed to $\alpha = \beta = 2$, however, to improve the STA performance, the optimal values of α and β should be updated according to the channel variation and not predefined as fixed parameters for all mobility scenarios.

3.1.3 CDP

The constructed data pilots (CDP) estimator is proposed in [Zhao et al., 2013] to improve DPA estimation. The CDP estimator uses only the $K_d = 48$ data subcarriers and it applies DPA estimation as an initial step. After that, CDP estimator updates the estimated channel according to the assumption that the time correlation of the channel response between two adjacent OFDM symbols is high. Following this assumption, the demapped data subcarriers using the current DPA estimated channel and previous CDP estimated channel should be equal, then, the DPA estimated channel is considered as reliable. In this context, the CDP estimator proceeds as follows:

1. **Equalization**: The previously received OFDM symbol is equalized by $\hat{\tilde{\boldsymbol{h}}}_{\mathrm{DPA}_i}[k]$ and $\hat{\tilde{\boldsymbol{h}}}_{\mathrm{CDP}_{i-1}}[k]$, where:

$$\tilde{\boldsymbol{y}}'_{\mathrm{eq}_{i-1}}[k] = \frac{\tilde{\boldsymbol{y}}_{i-1}[k]}{\hat{\tilde{\boldsymbol{h}}}_{\mathrm{DPA}_i}[k]},\ \tilde{\boldsymbol{y}}''_{\mathrm{eq}_{i-1}}[k] = \frac{\tilde{\boldsymbol{y}}_{i-1}[k]}{\hat{\tilde{\boldsymbol{h}}}_{\mathrm{CDP}_{i-1}}[k]},\ k \in \mathcal{K}_{\mathrm{d}}. \tag{3.6}$$

2. **Demapping**: The obtained $\tilde{\boldsymbol{y}}'_{\mathrm{eq}_{i-1}}[k]$ and $\tilde{\boldsymbol{y}}''_{\mathrm{eq}_{i-1}}[k]$ are demapped into $\tilde{\boldsymbol{d}}'_{i-1}[k]$ and $\tilde{\boldsymbol{d}}''_{i-1}[k]$, respectively.

3. **Final CDP estimation**: The final $\hat{\tilde{\boldsymbol{h}}}_{\mathrm{CDP}_i}[k]$ is updated using the demapping results from the previous step, such that:

$$\hat{\tilde{\boldsymbol{h}}}_{\mathrm{CDP}_i}[k] = \begin{cases} \hat{\tilde{\boldsymbol{h}}}_{\mathrm{CDP}_{i-1}}[k], & \tilde{\boldsymbol{d}}'_{i-1}[k] \neq \tilde{\boldsymbol{d}}''_{i-1}[k]. \\ \hat{\tilde{\boldsymbol{h}}}_{\mathrm{DPA}_i}[k], & \tilde{\boldsymbol{d}}'_{i-1}[k] = \tilde{\boldsymbol{d}}''_{i-1}[k]. \end{cases} \tag{3.7}$$

The CDP estimator outperforms STA estimator especially in high SNR region, but the additional demapping procedure applied in the CDP estimator leads to the increase of the total demapping error of both DPA and CDP estimators. Therefore, CDP suffers from performance degradation, especially in high mobility scenarios and when high modulation orders are employed.

3.1.4 iCDP

In [Wang, 2018] the authors proposed an improved version of the CDP estimator, where the pilot subcarriers are also included in the channel estimation in addition to the K_d data subcarriers. Therefore, Improved CDP (iCDP) proceeds similar to the CDP estimation as illustrated in (3.6), but with $k \in \mathcal{K}_{on}$, and $\hat{\tilde{\boldsymbol{h}}}_{\mathrm{iCDP}_{i-1}}[k]$ instead of $\hat{\tilde{\boldsymbol{h}}}_{\mathrm{CDP}_{i-1}}[k]$. Moreover, the iCDP estimator updates the final estimated channel for the current OFDM symbol by utilizing the STA estimated channel, such that:

$$\hat{\tilde{\boldsymbol{h}}}_{\mathrm{iCDP}_i}[k] = \begin{cases} \hat{\tilde{\boldsymbol{h}}}_{\mathrm{iCDP}_{i-1}}[k], & \tilde{\boldsymbol{d}}'_{i-1}[k] \neq \tilde{\boldsymbol{d}}''_{i-1}[k]. \\ \frac{\tilde{\boldsymbol{h}}_{\mathrm{STA}_i}[k] + \hat{\tilde{\boldsymbol{h}}}_{\mathrm{DPA}_i}[k]}{2}, & \tilde{\boldsymbol{d}}'_{i-1}[k] = \tilde{\boldsymbol{d}}''_{i-1}[k]. \end{cases} \tag{3.8}$$

It is clearly noticed that the iCDP is a combination of both DPA, CDP, and STA estimators. Although this combination improves the performance, the improvement is still limited since the enlarged DPA and CDP demapping error still exists, besides employing fixed time and frequency averaging parameters of the STA estimator as discussed in Section 3.1.2.

3.1.5 TRFI

Another methodology to improve the CDP estimator is proposed in [Yoon-Kyeong Kim et al., 2014], where frequency domain interpolation is utilized by time domain reliable test frequency domain interpolation (TRFI) estimator. Similarly to CDP estimator, the TRFI estimator applies DPA estimation followed by the equalization and demapping steps as (3.6) and (3.7). After that, the final channel estimates are updated by dividing the subcarriers into reliable subcarriers (RS) and unreliable subcarriers (URS) denoted by $\mathcal{RS}_i$ and $\mathcal{URS}_i$ respectively. The TRFI estimator applies first the demapping operations according to (3.6) where $\hat{\tilde{\boldsymbol{h}}}_{\mathrm{TRFI}_{i-1}}[k]$ is used instead of $\hat{\tilde{\boldsymbol{h}}}_{\mathrm{CDP}_{i-1}}[k]$. Then it proceeds as follows:

1. **Subcarriers selection**: The RS set contains the four pilot subcarriers, in addition to the data subcarriers that fulfills $\tilde{\boldsymbol{d}}'_{i-1}[k] = \tilde{\boldsymbol{d}}''_{i-1}[k]$. On the other hand, the URS set includes the data subcarriers where $\tilde{\boldsymbol{d}}'_{i-1}[k] \neq \tilde{\boldsymbol{d}}''_{i-1}[k]$.

2. **Final TRFI estimation**: After selecting the RS and URS sets, frequency-domain cubic interpolation [Marsden, 1974] is applied by using the channel estimates in $\mathcal{RS}_i$ to determine the channel estimates for the $\mathcal{URS}_i$.

The TRFI estimator outperforms both STA and CDP estimators, especially when high modulation orders like 16QAM are employed. Therefore, performing frequency-domain interpolation on top of the CDP estimator improves the performance. However, TRFI still suffers from the demapping and interpolation errors. We note that, the number of RS

subcarriers is inversely proportional to the channel variations. Since as the channel varies rapidly, the condition where $\tilde{\boldsymbol{d}}'_{i-1}[k] \neq \tilde{\boldsymbol{d}}''_{i-1}[k]$ will be more dominant. Therefore, only few RS subcarriers will be selected and the employed cubic interpolation performance will be degraded.

3.1.6 E-TRFI

Recently, the authors in [Han et al., 2020] proposed the E-TRFI estimator which is an enhanced version of the conventional TRFI estimator [Yoon-Kyeong Kim et al., 2014] described in Section 3.1.5. The main E-TRFI upgrades are in the reliable and unreliable subcarriers selection algorithm, where the Euclidean distance between the demapped subcarriers is used as a reliability condition. Moreover, E-TRFI employs the estimated channels at the $\mathcal{K}_{\mathrm{n}}$ guard band subcarriers in order to improve the cubic interpolation accuracy. The E-TRFI estimator proceeds as follows:

1. **Enhanced LS estimation**: The conventional LS estimation presented in (3.1) estimates the channel at $\mathcal{K}_{on}$ subcarriers. However, the E-TRFI employs the estimated channel also at the remaining $\mathcal{K}_{\mathrm{n}}$ subcarriers that are interpolated as follows:

$$\hat{\tilde{\boldsymbol{h}}}_{\text{E-LS}}[k] = \boldsymbol{F}_{64}(\boldsymbol{F}_{52}^{\mathrm{H}}\boldsymbol{F}_{52})^{-1}\boldsymbol{F}_{52}^{\mathrm{H}}\hat{\tilde{\boldsymbol{h}}}_{\mathrm{LS}}[k],\ k \in \mathcal{K}, \tag{3.9}$$

   where $\boldsymbol{F}_{64} \in \mathbb{C}^{K\times L}$ and $\boldsymbol{F}_{52} \in \mathbb{C}^{K_{\mathrm{on}}\times L}$ denote the truncated DFT matrix obtained by selecting K, K_{on} rows and L columns from the $K \times K$ DFT matrix, respectively. Here, L represents the number of channel taps.

2. **Equalization**: The i-th received OFDM symbol is equalized by $\hat{\tilde{\boldsymbol{h}}}_{\text{E-TRFI}_{i-1}}[k]$, where

$$\tilde{\boldsymbol{y}}'_{\mathrm{eq}_i}[k] = \frac{\tilde{\boldsymbol{y}}_i[k]}{\hat{\tilde{\boldsymbol{h}}}_{\text{E-TRFI}_{i-1}}[k]}, k \in \mathcal{K}_{\mathrm{d}}. \tag{3.10}$$

   Here, $\hat{\tilde{\boldsymbol{h}}}_{\text{E-TRFI}_0}[k] = \hat{\tilde{\boldsymbol{h}}}_{\text{E-LS}}[k]$. After that, the obtained $\tilde{\boldsymbol{y}}'_{\mathrm{eq}_i}[k]$ is demapped to the nearest constellation point denoted as $\tilde{\boldsymbol{d}}'_i[k]$.

3. **Euclidean distance reliability test**: In order to ensure that $\tilde{\boldsymbol{d}}'_i[k]$ is correctly demapped, E-TRFI estimator employs a reliability test where the Euclidean distances between $\tilde{\boldsymbol{y}}'_{\mathrm{eq}_i}[k]$ and the constellation points are calculated such that:

$$\boldsymbol{\delta}_i^{(m)}[k] = |\tilde{\boldsymbol{y}}'_{\mathrm{eq}_i}[k] - \boldsymbol{c}^{(m)}|^2,\ m = 1, 2, \ldots, M. \tag{3.11}$$

   $\boldsymbol{c}^{(m)}$ denotes the m-th constellation point with M standing for the employed modulation order. The obtained $\boldsymbol{\delta}_i^{(m)}[k]$ values are arranged in ascending order vector

denoted as $\bar{\boldsymbol{\delta}}_i^{(m)}[k]$, then the reliability ratio for each subcarrier $\boldsymbol{R}_i[k]$ is calculated employing the first and second minimum distances as follows:

$$\boldsymbol{R}_i[k] = \frac{\bar{\boldsymbol{\delta}}_i^{(1)}[k]}{\bar{\boldsymbol{\delta}}_i^{(2)}[k]},\ k \in \mathcal{K}_{\mathrm{d}}, \tag{3.12}$$

where $0 < \boldsymbol{R}_i[k] < 1$. Small $\boldsymbol{R}_i[k]$ value indicates that the demapped subcarrier $\tilde{\boldsymbol{d}}_i'[k]$ is close to actual transmitted modulated symbol. Therefore, the subcarrier is classified as a reliable subcarrier and inserted into the reliable subcarriers set $\mathcal{RS}_i$, otherwise it is considered unreliable and inserted into the unreliable subcarriers set $\mathcal{URS}_i$. The extensive simulations performed in [Han et al., 2020], show that $\boldsymbol{R}_i[k] = 0.5$ is the best predefined threshold for the reliability test.

4. **Frequency domain cubic interpolation**: After selecting the $\mathcal{RS}_i$ and $\mathcal{URS}_i$ sets, frequency-domain cubic interpolation [Marsden, 1974] is applied by using the channel estimates in $\mathcal{RS}_i$ to determine the channel estimates for the $\mathcal{URS}_i$, such that

$$\hat{\tilde{\boldsymbol{h}}}_{\mathrm{Int}_i}[k] = \begin{cases} \tilde{\boldsymbol{y}}_i[k]/\tilde{\boldsymbol{d}}_i'[k], & k \in \mathcal{RS}_i. \\ \tilde{\boldsymbol{y}}_i[k]/\tilde{\boldsymbol{x}}_{\mathrm{p}_i}[k], & k \in \mathcal{K}_{\mathrm{p}}. \\ \tilde{\boldsymbol{y}}_i[k]/\hat{\tilde{\boldsymbol{h}}}_{\text{E-TRFI}_{i-1}}[k], & k \in \mathcal{K}_{\mathrm{n}}. \\ \text{Cubic Interpolation}, & k \in \mathcal{URS}_i. \end{cases} \tag{3.13}$$

5. **Noise attenuation**: The last step in the E-TRFI estimator is applying a noise attenuation as follows

$$\hat{\tilde{\boldsymbol{h}}}_{\text{E-TRFI}_i}[k] = \boldsymbol{Q}\hat{\tilde{\boldsymbol{h}}}_{\mathrm{Int}_i}[k],\ k \in \mathcal{K}, \tag{3.14}$$

where $\boldsymbol{Q} = \boldsymbol{F}_{64}\boldsymbol{F}_{64}^{\mathrm{H}}$. We note that the computation of $\boldsymbol{Q}$ depends mainly on the L channel taps and it can be performed offline resulting in reducing the computational complexity.

The E-TRFI estimator outperforms other conventional estimators, but it still suffers from a considerable performance degradation as a result of the enlarged demapping error in high mobility scenarios. In addition, it suffers from high computational complexity due to the matrix multiplication in (3.9) and (3.14). Moreover, the Euclidean distance reliability test threshold is sensitive with respect to the channel variation and it is impractical to be fixed at 0.5 as proposed. Therefore, the E-TRFI performance degrades as the channel variation increase. On the other hand, even though increasing the number of subcarriers considered in the cubic interpolation improves the performance, but similar to TRFI estimator the numbers of $\mathcal{RS}_i$ and $\mathcal{URS}_i$ are directly related to the vehicular channel variations.

3.1.7 MMSE-VP

The minimum mean square error using virtual pilots (MMSE-VP) estimator is proposed to tackle the drawbacks caused by the channel variations [Joo-Young Choi et al., 2016]. Similarly to the previously presented channel estimators, MMSE-VP employs the DPA estimation performed in (3.2), and (3.3) followed by the frequency-domain minimum mean squared error (MMSE) estimation to compute the final estimate. First of all, MMSE-VP builds a virtual pilot vector $\hat{\hat{\boldsymbol{h}}}_{\mathrm{vp}_i} \in \mathbb{C}^{K_{\mathrm{on}} \times 1}$ by arranging the $\hat{\hat{\boldsymbol{h}}}_{\mathrm{DPA}_i}$ obtained from (3.3), where the two pilots are placed at the beginning ($k = -21, -7$) and two pilots are placed at the end ($k = 7, 21$), such that:

$$\hat{\hat{\boldsymbol{h}}}_{\mathrm{vp}_i} = [\hat{\hat{\boldsymbol{h}}}_{\mathrm{DPA}_i}[-21], \hat{\hat{\boldsymbol{h}}}_{\mathrm{DPA}_i}[k], ..., \hat{\hat{\boldsymbol{h}}}_{\mathrm{DPA}_i}[21]], \ k \in \mathcal{K}_{on}. \tag{3.15}$$

As noticed in (3.15), the virtual pilot vector $\hat{\hat{\boldsymbol{h}}}_{\mathrm{vp}_i}$ contains pilots subcarriers on its boundaries, such that the first two pilots $\hat{\hat{\boldsymbol{h}}}_{\mathrm{DPA}_i}[-21]$ and $\hat{\hat{\boldsymbol{h}}}_{\mathrm{DPA}_i}[-7]$ are added at the beginning while the other two pilots $\hat{\hat{\boldsymbol{h}}}_{\mathrm{DPA}_i}[7]$ and $\hat{\hat{\boldsymbol{h}}}_{\mathrm{DPA}_i}[21]$ are concatenated at the end. The final channel estimate for each received OFDM symbol is calculated using the frequency domain MMSE estimation as illustrated below:

$$\hat{\hat{\boldsymbol{h}}}_{\mathrm{MMSE}_i} = \boldsymbol{R}_{\hat{\hat{\boldsymbol{h}}}_{\mathrm{DPA}_i}\hat{\hat{\boldsymbol{h}}}_{\mathrm{vp}_i}} \left(\boldsymbol{R}_{\hat{\hat{\boldsymbol{h}}}_{\mathrm{vp}_i}\hat{\hat{\boldsymbol{h}}}_{\mathrm{vp}_i}} + \sigma_i^2 \boldsymbol{I}' \right)^{-1} \hat{\hat{\boldsymbol{h}}}_{\mathrm{vp}_i}, \tag{3.16}$$

where $\boldsymbol{R}_{\hat{\hat{\boldsymbol{h}}}_{\mathrm{DPA}_i}\hat{\hat{\boldsymbol{h}}}_{\mathrm{vp}_i}} \in \mathbb{C}^{K_{\mathrm{on}} \times K_{\mathrm{on}}}$ is the cross-correlation matrix between the DPA estimated channel vector $\hat{\hat{\boldsymbol{h}}}_{\mathrm{DPA}_i}$ and the virtual pilot vector $\hat{\hat{\boldsymbol{h}}}_{\mathrm{vp}_i}$, $\boldsymbol{R}_{\hat{\hat{\boldsymbol{h}}}_{\mathrm{vp}_i}\hat{\hat{\boldsymbol{h}}}_{\mathrm{vp}_i}} \in \mathbb{C}^{K_{\mathrm{on}} \times K_{\mathrm{on}}}$ is the auto-correlation matrix of the virtual pilot vector. $\boldsymbol{I}'_{K_{\mathrm{on}}}$ is the identity unit matrix, and σ_i^2 is the average noise power in the i-th received OFDM symbol. The MMSE-VP estimator uses the correlation characteristics between the DPA channel estimates and the virtual pilots vectors to reduce the channel estimation errors. This estimator provides a better performance than the existing conventional estimators, especially when low modulation orders e.g. QPSK are utilized, but it still suffers from performance degradation in high mobility scenarios and it has lower performance than the STA at a low SNRs. Moreover, it requires high computational complexity due to the matrix inversion applied in (3.16).

3.1.8 Adaptive STA-MMSE-VP

Adaptive channel estimation is proposed in [Choi et al., 2017] where the STA and MMSE-VP estimators are jointly employed resulting in an improved overall performance. The key element is the decision method based on preamble symbols. Therefore, the decision method utilizes the basic LS estimated channel $\hat{\hat{\boldsymbol{h}}}_{\mathrm{LS}}[k]$ obtained by (3.1) to select the best channel estimator. In this context, STA and MMSE-VP are implemented, and the mean squared error (MSE) with respect to $\hat{\hat{\boldsymbol{h}}}_{\mathrm{LS}}[k]$ is calculated such that:

$$\mathrm{MSE}_{\mathrm{STA}} = |E[|\hat{\hat{\boldsymbol{h}}}_{\mathrm{LS}}[k] - \hat{\hat{\boldsymbol{h}}}_{\mathrm{STA}}[k]|^2] - \sigma^2_{\mathrm{LS}}| \tag{3.17}$$

$$\mathrm{MSE}_{\text{MMSE-VP}} = |E[|\hat{\hat{\boldsymbol{h}}}_{\mathrm{LS}}[k] - \hat{\hat{\boldsymbol{h}}}_{\text{MMSE-VP}}|^2] - \sigma^2_{\mathrm{LS}}|, \tag{3.18}$$

where σ^2_{LS} is the average noise power of $\hat{\hat{\boldsymbol{h}}}_{\mathrm{LS}}[k]$. In low SNR region, $\hat{\hat{\boldsymbol{h}}}_{\mathrm{LS}}[k]$ includes high-noise components, and because the STA is effective in alleviating the noise impact, then, $\mathrm{MSE}_{\mathrm{STA}}$ records the lowest MSE value. Therefore, the STA estimator is employed in low SNR region. However, as the SNR increases, the noise impact becomes neglected. Thus, the accuracy of the channel estimates in the STA significantly decreases while the error performance of the MMSE-VP estimator is improved. Therefore, $\mathrm{MSE}_{\text{MMSE-VP}}$ becomes lower than $\mathrm{MSE}_{\mathrm{STA}}$ in high SNR region. Accordingly, the proposed estimator uses the STA when the MSE performance of the STA is lower than that of the MMSE-VP and vice versa. This adaptive estimator suffers from a considerable high computational complexity due to the MMSE-VP employment. Moreover, it suffers from a considerable performance degradation in high mobility scenarios especially when high modulation orders are used, due to the enlarged DPA demapping error impact.

3.1.9 LMMSE Estimator

The linear minimum mean square error (LMMSE) estimator [Choi et al., 2015] aims to linearly minimize the MSE error between the LMMSE estimated and real channel, given the LS estimated channel at the $\mathcal{K}_{\mathrm{p}}$ subcarriers defined as:

$$\hat{\hat{\boldsymbol{h}}}_{p_i}[k] = \frac{\tilde{\boldsymbol{y}}_{p_i}[k]}{\tilde{\boldsymbol{x}}_{p_i}[k]},\ k \in \mathcal{K}_{\mathrm{p}}. \tag{3.19}$$

Accordingly, the key element here is to find $\boldsymbol{W}_{\mathrm{LMMSE}_i}$ where $\hat{\boldsymbol{h}}_{\mathrm{LMMSE}_i} = \boldsymbol{W}_{\mathrm{LMMSE}_i}\hat{\hat{\boldsymbol{h}}}_{p_i}$, so that $\boldsymbol{\epsilon}_{\mathrm{LMMSE}}$ is minimized, such that:

$$\boldsymbol{\epsilon}_{\mathrm{LMMSE}} = \mathrm{E}\left[(\hat{\boldsymbol{h}}_{\mathrm{LMMSE}_i} - \tilde{\boldsymbol{h}}_i)^2\right] = \mathrm{E}\left[(\boldsymbol{W}_{\mathrm{LMMSE}_i}\hat{\hat{\boldsymbol{h}}}_{p_i} - \tilde{\boldsymbol{h}}_i)^2\right]. \tag{3.20}$$

The minimization of $\boldsymbol{\epsilon}_{\mathrm{LMMSE}}$ results in the following expression:

$$\boldsymbol{W}_{\mathrm{LMMSE}_i} = \boldsymbol{R}_{\tilde{\boldsymbol{h}}_i\tilde{\boldsymbol{h}}_{p_i}}\left(\boldsymbol{R}_{\tilde{\boldsymbol{h}}_{p_i}\tilde{\boldsymbol{h}}_{p_i}} + \sigma^2\boldsymbol{I}'\right)^{-1}. \tag{3.21}$$

$\boldsymbol{R}_{\tilde{\boldsymbol{h}}_i,\tilde{\boldsymbol{h}}_{p_i}} = \mathrm{E}\left[\tilde{\boldsymbol{h}}_i\tilde{\boldsymbol{h}}_{p_i}^H\right] \in \mathbb{C}^{K_d \times K_p}$ represents the cross correlation matrix of the real channel and the real channel vector at the $\mathcal{K}_{\mathrm{p}}$ pilot subcarriers within the i-th received OFDM symbol. Moreover, $\boldsymbol{R}_{\tilde{\boldsymbol{h}}_{p_i},\tilde{\boldsymbol{h}}_{p_i}} = \mathrm{E}\left[\tilde{\boldsymbol{h}}_{p_i}\tilde{\boldsymbol{h}}_{p_i}^H\right] \in \mathbb{C}^{K_p \times K_p}$ denotes the autocorrelation matrix of $\tilde{\boldsymbol{h}}_{p_i}$. I'_{K_p} is the identity matrix and σ^2 is the noise power. Therefore, the LMMSE estimated

channel at the $\mathcal{K}_{\mathrm{d}}$ data subcarriers within the i-th received OFDM symbol can be simply obtained as follows:

$$\hat{\hat{\boldsymbol{h}}}_{\mathrm{LMMSE}_i} = \boldsymbol{W}_{\mathrm{LMMSE}_i} \hat{\hat{\boldsymbol{h}}}_{p_i}. \tag{3.22}$$

According to the calculation of $\boldsymbol{W}_{\mathrm{LMMSE}}$ matrix we can define two LMMSE estimators: (i) conventional LMMSE estimator where the matrix inversion in (3.21) is performed online according to the real-time SNR. (ii) low-complexity LMMSE estimator where the $\boldsymbol{W}_{\mathrm{LMMSE}}$ matrix is computed offline resulting in decreasing the computational complexity. Moreover, several configurations of the LMMSE estimator can be defined, according to the number of considered pilot subcarriers. In this thesis, we define two configurations: (i) 1D-LMMSE estimator, where the K_p pilot subcarriers within the i-th received symbol are used to estimate the channel at the $\mathcal{K}_{\mathrm{d}}$ data subcarriers within the same symbol. (ii) 2D-LMMSE estimator, where all the pilot subcarriers within the received OFDM frame, i.e. K_pI, are employed in the channel estimation of each data subcarrier within the frame.

Finally, we note that the LMMSE performance highly depends on the pre-estimated $\boldsymbol{W}_{\mathrm{LMMSE}}$ matrix, where it suffers from a considerable performance degradation in case the channel employed in the $\boldsymbol{W}_{\mathrm{LMMSE}}$ estimation changes.

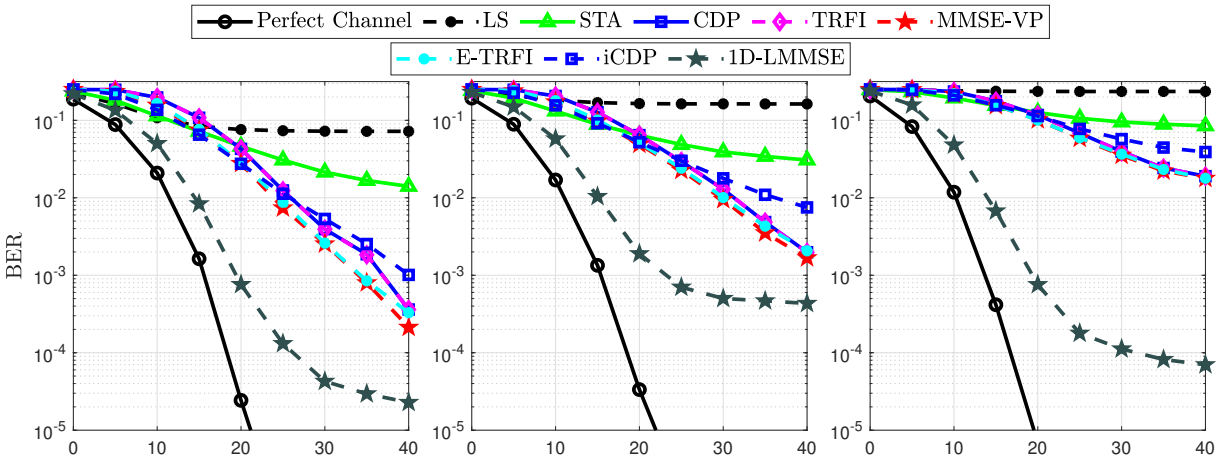
3.2 Simulation Results

In this section, normalized mean-squared error (NMSE) and bit error rate (BER) performance are evaluated using three vehicular channel models referring to low, high and very high mobility scenarios as defined in Section 2.3. We recall that QPSK and 16QAM modulation orders are employed with $\frac{1}{2}$ convolutional code rate. The frame size is $I = 100$ OFDM symbols, and the SNR range $\in [0\ \mathrm{dB}, 40\ \mathrm{dB}]$. The NMSE and BER performance evaluation of the studied estimators are performed over the chosen vehicular channel models according to three criteria: (i) modulation order, (ii) mobility, and (iii) frame length.

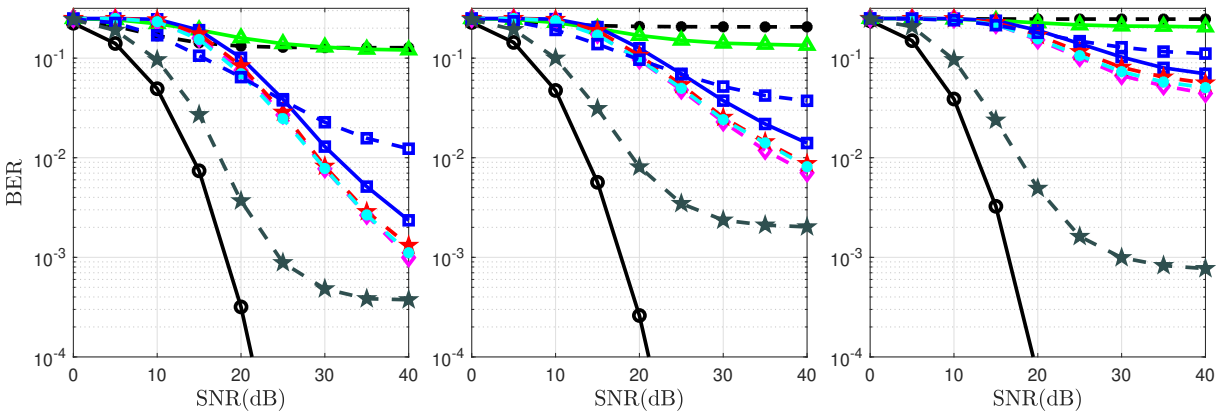
3.2.1 Modulation Order: QPSK vs. 16QAM

Simulation results show that the performance of the SoA conventional DPA-based estimators depends on the employed modulation order as depicted in Fig. 3.1, 3.2. Moreover, we can observe that employing the basic LS estimator without any channel tracking leads to a severe performance degradation in different vehicular scenarios when long frame size is employed, i.e. $I = 100$.

For QPSK modulation order, Fig. 3.1a, and 3.2a, the STA estimator outperforms other conventional estimators in low SNR region due to the frequency and time averaging operations used in STA (3.4), (3.5). Whereas in high SNR regions, conventional estimators express a significant improvement over the STA estimator, where CDP and TRFI record



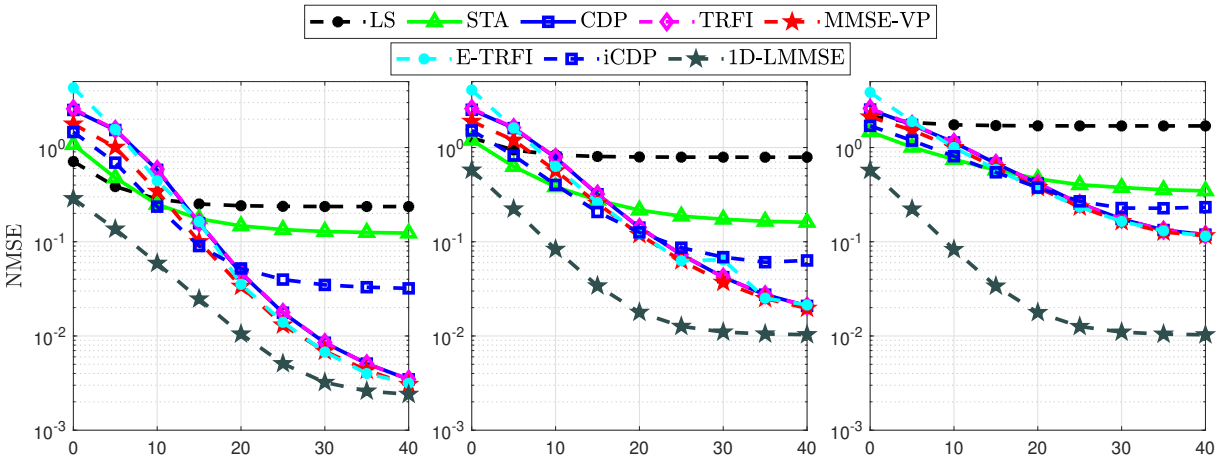
(a) BER performance employing QPSK.



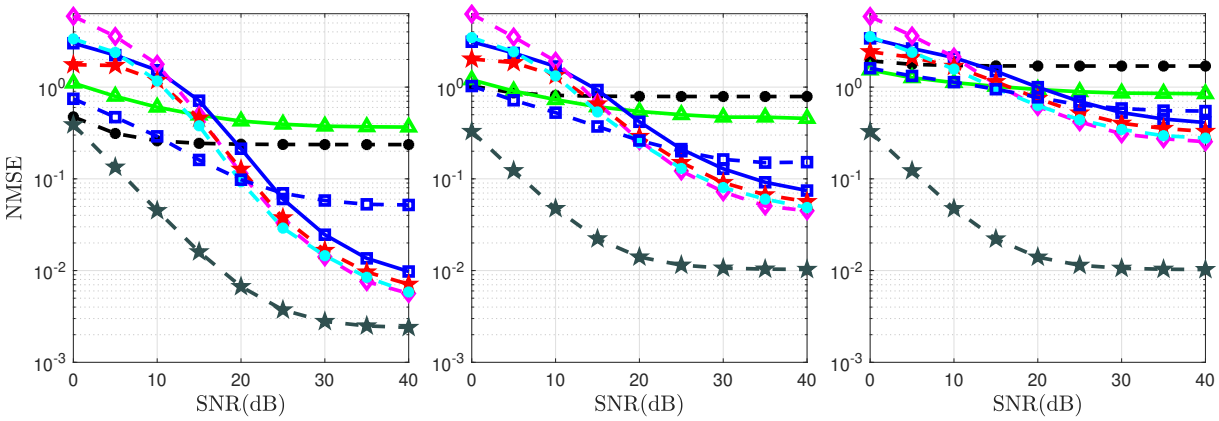
(b) BER performance employing 16QAM.

Figure 3.1: BER performance employing three scenarios: (*i*) first column - low mobility ($v = 45$ Kmph, $f_d = 250$ Hz) (*ii*) second column - high mobility ($v = 100$ Kmph, $f_d = 500$ Hz) (*iii*) third column - very high mobility ($v = 200$ Kmph, $f_d = 1000$ Hz).

similar performance. This is due to the fact that when the SNR is low, the impact of noise and interference are high and powerful enough to shift the equalized received OFDM symbol $\boldsymbol{y}_{\text{eq}_i}[k]$ to wrong regions and as a result, its demapping $\boldsymbol{d}_i[k]$ is shifted to incorrect constellation points. The STA estimator averaging operations are able to alleviate the impact of the noise. However, as the SNR increases, the aforementioned influence is reduced, and thus, the superiority of the conventional estimators emerges over STA. It is worth mentioning that the STA estimator frequency-domain averaging window β, and the time-domain averaging coefficient α are fixed to 2 as discussed in [Fernandez et al., 2012]. But, fixing these parameters instead of updating them according to the channel variation makes the smoothing in the time and frequency domains not effective under vehicular environment. Thus, the gradually accumulated demapping error of $\boldsymbol{d}_i[k]$ cannot be well mitigated using fixed β and α. Hence the emergence of the error floor for STA in high SNR



(a) NMSE performance employing QPSK.

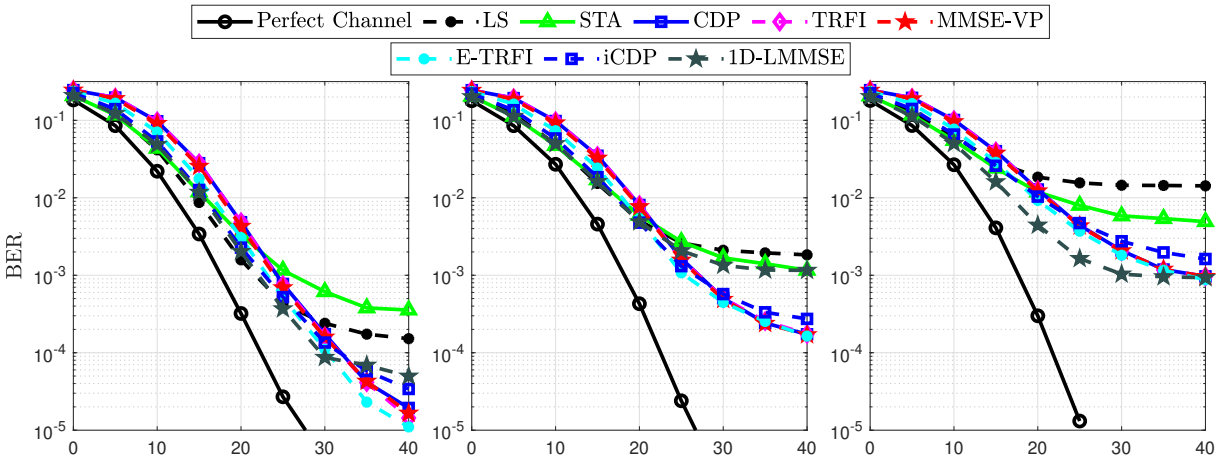


(b) NMSE performance employing 16QAM.

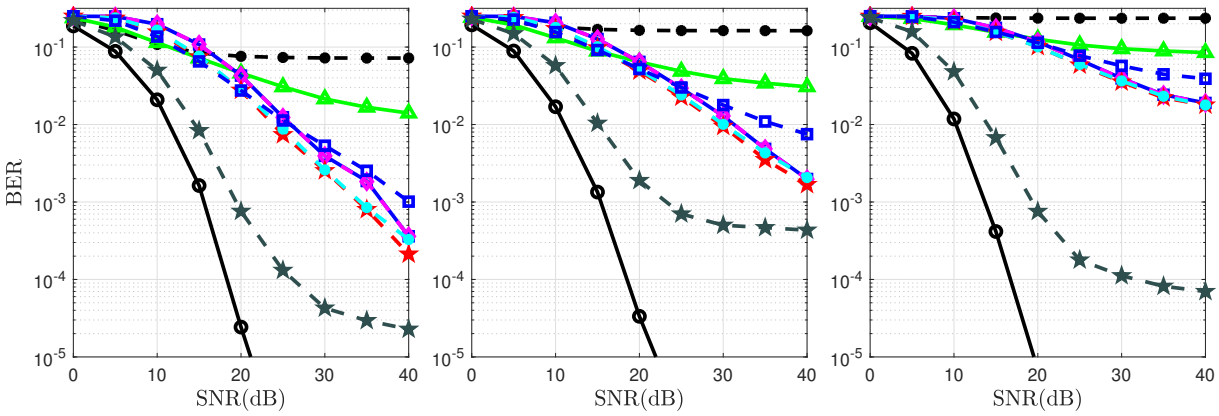
Figure 3.2: NMSE performance employing three scenarios: (*i*) first column - low mobility (v = 45 Kmph, f_d = 250 Hz) (*ii*) second column - high mobility (v = 100 Kmph, f_d = 500 Hz) (*iii*) third column - very high mobility (v = 200 Kmph, f_d = 1000 Hz).

region. On the other hand, the iCDP estimator improves the STA estimator performance by considering both STA and DPA estimations in updating the final iCDP channel estimates. We can notice that employing the iCDP estimator alleviates the STA estimator error floor in high SNR regions while recording almost similar performance as STA in low SNR regions.

