Department of Information Science and Technology

Channel Estimation with TCH Codes for Machine-Type Communications

Bruno Miguel de Carvalho Lopes

A Dissertation presented in partial fulfillment of the Requirements for the Degree of

Master of Science in Telecommunications and Computer Engineering

Supervisor

Prof. Doutor Francisco António Bucho Cercas

ISCTE-IUL

Co-Supervisor

Prof. Doutor Rui Miguel Henriques Dias Morgado Dinis

FCT-UNL

October 2017

Acknowledgements

Firstly, I would like to sincerely thank my supervisor, Professor Francisco Cercas for presenting this challenge to me and, above all, for all the support, orientation and readiness demonstrated throughout this last year.

Secondly, I would like to thank my co-supervisor, Professor Rui Dinis, for all his help in planning and executing the different tasks associated with this dissertation. I would also like to acknowledge Professor Nuno Souto, who always showed availability and patience to help in any question about this work.

I would also like to thank all my friends and colleagues - who, fortunately, are too many to mention here one by one - for all the friendship, motivation, support and camaraderie shown throughout all these years.

A big thank you to all my family: cousins, uncles, grandfather, aunt, even my cat (since he is also part of the family!) and last, but certainly not least, my mother, for all the unconditional love, support, education and everything she has done for me throughout my whole life. I wouldn't be where I am today if it wasn't for her.

Finally, I would like to pay a tribute to my grandmother, who unfortunately is no longer with us. May this dissertation serve as a dedication to her memory.

Resumo

Os códigos TCH possuem várias propriedades que nos permitem usá-los eficientemente em diversas aplicações. Uma delas é a estimação de canal e nesta dissertação é estudado o desempenho dos códigos TCH em estimação de canal num sistema OFDM, tendo em conta as comunicações Machine-Type. Resultados que ilustram a taxa de erro de bit foram obtidos através de simulações que permitem avaliar o impacto de usar diferentes técnicas de pilotos, nomeadamente multiplexados e implícitos, diferentes valores de potência para os pilotos e diferentes modulações, QPSK e 64-QAM. Também é feita a comparação entre os pilotos TCH e pilotos convencionais. Os resultados mostram que os pilotos TCH tem um desempenho muito positivo e confiável, dentro dos parâmetros testados.

Também é efetuado o estudo de sincronização e estimação de canal conjunta usando métodos esparsos como o OMP, o ℓ_1-regularized e o Iterative Reweighted ℓ_1. Os códigos TCH são comparados com outros tipos de sequências, tais como as sequências Zadoff-Chu e os códigos pseudo-aleatórios. São consideradas variações no tamanho dos pilotos, no comprimento do canal e no tamanho da janela de observação para perceber quais são os seus efeitos no desempenho. Os resultados demonstram que os códigos TCH podem ser utilizados com sucesso em estimação de canal e sincronização conjunta e conseguem aguentar condições adversas de simulação melhor que os outros pilotos utilizados. Também é provado que compressed sensing pode ser utilizado com sucesso em sincronização e estimação conjunta, que é uma área onde o seu uso ainda não foi explorado aprofundadamente.

Palavras-chave: códigos TCH, estimação de canal, pilotos multiplexados, pilotos implícitos, compressed sensing, recuperação de sinais esparsos.

Abstract

TCH codes possess several properties that allow us to use them efficiently in various applications. One of these applications is channel estimation and, in this dissertation, it is studied the performance of TCH codes to estimate the channel in an Orthogonal Frequency Division Multiplexing system, regarding Machine-Type Communications. Bit error rate performance results were obtained by executing simulations that allowed the evaluation of the impact of using two different pilot techniques, such as data multiplexed and implicit pilots, different pilot power levels and different modulations, QPSK and 64-QAM. Pilots based on TCH codes are also compared with other conventional pilots. Results show that TCH codes have a very positive and reliable performance.

Joint timing synchronization and channel estimation is also performed using different sparse based approaches, such as Orthogonal Matching Pursuit, ℓ_1-regularized and Iterative Reweighted ℓ_1. TCH codes are compared against different sequence types, namely Zadoff-Chu sequences and pseudorandom codewords, and variations in the pilot size, the channel length and the observation window size are executed in order to understand their effects. Results ultimately illustrate that TCH codes can be effectively used in joint channel estimation and synchronization, managing to withstand worst simulation conditions better than its counterparts. It is also proven that compressed sensing can successfully be utilized in joint synchronization and channel estimation, an area where its use has not been very explored.

Keywords: TCH codes, channel estimation, data multiplexed pilots, implicit pilots, compressed sensing, sparse signal recovery.