The E-TRFI estimator achieves similar performance as MMSE-VP where it outperforms the conventional TRFI estimator by 3 dB gain in terms of SNR for a BER = 10^{-3} in low mobility scenario. Moreover, the adaptive STA-MMSE-VP estimator switches between STA and MMSE-VP according to the MSE criteria shown in (3.17) and (3.18) which improves the performance in the whole SNR region. However, this improvement is also limited due to the DPA estimation error.



(a) BER performance with $I = 10$.

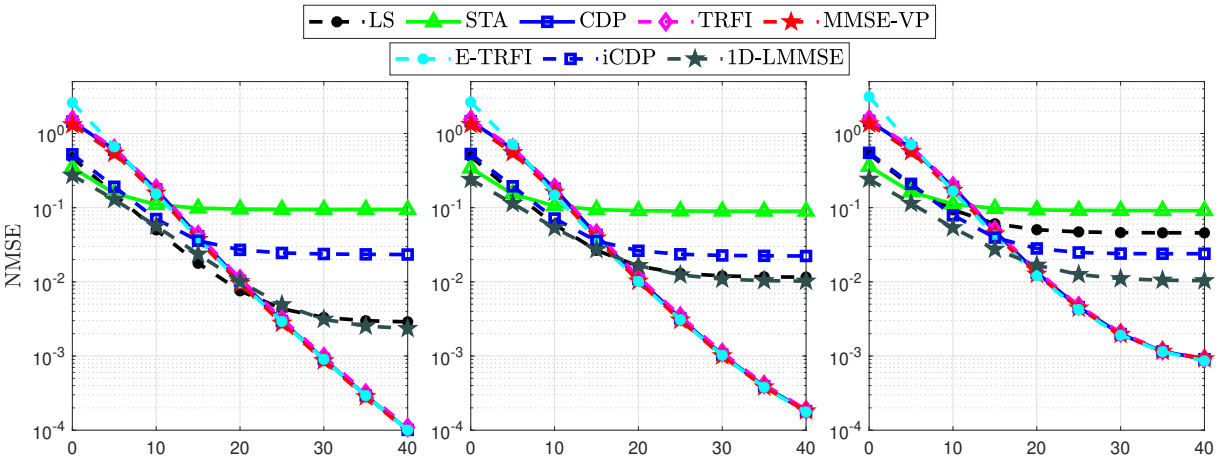


(b) BER performance with with $I = 100$.

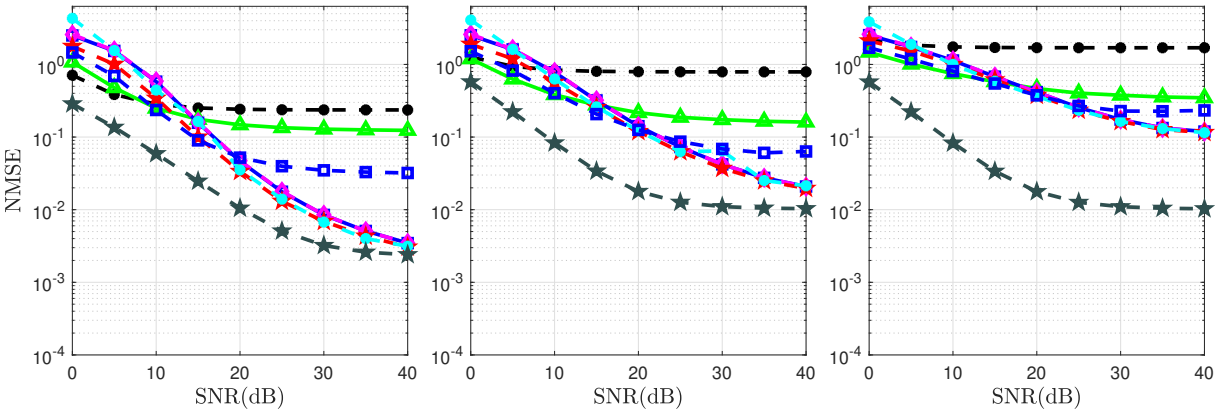
Figure 3.3: BER performance employing three scenarios: (*i*) first column - low mobility ($v = 45$ Kmph, $f_d = 250$ Hz) (*ii*) second column - high mobility ($v = 100$ Kmph, $f_d = 500$ Hz) (*iii*) third column - very high mobility ($v = 200$ Kmph, $f_d = 1000$ Hz).

When 16QAM modulation is employed, STA performance is severely degraded in the whole SNR region due to the huge DPA demapping error that increases as the modulation order increases. Therefore, the adaptive STA and MMSE-VP estimators suffer as well. Moreover, we can notice the impact of employing the cubic interpolation in TRFI, where it outperforms CDP by around 2dB gain. On the other hand, E-TRFI is not able to improve the TRFI estimator performance due to fixing the E-TRFI reliability test threshold to 0.5 as provided in [Han et al., 2020].

As a summary, we can conclude that conventional SBS estimators performance is limited due to their dependency on the DPA estimation and due to using fixed valued parameters. These two reasons are mainly responsible of the conventional estimators performance degradation in time variant channel estimation as shown in Fig. 3.2a and Fig. 3.2b. In contrast, we can notice the performance superiority of the 1D-LMMSE over the



(a) NMSE performance with $I = 10$.



(b) NMSE performance with $I = 100$.

Figure 3.4: NMSE performance employing three scenarios: (*i*) first column - low mobility ($v = 45$ Kmph, $f_d = 250$ Hz) (*ii*) second column - high mobility ($v = 100$ Kmph, $f_d = 500$ Hz) (*iii*) third column - very high mobility ($v = 200$ Kmph, $f_d = 1000$ Hz).

conventional SBS estimation due to employing the channel and noise statistics in the channel estimation process. On the other hand, the 1D-LMMSE estimator is independent of the employed modulation order where its NMSE performance records similar results for both QPSK and 16QAM modulation orders as illustrated in Fig. 3.2a and Fig. 3.2b.

3.2.2 Mobility

As shown in Fig. 3.1 and 3.2, we can notice the severe mobility impact on all the conventional estimators. This is mainly due to the increasing impact of the accumulated DPA demapping error that is directly affected by Doppler error. In contrast, the 1D-LMMSE estimator is robust against Doppler error but it also suffers from performance error floor in high mobility scenarios employing high modulation order.

3.2.3 Frame Length

Figures 3.3 and 3.4 show the BER and NMSE performance employing short and long frame size, respectively.

When short frame size is employed, the channel variation is negligible within the received OFDM frame. Therefore, we can notice that the performance degradation of the conventional DPA-based estimators is less than that when longer frame is employed and the basic LS estimator still performs well, where it records similar performance as 1D-LMMSE especially in low SNR regions. However, in very high mobility scenarios, the impact of Doppler error is dominant, thus, affecting the performance of conventional DPA-based estimators.

In contrast, we can notice that employing long frames leads to considerable performance challenge even in low mobility, where the basic LS estimator is not useful. Finally, we note that the performance of 1D-LMMSE depends mainly on the employed K_p pilot subcarriers that are insufficient since the channel may have more than 4 taps. In addition, the performance of the 1D-LMMSE in long frames is better than that in short frames due to the time diversity impact.

3.3 Computational Complexity Analysis

In this section, a detailed computational complexity analysis of the conventional SBS estimators is provided. The computational complexity analysis is performed according to the number of real-valued multiplication/division and summation/subtraction mathematical operations required to estimate the channel for one received OFDM symbol. Since we are working with complex-valued data, each complex-valued division requires 6 real-valued multiplications, 2 real-valued divisions, 2 real-valued summations, and 1 real-valued subtraction. Moreover, each complex-valued multiplication is equivalent to 4 real-valued multiplications, and 3 real-valued summations.

The least complex estimator is the basic LS estimator (3.1) where the received preamble symbols are added to each others, resulting in $2K_{\text{on}}$ summations. After that, the summation result is divided by the predefined preamble, thus $2K_{\text{on}}$ divisions are also required. Therefore, the total number of divisions and summations needed by the basic LS estimator is $2K_{\text{on}}$ and $2K_{\text{on}}$ respectively.

The DPA estimation that is implemented in the conventional SBS estimators as an initial step requires two equalization steps (3.2), and (3.3). Each equalization step consists of K_{on} complex-valued division, therefore the overall computational complexity of DPA is $16K_{\text{on}}$ multiplications/divisions and $6K_{\text{on}}$ summations/subtractions.

The STA estimator applies frequency and time-domain averaging on top of DPA. The frequency-domain averaging (3.4) coefficient is fixed ($\beta = 2$). Therefore, each

Table 3.1: Computational complexity analysis of the SoA SBS channel estimators.

Est.	Pre-Est.	Mul. / Div.	Sum. / Sub.
LS	-	$2K_{\text{on}}$	$2K_{\text{on}}$
DPA	LS	$18K_{\text{on}}$	$8K_{\text{on}}$
STA	DPA	$22K_{\text{on}} + 2K_{\text{d}}$	$10K_{\text{on}} + 10K_{\text{d}}$
CDP	DPA	$34K_{\text{d}}$	$14K_{\text{d}}$
TRFI	DPA	$34K_{\text{on}} + 26K_{\text{URS}}$	$14K_{\text{on}} + 30K_{\text{URS}}$
iCDP	DPA , STA	$58K_{\text{on}} + 2K_{\text{d}}$	$26K_{\text{on}} + 10K_{\text{d}}$
MMSE-VP	DPA	$26K_{\text{on}} + 6K_{\text{on}}^2 + 4K_{\text{on}}^3$	$14K_{\text{on}} + 6K_{\text{on}}^2 + 3K_{\text{on}}^3$
MMSE-VP & STA	MMSE-VP , STA	$4K_{\text{on}}^3 + 6K_{\text{on}}^2 + 30K_{\text{on}} + 2K_{\text{d}}$	$3K_{\text{on}}^3 + 6K_{\text{on}}^2 + 16K_{\text{on}} + 10K_{\text{d}}$
E-TRFI	DPA	$4K^2 + 4K_{\text{on}}K + 2MK_d + 26K_{\text{URS}}$	$5K^2 + 5K_{\text{on}}K + 3K_d + 4MK_d + 30K_{\text{URS}} - 4K$
1D-LMMSE (real-time)	$\hat{\tilde{\boldsymbol{h}}}_{p_i}$, $\boldsymbol{R}_{\tilde{\boldsymbol{h}}_i,\tilde{\boldsymbol{h}}_{p_i}}$, $\boldsymbol{R}_{\tilde{\boldsymbol{h}}_{p_i},\tilde{\boldsymbol{h}}_{p_i}}$	$4K_{\text{p}}^3 + 4K_{\text{p}}K_{\text{d}} + 2K_{\text{p}}$	$3K_{\text{p}}^3 + 2K_{\text{p}}^2 + 5K_{\text{p}}K_{\text{d}} - 2K_{\text{d}}$
1D-LMMSE (offline)	$\hat{\tilde{\boldsymbol{h}}}_{p_i}$, $\boldsymbol{W}_{\text{1D-LMMSE}}$	$4K_{\text{p}}K_{\text{d}} + 2K_{\text{p}}$	$5K_{\text{p}}K_{\text{d}} - 2K_{\text{d}}$

subcarrier requires 5 complex-valued summations multiplied by a real-valued weight, which are equivalent to 10 real-valued summations, and 2 real-valued multiplications. But the frequency averaging in STA estimator is applied on K_d since the subcarriers in the boundaries are excluded from the averaging operation. As a result, the STA frequency-domain averaging step requires $10K_d$ real-valued summations, and $2K_d$ real-valued multiplications. Moreover, the STA time-domain averaging step (3.5) requires $4K_{\text{on}}$ real-valued divisions, and $2K_{\text{on}}$ real-valued summations. Therefore, the computational complexity of STA is $4K_{\text{on}} + 2K_d$ real-valued multiplications/divisions, and $2K_{\text{on}} + 10K_d$ real-valued summations/subtractions, and the accumulated overall computational complexity of STA estimator is $22K_{\text{on}} + 2K_{\text{d}}$ multiplications/divisions and $10K_{\text{on}} + 10K_{\text{d}}$ summations/subtractions.

The CDP estimator applies two additional equalization steps after DPA on K_d subcarriers resulting in $16K_d$ multiplications/divisions and $6K_d$ summations/subtractions. Hence, CDP requires in total $34K_{\text{d}}$ multiplications/divisions and $14K_{\text{d}}$ summations/subtractions.

The iCDP estimator employs STA in order to update the final iCDP estimated channels. Therefore, the overall accumulated computational complexity of iCDP is $58K_{\text{on}} + 2K_{\text{d}}$ multiplications/divisions and $26K_{\text{on}} + 10K_{\text{d}}$ summations/subtractions.

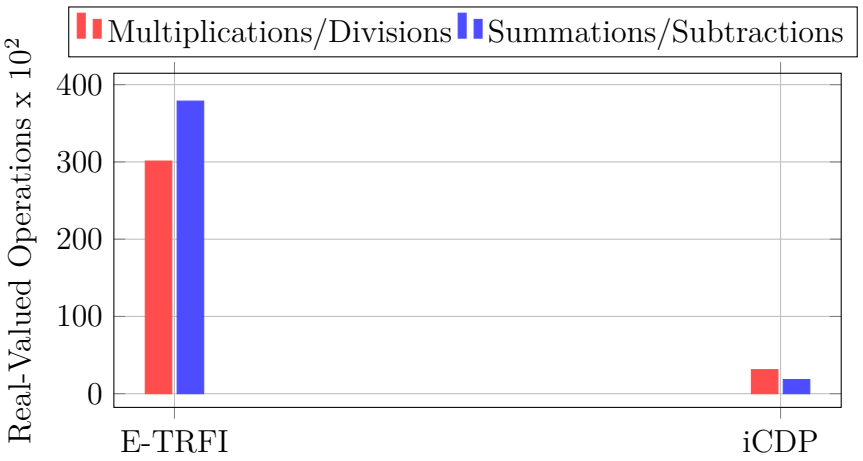
The computational complexity of TRFI relies mainly on the size of unreliable subcarriers set. In order to have a good approximation of the unreliable subcarriers set size, we ran simulations 10^4 times, and we found that an average of $K_{\text{int}} = 10$ subcarriers are considered as unreliable in each received OFDM symbol. Each unreliable subcarrier is bounded by two reliable subcarriers. Theoretically, the cubic interpolation of points located within a known interval requires the calculation of the third degree polynomial coefficients. This polynomial (3.23) expresses the behaviour of the interpolated curve within the specified interval.

$$f(x) = a.x^3 + b.x^2 + c.x + d. \tag{3.23}$$

As shown in Appendix A the cubic interpolation of one subcarrier between two reliable subcarriers requires 26 multiplications/divisions and 30 summations/subtractions. Thus, the computational complexity of TRFI is $34K_{\text{on}} + 26K_{\text{int}}$ multiplications/divisions and $14K_{\text{on}} + 30K_{\text{int}}$ summations/subtractions.

MMSE-VP scheme suffers from high computational complexity due to the correlation matrices manipulation in addition to the matrix inversion operation. $\boldsymbol{R}_{\hat{\boldsymbol{h}}_{\text{DPA}_i}\hat{\boldsymbol{h}}_{\text{vp}_i}}$ calculation requires $4K_{\text{on}}+2K_{\text{on}}^2$ real-valued multiplications and $3K_{\text{on}}+2K_{\text{on}}^2$ summations. The same calculation is performed for $\boldsymbol{R}_{\hat{\boldsymbol{h}}_{\text{vp}_i}\hat{\boldsymbol{h}}_{\text{vp}_i}}$ manipulation. Moreover, $\boldsymbol{R}_{\hat{\boldsymbol{h}}_{\text{vp}_i}\hat{\boldsymbol{h}}_{\text{vp}_i}}$ as presented in (3.16) is added to the noise power vector multiplied by the identity matrix. Thus, $2K_{\text{on}}^2$ multiplications and $2K_{\text{on}}^2$ summations are required in this step. Finally, matrix inversion followed by multiplication is applied to calculate the final MMSE-VP channel estimate. This requires K_{on}^3 complex multiplications, which is equivalent to $4K_{\text{on}}^3$ real-valued multiplications and $3K_{\text{on}}^3$ real-valued summations/subtractions. Therefore, the overall accumulated computational complexity of MMSE-VP is $26K_{\text{on}} + 6K_{\text{on}}^2 + 4K_{\text{on}}^3$ multiplications/divisions and $14K_{\text{on}} + 6K_{\text{on}}^2 + 3K_{\text{on}}^3$ summations/subtractions operations.

The adaptive STA-MMSE-VP estimator employs jointly both STA and MMSE-VP. Therefore, the required overall computational complexity is simply $4K_{\text{on}}^3+6K_{\text{on}}^2+30K_{\text{on}}+2K_{\text{d}}$ multiplications/divisions and $3K_{\text{on}}^3+6K_{\text{on}}^2+16K_{\text{on}}+10K_{\text{d}}$ summations/subtractions operations.



For the E-TRFI estimator, it starts with the E-LS estimation step (3.9) that requires $4K_{\text{on}}K$ real-valued multiplications and $5K_{\text{on}}K - 2K$ real-valued summations. Then equalization is employed only on K_d subcarriers where $8K_d$ multiplications/divisions and $3K_d$ summations/subtractions are required. The Euclidean distance reliability test employs $K_d + 2MK_d$ real-valued divisions and $4MK_d$ subtractions. After that, frequency domain cubic interpolation is performed. Our simulations show that for each OFDM symbol, $K_{\text{URS}} = 16$ subcarriers needed to be interpolated. The frequency domain cubic interpolation requires $26K_{\text{URS}}$ multiplications/divisions and $30K_{\text{URS}}$ summations/subtractions. Finally, the E-TRFI employs the noise attenuation step that

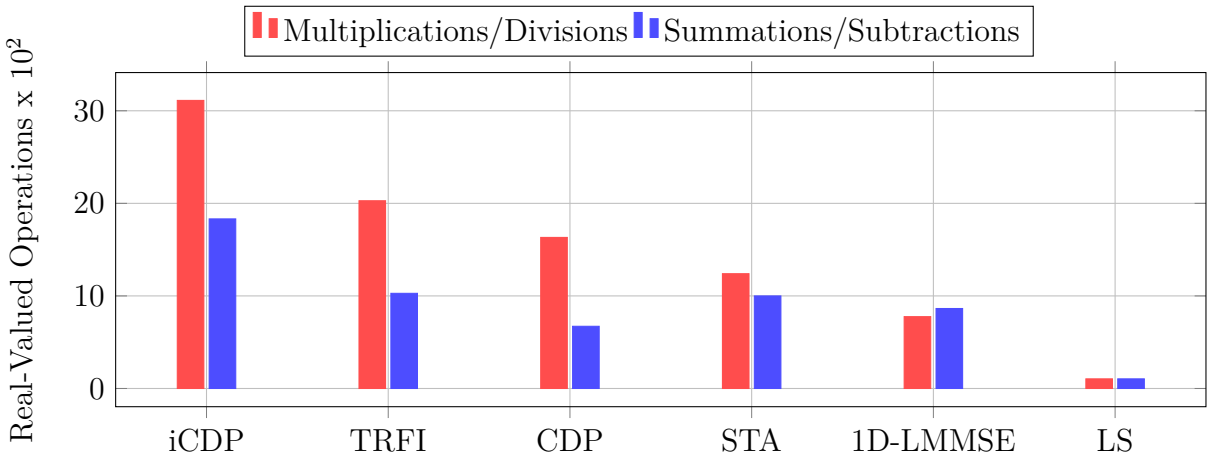


Figure 3.5: Detailed computational complexity of the SoA SBS channel estimators in terms of real-valued operations.

requires $4K^2$ real-valued multiplications and $5K^2 - 2K$ real-valued summations. Hence, the E-TRFI estimator's overall computational complexity is $4K^2+4KK_{\text{on}}+9K_{\text{d}}+26K_{\text{URS}}$ multiplications/divisions and $5K^2 + 5KK_{\text{on}} + 3K_{\text{d}} + 2MK_{\text{d}} + 30K_{\text{URS}} - 4K$ real-valued summations/subtractions.

The 1D-LMMSE estimator manipulates first the LS estimation at the pilot subcarriers within the received OFDM symbol which requires $2K_{\text{p}}$ real-valued divisions. After that, conventional 1D-LMMSE estimator where the matrix inversion in (3.21) is computed in real-time requires K_p^3 complex-valued multiplication. Accordingly, the overall computational complexity of the conventional 1D-LMMSE estimator is $4K_{\text{p}}^3+4K_{\text{p}}K_{\text{d}}+2K_{\text{p}}$ real-valued multiplications/divisions and $3K_{\text{p}}^3+2K_{\text{p}}^2+5K_{\text{p}}K_{\text{d}}-2K_{\text{d}}$ real-valued summations. Whereas, the computational complexity of the low-complexity 1D-LMMSE where $\boldsymbol{W}_{\text{1D-LMMSE}}$ is computed offline, requires $4K_{\text{p}}K_{\text{d}} + 2K_{\text{p}}$ real-valued multiplications/divisions and $5K_{\text{p}}K_{\text{d}} - 2K_{\text{d}}$ real-valued summations.

Table 3.1 and Fig. 3.5 summarizes the overall computational complexities of the studied SoA SBS estimators in terms of real valued operations. For clarity, the bar graphs in Fig. 3.5, show first the high complex estimators, i.e. E-TRFI compared to iCDP. Then, iCDP estimator is compared to the other SoA SBS estimators including the low-complexity 1D-LMMSE estimator. Moreover, the adaptive STA-MMSE-VP, MMSE-VP are not shown since their computational complexities are of order K_{on}^3. Finally, we note that, the adaptive STA-MMSE-VP, MMSE-VP, and E-TRFI estimators are more complex than the iCDP estimator by 99.6%, 99.5%, and 92.72% respectively.

3.4 Limitations and Proposed Strategies

Performance degradation of conventional vehicular channel estimators is mainly related to the following reasons: (i) conventional estimators are based initially on the basic LS

channel estimation applied at the received preamble. This basic estimation does not benefit from prior knowledge of the channel model such as the number of taps and power delay profile. Moreover, the impact of noise is not considered in the estimation process. Therefore, basic LS estimation is noisy and unreliable to be considered as a starting point of the conventional estimators. (*ii*) conventional estimators employ the DPA estimation, which is based on demapping the received data subcarriers and using them as pilots. This mechanism is also unreliable since the demapping error is enlarged from one symbol to another leading to an additional error in the estimation process, especially when high modulation orders are employed. (*iii*) conventional estimators either uses pre-defined parameters like the STA frequency and time averaging coefficients, E-TRFI reliability test threshold, or assume that the channel is highly correlated between two successive received OFDM symbols as the case in CDP and iCDP estimators. (*iv*) conventional 1D-LMMSE estimator suffers from a robustness challenge since its performance depends on the pre-estimated W_{LMMSE} matrix, while the number of channel taps may varies in real-time scenarios. These issues highly impact the channel estimation accuracy when the channel varies rapidly. As a result, the generalization ability of the estimator cannot be guaranteed in all mobility scenarios.

In order to overcome the conventional estimators drawbacks, we propose to: (*i*) Improve the conventional estimation by proposing a novel estimation method that depends mainly on employing the tracking pilots without considering the basic LS estimation at the preamble or the DPA estimation. The proposed method does not depends on estimating the channel statistics as required for the LMMSE estimator. As a result, the data subcarriers demapping step is not required and the demapping error is completely eliminated. (*ii*) Integrating different deep learning techniques with conventional channel estimators to improve the estimation accuracy due to the ability of deep learning to learn channel correlation and correcting the conventional estimation errors.

Chapter 4

Proposed DFT-Based Channel Estimators

Contents

In this chapter, we discuss our first strategy to overcome the limitations of conventional symbol-by-symbol (SBS) estimators, mainly due of the employment of the data-pilot aided (DPA) estimation that suffers from a significant demapping error. In this context, we propose a novel estimation method that is based on truncated discrete Fourier transform (T-DFT) interpolation, where the classical discrete Fourier transform (DFT) interpolation is adapted to the IEEE 802.11p standard structure. In this approach, only the tracking pilots are employed without a need to perform the basic least squares (LS) estimation at the preamble or the DPA estimation. This results in a significant performance improvement since the DPA demapping error is eliminated. Moreover, we show that updating the estimated channels for the received symbols using temporal averaging T-DFT (TA-TDFT) leads to considerable noise alleviation throughout the frame leading to enhanced bit error rate (BER) and normalized mean-squared error (NMSE) performance. The proposed estimators outperform conventional SBS estimators with less computational complexities. Moreover, they are robust against Doppler error and

do not depend on the pre-estimated channel statistics as it is the case for the LMMSE estimator. As a result, both T-DFT and TA-TDFT are robust and can be generalized to be employed in different mobility scenarios.

The remainder of this chapter is organized as follows: in Section 4.1, the classical DFT interpolation concept is illustrated. The proposed T-DFT and TA-TDFT estimators and their analytical NMSE derivations are described in Sections 4.2 and 4.3, respectively. In Section 4.4, the simulation results demonstrate the performance of the proposed schemes in different mobility scenarios. Detailed computational complexity analysis is provided in Section 4.5. Finally, the chapter is concluded in Section 4.6.

4.1 Classical DFT Estimator

Recall the i-th received frequency-domain orthogonal frequency division multiplexing (OFDM) symbol (2.7) and (2.10):

$$\tilde{\boldsymbol{y}}_i[k] = \tilde{\boldsymbol{h}}_i[k]\tilde{\boldsymbol{x}}_i[k] + \tilde{\boldsymbol{e}}_{i,\mathrm{d}}[k] + \tilde{\boldsymbol{v}}_i[k],\ k \in \mathcal{K}_{on}$$

where, $\tilde{\boldsymbol{h}}_i[k] = \frac{1}{K}\sum_{l=0}^{L-1} \bar{\boldsymbol{h}}_i[l,0]e^{-j2\pi\frac{kl}{K}}$, $\tilde{\boldsymbol{e}}_{i,\mathrm{d}}[k]$ is the Doppler interference, and $\tilde{\boldsymbol{v}}_i[k]$ the additive noise. The goal is to estimate $\tilde{\boldsymbol{h}}_i[k]$ at the data subcarriers based on the pilot subcarriers. Let $\boldsymbol{h}_{i,L} \in \mathbb{C}^{L\times 1}$ such that $\boldsymbol{h}_{i,L}[l] = \frac{1}{K}\bar{\boldsymbol{h}}_i[l,0], l = 0\cdots L-1$, the vector model corresponding to the $\mathcal{K}_\mathrm{p}$, and $\mathcal{K}_\mathrm{d}$ subcarriers can be expressed as follows:

$$\begin{cases} \tilde{\boldsymbol{y}}_{p_i} &= (\boldsymbol{F}_p\boldsymbol{h}_{i,L}) \odot \tilde{\boldsymbol{x}}_{\mathrm{p}_i} + \tilde{\boldsymbol{e}}_{p_i,d} + \tilde{\boldsymbol{v}}_{p_i}, \quad k \in \mathcal{K}_\mathrm{p}. \\ \tilde{\boldsymbol{y}}_{d_i} &= (\boldsymbol{F}_d\boldsymbol{h}_{i,L}) \odot \tilde{\boldsymbol{x}}_{\mathrm{d}_i} + \tilde{\boldsymbol{e}}_{d_i,d} + \tilde{\boldsymbol{v}}_{d_i}, \quad k \in \mathcal{K}_\mathrm{d}, \end{cases} \tag{4.1}$$

where $\odot$ denotes the element wise multiplication. $\boldsymbol{F}_d \in \mathbb{C}^{K_d\times L}$ represents the truncated DFT matrix obtained by selecting $\mathcal{K}_\mathrm{d}$ rows and L columns from the $K \times K$ DFT matrix. $\boldsymbol{F}_p \in \mathbb{C}^{K_\mathrm{p}\times L}$ denotes the truncated DFT matrices at $\mathcal{K}_\mathrm{p}$ subcarriers. The pilot signal is used to estimate $\boldsymbol{h}_{i,L}$. First, by dividing over the pilots, we get:

$$\begin{aligned} \hat{\tilde{\boldsymbol{h}}}_{p_i} &= \tilde{\boldsymbol{y}}_{p_i} \oslash \tilde{\boldsymbol{x}}_{\mathrm{p}_i} + \tilde{\boldsymbol{e}}_{p_i,d} \oslash \tilde{\boldsymbol{x}}_{\mathrm{p}_i} + \tilde{\boldsymbol{v}}_{p_i} \oslash \tilde{\boldsymbol{x}}_{\mathrm{p}_i} \\ &= \tilde{\boldsymbol{h}}_{p_i} + \tilde{\boldsymbol{e}}_{p_i,d} \oslash \tilde{\boldsymbol{x}}_{\mathrm{p}_i} + \tilde{\boldsymbol{v}}_{p_i} \oslash \tilde{\boldsymbol{x}}_{\mathrm{p}_i}, \end{aligned} \tag{4.2}$$

where $\oslash$ is the element wise division, and $\tilde{\boldsymbol{h}}_{p_i} = \boldsymbol{F}_p\boldsymbol{h}_{i,L}$. In our work, we consider that the transmitted pilots are equal to one for simplicity. Moreover, after estimating $\hat{\boldsymbol{h}}_{i,L}$ employing $\hat{\tilde{\boldsymbol{h}}}_{p_i}$, the final channel estimate at the K_d subcarriers of the i-th received OFDM symbol can be obtained according to the employed estimator as we discuss in the next subsections, where $\hat{\tilde{\boldsymbol{h}}}_{d_i} = \boldsymbol{F}_d\hat{\boldsymbol{h}}_{i,L}$.

The classical DFT estimator employs DFT interpolation [Biyyam and Bhuma, 2018] in order to obtain the final channel estimates at K_d subcarriers assuming that $\boldsymbol{F}_p$ is either tall or square matrix, i.e. $K_\mathrm{p} \geq L$. Therefore, $\hat{\boldsymbol{h}}_{i,L}$ can be estimated with the LS as:

$$\hat{\boldsymbol{h}}_{i,L} = \boldsymbol{F}_p^\dagger\hat{\tilde{\boldsymbol{h}}}_{p_i} = \boldsymbol{h}_{i,L} + \boldsymbol{F}_p^\dagger(\tilde{\boldsymbol{e}}_{p_i,d} + \tilde{\boldsymbol{v}}_{p_i}). \tag{4.3}$$

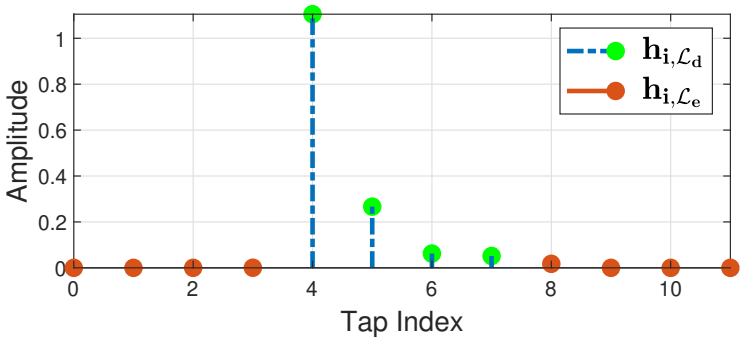


Figure 4.1: Vehicular channel dominant taps selection.

Here, $\boldsymbol{F}_p^{\dagger} = (\boldsymbol{F}_p^{\mathrm{H}}\boldsymbol{F}_p)^{-1}\boldsymbol{F}_p^{\mathrm{H}}$ is the pseudo inverse matrix of $\boldsymbol{F}_p$, and $(\cdot)^{\mathrm{H}}$ denotes the conjugate transpose. Thereafter, the estimate at the data subcarrier denoted as $\hat{\tilde{\boldsymbol{h}}}_{\mathrm{DFT}_i}$, is computed as follows:

$$\hat{\tilde{\boldsymbol{h}}}_{\mathrm{DFT}_i} = \boldsymbol{F}_d\hat{\boldsymbol{h}}_{i,L} = \boldsymbol{F}_d\boldsymbol{h}_{i,L} + \boldsymbol{W}_{\mathrm{DFT}}\boldsymbol{z}_{\mathrm{DFT}_i}, \tag{4.4}$$

where $\boldsymbol{W}_{\mathrm{DFT}} = \boldsymbol{F}_d\boldsymbol{F}_p^{\dagger}$ denotes the DFT interpolation matrix and $\boldsymbol{z}_{\mathrm{DFT}_i} = \tilde{\boldsymbol{e}}_{p_i,d} + \tilde{\boldsymbol{v}}_{p_i}$. The DFT estimation mean squared error (MSE) $\boldsymbol{\epsilon}_{\mathrm{DFT}}$ can be expressed as:

$$\begin{aligned}\epsilon_{\mathrm{DFT}_i} &= \mathrm{E}\left[\|\hat{\tilde{\boldsymbol{h}}}_{\mathrm{DFT}_i} - \tilde{\boldsymbol{h}}_{i,d}\|^2\right] \\ &= \mathrm{trace}\left\{\boldsymbol{W}_{\mathrm{DFT}}\left(\boldsymbol{\Lambda}_{p,d} + \sigma^2\boldsymbol{I}_{K_p}\right)\boldsymbol{W}_{\mathrm{DFT}}^{\mathrm{H}}\right\},\end{aligned} \tag{4.5}$$

where σ^2 denotes the noise variance, $\boldsymbol{\Lambda}_{p,d} = \mathrm{E}\left[\tilde{\boldsymbol{e}}_{p_i,d}\tilde{\boldsymbol{e}}_{p_i,d}^{\mathrm{H}}\right] \in \mathbb{C}^{K_p\times K_p}$ is the auto-correlation matrix of the Doppler error at $\mathcal{K}_\mathrm{p}$ subcarriers, and $\boldsymbol{I}_{K_p} \in \mathbb{C}^{K_p\times K_p}$ denotes the identity matrix. Based on (2.9), $\boldsymbol{\Lambda}_{p,d}$ is diagonal and independent of i.

4.2 Truncated DFT Estimator

IEEE 802.11p standard allocates only four pilots for each transmitted OFDM symbol, while the vehicular channel models may consist of more than 4 taps. For instance, according to the channel models discussed in [Acosta-Marum and Ingram, 2007b], it is found that $L = 12$ taps. However, considering the bandwidth of 10 MHz, the number of discrete significant taps is smaller than 12 as shown in Fig. 4.1. Thus, classical DFT cannot be implemented in IEEE 802.11p standard, since the condition $K_\mathrm{p} \geq L$ is not satisfied. To overcome this limitation, we propose a truncated DFT (T-DFT) estimator that targets the estimation of only $L_\mathrm{d} = K_\mathrm{p}$ dominant channel taps out of L. The indexes of those dominant taps are denoted as $\mathcal{L}_\mathrm{d}$, $L_d = |\mathcal{L}_\mathrm{d}|$, and they are selected based on the maximum values of $\rho[l,0]$ according to the measured channel profile as shown in Fig. 4.1. The remaining minor taps are represented by the set $\mathcal{L}_\mathrm{e}$, $L_e = |\mathcal{L}_\mathrm{e}|$ and they are considered as noise, such that $\mathcal{L}_\mathrm{d} \cup \mathcal{L}_\mathrm{e} = \{0, \cdots, L-1\}$. Let $\boldsymbol{h}_{i,\mathcal{L}_\mathrm{d}} \in \mathbb{C}^{L_d\times 1}$ and $\boldsymbol{h}_{i,\mathcal{L}_\mathrm{e}} \in \mathbb{C}^{L_e\times 1}$ be the

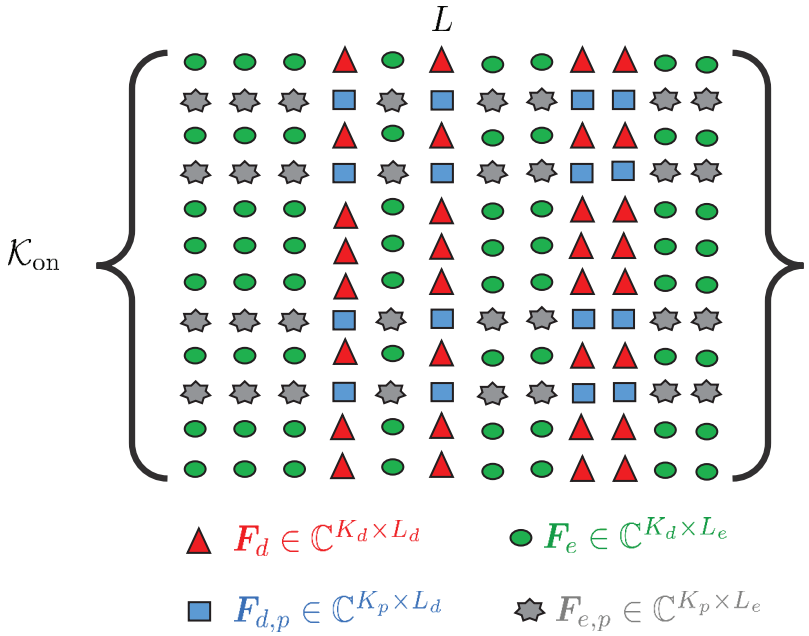


Figure 4.2: Proposed T-DFT truncated DFT matrices.

vectors corresponding to the significant and minor channel taps, the pilot signal (4.2) can be rewritten as:

$$\hat{\tilde{\boldsymbol{h}}}_{i,p} = \boldsymbol{F}_{d,p}\boldsymbol{h}_{i,\mathcal{L}_\text{d}} + \boldsymbol{F}_{e,p}\boldsymbol{h}_{i,\mathcal{L}_\text{e}} + \tilde{\boldsymbol{e}}_{p_i,d} + \tilde{\boldsymbol{v}}_{p_i}, \tag{4.6}$$

where ($\boldsymbol{F}_{d,p} \in \mathbb{C}^{K_\text{p} \times L_d}$, $\boldsymbol{F}_{e,p} \in \mathbb{C}^{K_\text{p} \times L_e}$) denote the truncated DFT matrices at $\mathcal{K}_\text{p}$ subcarriers and ($\mathcal{L}_\text{d}$, $\mathcal{L}_\text{e}$) channel taps respectively, as shown in Fig. 4.2.

Similarly, the channel gain at the data subcarriers can be expressed as:

$$\tilde{\boldsymbol{h}}_{i,d} = \boldsymbol{F}_d\boldsymbol{h}_{i,\mathcal{L}_\text{d}} + \boldsymbol{F}_e\boldsymbol{h}_{i,\mathcal{L}_\text{e}}. \tag{4.7}$$

Here $\boldsymbol{F}_d \in \mathbb{C}^{K_d \times L_d}$ and $\boldsymbol{F}_e \in \mathbb{C}^{K_d \times L_e}$ denote the truncated DFT matrices from selecting $\mathcal{K}_\text{d}$ rows and $\mathcal{L}_\text{d}$, $\mathcal{L}_\text{e}$ columns, respectively. First, $\boldsymbol{h}_{i,\mathcal{L}_\text{d}}$ is estimated using LS as:

$$\begin{aligned}\hat{\boldsymbol{h}}_{i,\mathcal{L}_\text{d}} &= \boldsymbol{F}_{d,p}^\dagger \hat{\tilde{\boldsymbol{h}}}_{i,p} \\ &= \boldsymbol{h}_{i,\mathcal{L}_\text{d}} + \boldsymbol{F}_{d,p}^\dagger \boldsymbol{F}_{e,p}\boldsymbol{h}_{i,\mathcal{L}_\text{e}} + \boldsymbol{F}_{d,p}^\dagger(\tilde{\boldsymbol{e}}_{p_i,d} + \tilde{\boldsymbol{v}}_{p_i}),\end{aligned} \tag{4.8}$$

where $\boldsymbol{F}_{d,p}^\dagger$ is the pseudo inverse matrix of $\boldsymbol{F}_{d,p}$. Therefore, the T-DFT estimator can be expressed as follows:

$$\hat{\tilde{\boldsymbol{h}}}_{\text{T-DFT}_i} = \boldsymbol{F}_d\hat{\boldsymbol{h}}_{i,\mathcal{L}_\text{d}} = \boldsymbol{F}_d\boldsymbol{h}_{i,\mathcal{L}_\text{d}} + \boldsymbol{W}_\text{TDFT}\boldsymbol{z}_{\text{TDFT}_i}, \tag{4.9}$$

where $\boldsymbol{W}_{\text{TDFT}_i} = \boldsymbol{F}_d\boldsymbol{F}_{d,p}^\dagger$ denotes the T-DFT interpolation matrix and $\boldsymbol{z}_{\text{TDFT}_i} = \boldsymbol{F}_{e,p}\boldsymbol{h}_{i,\mathcal{L}_\text{e}} + \tilde{\boldsymbol{e}}_{p_i,d} + \tilde{\boldsymbol{v}}_{p_i}$. The overall estimation error between $\tilde{\boldsymbol{h}}_{i,d}$ and $\hat{\tilde{\boldsymbol{h}}}_{\text{T-DFT}_i}$, denoted as $\hat{e}_\text{TDFT}$, can be expressed as:

$$\begin{aligned}\hat{\boldsymbol{e}}_\text{TDFT} &= \hat{\tilde{\boldsymbol{h}}}_{\text{T-DFT}_i} - \tilde{\boldsymbol{h}}_{i,d} \\ &= \boldsymbol{W}_{\mathcal{L}_\text{e}}\boldsymbol{h}_{i,\mathcal{L}_\text{e}} + \boldsymbol{W}_\text{TDFT}(\tilde{\boldsymbol{e}}_{p_i,d} + \tilde{\boldsymbol{v}}_{p_i}),\end{aligned} \tag{4.10}$$

where $\boldsymbol{W}_{\mathcal{L}_\mathrm{e}} = \boldsymbol{W}_{\mathrm{TDFT}}\boldsymbol{F}_{e,p} - \boldsymbol{F}_e$. Accordingly, the T-DFT MSE can be expressed as:

$$\begin{aligned}\epsilon_{\text{T-DFT}} &= \mathrm{E}\left[\|\hat{\boldsymbol{e}}_{\text{TDFT}_i}\|^2\right] \\ &= \mathrm{trace}\left\{\boldsymbol{W}_{\mathcal{L}_\mathrm{e}}\boldsymbol{\Lambda}_{\mathcal{L}_\mathrm{e}}\boldsymbol{W}_{\mathcal{L}_\mathrm{e}}^{\mathrm{H}}\right\} + \mathrm{trace}\left\{\boldsymbol{W}_{\mathrm{TDFT}}\left(\boldsymbol{\Lambda}_{p,d} + \sigma^2\boldsymbol{I}_{K_p}\right)\boldsymbol{W}_{\mathrm{TDFT}}^{\mathrm{H}}\right\},\end{aligned} \tag{4.11}$$

where $\boldsymbol{\Lambda}_{\mathcal{L}_\mathrm{e}} = \mathrm{E}\left[\boldsymbol{h}_{i,\mathcal{L}_\mathrm{e}}\boldsymbol{h}_{i,\mathcal{L}_\mathrm{e}}{}^{\mathrm{H}}\right] \in \mathbb{C}^{L_e\times L_e}$ represents the auto-correlation matrix of the $\mathcal{L}_\mathrm{e}$ neglected channel taps, which is independent of i. Note that, $\boldsymbol{\Lambda}_{\mathcal{L}_\mathrm{e}}$ is diagonal and $\mathrm{E}\left[\tilde{\boldsymbol{e}}_{p_i,d}\boldsymbol{h}_{i,\mathcal{L}_\mathrm{e}}^{\mathrm{H}}\right] = \boldsymbol{0}$ due to the uncorrelated Delay-Doppler assumption. The classical DFT estimation is a special case when all the taps are considered. In this case, the error term $\mathrm{trace}\left\{\boldsymbol{W}_{\mathcal{L}_\mathrm{e}}\boldsymbol{\Lambda}_{\mathcal{L}_\mathrm{e}}\boldsymbol{W}_{\mathcal{L}_\mathrm{e}}^{\mathrm{H}}\right\}$ is null, and the interpolation matrix $\boldsymbol{W}_{\mathrm{TDFT}}$ becomes $\boldsymbol{W}_{\mathrm{DFT}}$.

4.3 Temporal Averaging TDFT Estimator

Considering the high temporal correlation between the successive channels, $\tilde{\boldsymbol{h}}_{i,d}$ and $\tilde{\boldsymbol{h}}_{i+1,d}$, averaging has the potential of improving the estimation gradually. To demonstrate that, let $\tilde{\boldsymbol{h}}_{i,d} = \tilde{\boldsymbol{h}}_d + \tilde{\boldsymbol{\epsilon}}_i$, where $\tilde{\boldsymbol{h}}_d$ is static and $\tilde{\boldsymbol{\epsilon}}_i$ denotes the variation. Similarly, $\boldsymbol{h}_{i,\mathcal{L}_\mathrm{e}} = \boldsymbol{h}_{\mathcal{L}_\mathrm{e}} + \boldsymbol{\Delta}_{i,\mathcal{L}_\mathrm{e}}$. Thus, using (4.10), we get:

$$\hat{\tilde{\boldsymbol{h}}}_{\text{T-DFT}_i} = \tilde{\boldsymbol{h}}_d + \underbrace{\boldsymbol{W}_{\mathcal{L}_\mathrm{e}}\boldsymbol{h}_{\mathcal{L}_\mathrm{e}}}_{\boldsymbol{c}} + \underbrace{\tilde{\boldsymbol{\epsilon}}_i + \boldsymbol{W}_{\mathcal{L}_\mathrm{e}}\boldsymbol{\Delta}_{i,\mathcal{L}_\mathrm{e}}}_{\boldsymbol{z}_i} + \underbrace{\boldsymbol{W}_{\mathrm{TDFT}}(\tilde{\boldsymbol{e}}_{p_i,d} + \tilde{\boldsymbol{v}}_{p_i})}_{\boldsymbol{\eta}_i}.$$

Note that $\tilde{\boldsymbol{e}}_{p_i,d}\tilde{\boldsymbol{e}}_{p_j,d}^{H} = \delta[i-j]\boldsymbol{\Lambda}_{p,d}$ due to the assumption of uncorrelated data, and $\mathrm{E}\left[\tilde{\boldsymbol{v}}_{p_i}\tilde{\boldsymbol{v}}_{p_j}^{H}\right] = \delta[i-j]\sigma^2\boldsymbol{I}_{K_p}$. As a result, $\boldsymbol{\eta}_i = \boldsymbol{W}_{\mathrm{TDFT}}(\tilde{\boldsymbol{e}}_{p_i,d} + \tilde{\boldsymbol{v}}_{p_i})$ is the error term that can be reduced by means of averaging. On the contrary, the term $\boldsymbol{z}_i = \tilde{\boldsymbol{\epsilon}}_i + \boldsymbol{W}_{\mathcal{L}_\mathrm{e}}\boldsymbol{\Delta}_{i,\mathcal{L}_\mathrm{e}}$, which is a correlated error term with respect to the channel variation, and the fixed error term $\boldsymbol{c} = \boldsymbol{W}_{\mathcal{L}_\mathrm{e}}\boldsymbol{h}_{\mathcal{L}_\mathrm{e}}$ are not reduced by the averaging.

The temporal averaging T-DFT (TA-TDFT) can be achieved such that:

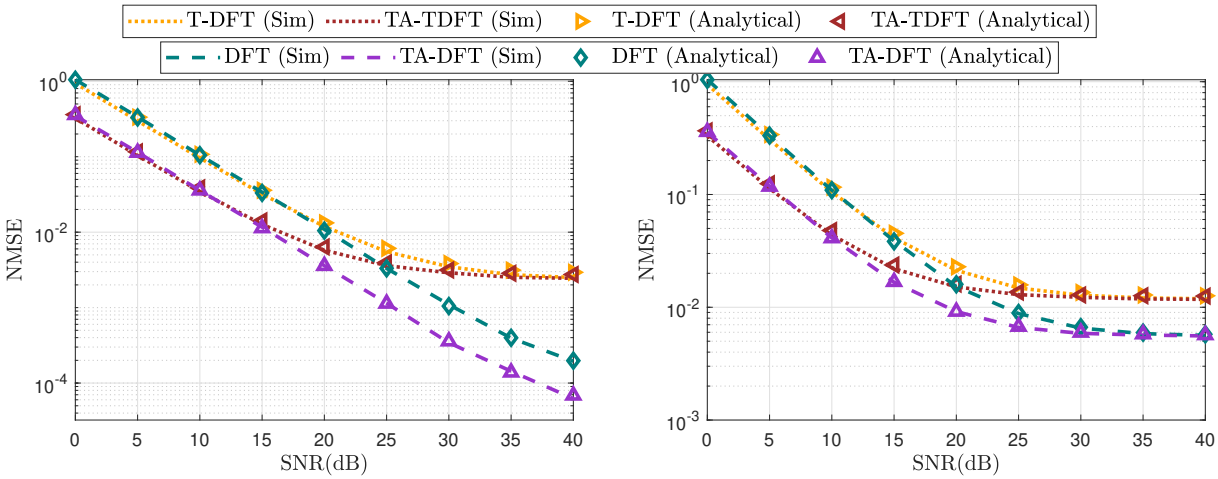
$$\hat{\tilde{\boldsymbol{h}}}_{\text{TA-TDFT}_i} = \begin{cases} \hat{\tilde{\boldsymbol{h}}}_{\text{T-DFT}_i}, & i = 1 \\ \gamma\hat{\tilde{\boldsymbol{h}}}_{\text{TA-TDFT}_{i-1}} + (1-\gamma)\hat{\tilde{\boldsymbol{h}}}_{\text{T-DFT}_i}, & 2 \leqslant i \leqslant I, \end{cases} \tag{4.12}$$

where γ defines the weights given to $\hat{\tilde{\boldsymbol{h}}}_{\text{TA-TDFT}_{i-1}}$ and $\hat{\tilde{\boldsymbol{h}}}_{\text{T-DFT}_i}$. In our simulations, we consider $\gamma = \frac{1}{2}$. Therefore, $\hat{\tilde{\boldsymbol{h}}}_{\text{TA-TDFT}_i}$ can be rewritten in terms of the previous estimated T-DFT channels $\hat{\tilde{\boldsymbol{h}}}_{\text{T-DFT}_i}$ as follows:

$$\begin{aligned}\hat{\tilde{\boldsymbol{h}}}_{\text{TA-TDFT}_i} &= \left(\frac{1}{2}\right)^{(i-1)}\hat{\tilde{\boldsymbol{h}}}_{\text{T-DFT}_1} + \sum_{j=2}^{i}\left(\frac{1}{2}\right)^{(i-j+1)}\hat{\tilde{\boldsymbol{h}}}_{\text{T-DFT}_j} \\ &= \left[\tilde{\boldsymbol{h}}_d + \tilde{\boldsymbol{\epsilon}}_i\right] + \boldsymbol{c} + [\boldsymbol{z}_{\text{TA-TDFT}_i} - \tilde{\boldsymbol{\epsilon}}_i] + \boldsymbol{\eta}_{\text{TA-TDFT}_i}.\end{aligned} \tag{4.13}$$

Here, the overall error terms are given by:

$$\begin{aligned}\boldsymbol{z}_{\text{TA-TDFT}_i} &= \left(\frac{1}{2}\right)^{(i-1)}\boldsymbol{z}_1 + \sum_{j=2}^{i}\left(\frac{1}{2}\right)^{(i-j+1)}\boldsymbol{z}_j, \\ \boldsymbol{\eta}_{\text{TA-TDFT}_i} &= \left(\frac{1}{2}\right)^{(i-1)}\boldsymbol{\eta}_1 + \sum_{j=2}^{i}\left(\frac{1}{2}\right)^{(i-j+1)}\boldsymbol{\eta}_j.\end{aligned} \tag{4.14}$$



(a) Low mobility vehicular channel model. **(b)** High mobility vehicular channel model.

Figure 4.3: NMSE analytical and simulation results for low and high mobility vehicular channel models respectively.

The overall estimation error of TA-TDFT depends on the i-th received OFDM symbol, the signal-to-noise ratio (SNR), and the Doppler shift. At low SNR, the estimation error is highly influenced by the noise, where $\boldsymbol{\eta}_i \approx \boldsymbol{W}_{\text{TDFT}}\tilde{\boldsymbol{v}}_{p_i}$. In this case, the averaging reduces the noise power when i increases, such that:

$$\epsilon^{(1)}_{\text{TA-TDFT}_i} \leq \boldsymbol{N}_{\text{TA-TDFT}_i}\sigma^2 \text{trace}\left\{\boldsymbol{W}_{\text{TDFT}}\boldsymbol{W}^H_{\text{TDFT}}\right\}, \tag{4.15}$$

where

$$\boldsymbol{N}_{\text{TA-TDFT}_i} = \left(\frac{1}{4}\right)^{(i-1)} + \sum_{j=2}^{i}\left(\frac{1}{4}\right)^{(i-j+1)} = \frac{4^{i-1}+2}{3\times 4^{i-1}}. \tag{4.16}$$

The full derivation of (4.16) is listed in Appendix B. We note that, in high SNR region and low mobility, the error is influenced by non-significant channel taps error $\boldsymbol{c}$, and thus,

$$\epsilon^{(2)}_{\text{TA-TDFT}_i} \leq \text{trace}\left\{\boldsymbol{W}_{\text{TDFT}}\boldsymbol{\Lambda}_{\mathcal{L}_\text{e}}\boldsymbol{W}^H_{\text{TDFT}}\right\}. \tag{4.17}$$

At high mobility, the error is influenced by $\boldsymbol{z}_{\text{TA-DFT}_i} - \tilde{\boldsymbol{\epsilon}}_i$, which leads to an increase of the averaging error because of the increase in $\|\tilde{\boldsymbol{\epsilon}}_i\|$. Accordingly, the gain of averaging is notable at low SNRs, not significant at high SNRs with low mobility, and worse at very high SNRs and very high mobility. We note that $\epsilon^{(1)}_{\text{TA-TDFT}_i} + \epsilon^{(2)}_{\text{TA-TDFT}_i}$ denotes the lower bound performance of the TA-TDFT estimator.

Figure 4.3 shows the analytical and simulated NMSE curves of the proposed estimators in low and high mobility scenarios. We note that DFT and TA-DFT estimators employ $K_p = 12$ pilot subcarriers, whereas, $K_p = 4$ is used in the proposed T-DFT and TA-TDFT estimators. It is clearly shown that applying temporal averaging on top of the T-DFT estimator improves the NMSE performance, especially in low SNR region, where the

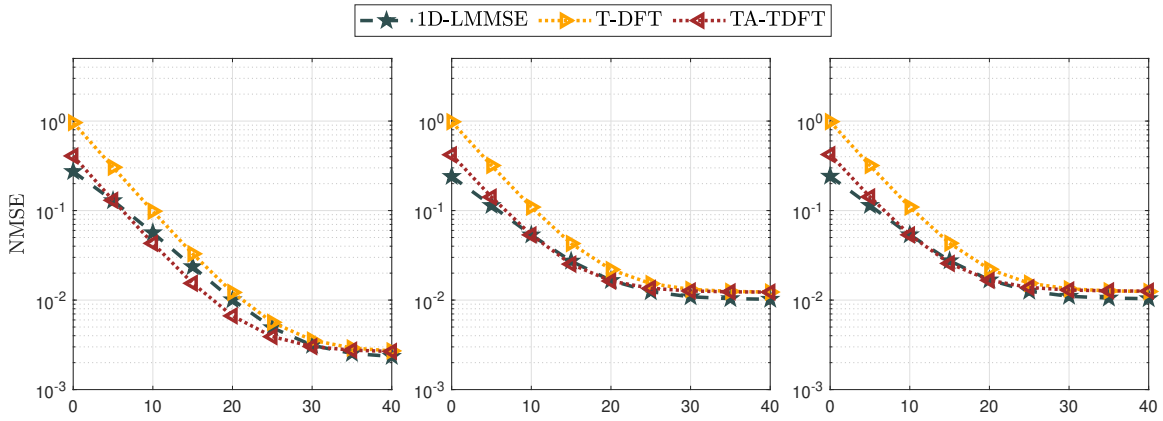
impact of noise is dominant. This is because adding the temporal averaging step to the T-DFT estimated channels, reduces the noise power iteratively over the frame, and thus, the SNR increases resulting in improving the total channel estimation accuracy. Nevertheless, TA-TDFT estimator suffers from an error floor in high SNR region, as a consequence of the channel model error resulting from the neglected $\mathcal{L}_{\mathrm{e}}$ channel taps, in addition to Doppler interference error. This modeling error can be clearly seen in Fig. 4.3a considering low mobility, and the influence of Doppler interference in Fig. 4.3b at high mobility. Moreover, employing the DFT and TA-DFT estimators leads to a significant NMSE performance improvement over the whole SNR region, especially in low mobility scenario, since L channel taps are estimated by the DFT estimator. However, it can be noticed that in high mobility scenario, the temporal averaging step is not effective in reducing the T-DFT estimation error, especially in high SNR region. This is because the influence of Doppler error is dominant over the noise impact. Finally, we note that if more pilot subcarriers are employed i.e. $K_{\mathrm{p}} = 12$, transmission data rates loss is recorded according to the modulation order and coding used. Since the number of data subcarriers K_{d} becomes 40 instead of 48 data subcarriers per OFDM symbol.

4.4 Simulation Results

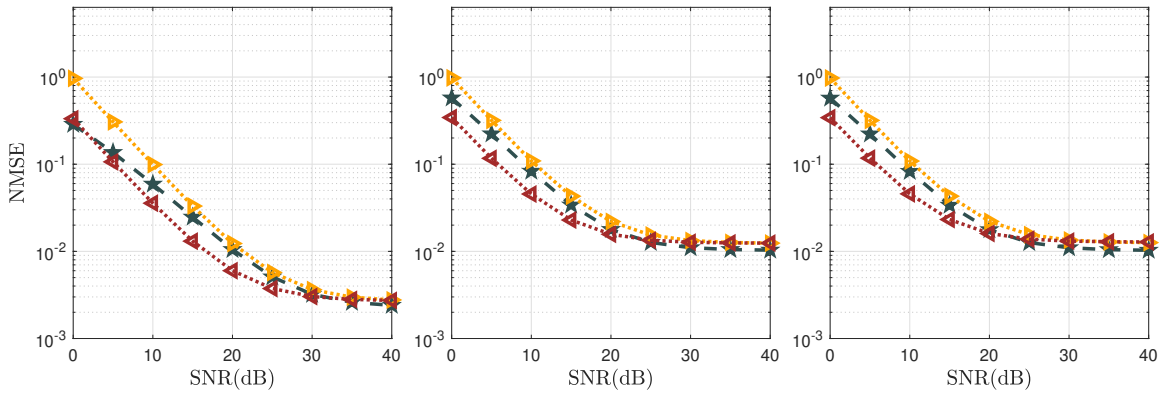
In this section, NMSE and BER simulations are conducted in order to evaluate the performance of the proposed estimators to 1D-LMMSE in the three vehicular channel models as discussed in 3.2 and considering the modulation order, mobility, and frame length criteria. Moreover, a robustness analysis is presented in order to evaluate the generalization ability of the studied estimators.

4.4.1 NMSE Evaluation

As the estimation techniques in this section are all linear and do not employ demapping, the estimation error is independent of the modulation order. In addition, as the estimation is performed in SBS fashion, only TA-TDFT experience dependency on the frame length since it employs averaging. This behavior can be seen in Figure. 4.4. In particular, when the frame length is $I = 100$, the estimation error of TA-TDFT is slightly reduced at low mobility and low SNRs compared with shorter frames, as expected. In comparison with 1D-LMMSE , the T-DFT only approaches 1D-LMMSE at high SNR, whereas TA-TDFT outperforms 1D-LMMSE when the averaging is sufficient, which is the case of using a long frame ($I = 100$) at low SNRs, or in medium SNR region with a short frame ($I = 10$). However, at high mobility and high SNRs, the estimation error is influenced by Doppler interference as well as the channel model mismatch resulting from neglecting $\mathcal{L}_{\mathrm{e}}$ channel taps in the estimation process for all of the presented estimators. Fig. 4.5 shows the average Doppler interference as f_{d} increases.



(a) NMSE performance with $I = 10$.



(b) NMSE performance with $I = 100$.

Figure 4.4: NMSE performance employing three scenarios: (i) first column - low mobility ($v = 45$ Kmph, $f_d = 250$ Hz) (ii) second column - high mobility ($v = 100$ Kmph, $f_d = 500$ Hz) (iii) third column - very high mobility ($v = 200$ Kmph, $f_d = 1000$ Hz).

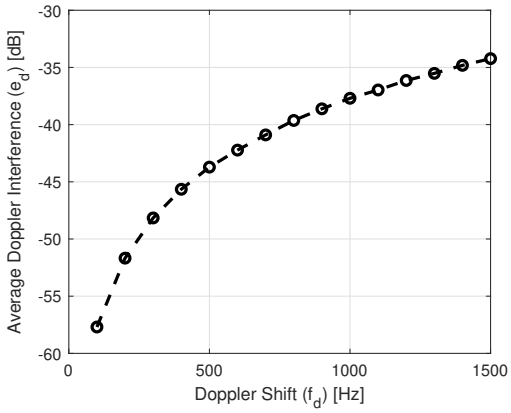
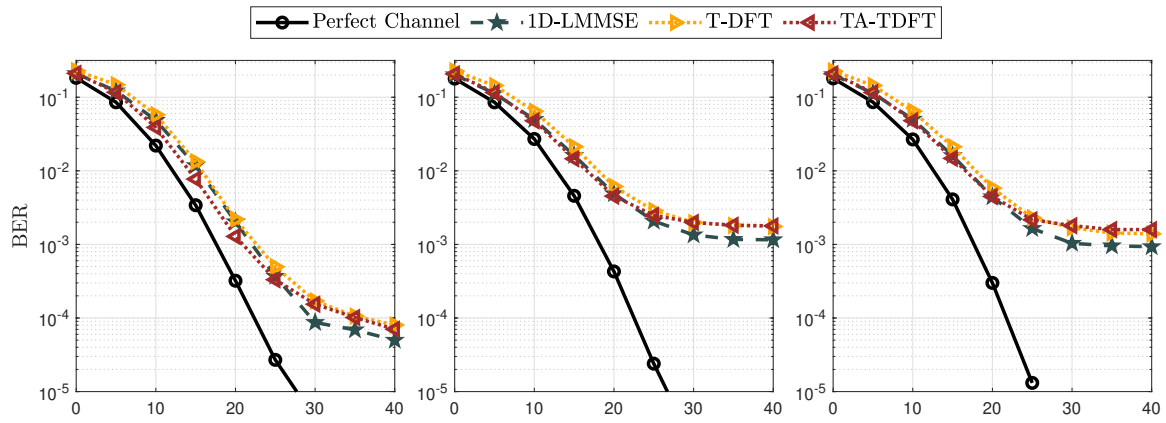
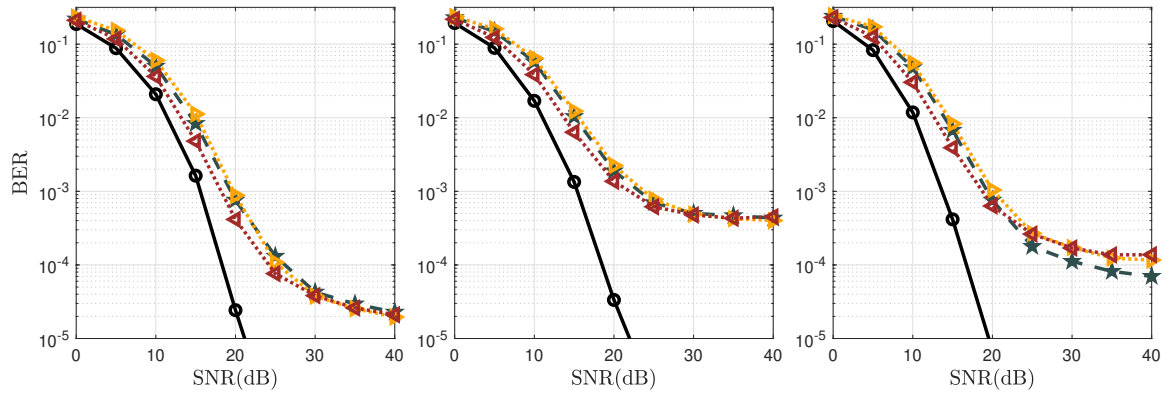


Figure 4.5: Doppler spectrum Interference (maximum speed $v = 290$ Kmph).



(a) BER performance with $I = 10$.



(b) BER performance with with $I = 100$.

Figure 4.6: BER performance employing QPSK: (*i*) first column - low mobility ($v = 45$ Kmph, $f_d = 250$ Hz) (*ii*) second column - high mobility ($v = 100$ Kmph, $f_d = 500$ Hz) (*iii*) third column - very high mobility ($v = 200$ Kmph, $f_d = 1000$ Hz).

For low mobility vehicular scenarios, where $f_{\mathrm{d}} < 200$ Hz, we can notice that ϵ_{d} is almost negligible. However, for high and very high mobility vehicular scenarios, where $f_{\mathrm{d}} > 500$ Hz, Doppler interference starts recording a considerable impact on the overall performance. We note that, the proposed estimators show a significant robustness against Doppler error similar to the conventional 1D-LMMSE estimator. Moreover, they adhere to the IEEE 802.11p standard structure, and the transmission data rates in all modulation orders are preserved. Moreover, employing T-DFT and TA-TDFT estimators outperform conventional estimators with a considerable performance improvement and significant decrease in the overall computational complexity as shown in Section 4.5.

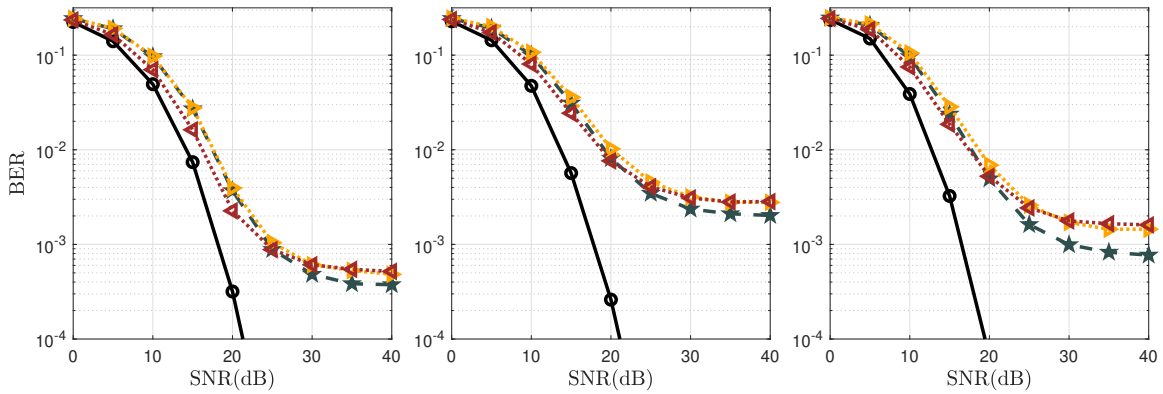


Figure 4.7: BER performance employing $I = 100$, 16QAM: (*i*) first column - low mobility ($v = 45$ Kmph, $f_d = 250$ Hz) (*ii*) second column - high mobility ($v = 100$ Kmph, $f_d = 500$ Hz) (*iii*) third column - very high mobility ($v = 200$ Kmph, $f_d = 1000$ Hz).

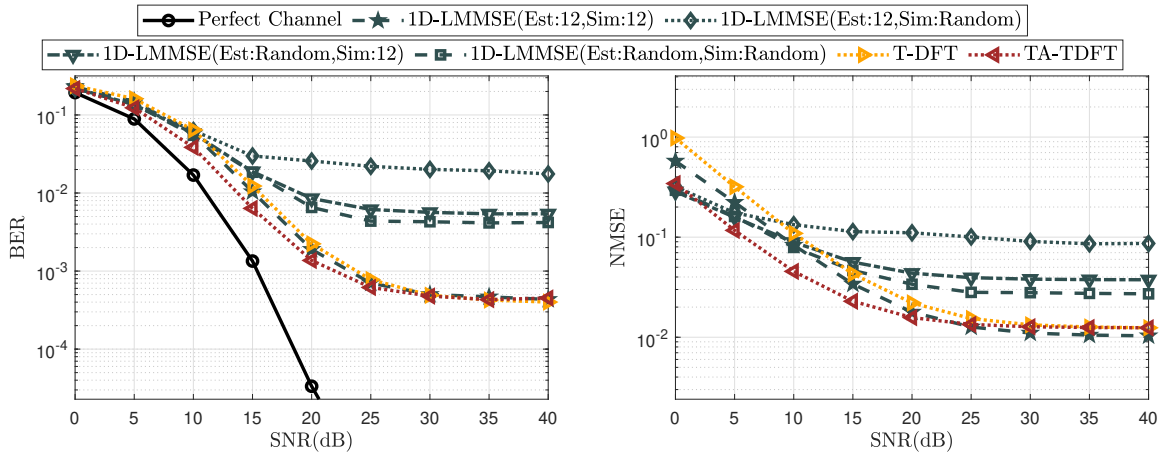
4.4.2 BER Evaluation

Figure 4.6 depicts the BER performance employing QPSK modulation and different frame length in low mobility, high mobility, and very high mobility vehicular scenarios. We can notice that the impact of employing temporal averaging on top of the T-DFT estimator to improve the overall performance can be observed, where the TA-TDFT estimator outperforms T-DFT estimator by up to 3 dB gain in terms of SNR for a BER $= 10^{-3}$ for a long frame ($I = 100$), whereas the impact in a short frame ($I = 10$) is less significant. These results are aligned with the estimation accuracy shown in Figure. 4.4.

The impact of mobility and frame length can be observed in the significant decrease of BER when using long frames. This is due to the use of a long codeword and the harvested time diversity gain. The time diversity gain increases with the increase of the Doppler spread, as can be seen by comparing the case of high mobility ($f_d = 500$) and very high mobility ($f_d = 1000$). Nevertheless, as the channel estimation error increases with the increase of Doppler frequency, the net gain from the time diversity and channel estimation loss depends also on the frame length. For instance, assuming a perfect channel, there is a small net gain with ($I = 100$), when the Doppler shift increases from $f_d = 250$ in low mobility to $f_d = 500$ in high mobility, but it turns to a small loss for short frames ($I = 10$). The perfect estimation is only influenced by the Doppler interference. With the induced estimation error, this impact becomes remarkable. The performance of the presented SBS estimators is significantly degraded by the mobility increase, however it is improved in high mobility for long frames. This observation is also valid for high modulation orders such as 16QAM, as shown in Figure. 4.7

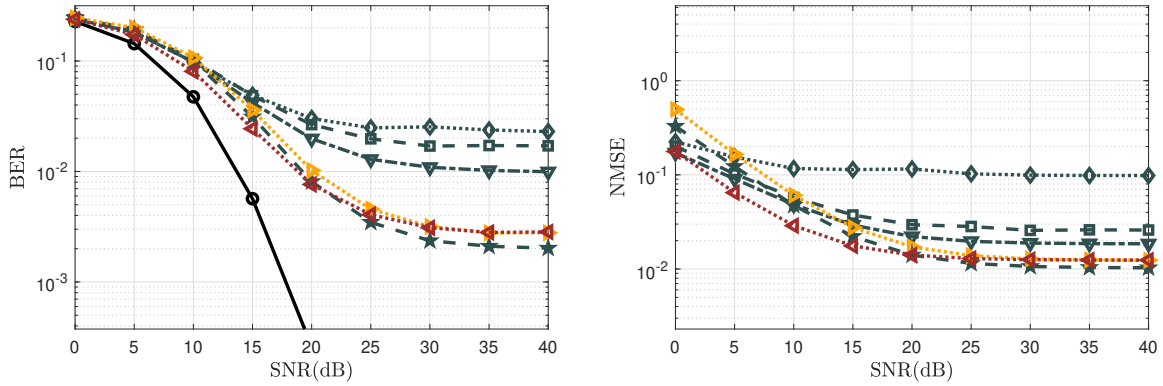
4.4.3 Robustness Analysis

This section compares the robustness and generalization ability of the proposed T-DFT and TA-TDFT estimators with the 1D-LMMSE estimator. As discussed in Section 3.1.9,



(a) BER performance employing QPSK.

(b) NMSE performance employing QPSK.



(c) BER performance employing 16QAM.

(d) NMSE performance employing 16QAM.

Figure 4.8: Robustness analysis of the proposed DFT-based estimators versus the 1D-LMMSE estimator in high mobility channel model ($f_d = 500$ Hz).

the 1D-LMMSE estimator depends mainly on the pre-estimated $\boldsymbol{W}_{\text{1D-LMMSE}}$ matrix that is calculated offline. This offline pre-estimation affects the robustness and generalization ability of the 1D-LMMSE estimator when the channel changes in real-time scenarios. To further illustrate this idea, we simulate the 1D-LMMSE estimator in four different cases, such that:

- ▷ Case 1 (Est:12,Sim:12): $\boldsymbol{W}_{\text{1D-LMMSE}}$ is pre-estimated using 12 taps channel model and we consider that the number of taps are fixed also in the real-time scenario.
- ▷ Case 2 (Est:12,Sim:Random): $\boldsymbol{W}_{\text{1D-LMMSE}}$ is pre-estimated using 12 taps channel model. However, the number of channel taps changes randomly in the real-time scenario. We note that the "Random" term means that the simulation is carried using a random number of paths, where in each iteration, a random number of paths between 1 and 11 is eliminated from the channel power delay profile. Therefore, we obtain a generalized channel model where the number of taps varies.

- ▷ Case 3 (Est:Random,Sim:12): $\boldsymbol{W}_{\text{1D-LMMSE}}$ is pre-estimated where the number of channel taps varies randomly, while the channel taps are fixed to 12 in the real-time scenario.
- ▷ Case 4 (Est:Random,Sim:Random): Random number of channel taps are used in both offline estimation of $\boldsymbol{W}_{\text{1D-LMMSE}}$ and in the real-time scenario.

Figure 4.8 depicts the simulation results of these four cases in high mobility scenario employing both QPSK and 16QAM modulation orders. We can observe that the 1D-LMMSE performance is severely degraded when the channel changes in real-time scenario, whereas, the proposed TDFT and TA-TDFT estimators reveal a good robustness since they are implemented without any dependency on the channel model statistics. Therefore, the proposed TDFT and TA-TDFT estimators are more robust than the 1D-LMMSE estimator in real-time scenarios.

4.5 Computational Complexity Analysis

Compared to the conventional estimators, T-DFT estimator does not require the DPA as a pre-estimation step. It only requires the LS estimated channel at pilot subcarriers $\hat{\hat{\boldsymbol{h}}}_{i,p}$ that requires $2K_{\text{p}}$ divisions. It is worth mentioning that $\boldsymbol{W}_{\text{MDFT}}$ is computed offline, thus the computational complexity of the T-DFT estimator lies in multiplying $\boldsymbol{W}_{\text{MDFT}}$ with $\hat{\hat{\boldsymbol{h}}}_{i,p}$ which requires $K_pK_d - K_d$ complex-valued summations, and K_pK_d complex-valued multiplications, this is equivalent to $5K_{\text{p}}K_d - 2K_d$ real-valued summations, and $4K_pK_d$ real-valued multiplications. Therefore, the overall computational complexity of the T-DFT estimator is $2K_p + 4K_pK_d$ real-valued multiplications/divisions, and $5K_pK_d - 2K_d$ real-valued summations.

TA-TDFT estimator applies simple time averaging on top of the T-DFT estimator that requires additional $4K_d$ real-valued multiplications, and $2K_d$ real-valued summations. As a result, TA-TDFT estimator requires $2K_p + 4K_d + 4K_pK_d$ real-valued multiplications/divisions, and $5K_pK_d$ real-valued summations. TA-TDFT estimator is also less complex than the conventional IEEE 802.11p estimators, while it records a complexity higher than T-DFT estimator by 17.56% as illustrated in 4.9.

Table 4.1 shows a detailed computational complexities percentages analysis for the proposed estimators compared to the conventional estimators, where the proposed

Table 4.1: Computational complexity analysis of the studied estimators.

Est.	E-TRFI	iCDP	TRFI	CDP	STA	1D-LMMSE	
						Conventional	Low-complexity
Proposed T-DFT	-97.59%	-66.82%	-46.33%	-29.81%	-26.78%	-54.13%	same
Proposed TA-TDFT	-97.16%	-61%	-36.91%	-16.31%	-13.92%	-46.08%	+17.56%

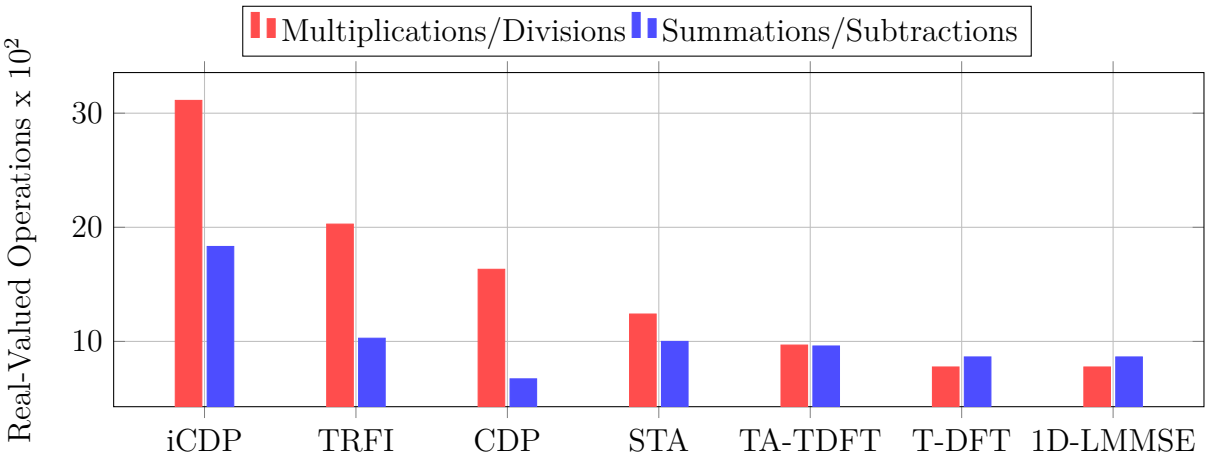


Figure 4.9: Detailed computational complexity of the studied vehicular channel estimators in terms of real-valued operations.

T-DFT and TA-TDFT estimators record 54.13% and 46.68% computational complexity decrease compared to the conventional 1D-LMMSE estimator, respectively. Whereas, the proposed T-DFT estimator requires similar complexity as the low-complexity 1D-LMMSE. Fig. 4.9, shows a bar graph for the required multiplications/divisions, and summations/subtractions by the studied estimators including the low-complexity 1D-LMMSE estimator. It can be noticed that employing both T-DFT and TA-TDFT instead of the conventional estimators leads to a significant decrease in the overall computational complexity, with a considerable gain in the BER and NMSE performance. In addition, the proposed T-DFT and TA-TDFT are more robust than the 1D-LMMSE estimator in highly dynamic real-time scenarios.

4.6 Conclusion

In this chapter, we have focused on overcoming the limitations of vehicular SoA SBS conventional estimators represented by their dependency on the DPA estimation, where the demapping error and the noise are enlarged during the estimation process. To overcome these limitations, we have proposed low-complexity and robust channel estimators for vehicular communications, namely T-DFT and TA-TDFT. Unlike classical estimators, the proposed estimators are based on the truncated DFT-based interpolation, and do not require the DPA estimation as an initial estimation step. Therefore, the enlarged DPA demapping error, and the noise enhancement that results from the basic LS estimator are totally eliminated in the proposed estimators. In addition to that, the proposed estimators do not depend on the pre-estimated channel statistics as it is the case for 1D-LMMSE, thus, they are insensitive against channel model changes in real-time scenarios. Therefore, they are more suitable to be employed in practical systems. Simulation and analytical results have shown the performance superiority of the proposed

T-DFT and TA-TDFT estimators over conventional estimators in low and high mobility vehicular scenarios with a significant computational complexity decrease. Finally, we note that more performance gain could be achieved by employing deep learning based estimators as we discuss in the next chapters. However, this performance gain leads to an increase in the required computational complexity, hence a performance-complexity trade-off must be taken into consideration to select the appropriate estimator according to the application requirements.

Chapter 5

Proposed DL-based SBS Channel Estimators

Contents

Practical wireless communication systems may encounter noise imperfections due to the multi-path and Doppler shift effects. In doubly-dispersive channel, conventional symbol-by-symbol (SBS) channel estimators suffer from a considerable performance degradation as we have discussed in Chapter 3. We recall that, in such SBS estimators, the previous estimated channel generated from the demapped data subcarriers is used as a preamble for the current orthogonal frequency division multiplexing (OFDM) symbol. This process is influenced by the demapping error, which depends on the accuracy of the previous estimation, the Doppler interference, and the noise level. Moreover, this error propagates and increases from one symbol to another all over the frame causing a considerable reliability degradation. Therefore, there is a crucial need to improve the channel estimation accuracy while preserving low computational complexity.

Recently, the rapid advancements in deep learning (DL) and their successful applications in several domains, have sparked significant interest to adopt DL techniques for

wireless communication applications including channel estimation. DL techniques are characterized by robustness, low-complexity, and good generalization ability making their integration into communication systems beneficial. For the SBS channel estimation, we employ both deep neural network (DNN) and long short-term memory (LSTM) networks. Both networks are able to learn the channel frequency correlation besides correcting conventional estimator errors. However, LSTM network is more beneficial when time correlation also exists between successive samples, as in the case of doubly-selective channels. This is because LSTM takes the previous output as an additional source with the current input to produce the current output. Motivated by these advantages, DL algorithms have been proposed to be integrated with conventional channel estimators in vehicular communications. Accordingly, this chapter presents our second strategy for improving the channel estimation using DL techniques in SBS mode.

In this context, DNN is first employed after the conventional spectral temporal averaging (STA) and time domain reliable test frequency domain interpolation (TRFI) estimators. The DNN is able to learn the channel correlation and corrects the estimation error of the conventional STA and TRFI estimators. After that, the proposed LSTM-based estimator is presented. We discover that employing LSTM instead of DNN in the channel estimation is more appropriate and lead to a significant performance improvement. This is due to the fact that LSTM network is able to learn the time correlation between successive OFDM symbols. However, this comes at the cost of higher complexity. This chapter presents the proposed DL-based SBS estimators for vehicular communications and compares them with the conventional estimators showing the advantage of employing DL techniques over the conventional ones. The results of this chapter have been published in [Gizzini et al., 2020a, Gizzini et al., 2020b, Gizzini et al., 2021a].

5.1 STA-DNN

The proposed STA-DNN channel estimator is mainly based on applying a DNN processing after the conventional STA channel estimation as shown in Fig. 5.1 to improve the estimation performance. As discussed in Section 3.1.2, the final STA estimated channel is a linear combination of the previous STA estimated channel and the current data-pilot aided (DPA) estimated channel. Therefore, by integrating DNN as an additional module in the STA estimator, we add a non-linear processing unit to capture the non-linear dependencies between the previous and the current channel. The DNN captures more features of the time-frequency correlations of the channel samples. Thus, the DNN implicitly corrects the estimation error of the conventional STA. This is achieved by minimizing the mean squared error (MSE) between the ideal channel $\tilde{\boldsymbol{h}}_i$ and STA estimated channel $\hat{\tilde{\boldsymbol{h}}}_{\text{STA}_i}$ as follows:

$$\text{MSE}_{\text{STA-DNN}} = \frac{1}{N_T} \sum_{i=1}^{N_T} \|\tilde{\boldsymbol{h}}_i - \hat{\tilde{\boldsymbol{h}}}_{\text{STA}_i}\|^2, \tag{5.1}$$

where N_T represents the number of samples considered during the DNN training. After performing STA estimation, the channel estimated gains are converted from the complex to real-valued domain to be introduced to the DNN input. Therefore, $\hat{\hat{\boldsymbol{h}}}_{\mathrm{STA}_i}$ is processed according to the vector realization function:

$$f_{\mathbb{R}}(V) = [\Re(V); \Im(V)]. \tag{5.2}$$

Thereafter, $\hat{\hat{\boldsymbol{h}}}^{(R)}_{\mathrm{STA}_i} \in \mathbb{R}^{2\mathcal{K}_{on}\times 1}$ is fed as an input to STA-DNN. Finally, the corrected STA channel estimates $\hat{\hat{\boldsymbol{h}}}_{\text{STA-DNN}_i}$, are converted back to K_{on} complex valued domain.

We note that the STA estimator takes into consideration the time and frequency correlation of successive received symbols. However, it suffers from performance degradation in real case scenarios due to fixing the coefficients α and β. Moreover, accurate estimation of α and β requires the knowledge of the channel characteristics, which is hard to obtain in practice. Thus, by employing DNN processing after the STA estimated channels, we implicitly overcome the performance loss of conventional STA, and the errors resulting from fixing α and β values.

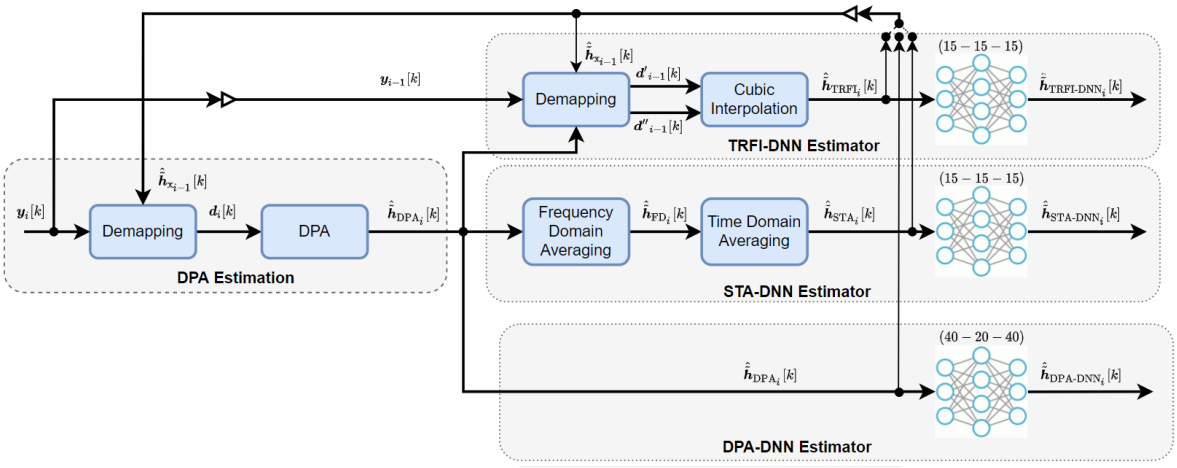


Figure 5.1: Proposed DNN-based estimators.

5.2 TRFI-DNN

Conventional TRFI estimator assumes high correlation between two successive received OFDM symbols to estimate the channel of the reliable subcarriers (RS) and improve the estimation of the unreliable subcarriers (URS) by applying frequency-domain cubic interpolation as presented in Section 3.1.5. When channel encounters fast fading, this assumption becomes invalid and interpolation errors occur. As a result, the number of RS subcarriers decreases as the channel variation increases. Therefore, adding DNN as a post processing unit to the conventional TRFI estimator, higher order channel statistics can be learned by the DNN as in the case of STA-DNN, in addition to correcting the

interpolation errors caused by the low-resolution interpolation. The proposed TRFI-DNN estimator proceeds similar as the STA-DNN estimator, where the conventional TRFI channel estimate is converted from complex to real valued domain by stacking the real and imaginary values in one vector such that $\hat{\tilde{\boldsymbol{h}}}^{(R)}_{\mathrm{TRFI}_i} \in \mathbb{R}^{2\mathcal{K}_{on}\times 1}$. Then, $\hat{\tilde{\boldsymbol{h}}}^{(R)}_{\mathrm{TRFI}_i}$ is fed as an input to the TRFI-DNN. Finally, the corrected TRFI complex-valued channel estimate $\hat{\tilde{\boldsymbol{h}}}_{\text{TRFI-DNN}_i}$ is obtained by converting the real-valued output vector of TRFI-DNN to the complex-valued vector. The proposed TRFI-DNN estimator aims to minimize the MSE given by

$$\mathrm{MSE}_{\text{TRFI-DNN}} = \frac{1}{N_{\mathrm{Train}}} \sum_{i=1}^{N_{\mathrm{Train}}} \left\| \tilde{\boldsymbol{h}}_i - \hat{\tilde{\boldsymbol{h}}}_{\mathrm{TRFI}_i} \right\|^2 . \tag{5.3}$$

Figure 5.1 shows the detailed block diagram of the proposed STA-DNN and TRFI-DNN estimators, besides the recently proposed SoA DNN-based estimator denoted as DPA-DNN [Han et al., 2019], that employs a three hidden layer DNN after the conventional DPA estimation.

5.3 LSTM-DPA-TA

Motivated by the fact that the vehicular channel can be modeled as a time series due to the time correlation between successive symbols, it is more convenient to employ the LSTM network in the channel estimation instead of simple DNN as proposed for STA-DNN and TRFI-DNN. Moreover, more performance gain can be achieved by using the LSTM network before the DPA estimation to reduce the demapping error. The proposed LSTM-based estimator proceeds as follows:

LSTM-based prediction

The first step is to estimate the channel for the current received OFDM symbol employing the previous estimated channel $\hat{\tilde{\boldsymbol{h}}}_{\text{DL-TA}_{i-1,d}[k]}$. Then, LSTM units are employed to implicitly learn the channel correlation between successive received OFDM symbols. Having LSTM as a pre-processing to the DPA estimation is able to significantly prevent having a high DPA demapping error. The i-th LSTM unit input is denoted by $\tilde{\bar{\boldsymbol{x}}}_i \in \mathbb{R}^{2K_{on}\times 1}$, which is obtained by applying complex to real-valued conversion to

$$\bar{\boldsymbol{x}}_i = \begin{cases} \hat{\tilde{\boldsymbol{h}}}_{\mathrm{LSTM}_{i-1,d}}[k], & k \in \mathcal{K}_{\mathrm{d}} \\ \hat{\tilde{\boldsymbol{h}}}_{i-1,p}[k], & k \in \mathcal{K}_{\mathrm{p}} \end{cases}, \tag{5.4}$$

and stacking its real and imaginary values in one vector. We note that $\hat{\tilde{\boldsymbol{h}}}_{i-1,p}[k]$ denotes the least squares (LS) estimated channel at the $\mathcal{K}_{\mathrm{p}}$ subcarriers. Then, $\tilde{\bar{\boldsymbol{x}}}_i$ is processed by the LSTM unit, such that

$$\hat{\tilde{\boldsymbol{h}}}_{\mathrm{LSTM}_{i,d}} = \Omega_{\mathrm{LSTM}}(\tilde{\bar{\boldsymbol{x}}}_i, \Theta), \tag{5.5}$$

where Ω_{LSTM} is the LSTM unit processing with overall weights denoted by Θ.

DPA estimation

The LSTM estimated channel undergoes DPA estimation using the i-th received OFDM symbol similarly as done in the conventional DPA estimation, however, here the previous LSTM-based estimated channel is used such that:

$$\hat{\boldsymbol{h}}_{\text{eq-LSTM}_i}[k] = \frac{\boldsymbol{y}_i[k]}{\hat{\hat{\boldsymbol{h}}}_{\text{LSTM}_{i-1}}[k]}, \ \hat{\hat{\boldsymbol{h}}}_{\text{LSTM}_0}[k] = \hat{\hat{\boldsymbol{h}}}_{\text{LS}}[k]. \tag{5.6}$$

Then, $\hat{\boldsymbol{h}}_{\text{eq-LSTM}_i}[k]$ is demapped to the nearest constellation points resulting in the $\hat{\boldsymbol{d}}_{\text{LSTM}_i}[k]$. The final DPA estimation is performed as follows:

$$\hat{\hat{\boldsymbol{h}}}_{\text{LSTM-DPA}_i}[k] = \frac{\boldsymbol{y}_i[k]}{\hat{\boldsymbol{d}}_{\text{LSTM}_i}[k]}. \tag{5.7}$$

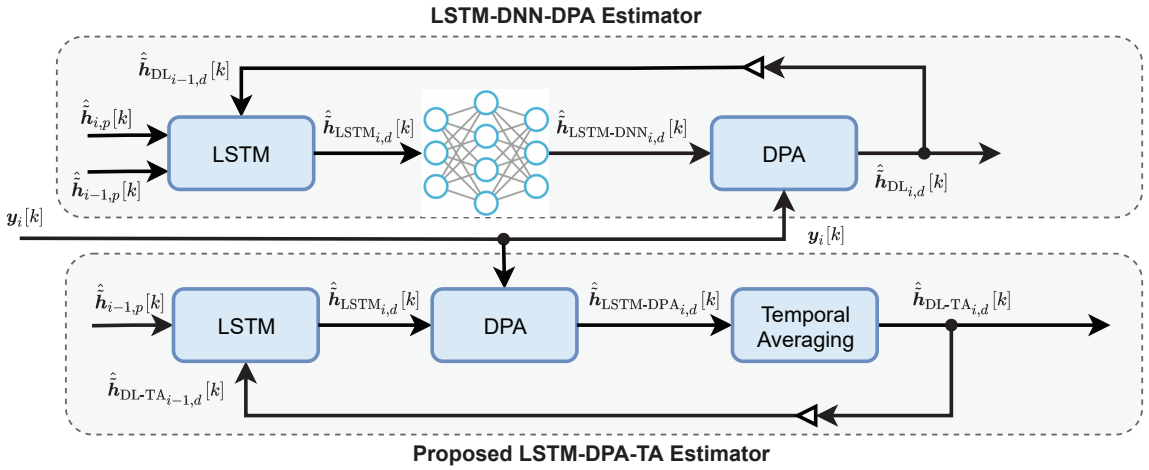


Figure 5.2: Proposed LSTM-based channel estimators block diagram.

TA processing

In order to alleviate the impact of noise, time averaging (TA) processing is applied to the $\hat{\hat{\boldsymbol{h}}}_{\text{LSTM-DPA}_i}[k]$ such that:

$$\hat{\hat{\boldsymbol{h}}}_{\text{DL-TA}_{i,d}} = (1 - \frac{1}{\alpha})\hat{\hat{\boldsymbol{h}}}_{\text{DL-TA}_{i-1,d}} + \frac{1}{\alpha}\hat{\hat{\boldsymbol{h}}}_{\text{LSTM-DPA}_{i,d}}. \tag{5.8}$$

Here we used a fixed $\alpha = 2$ for simplicity. Therefore, the AWGN noise power σ^2 is degraded iteratively within the received OFDM frame according to the ratio

$$R_{\text{DL-TA}_q} = \left(\frac{1}{4}\right)^{(q-1)} + \sum_{j=2}^{q}\left(\frac{1}{4}\right)^{(q-j+1)} = \frac{4^{q-1}+2}{3\times 4^{q-1}}. \tag{5.9}$$

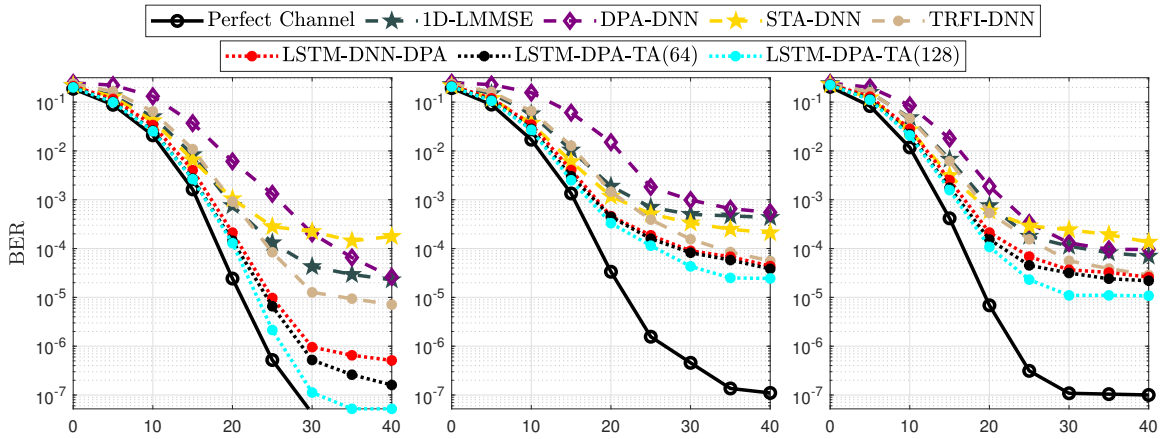
$R_{\text{DL-TA}_q}$ denotes the noise power ratio of the estimated channel at the q-th estimated channel, where $1 < q < I+1$ and $R_{\text{DL-TA}_1} = 1$ denotes the noise power ratio at $\hat{\hat{\boldsymbol{h}}}_{\text{LS}}[k]$.

Table 5.1: Proposed DL-based estimators parameters.

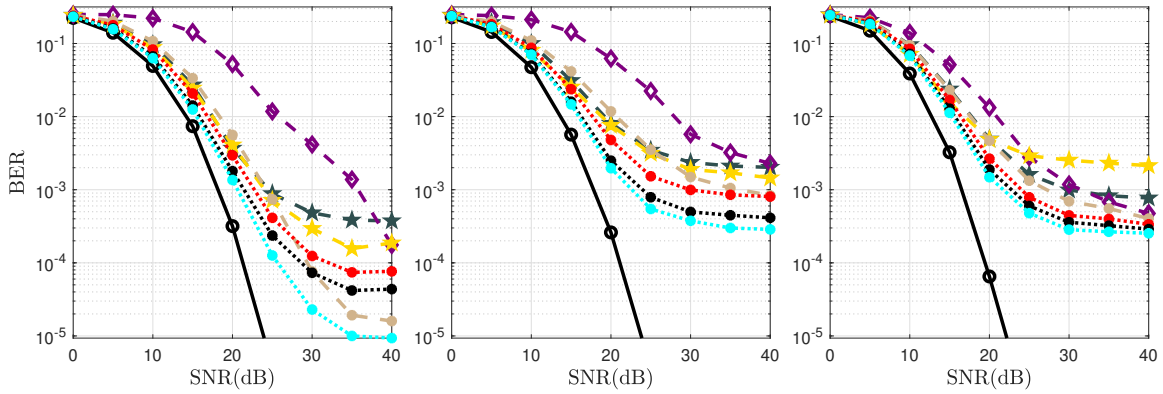
DNN (Hidden layers; Neurons per layer)	(3;15-15-15)
LSTM (Hidden layers; Neurons per layer)	(1;128)
Activation function	ReLU ($y = \max(0, x)$)
Number of epochs	500
Training samples	800000
Testing samples	200000
Batch size	128
Optimizer	ADAM
Loss function	MSE
Learning rate	0.001
Training SNR	40 dB

The full derivation of (5.9) is provided in Appendix A. It can be seen from the derivation of $R_{\text{DL-TA}_q}$ that the noise power decreases iteratively over the received OFDM frame, and hence, the SNR increases which leads to better overall performance. We note that, integrating DNN and LSTM processing blocks with the existing conventional estimators is compatible with the standard as it does not require a major change in the physical layer of the standard.

Intensive experiments show that the performance of DL networks highly depends on the SNR considered in the training. The training performed at the highest SNR value provides the best performance. In fact, when the training is performed at a high SNR value, the DL network is able to learn better the channel, because the impact of the channel is higher than the impact of the noise in this SNR range. Thanks to the good generalization properties of DL, trained networks can still estimate the channel even if the noise is increased i.e. low SNR values. Therefore, the proposed DNN and LSTM based estimators training is performed using signal-to-noise ratio (SNR) = 40 dB to achieve the best performance. Moreover, intensive experiments are performed using the grid search algorithm [Pontes et al., 2016] in order to select the best suitable DNN and LSTM hyper parameters in terms of both performance and complexity. Fig. 5.2 shows the proposed LSTM-based estimator with the recently proposed SoA LSTM-DNN-DPA estimator. Moreover, Table. 5.1 presents the proposed DNN and LSTM- based parameters.



(a) BER performance employing QPSK.



(b) BER performance employing 16QAM.

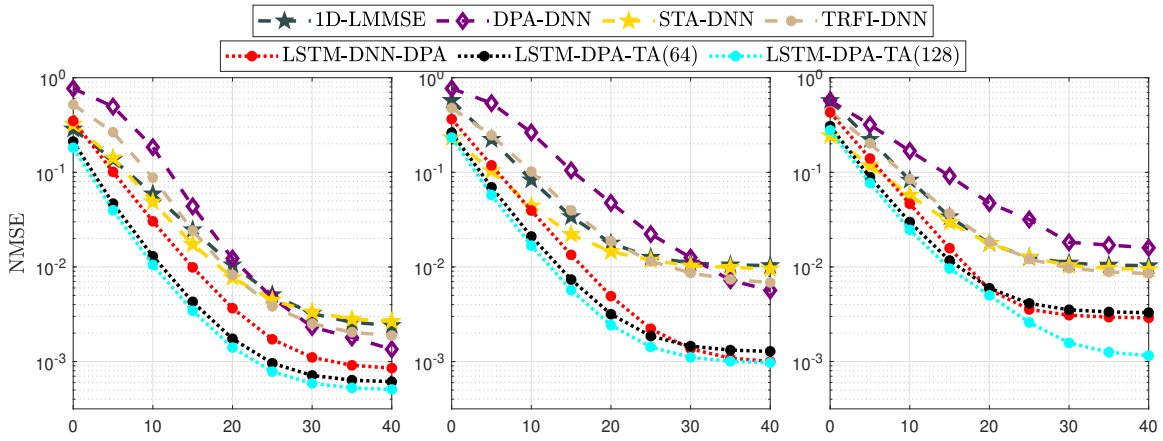
Figure 5.3: BER for $I = 100$, mobility from left to right: low ($v = 45$ Kmph, $f_d = 250$ Hz), high ($v = 100$ Kmph, $f_d = 500$ Hz), very high ($v = 200$ Kmph, $f_d = 1000$ Hz).

5.4 Simulation Results

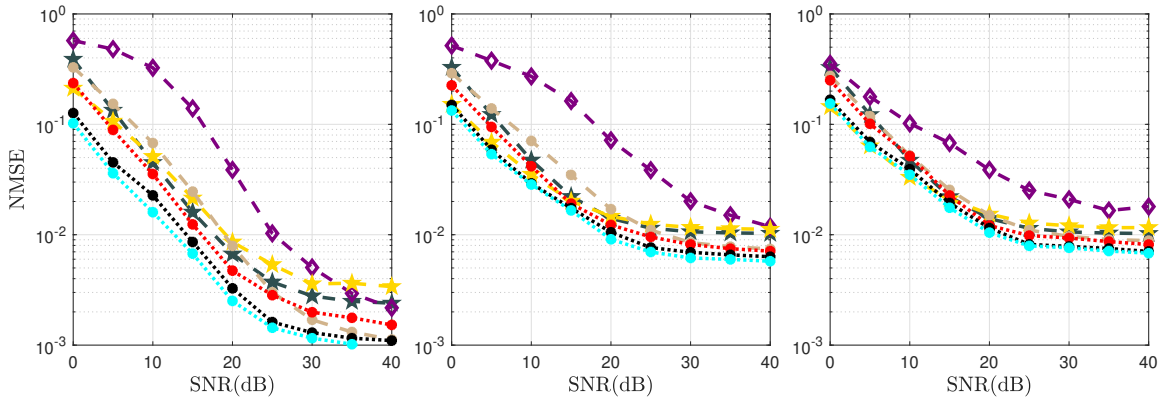
In this section, the proposed DL-based estimators is compared against the recently proposed SoA estimators in literature in terms of bit error rate (BER), normalized mean-squared error (NMSE) employing the low, high, and very high mobility scenarios. These are benchmarked with 1D-LMMSE as the best conventional linear SBS estimator, as shown in Chapter 3. The simulations are implemented using QPSK and 16QAM modulation orders. The SNR range is $[0, 5, \ldots, 40]$ dB. Moreover, the frame size is 100 OFDM symbols per frame and the performance evaluation is performed according to: (i) modulation order, (ii) mobility, (iii) frame length, and (iv) DL architecture. Finally, a robustness analysis is provided in order to validate the generalization ability of the proposed estimators.

5.4.1 Modulation Order: QPSK vs. 16QAM

For QPSK modulation order, we can notice from Fig. 5.3, and Fig. 5.4 that employing DL techniques in the channel estimation process leads to a significant overall performance



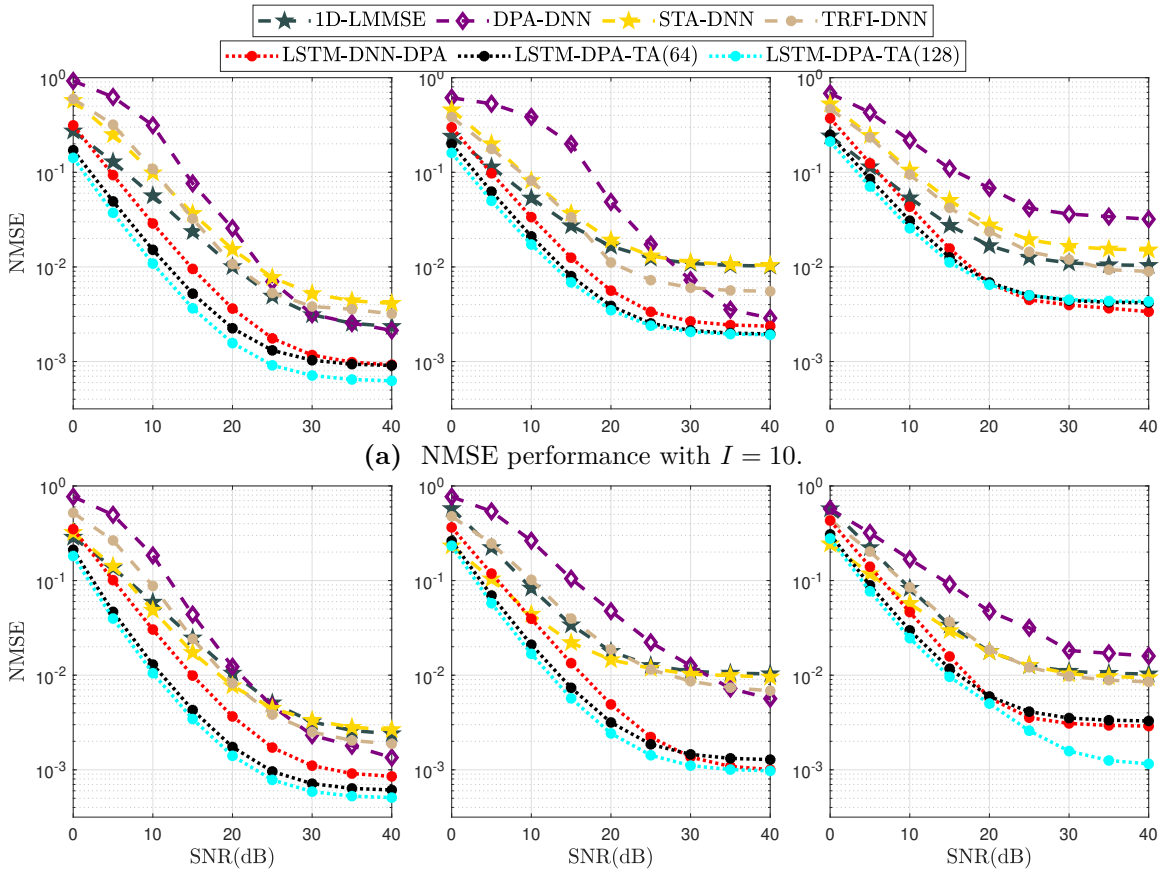
(a) NMSE performance employing QPSK.



(b) NMSE performance employing 16QAM.

Figure 5.4: NMSE for $I = 100$, mobility from left to right: low ($v = 45$ Kmph, $f_d = 250$ Hz), high ($v = 100$ Kmph, $f_d = 500$ Hz), very high ($v = 200$ Kmph, $f_d = 1000$ Hz).

improvement in different mobility scenarios. First of all, the DNN-based estimators where DNN is employed as a post-processing unit after conventional estimators are discussed. We note that DNN can implicitly learn the channel correlations besides preventing a high demapping error that results from conventional DPA estimation. The simulations show that further performance improvement can be achieved by employing DNN after the conventional STA and TRFI estimations, where STA-DNN estimator outperforms DPA-DNN estimator by around 5 dB gain in terms of SNR for BER $= 10^{-3}$. However, STA-DNN suffers from error floor starting from SNR $= 20$ dB especially in very high mobility scenario. This is due to the fact that conventional STA estimation outperforms DPA in low SNR region because of the frequency and time averaging operations that are able to alleviate the impact of noise and demapping error in low SNR regions. While in high SNR regions, the averaging operations are not useful since the impact of noise is low, and the STA averaging coefficients are fixed. Therefore, in order to compensate the STA-DNN performance degradation in high SNR region we propose to utilize the DNN processing after the conventional TRFI estimator, which improves the performance in high SNR region. It is worth mentioning that, STA-DNN and TRFI-DNN can be employed in an adaptive manner where STA-DNN and TRFI-DNN are used

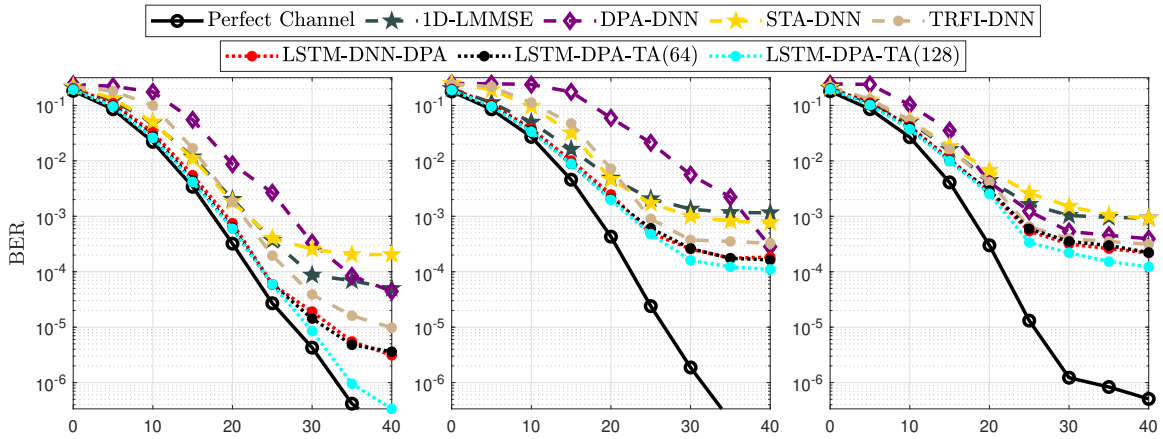


(a) NMSE performance with $I = 10$.

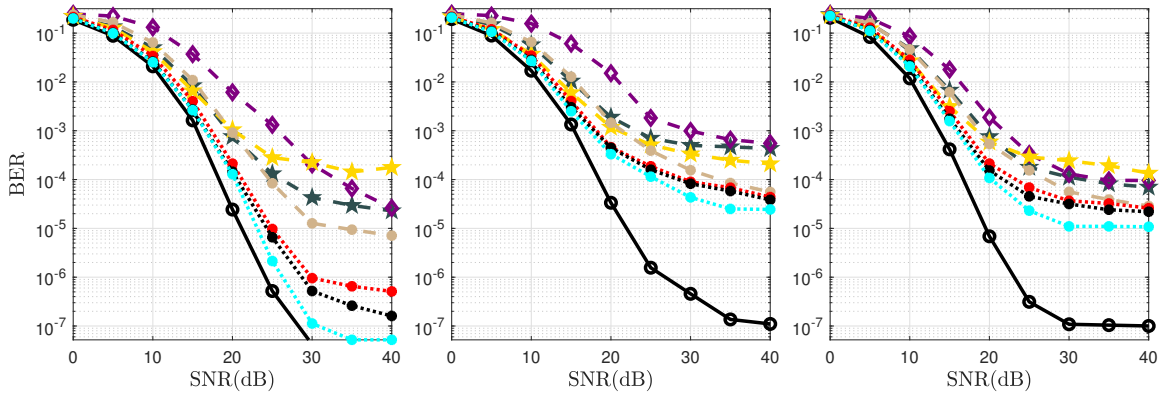
(b) NMSE performance with $I = 100$.

Figure 5.5: NMSE for QPSK, mobility from left to right: low ($v = 45$ Kmph, $f_d = 250$ Hz), high ($v = 100$ Kmph, $f_d = 500$ Hz), very high ($v = 200$ Kmph, $f_d = 1000$ Hz).

in low and high SNR regions, respectively. For the LSTM-based estimators, we can notice that employing LSTM as a prepossessing unit instead of just simple DNN in the channel estimation can significantly improve the overall performance. This is due to the fact that LSTM can efficiently learn the time correlations of the channel by taking the advantage of the previous output besides the current input to estimate the current output. LSTM-DNN-DPA estimator [Pan et al., 2021] outperforms our proposed STA-DNN and TRFI-DNN estimators by around 4 dB gain in terms of SNR for BER $= 10^{-3}$. However, this estimator suffers from high computational complexity as we discuss in the next section due to employing two DL networks, i.e, LSTM followed by DNN. On the other hand, our proposed LSTM-DPA-TA estimators performance gain in different scenarios can be explained by employing the TA processing which significantly reduces the noise impact. In addition to the high ability of the LSTM in learning the channel time correlations compared with a simple DNN architecture. Moreover, two LSTM architectures are employed using 64 and 128 LSTM hidden layer size, respectively. The proposed LSTM-DPA-TA estimators achieve almost similar NMSE performance as the 1D-LMMSE estimator in low SNR regions, while they outperform the recently proposed LSTM-DNN-DPA estimator by around 4 dB gain in terms of SNR for BER



(a) BER performance with $I = 10$.



(b) BER performance with with $I = 100$.

Figure 5.6: BER for QPSK, mobility from left to right: low ($v = 45$ Kmph, $f_d = 250$ Hz), high ($v = 100$ Kmph, $f_d = 500$ Hz), very high ($v = 200$ Kmph, $f_d = 1000$ Hz).

$= 10^{-4}$. However, the optimized LSTM architecture (64) achieves almost similar BER performance as the recently proposed LSTM-DNN-DPA estimator, with a significantly reduced computational complexity.

When adopting high modulation order (16QAM), our proposed DNN-based and LSTM-based estimators are also able to outperform the recently proposed estimators, where the proposed LSTM-DPA-TA (128) estimator outperforms the other benchmarked estimators by at least 7 dB and 3 dB gains in terms of SNR for BER $= 10^{-3}$ in high and very high mobility scenarios, respectively as shown in Fig. 5.3a and Fig. 5.4b.

5.4.2 Mobility and Frame Length

The impact of mobility and frame length is demonstrated in Figure. 5.5 and Figure. 5.6. The NMSE performance degradation with the increase of mobility of all the compared schemes can be noticed from Figure. 5.5, especially when a short frame ($I = 10$) is used. The LSTM-based estimators outperform the DNN-based ones, as a result of better exploiting the channel time correlations. This gain is significantly remarkable in the case of a long frame ($I = 100$), whereas the performance of 1D-LMMSE is independent of

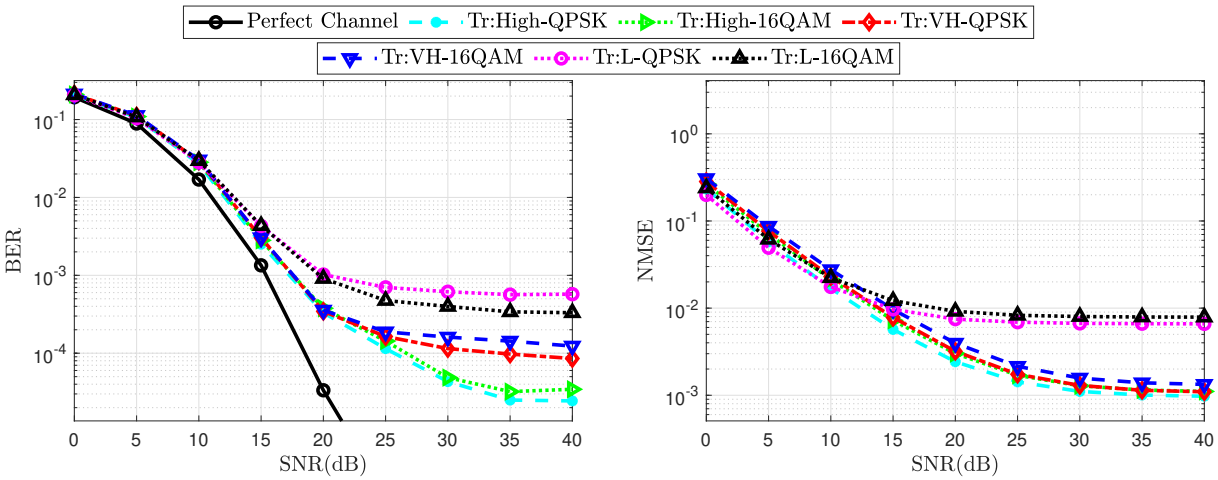
the frame length for the same mobility condition. The increase of estimation accuracy influences the reliability, as can be seen from Figure. 5.6. The LSTM-based estimators are more robust against Doppler error than the DNN-based estimators, and LSTM-DPA-TA estimator is still able to outperforms our proposed TRFI-DNN and the recently proposed LSTM-DNN-DPA by around 3 dB and 5 dB gains in terms of SNR for BER = 10^{-3}, respectively.

Although the NMSE of the LSTM-based increases with the increase of the mobility, the BER performance shows improvements when the mobility increases from high ($f_d = 500$) to very high mobility ($f_d = 1000$), especially for long frames. However, the BER gets worse in the case of high mobility than the case of low mobility. This behavior is attributed to the main following factors; i) channel estimation error, and ii) time diversity due to increased Doppler spread and frame length. The estimation error degrades the BER performance, whereas the time diversity improves it. As can be seen from Figure. 5.6, when perfect channel is used, which is only affected by the Doppler interference within the duration of one OFDM symbol, increasing the frame length increases the time diversity gain. In addition, with longer frame the codeword becomes longer, and therefore, the BER decreases with the increase of the frame length. Once the mobility increases, the time diversity increases as a result of the increase of Doppler spread, and this diversity compensates for Doppler error, leading to similar performance with a short frame ($I = 10$). Employing a longer frame, increases the time diversity and thus the BER decreases with the increase of mobility. The impact of channel estimation error overweights the diversity gain in the case of high mobility for all estimators. However, at very high mobility, the diversity gain becomes significant, especially with a long frame. In this case, the LSTM-based method performes similar to LSTM-DPA-TA(128) with slight gain, at the cost of high complexity in comparison with the least computationally-complex LSTM-DNN-DPA.

5.4.3 DL Architecture

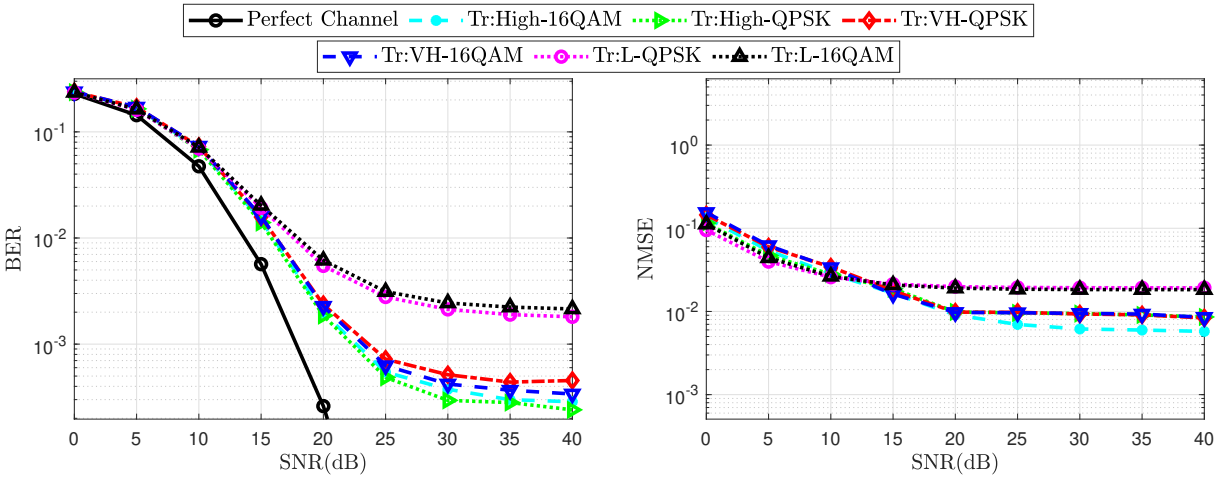
An intensive investigation has been performed on several DNN and LSTM architectures in order to select the most suitable hyper parameters in terms of both performance and complexity. The proposed STA-DNN and TRFI-DNN estimators have better optimized DNN architecture which definitely makes them less complex than the DPA-DNN estimator as we discuss in Section 5.5. The simulation results according to several mobility conditions show that integrating DNN with conventional estimators attains significant performance improvement. Correcting the estimation error of the DPA estimation which is the main idea of the recently proposed DPA-DNN estimator [Han et al., 2019] is not sufficient even if more neurons are included in the DNN hidden layers, since it just corrects the demapping error and neglects the frequency and time correlation of the received symbols. Moreover, employing DNN as a post processing module after the STA and TRFI reflect the

performance superiority with respect to DPA-DNN, due to the fact that STA takes into consideration both frequency and time correlation between the received OFDM symbols, whereas the conventional TRFI estimator employs frequency domain cubic interpolation to improve further the DPA estimation. Therefore, we can conclude that in order for the DNN processing to be more useful, the pre-estimation should be good enough. In other words, as the accuracy of the pre estimation increases, low complex DNN architecture is required while recording a considerable performance gain. In contrast, if the pre estimation is bad, employing DNN processing with high-complexity architecture leads to a limited performance gain while increasing the overall computational complexity.



(a) BER performance employing QPSK.

(b) NMSE performance employing QPSK.



(c) BER performance employing 16QAM.

(d) NMSE performance employing 16QAM.

Figure 5.7: Robustness analysis the proposed LSTM-DPA-TA (128) estimator in high mobility scenario ($f_d = 500$ Hz).

5.4.4 Robustness Analysis

In order to analyze the robustness and the generalization ability of the proposed DL-based estimators, we test the performance of the proposed LSTM-DPA-TA estimator in high mobility scenario according to the following study cases:

- ▷ Case 1: Robustness against mobility: where the LSTM models trained for low and very high mobility scenarios are used to estimate the channel in high mobility scenario. This gives more information about employing an LSTM model trained using different Doppler shifts in the testing phase.
- ▷ Case 2: Robustness against modulation order: here we consider in the testing phase, a trained DNN on same mobility but different modulations. Moreover, besides changing the Doppler shift in the trained LSTM model, we also change the modulation order. For example, we use the trained LSTM for very high mobility (QPSK) in estimating the channel for high mobility scenario (16QAM).

Figure 5.7 depicts the robustness analysis of the high mobility scenario including both QPSK and 16QAM modulation orders. We can clearly notice that employing a trained LSTM using the same Doppler but different modulation leads to almost similar performance as using a trained LSTM (same Doppler, same modulation). On the other hand, Figure. 5.7 illustrates the generalization ability of the LSTM network, where employing trained LSTMs on very high mobility scenario in the testing phase of high mobility scenario degrades slightly the performance, in contrast, when trained LSTMs on low mobility scenario are used for testing in high mobility scenario, a significant performance degradation is recorded. As a summary we can conclude the following:

- ▷ If LSTM is trained on high Doppler value and tested on low Doppler values, it gives the same performance as the LSTM model trained on low Doppler value.
- ▷ If LSTM is trained on low Doppler value and tested on high Doppler values, then, significant performance degradation is recorded.
- ▷ Training the LSTM on 16QAM modulation and testing it on QPSK modulation achieves better performance than training the LSTM on QPSK modulation and testing it on 16QAM modulation. This due to the fact that QPSK modulation is a part of the 16QAM modulation, thus, when the LSTM is trained on higher modulations it is able to generalize for the lower modulations.

5.5 Computational Complexity Analysis

For the computational complexity of the LSTM unit, it can be calculated in terms of the required real values operations performed by its four gates, where each gate applies $P^2 + PK_{in}$ real-valued multiplications, and $3P + K_{in} - 2$ real-valued summations.

In addition to $3P$ real-valued multiplications, and P real-valued summations required by (1.31), and (1.33). Therefore, the overall computational complexity for the LSTM becomes

$$C_{\text{LSTM}} = 4(P^2 + PK_{\text{in}} + 3P + K_{\text{in}} - 2) + 4P. \tag{5.10}$$

We note that, the DNN-based estimators achieve lower complexity compared to the LSTM-based estimators because LSTM processing requires more operations than DNN processing. Concerning the DNN-based estimators, the DPA-DNN architecture [Han et al., 2019] is composed of three hidden layers with $J_1 = J_5 = 2K_{\text{on}}$, $J_2 = J_4 = 40$, and $J_3 = 20$ neurons, respectively. Thus, the DPA-DNN requires $4K_{\text{on}}J_2 + 2J_2J_3$ multiplications, and $2K_{\text{on}} + 2J_2 + J_3$ summations. Moreover, the computational complexity of LS and the DPA estimation are accumulated for DPA-DNN computational complexity resulting in $178K_{\text{on}} + 1600$ multiplications and $168K_{\text{on}} + 1600$ summations/subtractions as an overall computational complexity required by the DPA-DNN estimator.

Table 5.2: Computation complexity in terms of real-valued operations.

Estimator	**Mul./Div.**	**Sum./Sub.**
DNN(J_2-J_3-J_4)	$2K_{\text{on}}J_2 + J_2J_3 + J_3J_4 + 2K_{\text{on}}J_4$	$2K_{\text{on}}J_2 + J_2J_3 + J_3J_4 + 2K_{\text{on}}J_4$
LSTM (P)	$P^2 + 3P + PK_{in}$	$4P + K_{in} - 2$
Overall channel estimation		
STA-DNN	$82K_{\text{on}} + 2K_d + 450$	$70K_{\text{on}} + 10K_d + 450$
TRFI-DNN	$94K_{\text{on}} + 26K_{\text{int}} + 450$	$74K_{\text{on}} + 30K_{\text{int}} + 450$
DPA-DNN	$178K_{\text{on}} + 1600$	$168K_{\text{on}} + 1600$
LSTM-DNN-DPA	$512K_{\text{in}} + 98K_d + 71040$	$4K_{\text{in}} + 88K_d + 6776$
LSTM-DPA-TA(64)	$514K_{on} + 18K_d + 16576$	$10K_{on} + 8K_d + 824$
LSTM-DPA-TA(128)	$1026K_{on} + 18K_d + 65920$	$10K_{on} + 8K_d + 1656$

The proposed STA-DNN and TRFI-DNN estimator employ a three hidden layer DNN architecture with 15 neurons each. This DNN architecture requires $4K_{\text{on}}J_2 + 2J_2^2$, besides $2K_{\text{on}} + 3J_2$ summations. It can be observed that this architecture is less complex than the DPA-DNN architecture. Therefore, the STA-DNN overall computational complexity is $82K_{\text{on}} + 2K_d + 450$ multiplications, and $70K_{\text{on}} + 10K_d + 450$ summations/subtractions. Moreover, the TRFI-DNN requires $94K_{\text{on}} + 26K_{\text{int}} + 450$ multiplications, and $74K_{\text{on}} + 30K_{\text{int}} + 450$ summations/subtractions. The proposed TRFI-DNN estimator decreases the number of multiplications and summations by 48% and 56%, respectively, in comparison with DPA-DNN, while it has almost similar computational complexity as STA-DNN.

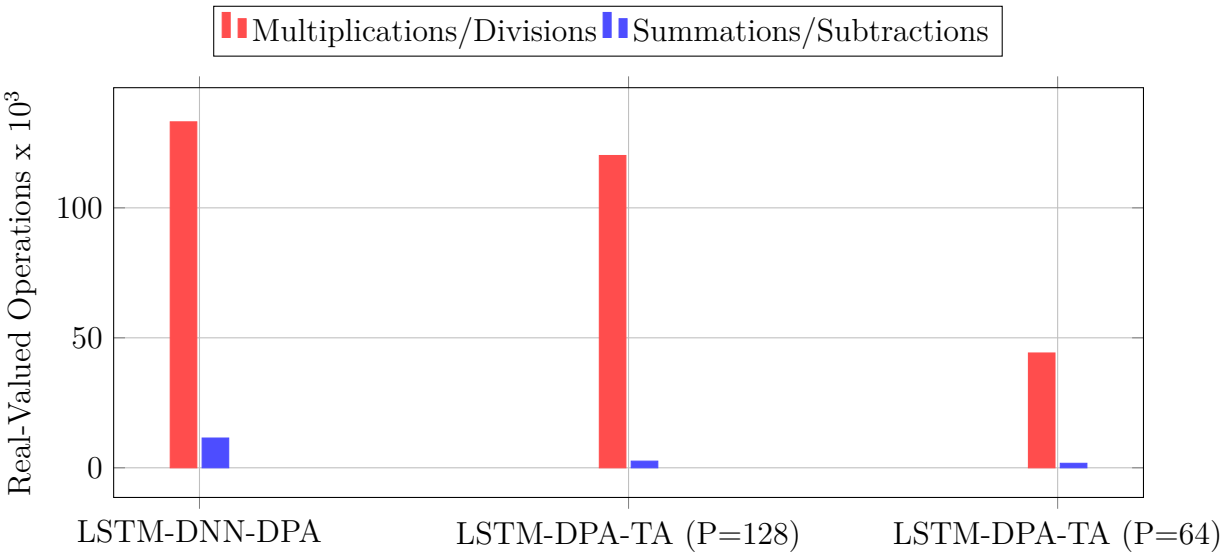


Figure 5.8: Computational complexity of the LSTM-based estimators.

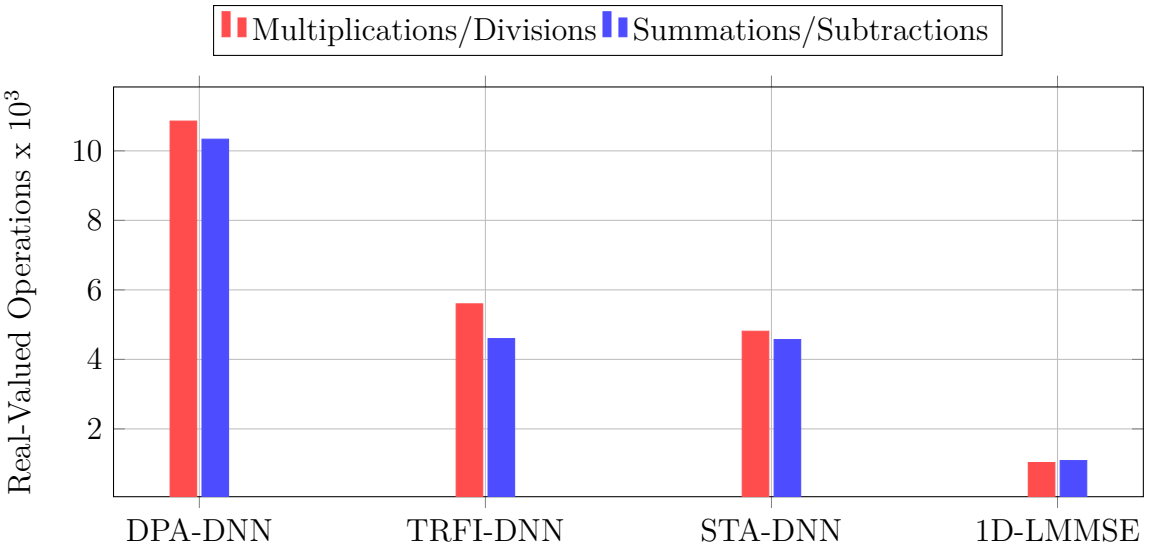


Figure 5.9: Computational complexity of the DNN-based estimators.

For the LSTM-based estimators, the LSTM-DNN-DPA estimator employs one LSTM unit with $P = 128$ and $K_{in} = 112$, followed by one hidden layer DNN network with $N_1 = 40$ neurons. Finally, the LSTM-DNN-DPA estimator applies the DPA estimation which requires $18K_d$ real-valued multiplication/division and $8K_d$ real-valued summation/subtraction. Therefore, the overall computational complexity of the LSTM-DNN-DPA estimator is $512K_{\text{in}} + 98K_d + 71040$ real-valued multiplication/division and $4K_{\text{in}} + 88K_d + 6776$ real-valued summation/subtraction. The proposed LSTM-DPA-TA employs one LSTM unit with $P = 128$ as LSTM-DNN-DPA estimator, or $P = 64$ when the optimized LSTM unit architecture is employed. Moreover, the proposed estimator uses $K_{in} = 2K_{on}$, and applies TA as a noise alleviation technique to the $\hat{\hat{\boldsymbol{h}}}_{\text{LSTM-DPA}_{i,d}}$ estimated channel, that requires only $2K_{on}$ real-valued

multiplication/division and $2K_{on}$ real-valued summation/subtraction. As a results, the proposed LSTM-DPA-TA estimator requires $4P^2+P(8K_{on}+3)+18K_d+2K_{on}$ real-valued multiplication/division and $13P+10K_{on}+8K_d-8$ real-valued summation/subtraction.

Based on this analysis, the proposed LSTM-based estimator achieves less computational complexity compared to the LSTM-DNN-DPA estimator. It records 9.73% and 77.63% computational complexity decrease in the required real-valued multiplication/division and summation/subtraction, respectively, when the LSTM unit is employed with $P = 128$ hidden size. On the other hand, more complexity reduction can be achieved when the optimized LSTM unit is used with $P = 64$ hidden size, where the proposed estimator is able to decrease the complexity of the required multiplication/division and summation/subtraction by 66.81% and 84.90%, respectively. It is worth mentioning that replacing the DNN network by the TA processing in order to alleviate the noise is the main factor in decreasing the overall computational complexity, where the proposed estimator outperforms the LSTM-DNN-DPA estimator while recording a significant computational complexity reduction. Moreover, we note that employing the proposed LSTM-based estimators instead of the proposed DNN-based estimators leads to 89.10% and 62.18% increase in the required real-valued multiplication/division and summation/subtraction, respectively. However, a significant performance gain can be achieved. Table 5.2, Figure. 5.9, and Figure. 5.8 show a detailed summary of the computational complexities for the different studied estimators.

5.6 Conclusion

In this chapter, we have investigated the challenging channel estimation in doubly-selective wireless channels. We have proposed two DNN-based estimators denoted as STA-DNN and TRFI-DNN. The basic idea behind them is to integrate DNN with conventional STA and TRFI estimators so that the DNN can capture more features of the time and frequency correlations between the channel samples, thus, ensuring more advanced tracking of the channel variations. Moreover, we have proposed another LSTM-based estimator, where we have shown that employing the LSTM network instead of the DNN network in the channel estimation leads to a significant performance improvement, especially in very high mobility scenarios. This is due to the ability of LSTM network in implicitly learning the time correlation between successive samples. Simulation results for different channel models of vehicular communications have demonstrated that the proposed DNN and LSTM based estimators significantly outperform the recently proposed SoA estimators, with a considerable computational complexity decrease.

Chapter 6

Proposed DL-based FBF Channel Estimators

Contents

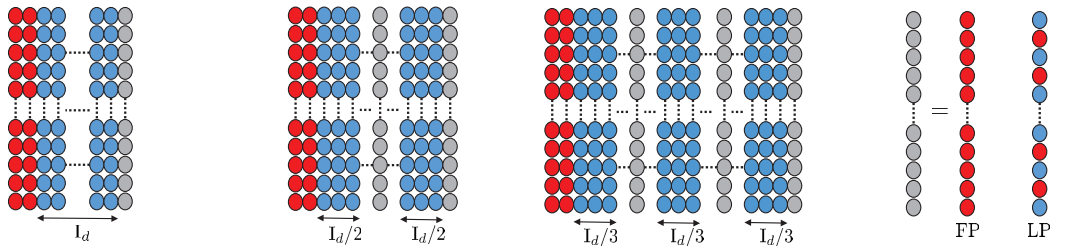
In general, frame-by-frame (FBF) estimation should provide better performance than symbol-by-symbol (SBS) estimation since the previous, current, and future pilot subcarriers are employed in the estimation process of each received orthogonal frequency division multiplexing (OFDM) symbol within the frame. The well-known FBF conventional estimator 2D linear minimum mean square error (LMMSE) utilizes the channel and noise statistics in the estimation, and thus, leading to comparable performance to the ideal case. However, the 2D-LMMSE suffers from high computational

complexity. Therefore, there is a need for new FBF estimators that are both robust and low complex in order to overcome the high complexity limitation of the 2D-LMMSE while achieving good estimation performance. In this context, we propose two FBF channel estimation techniques, one based on linear estimation and interpolation, and the other based on deep learning (DL).

The proposed hybrid and adaptive weighted interpolation (WI) channel estimators employ new pilot allocation schemes for the IEEE 802.11p standard, where pilot symbols are inserted into the transmitted frame. The number of inserted pilots is controlled and adapted according to the mobility condition. In particular, one OFDM symbol with all pilots is inserted at the end of the transmitted frame in low-mobility scenario and as the mobility increases, more pilot OFDM symbols are required. All the other symbols within the transmitted frame are fully allocated for data. The estimated channel for the data symbols is computed by means of weighted summation of the estimated channels at the inserted pilot symbols. To reduce the latency, each frame is divided into subframes, where the channel estimation starts upon receiving each subframe instead of waiting for the whole frame. Moreover, in order to gain more transmission data rate (TDR), few pilots can be inserted within the pilot OFDM symbols depending on the channel delay profile. On top of that, further performance improvement can be achieved by integrating optimized super resolution CNN (SR-CNN) or denoising CNN (DN-CNN) as post processing modules after the WI estimators. The proposed WI estimators enjoy low-complexity and robustness, and achieve good performance in high-mobility scenarios. Additionally, the proposed WI estimators contribute to latency reduction at the receiver in addition to gaining more transmission data rates. To the best of our knowledge, there is no recent FBF estimators that modify the IEEE 802.11p pilot allocation. However, the works proposed in [Kim et al., 2008, Cho et al., 2009] employ modified frame structure but for SBS channel estimation, where a midamble pilot symbol is inserted frequently within the transmitted frame and used for estimating the channel for all successive symbols.

Motivated by the advantages of DL in improving the overall system performance as discussed in Chapter 5, DL techniques, specifically convolutional neural networks (CNNs) are integrated in the FBF estimators as a good low-complexity alternative to the 2D LMMSE estimator that achieves considerable performance gain while preserving low computational complexity. In the CNN-based estimation the estimated channel for the whole frame is considered as a 2D low-resolution noisy image and CNN-based processing is applied as super resolution and denoising techniques.

This chapter sheds lights on FBF channel estimation in vehicular communications. First, the proposed linear and DL-based FBF estimators are presented and discussed in Section 6.1 and 6.2, respectively. Then, simulation results followed by computational complexity analysis are implemented in Section 6.3 and 6.4, respectively, where the performance superiority of the proposed estimators is illustrated. We note that the contributions presented in this chapter have been published in [Gizzini et al., 2021b].



(a) Low mobility. **(b)** High mobility. **(c)** Very high mobility. **(d)** Pilot schemes.

Figure 6.1: Proposed frame structure employing different pilot allocation schemes.

6.1 Proposed Weighted Interpolation Estimators

The proposed WI estimators employ different pilot allocation schemes to the IEEE 802.11p frame structure as shown in Fig. 6.1. The proposed frame structures are motivated by the fact that the vehicle velocity is a known exchanged parameter between all vehicular network nodes. For example, in urban environments (inside cities), the car velocity must not exceed 40 Kmph. The vehicles and road side units within this environment use a low-mobility frame structure, and similar approach can be decided for different environments such as highways. Therefore, the frame structure selection is performed in an adaptive manner according to the estimated vehicle's velocity. The proposed frame structures preserve the first two long training symbols (LTS) preamble symbols similar to IEEE 802.11p standard, so that the synchronization and frame detection processes will not be affected by the proposed modifications. Moreover, only P pilot OFDM symbols are required in the transmitted frame, such that $\tilde{\boldsymbol{Y}}_P = [\tilde{\boldsymbol{y}}_{i_1}^{(p)}, \ldots, \tilde{\boldsymbol{y}}_{i_q}^{(p)}, \ldots, \tilde{\boldsymbol{y}}_{i_P}^{(p)}] \in \mathbb{C}^{K_{\text{on}} \times P}$, where i_q denotes the index of the pilot symbol within the frame. The other $I_d = I - P$ OFDM data symbols are preserved for actual data transmission.

The estimation proceeds according to two criteria: (i) the pilot allocation scheme, Figure. 6.1d, for frequency-domain estimation, and (ii) the mobility condition to select the frame structure as shown in Figure. 6.1, for time interpolation.

6.1.1 Frequency-domain estimation

Full pilot allocation (FP)

In this scheme, all the K_{on} subcarriers are used as pilots formulating OFDM pilot symbol and simple frequency-domain least squares (LS) estimation, denoted as *simple LS (SLS)*, is applied to estimate the channel for each inserted pilot symbol. The SLS is applied on the two LTS preambles $\tilde{\boldsymbol{y}}_{\text{LTS}_1}[k]$, and $\tilde{\boldsymbol{y}}_{\text{LTS}_2}[k]$, and the received pilot symbol such that

$$\hat{\tilde{\boldsymbol{h}}}_{\text{SLS}_q}[k] = \frac{\tilde{\boldsymbol{y}}_q^{(p)}[k]}{\tilde{\boldsymbol{p}}[k]} = \tilde{\boldsymbol{h}}_q[k] + \tilde{\boldsymbol{v}}_q[k], \; k \in \mathcal{K}_{on}. \tag{6.1}$$

Here, $\tilde{\boldsymbol{v}}_q[k]$ represents the noise at the q-th received pilot symbol. Alternatively, the LS can be performed in the delay-domain followed by interpolation and it is denoted as *accurate LS (ALS)*. This relies on the fact that $\tilde{\boldsymbol{h}}_q = \boldsymbol{F}_{\text{on}}\boldsymbol{h}_{q,L}$, where $\boldsymbol{h}_{q,L} \in \mathbb{C}^{L\times 1}$ represents the channel impulse response at the q-th pilot symbol. First, $\boldsymbol{h}_{q,L}$ is estimated such that:

$$\hat{\boldsymbol{h}}_{q,L} = \boldsymbol{F}_{\text{on}}^{\dagger}\hat{\tilde{\boldsymbol{h}}}_{\text{LS}_q} = \boldsymbol{h}_{q,L} + \boldsymbol{F}_{\text{on}}^{\dagger}\tilde{\boldsymbol{v}}_q,\ k \in \mathcal{K}_{on}, \tag{6.2}$$

where $\boldsymbol{F}_{\text{on}}^{\dagger} = [(\boldsymbol{F}_{\text{on}}^{\text{H}}\boldsymbol{F}_{\text{on}})^{-1}\boldsymbol{F}_{\text{on}}^{\text{H}}]$ is the pseudo inverse matrix of $\boldsymbol{F}_{\text{on}}$. Then, the frequency-domain estimate produced by ALS is obtained by applying the discrete Fourier transform (DFT) interpolation of $\hat{\boldsymbol{h}}_{q,L}$ as follows:

$$\hat{\tilde{\boldsymbol{h}}}_{\text{ALS}_q} = \boldsymbol{F}_{\text{on}}\hat{\boldsymbol{h}}_{q,L} = \boldsymbol{W}_{\text{ALS}}\hat{\tilde{\boldsymbol{h}}}_{\text{LS}_q} = \boldsymbol{F}_{\text{on}}\boldsymbol{h}_{q,L} + \boldsymbol{W}_{\text{ALS}}\tilde{\boldsymbol{v}}_q,\ k \in \mathcal{K}_{on}. \tag{6.3}$$

The matrix $\boldsymbol{W}_{\text{ALS}} = \boldsymbol{F}_{\text{on}}\boldsymbol{F}_{\text{on}}^{\dagger}$ denotes the accurate ALS interpolation matrix.

L pilot allocation (LP)

This allocation is motivated by the fact that channel mainly consists of L taps, then calculating $\boldsymbol{h}_{q,L}$ requires only L pilots per pilot symbol, and can be computed as follows:

$$\hat{\boldsymbol{h}}_{q,L} = \boldsymbol{F}_{p}^{\dagger}\hat{\tilde{\boldsymbol{h}}}_{\text{LS}_q} = \boldsymbol{h}_{q,L} + \boldsymbol{F}_{p}^{\dagger}\tilde{\boldsymbol{v}}_q,\ k \in \mathcal{K}_{\text{p}}. \tag{6.4}$$

Here, $\hat{\tilde{\boldsymbol{h}}}_{\text{LS}_q}$ represents the LS estimated channel for the q-th pilot symbol at the L equally spaced inserted pilot subcarriers, and $K_p = L$. Moreover, $\boldsymbol{F}_{p}^{\dagger} = [(\boldsymbol{F}_{p}^{\text{H}}\boldsymbol{F}_{p})^{-1}\boldsymbol{F}_{p}^{\text{H}}]$ is the pseudo inverse matrix of $\boldsymbol{F}_{p} \in \mathbb{C}^{K_{\text{P}}\times L}$, which is the truncated DFT matrices obtained by selecting $\mathcal{K}_{\text{p}}$ rows, and L columns from the K-DFT matrix. Using DFT interpolation, the frequency-domain channel estimate for the q-th pilot symbol is given by

$$\hat{\tilde{\boldsymbol{h}}}_{\text{DFT}_q} = \boldsymbol{F}_{\text{on}}\hat{\boldsymbol{h}}_{q,L} = \boldsymbol{F}_{\text{on}}\boldsymbol{h}_{q,L} + \boldsymbol{W}_{\text{DFT}}\tilde{\boldsymbol{v}}_q,\ k \in \mathcal{K}_{on}. \tag{6.5}$$

The matrix $\boldsymbol{W}_{\text{DFT}} = \boldsymbol{F}_{\text{on}}\boldsymbol{F}_{\text{p}}^{\dagger}$ is the the interpolation matrix of the LP method.

Conventional pilot allocation (CP)

This case preserves the pilot structure of IEEE 802.11p standard, where only $K_p = 4$ pilots are employed. In this case, when $L > K_p$, truncated discrete Fourier transform (T-DFT) method, which is discussed in Section 4.2, is used to obtain the frequency-domain channel estimates for each pilot symbol.

6.1.2 Time-domain interpolation

After estimating the channel at the pilot symbols according to the selected configuration, the proposed WI estimators groups the estimated channels of the P pilot symbols in P matrices, each of two columns, defined by

$$\hat{\tilde{\boldsymbol{H}}}_q = [\hat{\tilde{\boldsymbol{h}}}_{q-1}, \hat{\tilde{\boldsymbol{h}}}_q],\ q = 1, \cdots P. \tag{6.6}$$

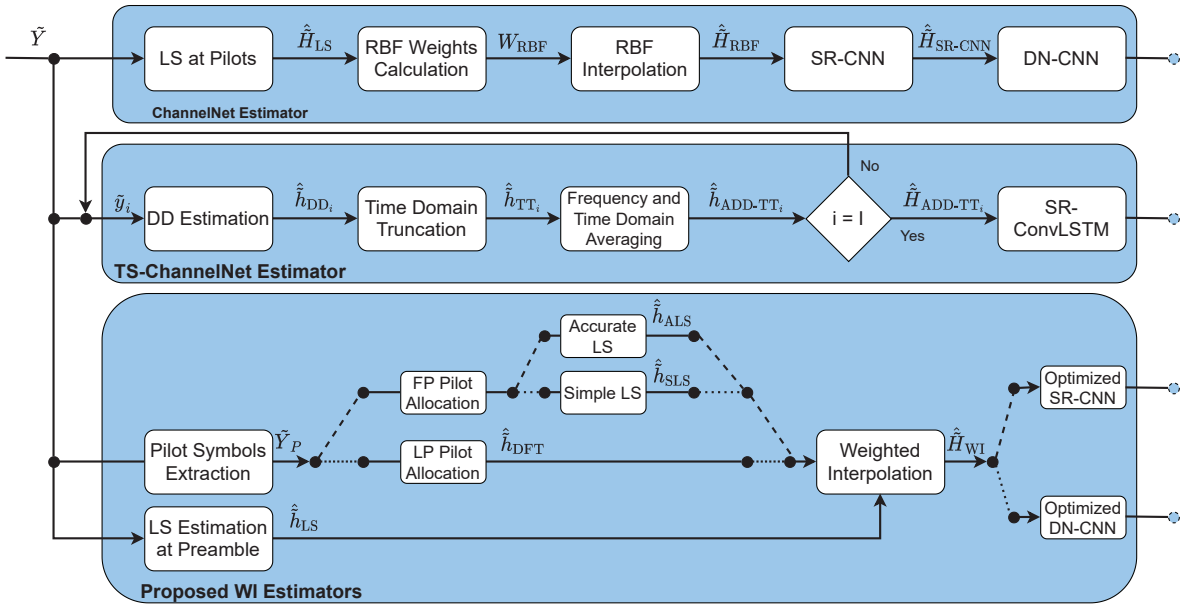


Figure 6.2: Proposed WI estimators block diagram.

We note that $\hat{\tilde{\boldsymbol{h}}}_0 = \hat{\tilde{\boldsymbol{h}}}_{\text{LTS}}$ and $\hat{\tilde{\boldsymbol{h}}}_q$ refers to the estimates obtained using one of the estimation techniques discussed in the previous section.

Weighted Interpolation

According to the grouped $\{\hat{\tilde{\boldsymbol{H}}}_q\}$ matrices, the received frame can be divided into P sub-frames, where f denotes the sub-frame index, such that $1 \leq f \leq P$. Therefore, the estimated channel for the i-th received OFDM symbol within each f-th sub-frame can be obtained using interpolation matrix $\boldsymbol{C}_f \in \mathbb{R}^{2\times I_f}$, such that:

$$\hat{\tilde{\boldsymbol{H}}}_{\text{WI}_f} = \hat{\tilde{\boldsymbol{H}}}_f \boldsymbol{C}_f, \tag{6.7}$$

$$\begin{aligned}
\boldsymbol{C}_f &= \mathrm{E}\left[\tilde{\boldsymbol{H}}_f \hat{\tilde{\boldsymbol{H}}}_f^H\right] \left[\mathrm{E}\left[\hat{\tilde{\boldsymbol{H}}}_f \hat{\tilde{\boldsymbol{H}}}_f^H\right]\right]^{-1} \\
&= \begin{bmatrix} \mathrm{E}\left[\tilde{\boldsymbol{H}}_f \hat{\tilde{\boldsymbol{h}}}_q^H\right] & \mathrm{E}\left[\tilde{\boldsymbol{H}}_i \hat{\tilde{\boldsymbol{h}}}_{q+1}^H\right] \end{bmatrix} \begin{bmatrix} \mathrm{E}\left[\|\tilde{\boldsymbol{h}}_q\|^2\right] + E_q & \mathrm{E}\left[\tilde{\boldsymbol{h}}_q \tilde{\boldsymbol{h}}_{q+1}^H\right] \\ \mathrm{E}\left[\tilde{\boldsymbol{h}}_{q+1} \tilde{\boldsymbol{h}}_q^H\right] & \mathrm{E}\left[\|\tilde{\boldsymbol{h}}_{q+1}\|^2\right] + E_{q+1} \end{bmatrix}^{-1} \\
&= \begin{bmatrix} J_0(2\pi f_{\text{d}}(f-1)T_{\text{s}}) & J_0(2\pi f_{\text{d}}(I_f+1-f)T_{\text{s}}) \end{bmatrix} \begin{bmatrix} 1+E_{\Phi_q} & J_0(2\pi f_{\text{d}} I_f T_{\text{s}}) \\ J_0(2\pi f_{\text{d}} I_f T_{\text{s}}) & 1+E_{q+1} \end{bmatrix}^{-1}.
\end{aligned} \tag{6.8}$$

This interpolation matrix is used to estimate the OFDM data symbols within the f-th sub-frame whose length is I_f. We note that I_f is calculated according to the employed frame structure where it is equal to I_d, $\frac{I_d}{2}$, and $\frac{I_d}{3}$ in low, high, and very high mobility scenarios, respectively. The entries of $\boldsymbol{C}_f$ are calculated by minimizing the mean squared error (MSE) between the ideal channel $\tilde{\boldsymbol{H}}_f$, and the LS estimated channel at the OFDM pilot symbols $\hat{\tilde{\boldsymbol{H}}}_f$ as derived in [Zheng and Xiao, 2009]. This minimization results in $\boldsymbol{C}_f$

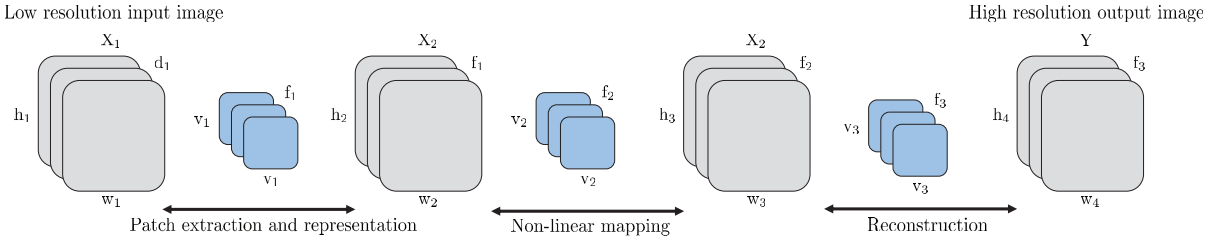


Figure 6.3: SR-CNN architecture.

expressed in (6.8), where $J_0(.)$ is the zeroth order Bessel function of the first kind, T_s is the received OFDM data symbol duration, and E_q denotes the overall noise of the estimated channel at the q-th pilot symbol. E_q is calculated according to the chosen pilot allocation scheme, and employed LS estimation, where it equals to σ^2, $\sigma^2\mathrm{trace}\left\{\boldsymbol{W}_{\mathrm{ALS}}\boldsymbol{W}_{\mathrm{ALS}}^{\mathrm{H}}\right\}$, and $\sigma^2\mathrm{trace}\left\{\boldsymbol{W}_{\mathrm{DFT}}\boldsymbol{W}_{\mathrm{DFT}}^{\mathrm{H}}\right\}$ for SLS, ALS and LP LS estimation respectively.

In fact, the weight elements of $\boldsymbol{C}_f$ for all the sub-frames can be calculated offline for several vehicular scenarios by employing different f_d and I_d values, Therefore, decreasing the online complexity and making the proposed WI estimators more practical. Moreover, a trade-off between the mobility condition controlled by f_d, the inserted pilot symbols P, and the employed frame length I_d should be considered. As the mobility increases, more pilot symbols should be inserted within the transmitted frame. This trade-off is managed by the vehicular application requirements, so that the transmitter adapts the transmission parameters according to these requirements. Fig. 6.2 shows the block diagram of the proposed WI estimators besides the recently proposed SOA DL-based FBF estimators presented in Appendix D.

6.2 Proposed CNN-based Channel Estimation

In this section, to enhance the performance of the linear WI approaches, we propose to perform DL-based post processing. Namely, SR-CNN and DN-CNN. SR-CNN [Dong et al., 2016] has been widely used in the literature to improve the quality of the low-resolution images, where the input-output mapping is represented as a deep CNN that takes the low-resolution image as the input and outputs the high-resolution one. SR-CNN consists of three parts as shown in Fig. 6.3: (*i*) Patch extraction and representation, where the low-resolution input image features are extracted (*ii*) non-linear mapping, and (*iii*) reconstruction. Let $\boldsymbol{X}_m \in \mathbb{R}^{h_m \times w_m \times d_m}$ be the input image to the m-th SR-CNN convolutional layer, where h_m, w_m, and d_m denote the image height, width, and depth respectively, and $\boldsymbol{X}_1 \in \mathbb{R}^{h_1 \times w_1 \times d_1}$ represents the input image. The convolutional filters are denoted by the weights and biases matrices $\boldsymbol{W}_m \in \mathbb{R}^{v_m \times v_m \times d_m \times f_m}$ and $\boldsymbol{B}_m \in \mathbb{R}^{f_m \times 1}$, where v_m refers to the convolutional filter size, and f_m denotes the total number of filters employed in the SR-CNN m-th convolutional

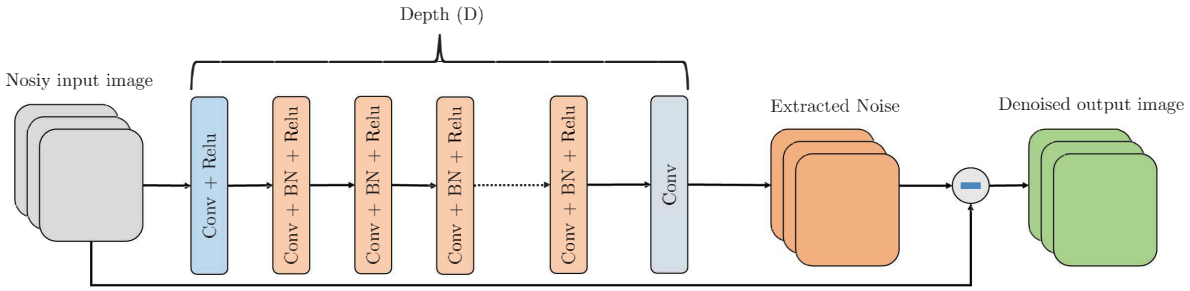


Figure 6.4: DN-CNN architecture.

layer. The m-th convolution layer applies f_m convolution operation to its $\boldsymbol{X}_m$ input image as follows

$$\boldsymbol{Y}_{(m)} = \boldsymbol{F}_{(m)}\Big(\boldsymbol{W}_{(m)} * \boldsymbol{X}_{(m)}\Big), \;\; \boldsymbol{X}_{(m+1)} = \boldsymbol{Y}_{(m)}, \tag{6.9}$$

where $\boldsymbol{F}_{(m)}$ represents the activation function applied to the m-th convolutional layer output. We note that, it is possible to add more convolutional layers to increase the non-linearity mapping between the SR-CNN convolutional layers, but this increases the complexity of the SR-CNN model and thus demands more training time. The employed SR-CNN architecture related mainly to the investigated problem and the dataset structure, whereas the best SR-CNN architecture and parameters can be fine tuned by intensive experiments or by using the grid search algorithm [Pontes et al., 2016] that selects the best suitable SR-CNN hyperparameters in terms of both performance and complexity. The loss function of the SR-CNN is represented by the MSE between the SR-CNN reconstructed high resolution images $\boldsymbol{Y}$, and the true images $\boldsymbol{Y}^{(T)}$, such that

$$\text{MSE}_{\text{SR-CNN}} = \frac{1}{N_{\text{Train}}} \sum_{i=1}^{N_{\text{Train}}} \left\| \boldsymbol{Y}_i - \boldsymbol{Y}_i^{(T)} \right\|^2, \tag{6.10}$$

where N_{Train} denotes the number of training samples.

On the other hand, DN-CNN [Zhang et al., 2017] improves the image quality by separating the noise from the input noisy image using a special CNN architecture. After that, the input noisy image is subtracted from the extracted noise resulting in the denoised image. In order to extract the noise from a noisy image, residual learning [He et al., 2015] is employed so that the noise included in the input image is learned in the DN-CNN training phase by minimizing the following DN-CNN loss function

$$\text{MSE}_{\text{DN-CNN}} = \frac{1}{N_{\text{Train}}} \sum_{i=1}^{N_{\text{Train}}} \left\| \boldsymbol{V}_i - \boldsymbol{V}_i^{(T)} \right\|^2, \tag{6.11}$$

where $\boldsymbol{V}_i$ represents the extracted noise included in the noisy input image $\boldsymbol{X}_i$, and $\boldsymbol{V}_i^{(T)}$ denotes the exact noise, such that $\boldsymbol{V}_i^{(T)} = \boldsymbol{Y}_i^{(N)} - \boldsymbol{X}_i$. As shown in Fig. 6.4, the DN-CNN employs in the first layer convolution and ReLU activation function to

Table 6.1: Optimized SR-CNN and DN-CNN parameters.

Parameter	**Values**
Input/Output dimensions	$2K_{\text{on}} \times I \times 1$
SRCNN ($v_1,\ f_1;\ v_2, f_2;\ v_3,\ f_3$)	(9,32; 1,16; 5,1)
DNCNN ($v,\ f$)	(3, 16)
DNCNN D	7
Activation function	ReLU
Number of epochs	250
Training samples	8000
Testing samples	2000
Batch size	128
Optimizer	ADAM
Loss function	MSE
Learning rate	0.001
Training SNR	30 dB

generate the initial feature maps from the input noisy image. After that, successive (Conv + BN + ReLU) layers are employed to extract the noise features, since the analysis performed in [Zhang et al., 2017] shows that integrating convolution with batch normalization followed by ReLU, can gradually separate the clean image structure from the noisy observation through the DN-CNN hidden layers. Finally, a convolutional layer is used for output image reconstruction. In summary, the DN-CNN architecture has two main tasks: the residual learning is used to learn the noise features, and the batch normalization which is incorporated to speed up training as well as to boost the denoising performance. The main complexity of the DN-CNN architecture lies in its depth (D) that denotes the number of employed (Conv+ BN + ReLU) layers. As D increases the complexity of the DN-CNN architecture increases. Similarly to SR-CNN, all the DN-CNN parameters can be tuned using grid search algorithm.

Unlike ChannelNet estimator, where both SR-CNN and DN-CNN networks are used on top of the radial basis function (RBF) interpolation, extensive experiments are applied in this paper using the hyper parameters tuning grid search algorithm [Pontes et al., 2016] in order to select the best CNN network configuration that is suitable with the mobility condition. Based on the selected CNN parameters, an optimized SR-CNN is employed on top of the WI estimators in low mobility scenario, while optimized DN-CNN is employed in high mobility scenario. By doing so, better performance can be achieved with a significant decrease in the overall computational complexity as we discuss in the next sections.

We note that, employing low-complexity SR-CNN and DN-CNN architectures is due to the good performance of the initial WI estimation. Moreover, our investigations show that by employing the proposed WI estimators, there is no need to use both SR-CNN and DN-CNN, since the computational complexity increases without any significant performance gain.

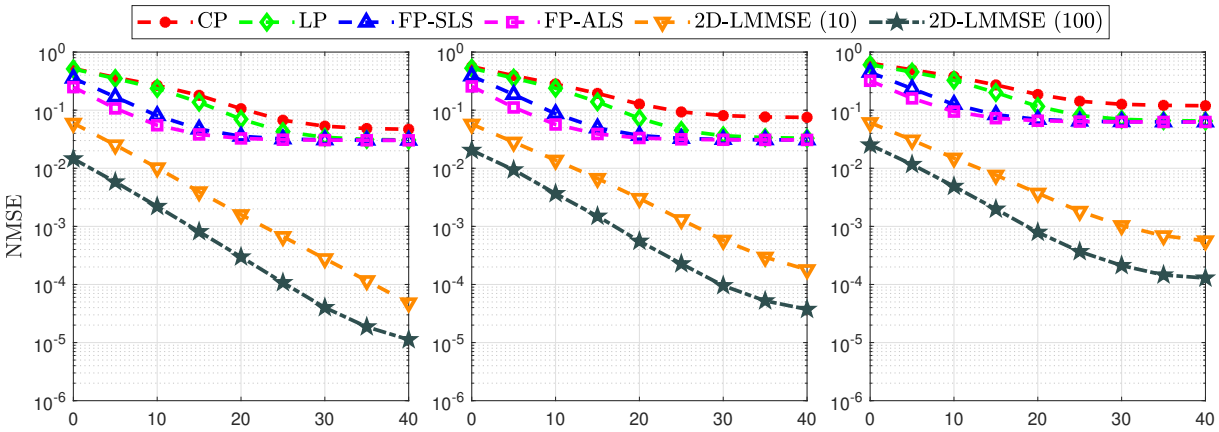
We note that CNNs networks work with real valued data, therefore after applying the WI interpolation estimator, $\hat{\hat{\boldsymbol{H}}}_{\mathrm{WI}} \in \mathbb{C}^{K_{\mathrm{on}} \times I_d}$ is converted from complex to real valued domain by stacking the real and imaginary values vertically in one matrix, such that $\hat{\hat{\boldsymbol{H}}}_{\mathrm{WI}}^{(R)} \in \mathbb{R}^{2K_{\mathrm{on}} \times I_d}$. Then $\hat{\hat{\boldsymbol{H}}}_{\mathrm{WI}}^{(R)}$ is fed as an input to the optimized SR-CNN or DN-CNN according to the mobility scenario. Finally, the output of the employed network is converted back to the complex domain. The optimized SR-CNN and DN-CNN networks are trained on signal-to-noise ratio (SNR)= 30 dB for each mobility scenario, since in high SNR region, the CNN network is able to learn better the channel due to the low noise impact in high SNR region. Moreover, ADAM optimizer is used with MSE loss function. Table 6.1 shows the proposed optimized SR-CNN and DN-CNN parameters.

6.3 Simulation Results

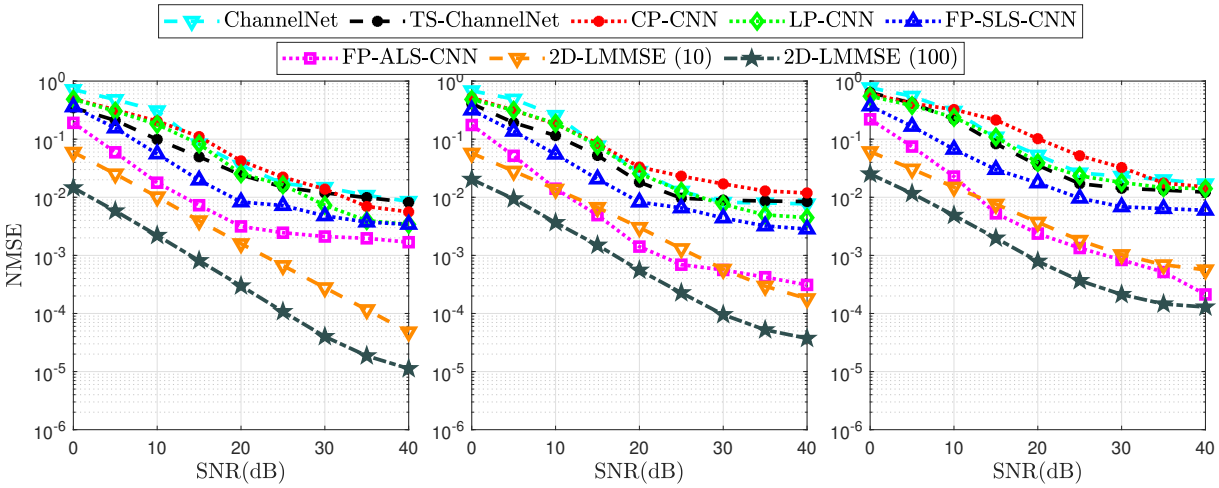
In this section, the performance of the proposed linear WI estimators is evaluated compared to the conventional 2D LMMSE that exploits all the pilots defined in the IEEE 802.11p standard, channel network (ChannelNet), and Temporal spectral ChannelNet (TS-ChannelNet) estimators using bit error rate (BER) and normalized mean-squared error (NMSE). In addition, we also include the WI estimator with $K_p = 4$ pilot subcarriers only denoted as WI-CP, in this case, the proposed T-DFT estimation in Chapter 4 is applied at the inserted pilot symbols within the OFDM frame. The simulations are conducted employing three vehicular scenarios and the criteria that are defined in Chapter 5.

6.3.1 NMSE Evaluation

The NMSE performance of the proposed estimators depends mainly on the employed WI estimation, where using low number of pilots as the case in the proposed WI-LP and WI-CP estimators leads to considerable performance degradation as we can notice in Figure. 6.5a. Whereas, the proposed WI-FP-SLS and WI-FP-ALS achieve better performance than WI-CP and WI-LP due to the employment of full pilots symbols within the frame. The accuracy of WI-FP-ALS is higher at low SNR due to the exploitation of the frequency correlation via interpolation at the cost of increased complexity. By employing all the pilots in the frame (4 pilots per OFDM symbol), 2D-LMMSE (100) significantly outperforms the WI estimators. In order to reduce the latency, we consider to apply the



(a) NMSE for linear methods.



(b) NMSE with DL post processing.

Figure 6.5: NMSE employing $I = 100$, mobility from left to right: low ($v = 45$ Kmph, $f_d = 250$ Hz), high ($v = 100$ Kmph, $f_d = 500$ Hz), very high ($v = 200$ Kmph, $f_d = 1000$ Hz). The CNN refers to SRCNN and DNCNN in low and high/very high) mobility scenarios, respectively.

2D-LMMSE on subframe basis with a subframe length of 10 OFDM symbols denoted as 2D-LMMSE (10). The estimation accuracy in this case is reduced by 10 dB with the decrease of the subframe length. Nevertheless, the accuracy of 2D-LMMSE (10) is better than the WI estimators due to the high correlation of the pilots.

The performance of the WI approaches is reduced by the increase of mobility. This mainly depends on the frame structure, and the Doppler spread through the coherence interval between the pilots symbols, which can be defined as

$$\Delta_C = \Delta_p I f_d T_s, \tag{6.12}$$

where T_s is the OFDM symbol duration, $\Delta_p I$ number of symbols between successive pilot symbols. A smaller value indicates more correlation between the pilots that can be exploited in the time-domain interpolation. Based on the frame structure shown in Figure. 6.2, at very high mobility ($f_d = 1000$ Hz), $\Delta_p = 33$ symbols, which corresponds to

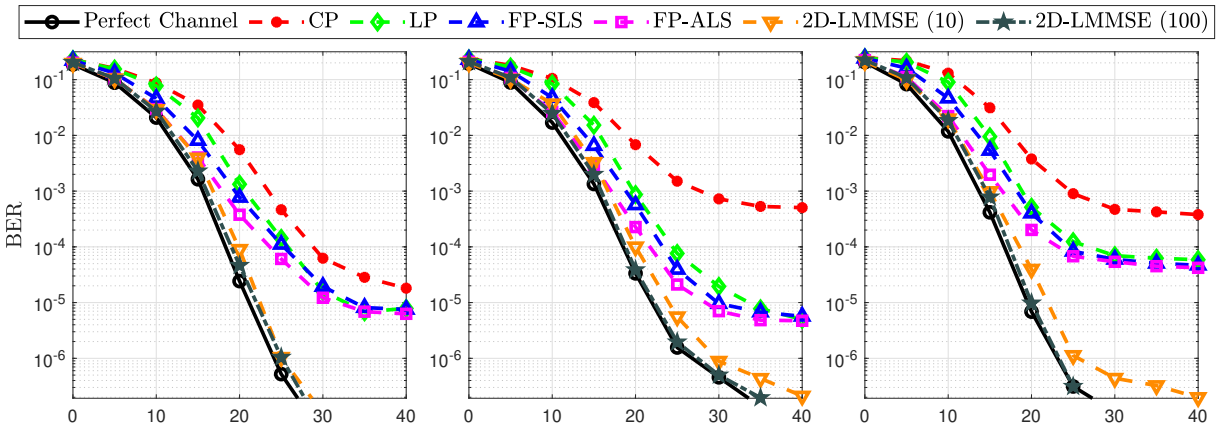
$\Delta_C = 33 \times 10^3 T_s$, whereas in low and high mobility, $\Delta_C = 25 \times 10^3 T_s$. As a consequence, it is clear that the NMSE at very high mobility increases compared to the other cases because of the larger coherence interval between the pilots. The high mobility case is only influenced by the Doppler interference, which can be observed from the error floor at high SNRs. In addition, WI-CP is affected by using less pilots compared to WI-LP. This is also the situation in both 2D-LMMSE estimators as the NMSE increases with the increase of mobility at high SNRs.

Figure. 6.5b shows the NMSE performance of the CNN-based estimators. It can be noticed that the proposed estimators are able to outperform the ChannelNet and TS-ChannelNet estimators in different mobility scenarios. It is worth mentioning that using CNN post processing after the WI-based estimators reveals a considerable robustness against mobility. This is due to the ability of the optimized SR-CNN and DN-CNN in significantly alleviating the impact of Doppler interference. The DL-based post processing networks provide a performance trade-off between the linear WI and 2D-LMMSE using the full pilots in the frame.

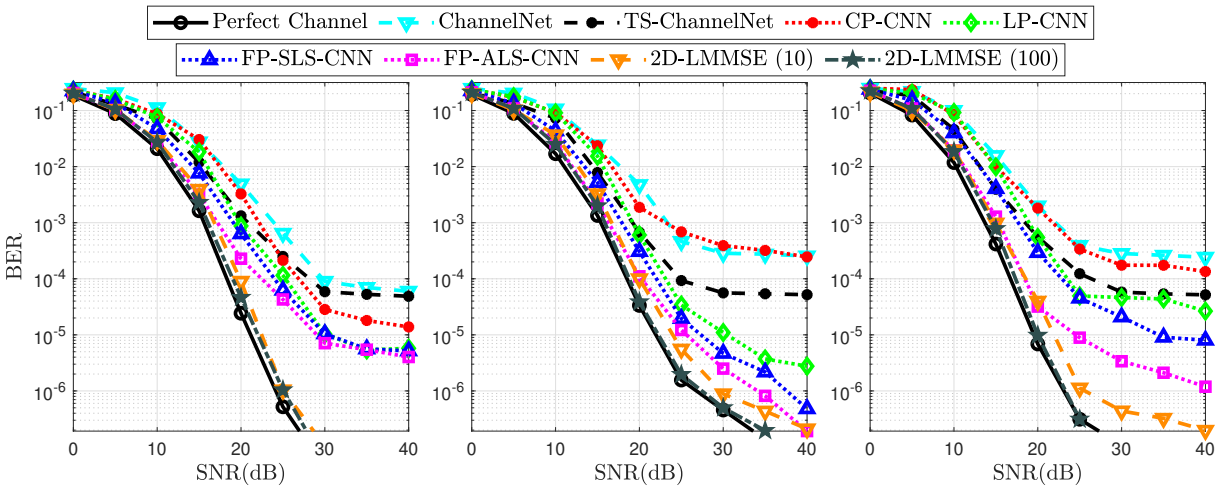
6.3.2 BER Evaluation

Figure. 6.9a depicts the BER using the discussed linear estimation techniques employing QPSK. The relative performance follows the same trend of the channel estimation performance shown in Figure. 6.5a in each mobility case. In general, the impact of the estimation error is lower in low SNR region and this impact increases with the increase of the SNR. In particular, the BER error floor of WI-CP is higher than the others as its estimation error floor is higher. Although the NMSE gap between WI-FP-ALS, WI-FP-SLS, WI-LP decreases with the increase of the SNR, the BER gain achieved by WI-FP-ALS is maintained until reaching an error floor. In different mobility scenarios, the trade-off between estimation error and time diversity gain, as detailed in Section 5.4.2, can be observed. For instance, although the NMSE increases at very high-mobility compared to low-mobility case, the BER of WI-CP does not highly degrade, this is due to the long codeword. The other methods employ more pilots at very high mobility, which reduces the codeword length. In total, the estimation error dominates over the diversity gain. Nevertheless, the case of 2D-LMMSE experiences improvement due to better channel estimation, although the codeword is smaller than that used in WI-FP-ALS.

The impact of the proposed DL-based post processing is shown in Figure. 6.9b. First, it can be clearly seen that the post processing enhances the BER as a result of enhancing the channel estimation, Figure. 6.5b. Next, we compare our proposed linear, and DL-enhanced estimation with the SoA DL-based ChannelNet and TS-ChannelNet. We can observe the significant BER performance superiority of the proposed estimators, where WI-LP records similar performance as TS-ChannelNet, while WI-FP-SLS estimator slightly outperforms WI-LP by around 1 dB gain in terms of SNR for a BER = 10^{-3}. On the other hand, the



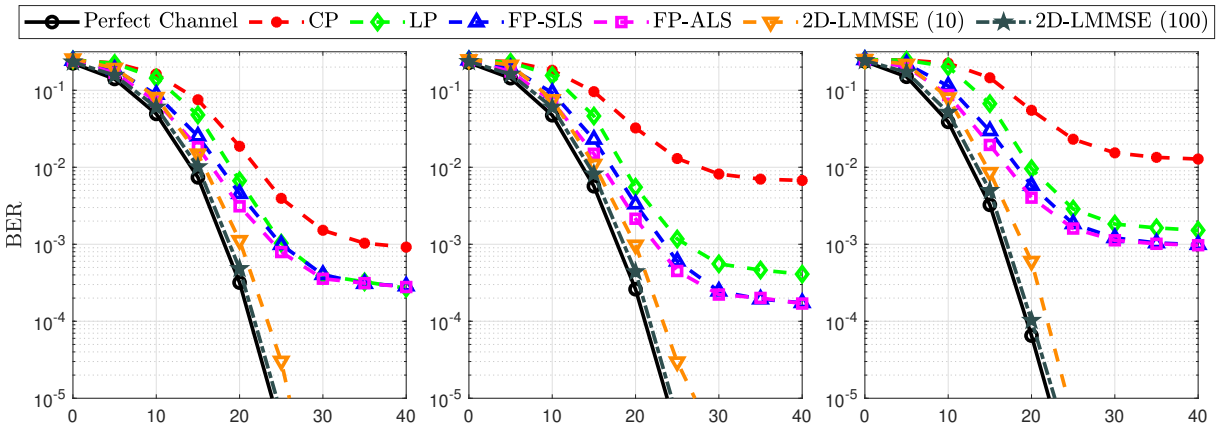
(a) BER with linear estimation.



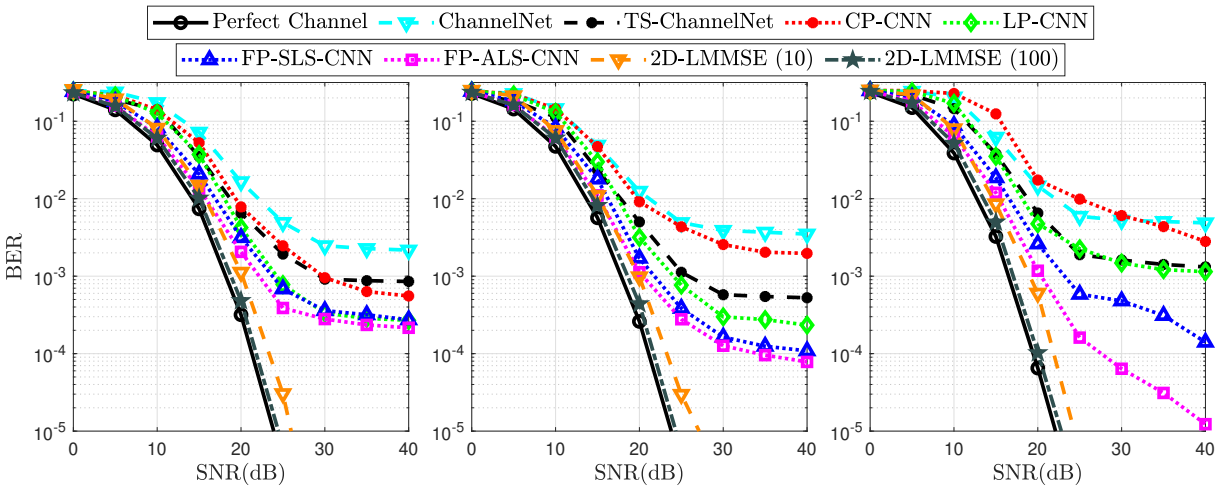
(b) BER with DL post processing.

Figure 6.6: BER for $I = 100$, QPSK, mobility from left to right: low ($v = 45$ Kmph, $f_d = 250$ Hz), high ($v = 100$ Kmph, $f_d = 500$ Hz), very high ($v = 200$ Kmph, $f_d = 1000$ Hz). The CNN refers to SRCNN and DNCNN in low and high/very high) mobility scenarios, respectively.

proposed WI-FP-ALS estimator outperforms both ChannelNet and TS-ChannelNet by around 6 dB and 3 gain in terms of SNR for a BER $= 10^{-3}$, respectively. The performance degradation of the proposed WI-CP estimator is due to employing only 4 pilot subcarriers with T-DFT estimation, and thus, the error of the neglected channel taps appears, causing this performance degradation. The performance of ChannelNet and TS-ChannelNet accounts of the predefined fixed parameters in the applied interpolation scheme, where the RBF interpolation function and the ADD-TT frequency and time averaging parameters must be updated in a real-time manner. Moreover, the ADD-TT interpolation employs only the previous and the current pilot subcarriers for the channel estimation at each received OFDM symbol. On the contrary, in the proposed WI estimators there are no fixed parameters, the time correlation between the previous and the future pilot symbols is considered in the WI interpolation matrix (6.8), and the estimated channel at all channel taps is considered in the overall estimation. These aspects lead to the proposed estimators



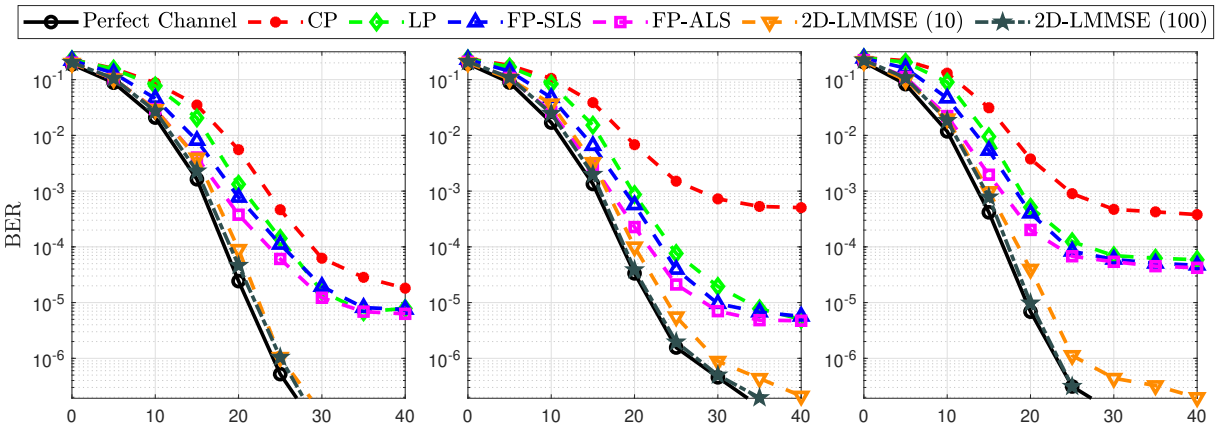
(a) BER with linear estimation.



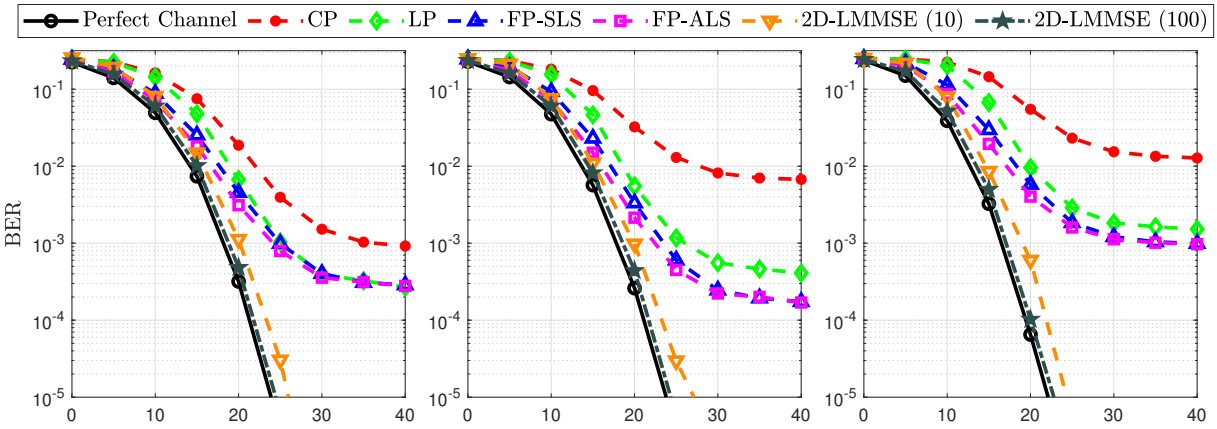
(b) BER with DL post processing.

Figure 6.7: BER for $I = 100$, 16QAM, mobility from left to right: low ($v = 45$ Kmph, $f_d = 250$ Hz), high ($v = 100$ Kmph, $f_d = 500$ Hz), very high ($v = 200$ Kmph, $f_d = 1000$ Hz). The CNN refers to SRCNN and DNCNN in low and high/very high) mobility scenarios, respectively.

performance superiority. In addition, ChannelNet and TS-ChannelNet estimators suffer from a considerable performance degradation that is dominant in very high mobility scenario. However, the proposed estimators show a robustness against high mobility, this is mainly due to the accuracy of the WI interpolation, combined with optimized SR-CNN and DN-CNN. Although CNN processing is applied in the ChannelNet and TS-ChannelNet, this post CNN processing is not able to perform well due to the high estimation error of the 2D RBF and ADD-TT interpolation techniques in the initial estimation. As a result, we can conclude that employing robust initial estimation as the proposed WI interpolation schemes allows the CNN to learn better the channel correlation with lower complexity, thus improving the channel estimation. Finally, we note that the performance of the 2D-LMMSE estimator is comparable to the performance of ideal channel but it requires huge complexity as we discuss in the next section, which is impractical in real scenario.



(a) BER of WI estimators employing QPSK.

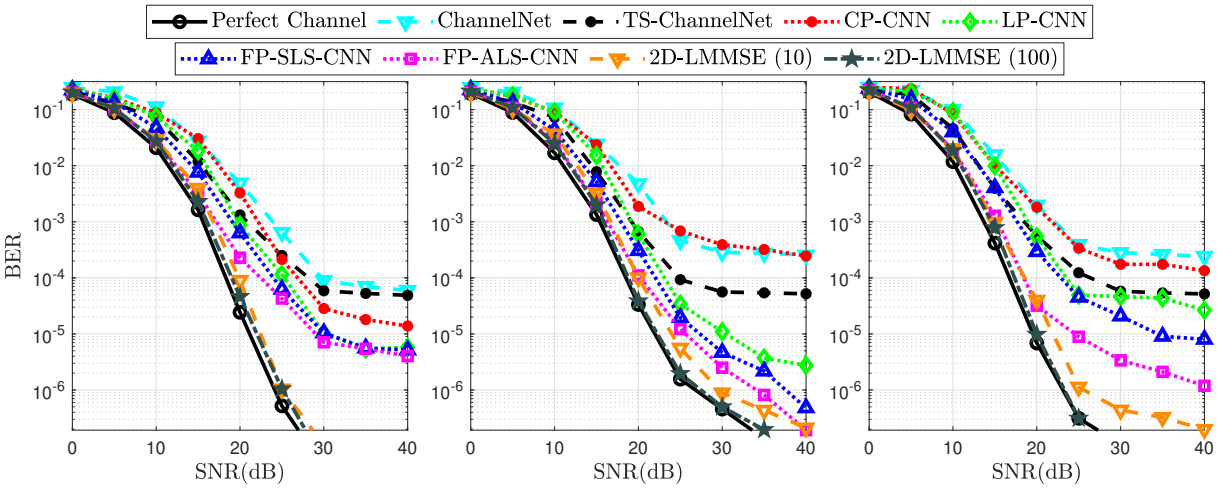


(b) BER of WI estimators employing 16QAM.

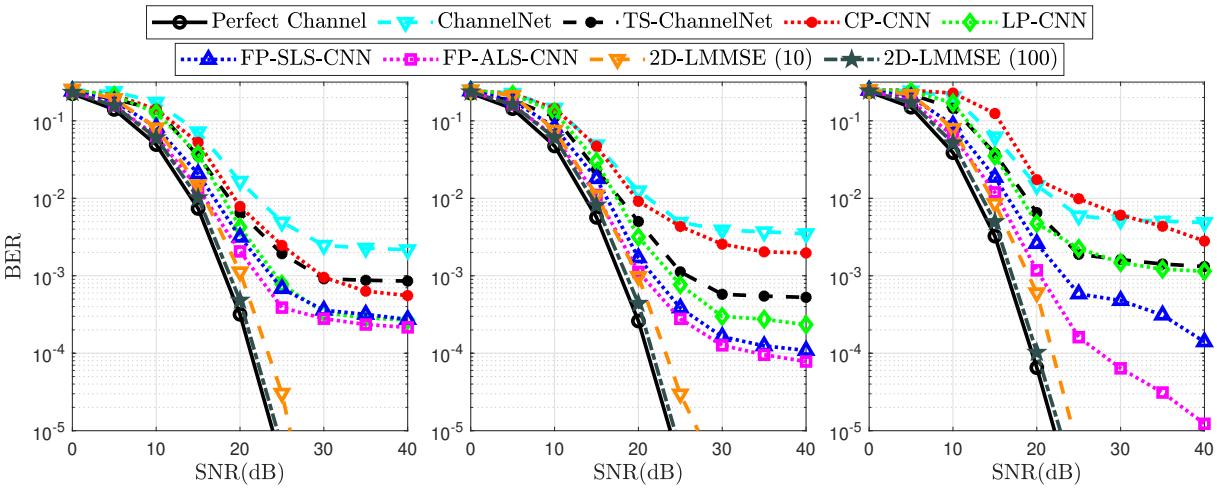
Figure 6.8: BER for $I = 100$, QPSK, mobility from left to right: low ($v = 45$ Kmph, $f_d = 250$ Hz), high ($v = 100$ Kmph, $f_d = 500$ Hz), very high ($v = 200$ Kmph, $f_d = 1000$ Hz). The CNN refers to SRCNN and DNCNN in low and high/very high) mobility scenarios, respectively.

6.3.3 Frame Length

Figure 6.10 shows the BER performance of high mobility vehicular scenario employing QPSK modulation and different frame lengths. It can be clearly noticed that the proposed WI-FP-ALS estimator is able to outperform ChannelNet and TS-ChannelNet for different frame lengths, this is due to the robustness of the proposed WI-FP-ALS estimator, unlike the 2D RBF and ADD-TT interpolation techniques that suffer from a considerable estimation error even when a short frame is considered, which affects the performance of ChannelNet and TS-ChannelNet. Moreover, employing the optimized DN-CNN after the WI-FP-ALS estimator improves significantly the BER performance. Table 6.2 illustrates the performance gain of the proposed WI-FP-ALS-DN-CNN estimator compared to the TS-ChannelNet estimator in high mobility scenario, where employing the optimized DN-CNN leads to 5 dB and 10 dB gain in terms of SNR for a BER $= 10^{-3}$ for QPSK and 16QAM modulation orders respectively.



(a) BER of CNN post processing employing QPSK.



(b) BER of CNN post processing employing 16QAM.

Figure 6.9: BER for $I = 100$, 16QAM, mobility from left to right: low ($v = 45$ Kmph, $f_d = 250$ Hz), high ($v = 100$ Kmph, $f_d = 500$ Hz), very high ($v = 200$ Kmph, $f_d = 1000$ Hz). The CNN refers to SRCNN and DNCNN in low and high/very high) mobility scenarios, respectively.

Table 6.2: BER performance gain (dB) of the proposed WI-FP-ALS compared to the TS-ChannelNet estimator in different mobility scenarios.

Scheme	**Low**		**High**		**Very High**	
	WI	SRCNN	WI	DNCNN	WI	DNCNN
QPSK	4	5	3	4	2	5
16QAM	2	6	3	4	5	10

6.3.4 CNN Architecture

The ChannelNet estimator employs SR-CNN and DN-CNN after the 2D RBF interpolation. The used SR-CNN consists of three convolutional layers with

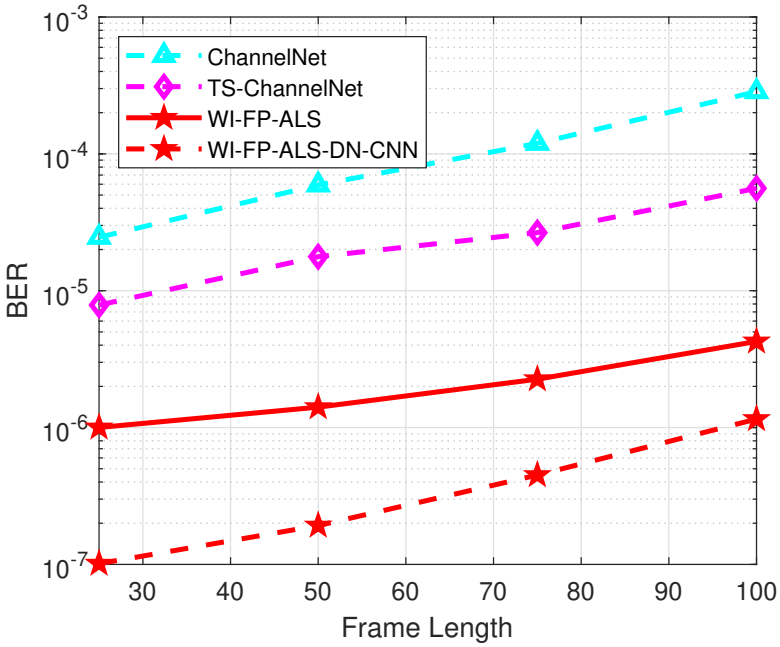


Figure 6.10: BER performance of VTV-SDWW high mobility vehicular channel model employing QPSK modulation and different frame lengths.

$(v_1 = 9; f_1 = 64), (v_2 = 1, f_2 = 32)$ and $(v_3 = 5, f_3 = 1)$ respectively. Moreover, the DN-CNN depth is $D = 18$ with $3 \times 3 \times 32$ kernels in each layer. On the other hand, super resolution convolutional long short-term memory (SR-ConvLSTM) network consists of three ConvLSTM layers of $(v_1 = 9; f_1 = 64), (v_2 = 1, f_2 = 32)$ and $(v_3 = 5, f_3 = 1)$ respectively is integrated after the ADD-TT interpolation in the TS-ChannelNet estimator. We note that, the SR-ConvLSTM network combines both the CNN and the LSTM networks [Shi et al., 2015], which increases the overall computational complexity as we discuss later. In contrast, the employed optimized SR-CNN and DN-CNN decreases significantly the complexity due to the accuracy of the proposed WI estimators. In conclusion, we can observe that, as the accuracy of the pre-estimation increases, the complexity of the employed CNN decreases, since low-complexity architectures can be used and vise versa.

6.4 Computational Complexity Analysis

In this section, a detailed computational complexity, TDR, and latency analysis of the 2D LMMSE estimator, ChannelNet, TS-ChannelNet, and the proposed WI estimators are presented.

6.4.1 Computational Complexity Analysis

6.4.1.1 2D LMMSE estimator

The conventional 2D LMMSE estimator requires first the LS estimation as that requires $2K_pI$ divisions. Then, the matrix inverse operation requires $4K_p^3I^3$ multiplications and $3K_p^3I^3$ summations. Finally, the correlation matrices are multiplied by the LS estimated channel vector resulting in $K_p^2I^2 + K_d^2K_p^2I^4$ multiplications. Therefore, the overall computational complexity of the conventional 2D LMMSE estimator is $4K_p^3I^3 + K_p^2I^2 + K_d^2K_p^2I^4 + 2K_pI$ multiplications and $3K_p^3I^3 + 2K_pI$ summations. We note that, in case the full $\boldsymbol{W}_{\text{2D-LMMSE}}$ matrix is calculated offline, the computational complexity of the 2D-LMMSE estimator is reduced to $4K_dK_p^2I^2 + 2K_pI$ multiplications and $3K_dK_p^2I^2 + 2K_dK_pI^2 - 2K_dI$ summations. We can notice that the 2D LMMSE suffers from very high computational complexity that make it impractical estimator in real-time scenarios.

6.4.1.2 ChannelNet estimator

The ChannelNet estimator employs the RBF interpolation followed by SR-CNN and DN-CNN networks. Thus, the overall computational complexity of the ChannelNet estimator can be expressed as follows

$$\text{CC}_{\text{ChannelNet}} = \text{CC}_{\text{RBF}} + \text{CC}_{\text{SR-CNN}} + \text{CC}_{\text{DN-CNN}}. \tag{6.13}$$

The calculation of $\hat{\hat{\boldsymbol{H}}}_{\text{LS}}$ requires $2K_pI$ divisions. $\boldsymbol{w}_{\text{RBF}}$ calculation requires $4K_p^2I^2$ multiplications/divisions and $5K_p^2I^2 - 2K_pI$ summations/subtractions. On the other hand, $\hat{\hat{\boldsymbol{H}}}_{\text{RBF}}$ computation requires $K_dI(K_p^2I^2 + 3K_pI)$ multiplications/divisions and $5K_dK_pI^2$ subtractions/summations. Therefore the total computational complexity of the RBF interpolation can be expressed by $K_p^2I^2(4+K_dI)+K_pI(2+3K_dI)$ multiplications/divisions and $K_pI(5K_pI + 5K_dI - 2)$ summations/subtractions. Thereafter, the ChannelNet estimator applies SR-CNN followed by DN-CNN on top of the RBF interpolation. $\text{CC}_{\text{SR-CNN}}$ and $\text{CC}_{\text{DN-CNN}}$ can be computed as follows

$$\begin{aligned}\text{CC}_{\text{SR-CNN}} &= \sum_{j=1}^{J} h_jw_jd_jv_j^2f_j + h_jw_jd_jf_j \\ &= \sum_{j=1}^{J} h_jw_jd_jf_j(v_j^2+1).\end{aligned} \tag{6.14}$$

$$\text{CC}_{\text{DN-CNN}} = \sum_{j=1}^{J} h_jw_jd_jf_j(v_j^2+1) + \sum_{j=1}^{D} 4h_jw_jd_j, \tag{6.15}$$

where J denotes the number of employed CNN layers. We note that the second term in $\text{CC}_{\text{DN-CNN}}$ represents the number of operations required by the batch

Table 6.3: CNN-based estimators overall computation complexity.

Scheme	Interpolation		CNN	
	Mul./Div.	**Sum./Sub.**	**Mul./Div.**	**Sum./Sub.**
ChannelNet	$K_p^2 I^2(4+K_dI)$ $+ K_pI(2+3K_dI)$	$K_pI(5K_pI$ $+ 5K_dI-2)$	$350144K_{\text{on}}I$	$42432K_{\text{on}}I$
TS-ChannelNet	$24K_{\text{on}}I + 4LK_{\text{on}}I$	$18K_{\text{on}}I$ $+ 5K_{\text{on}}IL$	$226880K_{\text{on}}I$	$81472K_{\text{on}}I$
FP-SLS-SR-CNN	$2K_{\text{on}}P + 2K_{\text{on}}$ $+ 4K_{\text{on}}I_d$	$2K_{\text{on}}$ $+ 2K_{\text{on}}I_d$	$7008K_{\text{on}}I_d$	$1120K_{\text{on}}I_d$
FP-ALS-SR-CNN	$4K_{\text{on}}^2P + 2K_{\text{on}}P$ $+ 2K_{\text{on}} + 4K_{\text{on}}I_d$	$5K_{\text{on}}^2P$ $+ 2K_{\text{on}}I_d$		
LP-SR-CNN	$2LP + 4K_{\text{on}}LP$ $+ 2K_{\text{on}} + 4K_{\text{on}}I_d$	$5K_{\text{on}}LP$ $+ 2K_{\text{on}}I_d$		
CP-SR-CNN	$8P + 16K_{\text{on}}P$ $+ 2K_{\text{on}} + 4K_{\text{on}}I_d$	$20K_{\text{on}}P$ $+ 2K_{\text{on}}I_d$		
FP-SLS-DN-CNN	$2K_{\text{on}}P + 2K_{\text{on}}$ $+ 4K_{\text{on}}I_d$	$2K_{\text{on}}$ $+ 2K_{\text{on}}I_d$	$84096K_{\text{on}}I_d$	$9856K_{\text{on}}I_d$
FP-ALS-DN-CNN	$4K_{\text{on}}^2P + 2K_{\text{on}}P$ $+ 2K_{\text{on}} + 4K_{\text{on}}I_d$	$5K_{\text{on}}^2P$ $+ 2K_{\text{on}}I_d$		
LP-DN-CNN	$2LP + 4K_{\text{on}}LP + 2K_{\text{on}}$ $+ 4K_{\text{on}}I_d$	$5K_{\text{on}}LP$ $+ 2K_{\text{on}}I_d$		
CP-DN-CNN	$8P + 16K_{\text{on}}P$ $+ 2K_{\text{on}} + 4K_{\text{on}}I_d$	$20K_{\text{on}}P$ $+ 2K_{\text{on}}I_d$		

normalization employed in the DN-CNN network. Therefore, the SR-CNN employed in the ChannelNet estimator requires $16064K_{\text{on}}I$ multiplications/divisions and $4288K_{\text{on}}I$ summations/subtractions, while the ChannelNet DN-CNN computations require $334080K_{\text{on}}I$ multiplications/divisions and $38144K_{\text{on}}I$ summations/subtractions.

6.4.1.3 TS-ChannelNet estimator

The TS-ChannelNet estimator applies the average decision-directed with time truncation (ADD-TT) interpolation followed by the SR-ConvLSTM network. Thus, the overall computational complexity of the TS-ChannelNet estimator can be expressed as follows

$$\text{CC}_{\text{TS-ChannelNet}} = \text{CC}_{\text{ADD-TT}} + \text{CC}_{\text{SR-ConvLSTM}}. \tag{6.16}$$

The ADD-TT interpolation applies first the data-pilot aided (DPA) estimation that requires $18K_{\text{on}}$ multiplications/divisions and $8K_{\text{on}}$ summations/subtractions. The time domain truncation operation applied in (C.6) requires $4LK_{\text{on}}$ multiplications and $5K_{\text{on}}L - 2K_{\text{on}}$ summations. After time domain truncation, the ADD-TT interpolation applies frequency and time domain averaging. The frequency domain averaging (C.7) requires $10K_{\text{on}}$ summations, and $2K_{\text{on}}$ multiplications. Moreover, the time domain averaging step (C.8), requires $4K_{\text{on}}$ real valued divisions, and $2K_{\text{on}}$ real valued summations. Therefore, the overall computational complexity of the ADD-TT interpolation for the whole received OFDM frame requires $24K_{\text{on}}I + 4LK_{\text{on}}I$ real valued multiplications/divisions, and $18K_{\text{on}}I + 5K_{\text{on}}IL$ real valued summations/subtractions, and its computational complexity is expressed in terms of the overall operations applied in the input, forget, and output gates of the SR-ConvLSTM network, such that

$$\text{CC}_{\text{ConvLSTM}} = \sum_{j=1}^{J} h_j w_j d_j f_j (8v_j^2 + 30). \tag{6.17}$$

Based on (6.17), the SR-ConvLSTM network employed in the TS-ChannelNet estimator requires $226880K_{\text{on}}I$ multiplications/divisions and $81472K_{\text{on}}I$ summations/subtractions.

TS-ChannelNet estimator is less complex than the ChannelNet estimator, since it employs only one CNN on top of the ADD-TT interpolation, unlike the ChannelNet estimator where both SR-CNN and DN-CNN are employed.

6.4.1.4 Proposed WI estimators

The proposed WI estimators computational complexity depends mainly on the selected frame structure, the pilot allocation scheme, and the selected optimized CNN. The overall computational complexity of the proposed WI estimators can be expressed as follows

$$\text{CC}_{\text{WI}} = \text{CC}_{\hat{\boldsymbol{H}}_{\text{WI}}} + \text{CC}_{\text{O-CNN}}. \tag{6.18}$$

In the case of inserting full pilot symbols two options are considered. The first option is the SLS estimator where it requires only $2K_{\text{on}}P + 2K_{\text{on}}$ divisions, and $2K_{\text{on}}$ summations. The second option is employing the ALS estimator, where $2K_{\text{on}}P + 2K_{\text{on}}$ divisions, followed by $4K_{\text{on}}^2P$ multiplications, and $5K_{\text{on}}^2P$ summations are required. On the other hand, when $K_{\text{p}} = L$ pilots are inserted with each pilot symbol, then the LS estimation requires $2LP + 2K_{\text{on}}$ divisions, $4K_{\text{on}}LP$ multiplications, and $5K_{\text{on}}LP$ summations. Similarly for employing only $K_p = 4$ pilot subcarriers within the inserted pilot symbols, the WI-CP estimator requires $8P + 2K_{\text{on}}$ divisions, $16K_{\text{on}}P$ multiplications, and $20K_{\text{on}}P$ summations. After selecting the required frame structure and pilot allocation scheme, the proposed estimators apply the weighted interpolation as shown in (6.7), where the channel estimation for each received OFDM frame requires $4K_{\text{on}}I_d$ divisions and $2K_{\text{on}}I_d$ summations. Finally, the optimized SR-CNN is employed in low mobility scenario and

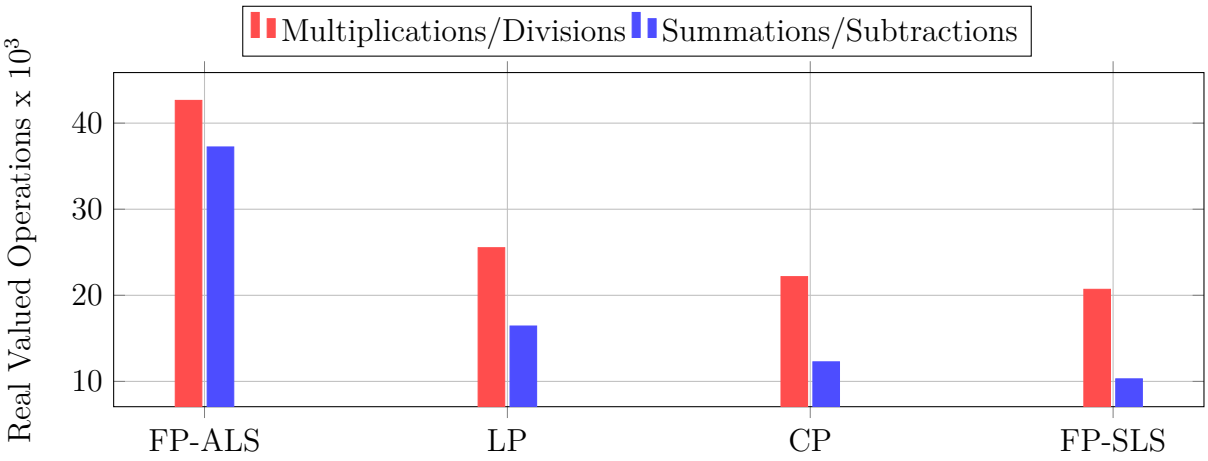


Figure 6.11: Computational complexities of the proposed WI estimators employing P = 2.

it requires $7008K_{\text{on}}I_d$ multiplications/divisions and $1120K_{\text{on}}I_d$ summations/subtractions, while the optimized DN-CNN employed in high and very high mobility scenarios requires $84096K_{\text{on}}I_d$ multiplications/divisions and $9856K_{\text{on}}I_d$ summations/subtractions

The proposed WI-FP-ALS records the higher computational complexity among the other proposed estimators in all mobility scenarios, due to the $\boldsymbol{W}_{\text{ALS}}$ calculation in (6.3). Moreover, the proposed WI-FP-SLS estimator is the simplest one. Fig. 6.11 shows the computational complexity of the proposed WI estimators employing $P = 2$ pilot symbols within the transmitted frame.

Table 6.3 shows the overall computational complexity of the studied estimators in terms of real valued operations. It is worth mentioning that the proposed WI estimators achieve significant computational complexity decrease compared to ChannelNet and TS-ChannelNet estimators. Fig. 6.12 shows the computational complexity of ChannelNet and TS-ChannelNet compared to FP-ALS-SR-CNN. The ChannelNet and TS-ChannelNet estimators are 70 and 39 times more complex than the proposed FP-ALS-SR-CNN respectively. Moreover, the proposed estimators achieve at least 7027.35 times less complexity than the 2D LMMSE estimator, with an acceptable BER performance, thus making them a good alternative to the 2D LMMSE. We note that FP-ALS-DN-CNN is 12 times more complex than FP-ALS-SR-CNN since the optimized DN-CNN architecture complexity that is employed in high and very high scenarios is higher than the optimized SR-CNN architecture which is used in low mobility scenario.

6.4.2 Transmission Data Rate and Latency Analysis

The TDR and the latency introduced at the receiver in order to recover the transmitted bits are important issues in vehicular communications, especially when real time applications are employed [Ashraf et al., 2017]. The transmission data rate is influenced

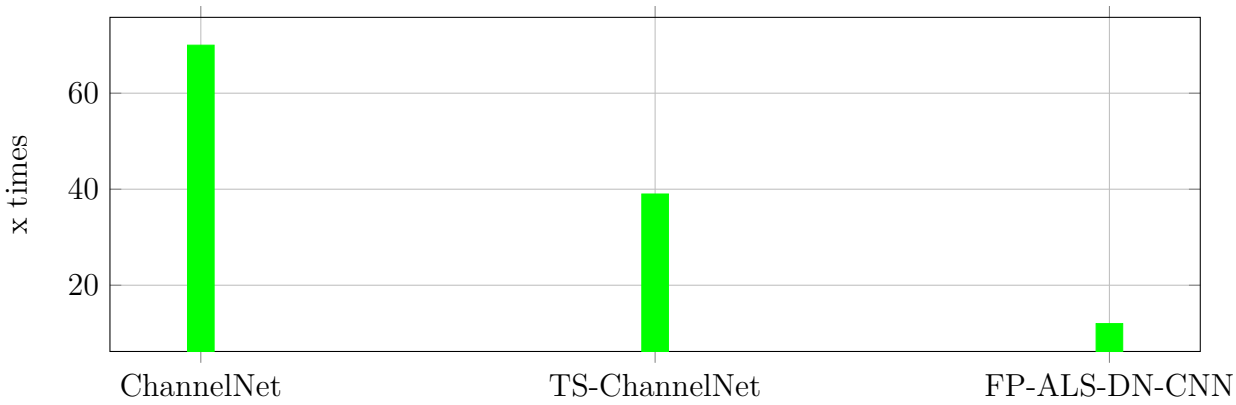


Figure 6.12: Computational complexities of the SoA CNN-based estimators compared to the proposed FP-ALS-SR-CNN.

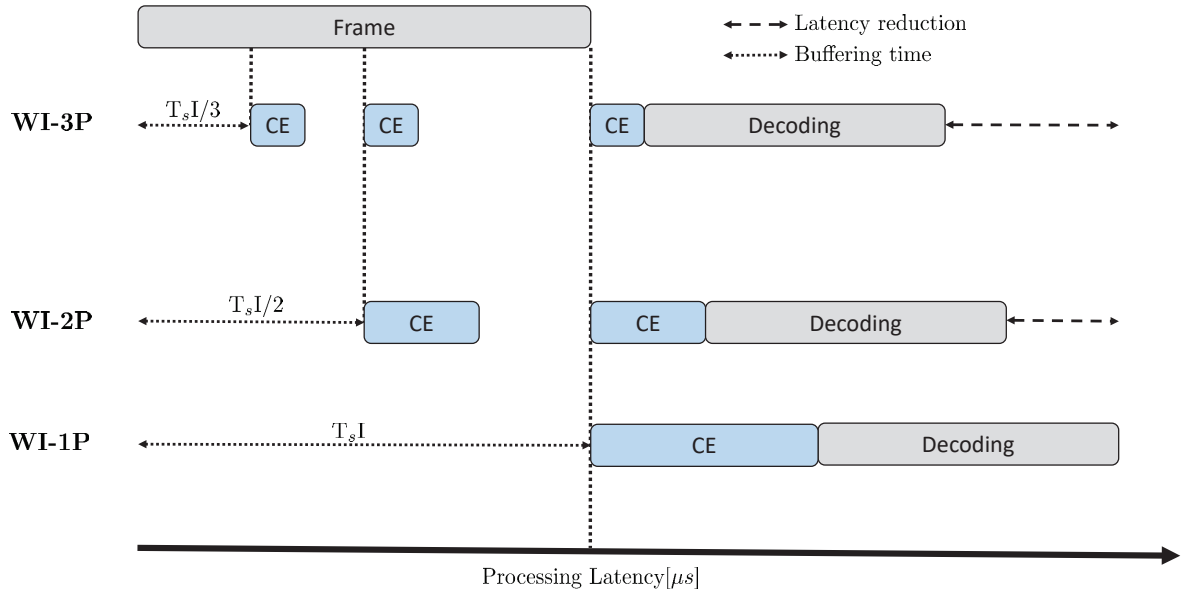


Figure 6.13: Processing latency of the proposed WI estimators.

by the number of allocated data subcarriers within the transmitted frame, such that

$$\mathrm{TDR} = \frac{K_{\mathrm{DF}} \log_2(M)\rho}{T_{\mathrm{s}} I}, \tag{6.19}$$

Table 6.4: Transmission data rate and buffering time analysis for the proposed WI estimators.

Estimator	**WI-1P**			**WI-2P**			**WI-3P**		
	FP	LP	CP	FP	LP	CP	FP	LP	CP
TDR gain	7.25%	8.08%	8.25%	6.16%	7%	7.16%	5.1%	7.58%	8.08%
φ [μs]	800			400			265		

where K_{DF} and ρ denote the total allocated data subcarriers within the transmitted frame, and the employed code rate respectively, and M represents the employed modulation order. Moreover, the buffering time φ can be expressed by the total duration that the receiver should wait before starting the channel estimation, such that

$$\varphi = T_{\mathrm{s}} I. \tag{6.20}$$

where T_{s} represents the received OFDM data symbol duration.

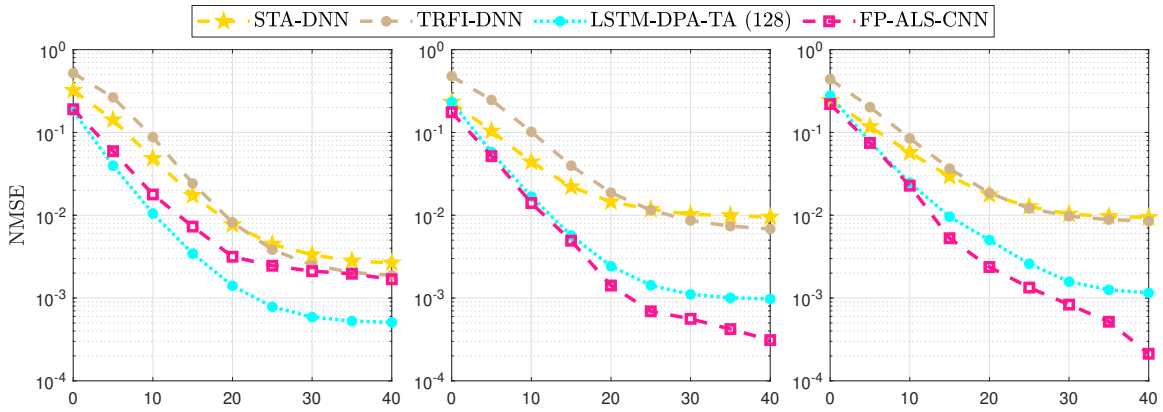
ChannelNet and TS-ChannelNet estimators have no TDR gain since they employ $K_p = 4$ pilot subcarriers within each transmitted OFDM symbol. Moreover, they also require the full reception time ($\varphi = 800$ μs in our case) before starting the channel estimation process. Therefore, ChannelNet and have similar transmission data rate as defined in the IEEE 802.11p standard. On the other hand, ChannelNet and TS-ChannelNet suffer from high buffering time at the receiver, since the full frame should be received before the channel estimation starts, thus leading to high latency.

As shown in Table 6.4, the proposed WI estimators record different TDRs gains according to the employed frame structure. Moreover, the proposed WI-2P and WI-3P estimators require lower buffering time than the proposed WI-1P, ChannelNet, and TS-ChannelNet estimators, since it divides the frame into several sub-frames as shown in Fig. 6.13. Thus, the channel estimation process starts earlier. Therefore, the proposed WI estimators contribute in reducing the total required latency. Finally, we note that the transmission parameters and the chosen frame structure should be adapted according to the mobility condition, required data rate, and the acceptable latency by the vehicular application.

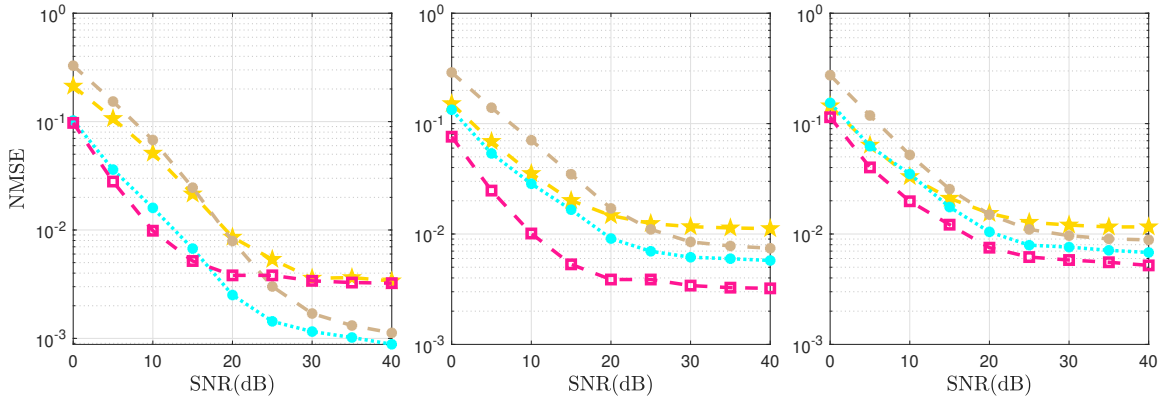
6.4.3 SBS vs. FBF Estimators

In this section, the proposed DL-based SBS and FBF estimators are compared and discussed using the same simulation setup as defined in the previous simulations. Figure. 6.15 and Figure. 6.14 show the BER and NMSE performance of the proposed DL-based estimators in low, high, and very high mobility scenarios employing QPSK and 16QAM modulation orders.

In low-mobility scenario, the proposed LSTM-based SBS estimator outperforms the proposed WI-FP-ALS-DNCNN FBF estimator. This can be explained by the ability of LSTM to learn the channel time correlation than the DNCNN, since in low mobility scenario, Doppler error is somehow negligible. However, in high and very high mobility scenarios, WI-FP-ALS-DNCNN shows a significant performance superiority, where it outperforms the proposed LSTM-based SBS estimator by 3 dB gain in terms of SNR for a BER = 10^{-4}. We note that in high mobility vehicular scenarios, where the Doppler error impact is high, LSTM records some performance degradation since learning the time correlation between successive samples is not achievable as the low mobility scenario case. On the other hand, DNCNN network is able to significantly alleviate the impact



(a) NMSE performance employing QPSK.

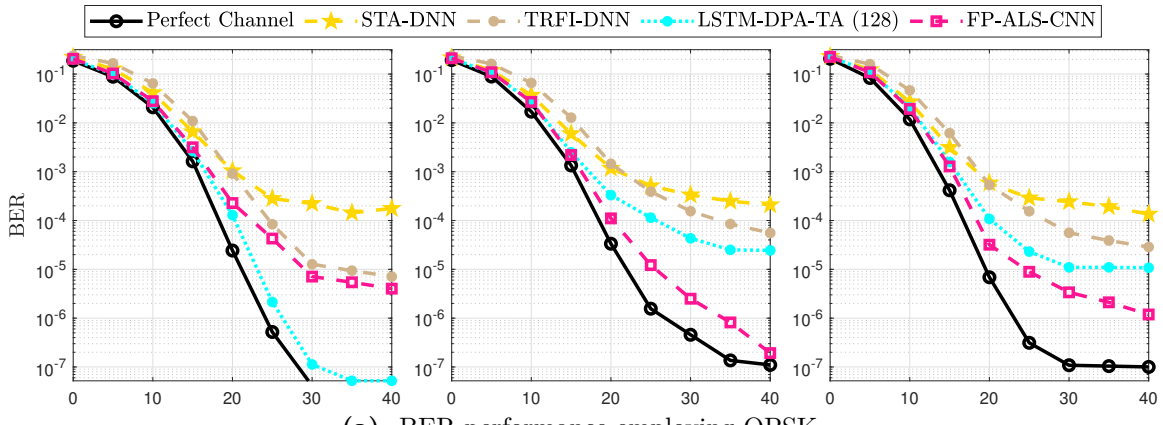


(b) NMSE performance employing 16QAM.

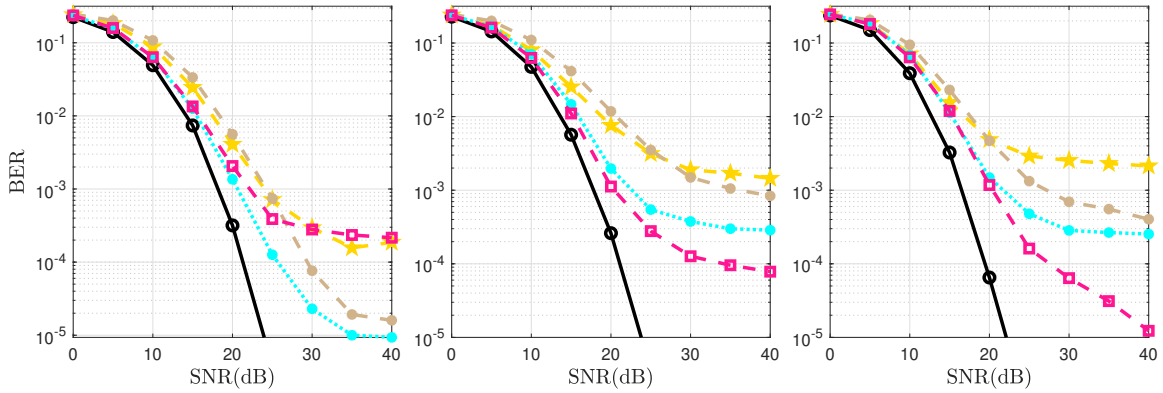
Figure 6.14: NMSE performance employing three scenarios: (*i*) first column - low mobility (v = 45 Kmph, f_d = 250 Hz) (*ii*) second column - high mobility (v = 100 Kmph, f_d = 500 Hz) (*iii*) third column - very high mobility (v = 200 Kmph, f_d = 1000 Hz). The CNN refers to SRCNN and DNCNN in low and high/very high) mobility scenarios, respectively.

of noise and Doppler error, where it records at least 5 dB gain in terms of SNR for a BER = 10^{-4}. As a conclusion, we can say that employing LSTM network instead of DNN and DNCNN networks gives better performance in low-mobility scenarios. In contrast, DNCNN is more useful in high and very high mobility scenarios. This is because DNCNN takes the advantage of using the whole pilot subcarriers within the received frame. As a summary, we can say that as the mobility increases, the time correlation between successive received OFDM symbols decreases. Therefore, the performance of LSTM degrades compared to CNN. Whereas, in high mobility scenarios the proposed CNN-based estimators become more useful than the proposed LSTM-based estimators.

However, we should take into consideration that even though the performance is improved by using LSTM and CNN networks, but more computational complexity is required where the proposed LSTM-based SBS estimator is 23.6 times more complex than the proposed DNN-based SBS estimators. On the other hand, the proposed CNN-based FBF estimator is more complex than the proposed LSTM-based SBS estimator by around 3450 times. This is due to the great difference in terms of required operations between the CNN and



(a) BER performance employing QPSK.



(b) BER performance employing 16QAM.

Figure 6.15: BER performance employing three scenarios: (*i*) first column - low mobility (v = 45 Kmph, f_d = 250 Hz) (*ii*) second column - high mobility (v = 100 Kmph, f_d = 500 Hz) (*iii*) third column - very high mobility (v = 200 Kmph, f_d = 1000 Hz). The CNN refers to SRCNN and DNCNN in low and high/very high) mobility scenarios, respectively.

LSTM networks. Finally, we note that, the choice of the vehicular channel estimator is mainly related to the vehicular applications requirements and the allowed computational complexity limits, if the application is sensitive to latency, than SBS estimators are more useful, while if some latency can be accepted, FBF estimators can be employed. As a summary, a trade off between the required performance, computational complexity, and the accepted latency should be first defined in order to select what is the best vehicular channel estimator to choose.

6.5 Conclusion

In this Chapter, FBF channel estimation in vehicular communication is studied, where the limitations of the conventional 2D LMMSE estimator and the motivation behind employing CNN processing in the channel estimation are presented. Moreover, The recently proposed CNN-based channel estimators have been extensively surveyed. In this

context, we have proposed a hybrid, adaptive, and robust WI channel estimators for the IEEE 802.11p standard, where pilot symbols are inserted within the transmitted frame, with several pilot allocation schemes adapted according to the mobility condition. Unlike the recently proposed ChannelNet and TS-ChannelNet estimators that suffer from high computational complexity, performance degradation in high mobility vehicular scenarios, and high latency at the receiver, the proposed WI estimators have reduced computational complexity and robustness in high mobility scenarios. Moreover, they require low buffering time at the receiver and more TDR gain is achieved since all the OFDM symbols within the transmitted frame are fully allocated to data. Additionally, the employed SR-CNN and DN-CNN architectures are optimized through intensive experiments in order to alleviate the high complexity problem. Simulation results have shown the performance superiority of the proposed WI estimators over ChannelNet and TS-ChannelNet estimators in all vehicular scenarios with a substantial reduction in computational complexity, where the ChannelNet and the TS-ChannelNet estimators are more complex than the proposed WI-FP-ALS-SR-CNN by 70 and 39 times respectively. On the other hand, the proposed estimators are less complex than the 2D LMMSE estimator by at least 7027.35 times while recording a convenient BER performance especially in high mobility vehicular scenarios, which makes them suitable alternatives with a good performance vs. complexity tradeoff.

Chapter 7

Conclusions and Perspectives

Summary of the Contributions

This thesis has addressed the topic of doubly-dispersive channel estimation, with specific focus on vehicular communications as an application. In order to investigate this topic, we have given a detailed background of time and frequency selective channels, vehicular communications system model and a state of art overview of the classical channel estimation methods. In order to improve the conventional channel estimators accuracy, we have first analyzed and defined their limitations, then we have presented the two following approaches:

- ▷ **Propose new DFT-based symbol-by-symbol (SBS) estimators:** as illustrated in Chapter 4, we have focused on overcoming the conventional estimators drawbacks represented by the data-pilot aided (DPA) demapping error that is enlarged during the estimation process throughout the frame. In this context, we have proposed low complex and robust channel estimators namely truncated discrete Fourier transform (T-DFT) and temporal averaging T-DFT (TA-TDFT). Unlike conventional estimators, the proposed estimators are based on the truncated discrete Fourier transform (DFT)-based interpolation, and do not require the DPA estimation. Therefore, the enlarged DPA demapping error is totally eliminated in the proposed estimators. Moreover, the simulation results are analytically verified by deriving the proposed estimators NMSE error derivations. The performance superiority of the proposed T-DFT and TA-TDFT estimators over conventional estimators in low and high mobility vehicular scenarios are achieved with a significant computational complexity decrease.
- ▷ **Integrate DL techniques with conventional estimators**: In this approach, we have employed DL techniques for SBS and FBF estimators, where DNN and LSTM networks are integrated with the conventional STA, TRFI and DPA estimators. Simulation results performed using high mobility vehicular channel

model have shown that the proposed spectral temporal averaging (STA)-deep neural network (DNN) and time domain reliable test frequency domain interpolation (TRFI)-DNN outperform conventional estimators, and auto-encoder deep neural network (AE-DNN) while reducing the computational complexity by around 50% compared with the recently proposed AE-DNN. On the other hand, the proposed LSTM-based estimators record a significant performance improvement over the conventional estimators and the DNN-based estimators. This can be explained by the ability of LSTM in learning time correlations better than DNN since it employs the previous output and the current input in estimating the current output.
In Chapter 6, frame-by-frame (FBF) channel estimators are studied, where the limitations of the conventional 2D linear minimum mean square error (LMMSE) estimator and the motivation behind employing convolutional neural network (CNN) processing in the channel estimation are presented. In this context, we have proposed a hybrid, adaptive, and robust weighted interpolation (WI) channel estimators for the IEEE 802.11p standard, where pilot symbols are inserted within the transmitted frame, with several pilot allocation schemes adapted according to the mobility condition. Unlike the recently proposed channel network (ChannelNet) and Temporal spectral ChannelNet (TS-ChannelNet) estimators that suffer from high computational complexity, performance degradation in high mobility vehicular scenarios, and high latency at the receiver, the proposed WI estimators have reduced computational complexity and robustness in high mobility scenarios. Moreover, they require low buffering time at the receiver and more transmission data rate (TDR) gain is achieved since all the orthogonal frequency division multiplexing (OFDM) symbols within the transmitted frame are fully allocated to data. Simulation results have shown the performance superiority of the proposed WI estimators over ChannelNet and TS-ChannelNet estimators in all vehicular scenarios with a substantial reduction in computational complexity, where the ChannelNet and the TS-ChannelNet estimators are more complex than the proposed WI-FP-ALS-SR-CNN by 70 and 39 times respectively. On the other hand, the proposed estimators are less complex than the 2D LMMSE estimator by at least 10^4 times while recording a convenient BER performance especially in high mobility vehicular scenarios, which makes them good alternatives to the 2D LMMSE estimator. Finally, we note that the employed DNN, long short-term memory (LSTM), super resolution CNN (SR-CNN) and denoising CNN (DN-CNN) architectures are optimized through intensive experiments employing the grid search algorithm in order to alleviate the high complexity problem.

Finally, we note that the proposed estimators show a good robustness and generalization ability in fast fading environments even when some of the training and testing conditions change. Moreover, the choice of the appropriate estimator should be based on a trade off between the application TDR requirement, the desired performance, the computational

complexity cost, and the latency sensitivity. Studying deeply this trade off can optimize the use of the proposed SBS and FBF vehicular estimators.

Perspectives

The contributions of this work open several perspectives. In particular, it would be interesting to study the following ideas:

- ▷ Doubly dispersive channel prediction: in order to allow more data gain, the channel in the current symbol can be predicted from previous symbols without the need of using pilots for each single symbol, especially in low mobility scenarios, when the channel time correlation is high.
- ▷ Deep Learning Hardware Implementation: The hardware implementation of the proposed DL-based channel estimators is important for further validating the numerical simulation results. It is worth mentioning that, we have already started a collaborative project with **Indraprastha Institute of Information Technology Delhi** in India and the DNN hardware implementation is almost ready, where the simulation results obtained in Chapter 5 for the frequency-selective LS-DNN channel estimator have been validated by the developed DNN hardware. We note that, now we are working on the hardware implementation of the LSTM and CNN networks in order to validated our doubly-selective DL-based channel estimators.
- ▷ MIMO Channel estimation: an extension of this work would consider a MIMO channel where the high dimension of the channel would justify even more the use of deep learning.
- ▷ Power amplifier non-linear distortions: while we assume in this thesis a perfect RF front-end, it would be worth studying how the non linear distortions resulting from the high power amplifier will affect the performance, and if the neural networks architectures are able to jointly estimate the channel and the RF nonlinearities.
- ▷ Wideband channel estimation in high frequencies: the deep learning based channel estimation methods can be extended to cover channel estimation for the mmWave and Terahertz spectrum, where the propagation environment is more challenging

Appendix A

Cubic interpolation

Let $h[k] = h_Q[k] + jh_I[k]$ be the channel function. Assume, the channel is known at the subcarriers k_1 and k_2. In order to find the channel at the unknown subcarrier $k_1 < k < k_2$, we consider cubic interpolation per real and imaginary component. First, we change the variable $x = \frac{k-k_1}{k_2-k_1}$ and considering the real part, we define $f(x) = h_Q[k]$ such that:

$$f(x) = a_3x^3 + a_2x^2 + a_1x + a_0,\ x \in [0,1]. \tag{A.1}$$

In order to compute the coefficients $\{a_k\}$, four equations are required. Two equations are obtained by:

$$\begin{aligned} h_Q[k_1] &= f(0) = a_0, \\ h_Q[k_2] &= f(1) = a_3 + a_2 + a_1 + a_0, \end{aligned} \tag{A.2}$$

and another two equations by using the derivative:

$$\begin{aligned} f'(0) &= a_1, \\ f'(1) &= 3a_3 + 2a_2 + a_1. \end{aligned} \tag{A.3}$$

As a result, $a_0 = f(0)$ and $a_1 = f'(0)$. In addition, a_2 and a_3 can be computed from the equations:

$$\begin{aligned} a_3 + a_2 &= f(1) - f(0) - f'(0), \\ 3a_3 + 2a_2 &= f'(1) - f'(0). \end{aligned}$$

The solution is given by:

$$\begin{aligned} a_2 &= 3[f(1) - f(0)] - f'(1) - 2f'(0), \\ a_3 &= -2[f(1) - f(0)] + f'(1) + f'(0). \end{aligned}$$

As assuming that the channel is known at $k_0 \leq k_1$ and $k_3 \geq k_2$, the derivatives can be computed as:

$$\begin{aligned} f'(0) &= \frac{h_Q[k_2] - h_Q[k_0]}{k_2 - k_0}, \\ f'(1) &= \frac{h_Q[k_3] - h_Q[k_1]}{k_3 - k_1}. \end{aligned} \tag{A.4}$$

This guarantees that the polynomials in different segments are continuous. In the case where k_1 is a boundary subcarrier, then $k_0 = k_1$, similar for boundary k_2; $k_3 = k_1$.

Based on that, in the worst case, the computation efforts required to interpolate the real part of the channel gain requires 4 subtractions and 4 divisions to compute the derivatives. To compute the polynomial coefficients, we need 3 multiplications and 6 summations/subtractions. To compute the gain, first x is computed by 2 subtractions and 1 division. Afterwards, computing $f(x)$ is achieved by 5 multiplications and 3 summations. In total, it requires 13 multiplications/divisions and 15 summations/subtractions. Similar process is repeated for the imaginary part, and therefore, to interpolate the channel gain at one subcarrier between two known subcarriers, it is required to perform 26 multiplications/divisions and 30 summations/subtractions.

Appendix B

time averaging (TA) noise power alleviation ratio

According to the TA-TDFT channel estimation applied in (4.12), we consider that $\hat{\tilde{\boldsymbol{h}}}_{\text{TA-TDFT}_1} = \hat{\tilde{\boldsymbol{h}}}_{\text{TDFT}_1}$, therefore the noise power of the first TA-TDFT estimated channel is σ^2, and the noise power enhancement ratio for the successive TA-TDFT estimated channels can be computed as follows:

$$N_{\text{TA-TDFT}_i} = \begin{cases} 1, & i = 1 \\ \frac{1}{4} + \frac{1}{4} = \frac{1}{2}, & i = 2 \\ \frac{1}{8} + \frac{1}{4} = \frac{3}{8}, & i = 3 \\ \vdots & \\ \frac{N_{\text{TA-TDFT}_{i-1}}}{4} + \frac{1}{4}, & i = I \end{cases} \tag{B.1}$$

The generalization formula of (B.1) can be written as a sequence where the first element $N_{\text{TA-TDFT}_1} = 1$ as follows:

$$
\begin{aligned}
N_{\text{TA-TDFT}_i} &= \frac{1}{4} N_{\text{TA-TDFT}_{i-1}} + \frac{1}{4} \\
&= \frac{1}{4} \left(N_{\text{TA-TDFT}_{i-1}} + 1 \right) \\
&= \frac{1}{4} \left(\frac{1}{4} N_{\text{TA-TDFT}_{i-2}} + \frac{1}{4} + 1 \right) \\
&= \frac{1}{4} \left(\frac{1}{4} \left(\frac{1}{4} N_{\text{TA-TDFT}_{i-3}} + \frac{1}{4} \right) + \frac{1}{4} + 1 \right) \\
&= \frac{1}{4} \left(\frac{1}{4^2} N_{\text{TA-TDFT}_{i-3}} + \frac{1}{4^2} + \frac{1}{4} + 1 \right) \\
&= \frac{1}{4} \left(\frac{1}{4^{3-1}} N_{\text{TA-TDFT}_{i-3}} + \frac{1}{4^{3-1}} + \frac{1}{4^{3-2}} + \frac{1}{4^{3-3}} \right) \\
&\vdots \\
&= \frac{1}{4} \left(\frac{1}{4^{(i-1)-1}} N_{\text{TA-TDFT}_{i-(i-1)}} + \frac{1}{4^{(i-1)-1}} + \cdots + \frac{1}{4^0} \right) \\
&= \frac{1}{4} \left(\frac{1}{4^{i-2}} N_{\text{TA-TDFT}_1} + \frac{1}{4^{i-2}} + \frac{1}{4^{i-3}} + \cdots + \frac{1}{4^0} \right) \\
&= \frac{1}{4} \left(\frac{1}{4^{i-2}} + \frac{1}{4^{i-2}} + \frac{1}{4^{i-3}} + \cdots + \frac{1}{4^0} \right).
\end{aligned}
\tag{B.2}
$$

The sequence derived in (B.2) can be written as follows:

$$
\begin{aligned}
N_{\text{TA-TDFT}_i} &= \frac{1}{4} \left(\frac{1}{4^{i-2}} + \frac{1}{4^{i-2}} + \frac{1}{4^{i-3}} + \cdots + \frac{1}{4^0} \right) \\
&= \frac{1}{4} \left(\frac{1}{4^{i-2}} + \sum_{j=2}^{i} \left(\frac{1}{4} \right)^{i-j} \right)
\end{aligned}
\tag{B.3}
$$

Let $j' = i - j$, then (B.3) can be written in terms of j' such that:

$$
N_{\text{TA-TDFT}_i} = \frac{1}{4} \left(\frac{1}{4^{i-2}} + \sum_{j'=0}^{i-2} \left(\frac{1}{4} \right)^{j'} \right). \tag{B.4}
$$

According to the summation of geometric sequence rule [Orosi, 2016], (B.4) can be further simplified to:

$$
\begin{aligned}
N_{\text{TA-MDFT}_i} &= \frac{1}{4} \left(\frac{1}{4^{i-2}} + \sum_{j'=0}^{i-2} \left(\frac{1}{4} \right)^{j'} \right) = \frac{1}{4} \left(\frac{1}{4^{i-2}} + \frac{1 - \frac{1}{4}^{i-1}}{\frac{3}{4}} \right) \\
&= \frac{4^{i-1} + 2}{3 \times 4^{i-1}}.
\end{aligned}
\tag{B.5}
$$

Appendix C

SoA CNN-based Channel Estimators

ChannelNet Estimator

In [Soltani et al., 2019], the authors propose a CNN-based channel estimator denoted as ChannelNet scheme, where 2D radial basis function (RBF) interpolation is applied as an initial channel estimation. The basic motivation of the 2D RBF interpolation is to approximate multidimensional scattered unknown data, from their surrounding neighbors known data, employing the the radial basis function [Toit, 2008]. To do so, the distance function is calculated between every data point to be interpolated and its neighbours, where closer neighbors are assigned higher weights. After that, the RBF interpolated frame is considered as a low resolution image, where SR-CNN is utilized to get better estimation. Finally, in order to alleviate the impact of noise within the high resolution estimated frame, DN-CNN is implemented on top of the SR-CNN resulting in a high resolution, noise alleviated estimated channels. The ChannelNet estimator considers sparsed allocated pilots within the IEEE 802.11p frame and it first applies the least squares (LS) estimation to the pilot subcarriers within the received OFDM frame. After that, The 2D RBF interpolation is obtained by the weighted summation of the distance between each data subcarrier to be interpolated and all the pilot subcarriers within the received OFDM frame, such that:

$$\hat{\hat{\boldsymbol{H}}}_{\mathrm{RBF}}[k,i] = \sum_{j=1}^{K_p I} \omega_j \Phi(|k - \mathcal{K}_f[j]|, |i - \mathcal{K}_t[j]|). \tag{C.1}$$

$\mathcal{K}_f = [\mathcal{K}_{\mathrm{p}_1}, \ldots, \mathcal{K}_{\mathrm{p}_I}] \in \mathbb{R}^{1\times K_p I}$ and $\mathcal{K}_t = [(1)_{\times K_p}, \ldots, (I)_{\times K_p}] \in \mathbb{R}^{1\times K_p I}$ denote the frequency and time indices vectors of the sparsed allocated pilot subcarriers within the received OFDM frame. ω_j represents the RBF weight multiplied by the RBF interpolation function $\Phi(.)$ between the (k,i) data subcarrier and the $(\mathcal{K}_f[j], \mathcal{K}_t[j])$ pilot subcarrier. In [Soltani et al., 2019], the RBF gaussian function is applied, such that

$$\Phi(x,y) = e^{-\frac{(x+y)^2}{r_0}}, \tag{C.2}$$

where r_0 is the 2D RBF scale factor and it varies according to the used RBF function. We note that changing the value of r_0 changes the shape of the interpolation function. Moreover, the RBF weights $\boldsymbol{w}_{\text{RBF}} = [\omega_1, \ldots, \omega_{K_pI}] \in \mathbb{R}^{K_pI \times 1}$ are calculated using the following relation

$$\boldsymbol{A}_{\text{RBF}} \boldsymbol{w}_{\text{RBF}} = \bar{\boldsymbol{h}}_{\text{LS}}, \tag{C.3}$$

where $\boldsymbol{A}_{\text{RBF}} \in \mathbb{R}^{K_pI \times K_pI}$ is the RBF interpolation matrix of the pilots subcarriers, with entries $a_{i,j} = \Phi(\mathcal{K}_f[i], \mathcal{K}_t[j])$ where $i, j = 1, \ldots, K_pI$. We note that, $\bar{\boldsymbol{h}}_{\text{LS}}$ is a vector containing the LS estimated channels at all the pilot subcarriers within the received OFDM frame. After computing $\boldsymbol{W}_{\text{RBF}}$, the RBF estimated channel for every data subcarriers within the received OFDM frame can be calculated as shown in (C.1). Finally, the RBF interpolation estimated frame $\hat{\bar{\boldsymbol{H}}}_{\text{RBF}}$ is fed as an input to SR-CNN and DN-CNN in order to improve the channel estimation accuracy, and alleviate the noise impact.

The ChannelNet estimator limitations lie in: (*i*) 2D RBF interpolation high computational complexity that results from the computation of (C.3) for the channel estimation of each data subcarrier. (*ii*) The 2D RBF function and scale factor should be optimized according to the channel variations. (*iii*) The integrated SR-CNN and DN-CNN architectures have considerable computational complexity. We note that, the ChannelNet estimator uses a fixed RBF function and scale factor, therefore, it suffers from considerable performance degradation especially in low signal-to-noise ratio (SNR) regions where the impact of noise is dominant, and high mobility vehicular scenarios, where the channel varies rapidly within the OFDM frame.

TS-ChannelNet Estimator

TS-ChannelNet [Zhu et al., 2020] is based on applying average decision-directed with time truncation (ADD-TT) interpolation to the received OFDM frame. After that, an accurate estimation is obtained by implementing super resolution convolutional long short-term memory (SR-ConvLSTM) network in order to track vehicular channel variations by learning the time and frequency correlations of the vehicular channel. We note that, the ADD-TT interpolation is a SBS estimator, where DPA estimation is applied first as described in the previous chapters. After that, the enlarged DPA demapping error is reduced by applying time domain truncation as follows:

$$\hat{\boldsymbol{h}}_{\text{DPA}_i} = \boldsymbol{F}_K^{\text{H}} \hat{\bar{\boldsymbol{h}}}_{\text{DPA}_i}, \tag{C.4}$$

where $\boldsymbol{F}_K \in \mathbb{C}^{K \times K}$ denotes the K-DFT matrix, and $\hat{\boldsymbol{h}}_{\text{DPA}_i}$ is the time domain DPA estimated channel. Then, $\hat{\boldsymbol{h}}_{\text{DPA}_i}$ truncation is applied to the significant L channel taps, such that:

$$\hat{\boldsymbol{h}}_{\text{DPA}_{i,L}} = \hat{\boldsymbol{h}}_{\text{DPA}_i}(1 : L). \tag{C.5}$$

After that, $\hat{\boldsymbol{h}}_{\mathrm{DPA}_{i,L}}$ is converted back to the frequency domain such that:

$$\hat{\hat{\boldsymbol{h}}}_{\mathrm{TT}_i} = \boldsymbol{F}_{\mathrm{on}} \hat{\boldsymbol{h}}_{\mathrm{DPA}_{i,L}}, \tag{C.6}$$

where $\boldsymbol{F}_{\mathrm{on}} \in \mathbb{C}^{K_{\mathrm{on}} \times L}$ represents the scaled DFT matrix that is used to convert $\hat{\boldsymbol{h}}_{\mathrm{DPA}_{i,L}}$ back to frequency domain, which is obtained by selecting $\mathcal{K}_{on}$ rows, and L columns from the K-DFT matrix. Applying the average time truncation operation to $\hat{\hat{\boldsymbol{h}}}_{\mathrm{DPA}_i}[k]$ alleviates the effect of noise and enlarged demapping error. Moreover, $\hat{\hat{\boldsymbol{h}}}_{\mathrm{TT}_i}[k]$ estimated channel is further improved by applying frequency and time domain averaging consecutively as follows:

$$\hat{\hat{\boldsymbol{h}}}_{\mathrm{FTT}_i}[k] = \sum_{\lambda=-\beta}^{\lambda=\beta} \omega_\lambda \hat{\hat{\boldsymbol{h}}}_{\mathrm{TT}_i}[k+\lambda], \; \omega_\lambda = \frac{1}{2\beta+1}. \tag{C.7}$$

The final ADD-TT channel estimates is updated using time averaging between the previously ADD-TT estimated channel and the frequency averaged channel in (C.7), such that:

$$\hat{\hat{\boldsymbol{h}}}_{\text{ADD-TT}_i}[k] = (1-\alpha)\hat{\hat{\boldsymbol{h}}}_{\text{ADD-TT}_{i-1}}[k] + \alpha \hat{\hat{\boldsymbol{h}}}_{\mathrm{FTT}_i}[k]. \tag{C.8}$$

Motivated by the fact that the vehicular time-variant channel can be modeled as a time-series forecasting problem, where historical data can be used to predict future observations [Chatfield, 2000]. The authors in [Zhu et al., 2020] apply SR-ConvLSTM network on top of the ADD-TT interpolation, where convolutional layers are added to the LSTM network in order to capture more vehicular channel features, hence improving the estimation performance. Therefore, the ADD-TT estimated channel for the whole received frame is modeled as a low resolution image, and then SR-ConvLSTM network is used after the ADD-TT interpolation. Unlike ChannelNet estimator where two CNNs are employed, TS-ChannelNet estimator uses only SR-ConvLSTM network, thus the overall computational complexity is relatively decreased. However, TS-ChannelNet still suffers from high computational complexity due to integrating both LSTM and CNN in one network.

Publications

Journal articles

[J1] **A. K. Gizzini**, M. Chafii, A. Nimr and G. Fettweis, "*Low Complex Methods for Robust Channel Estimation in IEEE 802.11p Standard*", Submitted to the IEEE Access.

[J2] **A. K. Gizzini**, M. Chafii, A. Nimr, R. Shubair, and G. Fettweis, "*CNN aided Weighted Interpolation for Channel Estimation in Vehicular Communications*", IEEE Transactions on Vehicular Technology (TVT), October 2021.

[J3] **A. K. Gizzini**, M. Chafii, A. Nimr and G. Fettweis, "*Deep Learning Based Channel Estimation Schemes for IEEE 802.11p Standard*," in IEEE Access, September 2020

[J4] G. Aquino, L. Mendes, T. Barbosa, M. Chafii, and **A. K. Gizzini**, "*MUSA Grant-Free Access Framework and Blind Detection Receiver*". Journal of Communication and Information Systems, July 2021.

International conferences

[C1] S. Ehsanfar, K. Moessner,**A. K. Gizzini**, M. Chafii, "*Performance Comparison of IEEE 802.11p, 802.11bd-draft and a Unique Word-based PHY in Doubly-Dispersive Channels*". Submitted to IEEE WCNC 2022, Austin, USA.

[C2] **A. K. Gizzini**, M. Chafii, S. Ehsanfar, R. Shubair, "*Temporal Averaging LSTM-based Channel Estimation Scheme for IEEE 802.11p Standard*". in: Proceedings of the IEEE Global Communications Conference (GLOBECOM) 2021, Madrid, Spain.

[C3] **A. K. Gizzini**, M. Chafii, A. Nimr and G. Fettweis, "*Joint TRFI and Deep Learning for IEEE 802.11p Channel Estimation*". in: Workshops Proceedings of the IEEE Global Communications Conference (GLOBECOM) 2020, Taipei, Taiwan.

[C4] **A. K. Gizzini**, M. Chafii, A. Nimr and G. Fettweis, "*Adaptive Channel Estimation based on Deep Learning,*" in: Proceedings of the IEEE Vehicular Technology Conference (VTC2020-Fall).

[C5] **A. K. Gizzini**, M. Chafii, A. Nimr and G. Fettweis, "*Enhancing Least Square Channel Estimation Using Deep Learning*" in: Proceedings of the IEEE Vehicular Technology Conference (VTC2020-Spring).

Bibliography

[Abdelgader and Wu, 2014] Abdelgader, A. and Wu, L. (2014). The Physical Layer of the IEEE 802.11 p WAVE Communication Standard: The Specifications and Challenges. *Lecture Notes in Engineering and Computer Science*, 2.

[Acosta-Marum, 2007] Acosta-Marum, G. (2007). Measurement, Modeling, and OFDM Synchronization for the Wideband Mobile-to-Mobile Channel. *Ph.D. dissertation, Georgia Inst. Technol., Atlanta, GA*.

[Acosta-Marum and Ingram, 2006] Acosta-Marum, G. and Ingram, M. A. (2006). Doubly selective vehicle-to-vehicle channel measurements and modeling at 5.9 ghz. In *Doubly Selective Vehicle-to-Vehicle Channel Measurements and Modeling at 5.9 GHz*.

[Acosta-Marum and Ingram, 2007a] Acosta-Marum, G. and Ingram, M. A. (2007a). Six Time and Frequency Selective Empirical Channel Models for Vehicular Wireless LANs. *IEEE Vehicular Technology Magazine*, 2(4):4–11.

[Acosta-Marum and Ingram, 2007b] Acosta-Marum, G. and Ingram, M. A. (2007b). Six Time and Frequency Selective Empirical Channel Models for Vehicular Wireless LANs. *IEEE Vehicular Technology Magazine*, 2(4):4–11.

[Ahmed et al., 2013] Ahmed, S. A. M., Ariffin, S. H. S., and Fisal, N. (2013). Overview of wireless access in vehicular environment (wave) protocols and standards. *Indian Journal of Science and Technology*, 6(7).

[Albawi et al., 2017] Albawi, S., Mohammed, T. A., and Al-Zawi, S. (2017). Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)*, pages 1–6.

[Ashraf et al., 2017] Ashraf, M. I., Chen-Feng Liu, Bennis, M., and Saad, W. (2017). Towards low-latency and ultra-reliable vehicle-to-vehicle communication. In *2017 European Conference on Networks and Communications (EuCNC)*, pages 1–5.

[Bello, 1963] Bello, P. (1963). Characterization of randomly time-variant linear channels. *IEEE Transactions on Communications Systems*, 11(4):360–393.

[Biyyam and Bhuma, 2018] Biyyam, S. and Bhuma, A. (2018). Optimal Channel Estimation Using DFT-Based Interpolation with Comb-Type Pilots for OFDM Systems. In *Innovations in Electronics and Communication Engineering*, pages 473–484.

[Blum et al., 2004] Blum, J., Eskandarian, A., and Hoffman, L. (2004). Challenges of intervehicle ad hoc networks. *IEEE Transactions on Intelligent Transportation Systems*, 5(4):347–351.

[Bomfin et al., 2021a] Bomfin, R., Chafii, M., Nimr, A., and Fettweis, G. (2021a). Channel estimation for mimo space time coded otfs under doubly selective channels. In *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 1–6.

[Bomfin et al., 2021b] Bomfin, R., Chafii, M., Nimr, A., and Fettweis, G. (2021b). A robust baseband transceiver design for doubly-dispersive channels. *IEEE Transactions on Wireless Communications*, 20(8):4781–4796.

[Brendha and Prakash, 2017] Brendha, R. and Prakash, V. S. J. (2017). A survey on routing protocols for vehicular ad hoc networks. In *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pages 1–7.

[Carona et al., 2010] Carona, D., Serrador, A., Mar, P., Abreu, R., Ferreira, N., Meireles, T., Matos, J., and Lopes, J. (2010). A 802.11p prototype implementation. In *2010 IEEE Intelligent Vehicles Symposium*, pages 1116–1121.

[Chafii et al., 2018a] Chafii, M., Bader, F., and Palicot, J. (2018a). Sc-fdma with index modulation for m2m and iot uplink applications. In *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–5.

[Chafii et al., 2018b] Chafii, M., Palicot, J., Gribonval, R., and Bader, F. (2018b). Adaptive wavelet packet modulation. *IEEE Transactions on Communications*, 66(7):2947–2957.

[Chatfield, 2000] Chatfield, C. (2000). *Time-Series Forecasting*. Chapman and Hall/CRC.

[Chen et al., 2017] Chen, S., Hu, J., Shi, Y., Peng, Y., Fang, J., Zhao, R., and Zhao, L. (2017). Vehicle-to-everything (v2x) services supported by lte-based systems and 5g. *IEEE Communications Standards Magazine*, 1(2):70–76.

[Chen and Wyglinski, 2010] Chen, S. and Wyglinski, A. M. (2010). Chapter 3 - digital communication fundamentals for cognitive radio. In Wyglinski, A. M., Nekovee, M., and Hou, Y. T., editors, *Cognitive Radio Communications and Networks*, pages 41–83. Academic Press, Oxford.

[Cho et al., 2009] Cho, W., Kim, S. I., Choi, H. k., Oh, H. S., and Kwak, D. Y. (2009). Performance evaluation of v2v/v2i communications: The effect of midamble insertion. In *2009 1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace Electronic Systems Technology*, pages 793–797.

[Choi et al., 2017] Choi, J.-Y., Mun, C., and Yook, J.-G. (2017). Adaptive channel estimation based on a decision method using a long preamble for the ieee 802.11p. In *2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*, pages 1–5.

[Choi et al., 2015] Choi, Y., Bae, J. H., and Lee, J. (2015). Low-complexity 2d lmmse channel estimation for ofdm systems. In *2015 IEEE 82nd Vehicular Technology Conference (VTC2015-Fall)*, pages 1–5.

[Dany et al., 2003] Dany, J., Antoine, J., Husson, L., Paul, N., Wautier, A., and Brouet, J. (2003). Low complexity algorithm for optimal hard decoding of convolutional codes. In *Low complexity algorithm for optimal hard decoding of convolutional codes*.

[De et al., 2018] De, S., Mukherjee, A., and Ullah, E. (2018). Convergence guarantees for rmsprop and adam in non-convex optimization and an empirical comparison to nesterov acceleration.

[de Almeida et al., 2021] de Almeida, I. B. F., Chafii, M., Nimr, A., and Fettweis, G. (2021). Blind transmitter localization in wireless sensor networks: A deep learning approach. In *2021 IEEE 32nd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pages 1241–1247.

[Djenouri, 2008] Djenouri, D. (2008). Preventing vehicle crashes through a wireless vehicular sensor network. In *2008 24th Biennial Symposium on Communications*, pages 320–323.

[Dodge, 2008] Dodge, Y. (2008). *Central Limit Theorem*, pages 66–68. Springer New York, New York, NY.

[Dong et al., 2016] Dong, C., Loy, C. C., He, K., and Tang, X. (2016). Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2):295–307.

[Ehsanfar et al., 2019] Ehsanfar, S., Chafii, M., and Fettweis, G. (2019). A frame design for mimo uw based systems: Overhead analysis amp; channel estimation. In *2019 IEEE 2nd 5G World Forum (5GWF)*, pages 173–178.

[Ehsanfar et al., 2020] Ehsanfar, S., Chafii, M., and Fettweis, G. (2020). A study on unique-word based synchronization for mimo systems over time-varying channels. In *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–7.

[Fernandez et al., 2012] Fernandez, J. A., Borries, K., Cheng, L., Vijaya Kumar, B. V. K., Stancil, D. D., and Bai, F. (2012). Performance of the 802.11p Physical Layer in Vehicle-to-Vehicle Environments. *IEEE Transactions on Vehicular Technology*, 61(1):3–14.

[G Vega et al., 2019] G Vega, C. A., Guti A, C. A., Jaime-Rodr A-guez, J. A. J., V A zquez Castillo, J., Campos-Delgado, D. U., M., L.-R., and DAaz-Ibarra, M. A. (2019). Doppler spectrum measurements of vehicular radio channels using a narrowband sounder. *Revista Facultad de Ingenier Universidad de Antioquia*, pages 32 – 40.

[Gizzini et al., 2021a] Gizzini, A. K., Chafii, M., Ehsanfar, S., and Shubair, R. M. (2021a). Temporal averaging lstm-based channel estimation scheme for ieee 802.11p standard.

[Gizzini et al., 2020a] Gizzini, A. K., Chafii, M., Nimr, A., and Fettweis, G. (2020a). Deep learning based channel estimation schemes for ieee 802.11p standard. *IEEE Access*, 8:113751–113765.

[Gizzini et al., 2020b] Gizzini, A. K., Chafii, M., Nimr, A., and Fettweis, G. (2020b). Joint trfi and deep learning for vehicular channel estimation. In *2020 IEEE Globecom Workshops (GC Wkshps*, pages 1–6.

[Gizzini et al., 2021b] Gizzini, A. K., Chafii, M., Nimr, A., Shubair, R. M., and Fettweis, G. (2021b). Cnn aided weighted interpolation for channel estimation in vehicular communications.

[Grami, 2016] Grami, A. (2016). Chapter 10 - error-control coding. In Grami, A., editor, *Introduction to Digital Communications*, pages 409–455. Academic Press, Boston.

[Greff et al., 2017] Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., and Schmidhuber, J. (2017). Lstm: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 28(10):2222–2232.

[Hamamreh et al., 2019] Hamamreh, J. M., Furqan, H. M., and Arslan, H. (2019). Classifications and applications of physical layer security techniques for confidentiality: A comprehensive survey. *IEEE Communications Surveys Tutorials*, 21(2):1773–1828.

[Han et al., 2019] Han, S., Oh, Y., and Song, C. (2019). A Deep Learning Based Channel Estimation Scheme for IEEE 802.11p Systems. In *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, pages 1–6.

[Han et al., 2020] Han, S., Park, J., and Song, C. (2020). Virtual subcarrier aided channel estimation schemes for tracking rapid time variant channels in ieee 802.11p systems. In *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, pages 1–5.

[Hassan et al., 2018] Hassan, K., Michael, K., and Mrutu, S. I. (2018). Performance of soft viterbi decoder enhanced with non-transmittable codewords for storage media. *Cogent Engineering*, 5(1):1426538.

[He et al., 2015] He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition.

[Hosseinzadeh Aghdam and Sharifi, 2019] Hosseinzadeh Aghdam, M. and Sharifi, A. A. (2019). Papr reduction in ofdm systems: An efficient pts approach based on particle swarm optimization. *ICT Express*, 5(3):178–181.

[Hyadi et al., 2016] Hyadi, A., Rezki, Z., and Alouini, M.-S. (2016). An overview of physical layer security in wireless communication systems with csit uncertainty. *IEEE Access*, 4:6121–6132.

[ichi Amari, 1993] ichi Amari, S. (1993). Backpropagation and stochastic gradient descent method. *Neurocomputing*, 5(4):185 – 196.

[Joo-Young Choi et al., 2016] Joo-Young Choi, Kang-Hee Yoo, and Cheol Mun (2016). MMSE Channel Estimation Scheme using Virtual Pilot Signal for IEEE 802.11p. *Journal of Korean Institute of Information Technology*, 27(1):27–32.

[Karedal et al., 2009] Karedal, J., Tufvesson, F., Czink, N., Paier, A., Dumard, C., Zemen, T., Mecklenbrauker, C. F., and Molisch, A. F. (2009). A Geometry-based Stochastic MIMO Model for Vehicle-to-Vehicle Communications. *IEEE Transactions on Wireless Communications*, 8(7):3646–3657.

[Kenney, 2011] Kenney, J. B. (2011). Dedicated short-range communications (dsrc) standards in the united states. *Proceedings of the IEEE*, 99(7):1162–1182.

[Kim et al., 2008] Kim, S. I., Oh, H. S., and Choi, H. K. (2008). Mid-amble aided ofdm performance analysis in high mobility vehicular channel. In *2008 IEEE Intelligent Vehicles Symposium*, pages 751–754.

[Lostanlen, 2017] Lostanlen, V. (2017). Convolutional operators in the time-frequency domain. In *Convolutional operators in the time-frequency domain.*

[Ma et al., 2018] Ma, X., Ye, H., and Li, Y. (2018). Learning assisted estimation for time-varying channels. In *2018 15th International Symposium on Wireless Communication Systems (ISWCS)*, pages 1–5.

[Marsden, 1974] Marsden, M. (1974). Cubic Spline Interpolation of Continuous Functions. *Journal of Approximation Theory*, 10(2):103 – 111.

[Matz and Hlawatsch, 2011] Matz, G. and Hlawatsch, F. (2011). Chapter 1 - fundamentals of time-varying communication channels. In *Wireless Communications Over Rapidly Time-Varying Channels*, pages 1–63. Elsevier Ltd.

[Mecklenbrauker et al., 2011] Mecklenbrauker, C. F., Molisch, A. F., Karedal, J., Tufvesson, F., Paier, A., Bernado, L., Zemen, T., Klemp, O., and Czink, N. (2011). Vehicular Channel Characterization and Its Implications for Wireless System Design and Performance. *Proceedings of the IEEE*, 99(7):1189–1212.

[National Center for Statistics and Analysis, 2004] National Center for Statistics and Analysis (2004). Traffic safety facts 2003. Technical report, National Center for Statistics and Analysis, U.S. DoT, Washington, DC, USA.

[Nekovee, 2005] Nekovee, M. (2005). Sensor networks on the road: the promises and challenges of vehicular ad hoc networks and grids. In *J. Netw. Comput. Appl.*

[Njima et al., 2021a] Njima, W., Chafii, M., Chorti, A., Shubair, R. M., and Poor, H. V. (2021a). Indoor localization using data augmentation via selective generative adversarial networks. *IEEE Access*, 9:98337–98347.

[Njima et al., 2020] Njima, W., Chafii, M., Nimr, A., and Fettweis, G. (2020). Deep learning based data recovery for localization. *IEEE Access*, 8:175741–175752.

[Njima et al., 2021b] Njima, W., Chafii, M., Nimr, A., and Fettweis, G. (2021b). Convolutional neural networks based denoising for indoor localization. In *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*, pages 1–6.

[Njima et al., 2021c] Njima, W., Chafii, M., and Shubair, R. M. (2021c). Gan based data augmentation for indoor localization using labeled and unlabeled data. In *2021 International Balkan Conference on Communications and Networking (BalkanCom)*, pages 36–39.

[Orosi, 2016] Orosi, G. (2016). The arithmetico-geometric sequence: an application of linear algebra. *International Journal of Mathematical Education in Science and Technology*, 47(5):766–772.

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

[Pan et al., 2021] Pan, J., Shan, H., Li, R., Wu, Y., Wua, W., and Quek, T. Q. S. (2021). Channel estimation based on deep learning in vehicle-to-everything environments. *IEEE Communications Letters*, pages 1–1.

[Pontes et al., 2016] Pontes, F., Amorim, G., Balestrassi, P., Paiva, A., and Ferreira, J. (2016). Design of experiments and focused grid search for neural network parameter optimization. *Neurocomputing*, 186:22 – 34.

[Pontes et al., 2016] Pontes, F., Amorim, G., Balestrassi, P., Paiva, A., and Ferreira, J. (2016). Design of experiments and focused grid search for neural network parameter optimization. *Neurocomputing*, 186:22 – 34.

[Qi, 2016] Qi, H. (2016). Derivation of backpropagation in convolutional neural network (cnn). In *Derivation of Backpropagation in Convolutional Neural Network (CNN)*.

[Rice, 1948] Rice, S. O. (1948). Statistical properties of a sine wave plus random noise. *The Bell System Technical Journal*, 27(1):109–157.

[Ruder, 2017] Ruder, S. (2017). An overview of multi-task learning in deep neural networks.

[Schmidhuber, 2015a] Schmidhuber, J. (2015a). Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117.

[Schmidhuber, 2015b] Schmidhuber, J. (2015b). Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117.

[Sempere-García et al., 2021] Sempere-García, D., Sepulcre, M., and Gozalvez, J. (2021). Lte-v2x mode 3 scheduling based on adaptive spatial reuse of radio resources. *Ad Hoc Networks*, 113:102351.

[Shi et al., 2015] Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., kin Wong, W., and chun Woo, W. (2015). Convolutional lstm network: A machine learning approach for precipitation nowcasting.

[Shi et al., 2004] Shi, Y., Zhang, X. M., Ni, Z.-C., and Ansari, N. (2004). Interleaving for combating bursts of errors. *IEEE Circuits and Systems Magazine*, 4(1):29–42.

[Sjoberg et al., 2017] Sjoberg, K., Andres, P., Buburuzan, T., and Brakemeier, A. (2017). Cooperative intelligent transport systems in europe: Current deployment status and outlook. *IEEE Vehicular Technology Magazine*, 12:89–97.

[Sobolewski, 2003] Sobolewski, J. S. (2003). Data transmission media. In Meyers, R. A., editor, *Encyclopedia of Physical Science and Technology (Third Edition)*, pages 277–303. Academic Press, New York, third edition edition.

[Soltani et al., 2019] Soltani, M., Pourahmadi, V., Mirzaei, A., and Sheikhzadeh, H. (2019). Deep learning-based channel estimation. *IEEE Communications Letters*, 23(4):652–655.

[Sun et al., 2017] Sun, M., Song, Z., Jiang, X., Pan, J., and Pang, Y. (2017). Learning pooling for convolutional neural network. *Neurocomputing*, 224:96–104.

[Toit, 2008] Toit, W. (2008). Radial basis function interpolation. *Master Thesis dissertation, Department of Mathematical Sciences University of Stellenbosch, Matieland, South Africa.*

[US Department of Transportation, 2006] US Department of Transportation (April, 2006). Vehicle safety communications project. Technical report, US Department of Transportation.

[Viriyasitavat et al., 2015] Viriyasitavat, W., Boban, M., Tsai, H., and Vasilakos, A. (2015). Vehicular Communications: Survey and Challenges of Channel and Propagation Models. *IEEE Vehicular Technology Magazine*, 10(2):55–66.

[Wang et al., 2009] Wang, C., Cheng, X., and Laurenson, D. I. (2009). Vehicle-to-Vehicle Channel Modeling and Measurements: Recent Advances and Future Challenges. *IEEE Communications Magazine*, 47(11):96–103.

[Wang, 2018] Wang, Hussain, C. G. (2018). An Improved Channel Estimation Technique for IEEE 802.11p Standard in Vehicular Communications. *Sensors (Basel, Switzerland)*, 19(1).

[Wang et al., 2017] Wang, T., Wen, C.-K., Wang, H., Gao, F., Jiang, T., and Jin, S. (2017). Deep learning for wireless physical layer: Opportunities and challenges.

[Weidong et al., 2009] Weidong, X., Javier, G., Zhisheng, N., Onur, A., and Eylem, E. (2009). Wireless Access in Vehicular Environments. *EURASIP Journal on Wireless Communications and Networking*, 1:1687–1499.

[Wu et al., 2018] Wu, Z., Jiang, L., Jiang, Z., Chen, B., Liu, K., Xuan, Q., and Xiang, Y. (2018). Accurate indoor localization based on csi and visibility graph. *Sensors*, 18(8).

[Yamada, 1990] Yamada, T. (1990). 2 - principles of error detection and correction. In Imai, H., editor, *Essentials of Error-Control Coding Techniques*, pages 11–37. Academic Press.

[Yang et al., 2019] Yang, Y., Gao, F., Ma, X., and Zhang, S. (2019). Deep learning-based channel estimation for doubly selective fading channels. *IEEE Access*, 7:36579–36589.

[Ye et al., 2018] Ye, H., Li, G. Y., and Juang, B. (2018). Power of deep learning for channel estimation and signal detection in ofdm systems. *IEEE Wireless Communications Letters*, 7(1):114–117.

[Yoon-Kyeong Kim et al., 2014] Yoon-Kyeong Kim, Jang-Mi Oh, Yoo-Ho Shin, and Cheol Mun (2014). Time and Frequency Domain Channel Estimation Scheme for IEEE 802.11p. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 1085–1090.

[Zhang et al., 2017] Zhang, K., Zuo, W., Chen, Y., Meng, D., and Zhang, L. (2017). Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155.

[Zhao et al., 2013] Zhao, Z., Cheng, X., Wen, M., Jiao, B., and Wang, C. (2013). Channel Estimation Schemes for IEEE 802.11p Standard. *IEEE Intelligent Transportation Systems Magazine*, 5(4):38–49.

[Zheng and Xiao, 2009] Zheng, Y. R. and Xiao, C. (2009). Channel Estimation for Frequency-Domain Equalization of Single-Carrier Broadband Wireless Communications. *IEEE Transactions on Vehicular Technology*, 58(2):815–823.

[Zhu et al., 2020] Zhu, X., Sheng, Z., Fang, Y., and Guo, D. (2020). A deep learning-aided temporal spectral channelnet for ieee 802.11p-based channel estimation in vehicular communications. *EURASIP Journal on Wireless Communications and Networking*, 1(94).