Contents

List of Figures

List of Tables

List of Acronyms

BER	**B**it **E**rror **R**ate
BPDN	**B**asis **P**ursuit **De**noising
CD	**C**hromatic **D**ispersion
CIR	**C**hannel **I**mpulse **R**esponse
CO-OFDM	**C**oherent **O**ptical **O**rthogonal **F**requency **D**ivision **M**ultiplexing
CS	**C**ompressed **S**ensing
DFT	**D**iscrete **F**ourier **T**ransform
FDMA	**F**requency **D**ivision **M**ultiple **A**ccess
FFT	**F**ast **F**ourier **T**ransform
GAIC	**G**eneralized **A**kaike **I**nformation **C**riterion
ICI	**I**nter**c**arrier **I**nterference
IDFT	**I**nverse **D**iscrete **F**ourier **T**ransform
IEEE	**I**nstitute of **E**lectrical and **E**lectronics **E**ngineers
IFFT	**I**nverse **F**ast **F**ourier **T**ransform
ISI	**I**nter**s**ymbol **I**nterference
IoT	**I**nternet **o**f **T**hings
LLR	**L**og-**L**ikelihood **R**atio
LMMSE	**L**inear **M**inimum **M**ean-**S**quare-**E**rror
LS	**L**east-**S**quare
LTE	**L**ong-**T**erm **E**volution
M2M	**M**achine **to** **M**achine
MF	**M**atched **F**ilter
MIMO	**M**ultiple-**I**nput and **M**ultiple-**O**utput

ML	**M**aximum **L**ikelihood
MMSE	**M**inimum **M**ean-**S**quare-**E**rror
MP	**M**atching **P**ursuit
MRI	**M**agnetic **R**esonance **I**maging
MSE	**M**ean **S**quared **E**rror
MTC	**M**achine-**T**ype **C**ommunications
OFDM	**O**rthogonal **F**requency **D**ivision **M**ultiplexing
OMP	**O**rthogonal **M**atching **P**ursuit
OSNR	**O**ptical **S**ignal-to-**N**oise **R**atio
PAN	**P**ersonal **A**rea **N**etworks
PAPR	**P**eak-to-**A**verage-**P**ower **R**atio
PDF	**P**robability **D**ensity **F**unction
PDP	**P**ower **D**elay **P**rofile
PSAM	**P**ilot **S**ymbol **A**ided **M**odulation
QAM	**Q**uadrature **A**mplitude **M**odulation
QPSK	**Q**uadrature **P**hase **S**hift **K**eying
QoS	**Q**uality **o**f **S**ervice
STO	**S**ymbol **T**iming **O**ffset
UMTS	**U**niversal **M**obile **T**elecommunications **S**ystem
WLAN	**W**ireless **L**ocal **A**rea **N**etwork

List of Symbols

$(.)^H$	Conjugate of a matrix/vector
$(.)^T$	Transpose of a matrix/vector
$\|.\|$	Absolute value operation
$\|\|.\|\|_p$	ℓ_p-norm of a vector
$\|\|.\|\|_0$	Cardinality of a vector
$\mathbf{0}$	Matrix of zeros
Δ_{CP}	Cyclic prefix duration
ΔN_F	Frequency domain spacing
ΔN_T	Time domain spacing
ϵ'	Tunable positive number
ε	Stability provider
ε_{loss}	Loss of signal-to-noise ratio
λ	Penalizing parameter
Λ	Diagonal matrix
Λ_k	Set containing positions of non-zero elements
Ω	Support set
$\mathbf{\Phi}$	Measurement matrix
$\Phi_k(t)$	baseband subcarrier
$\mathbf{a}$	Signal
a_i	Finite field
$\mathbf{C}$	Pilot symbol
d_{min}^{TCH}	Minimum distance between TCH codes
$\mathbf{F}$	DFT matrix

F_i	Fermat number
f_k	Frequency of k^{th} subcarrier
$f(t)$	Bandlimited signal
$\mathbf{h}$	Finite response filter
$\mathbf{H}$	Channel frequency response
H_d	Hamming distance between two polynomials
$\mathbf{h}^{ext}$	Channel impulse response augmented with zeros
$\mathbf{H}_{k,l}$	Overall channel frequency response between the transmitter and the receiver
$h^j_{t_{OFDM}}$	Interpolation coefficient
$\widehat{\mathbf{h}}$	Finite response filter estimate
$\widehat{\mathbf{H}}$	Channel estimate
$\widehat{\mathbf{h}}^{ext}$	Channel impulse response augmented with zeros estimate
$\widehat{\mathbf{h}}^{ext}_{\Omega}$	Channel estimate at the support position
$\widehat{\mathbf{H}}_{k,l}$	Channel response estimate that does not carry a pilot symbol
$\widetilde{\mathbf{h}}$	Diagonal of $\widehat{\mathbf{H}}$
$\widetilde{\mathbf{H}}$	Channel frequency response estimate
$\widetilde{\mathbf{H}}_{k,l}$	Channel estimate between the transmitter and the receiver
$\mathbf{I}$	Identity matrix
k	Subcarrier
K_i	Exponent
K_s	Sparsity level of a signal
k_{TCH}	Number of data bits of TCH codes
l	Time-domain position
L	Maximum expected channel length
M_o	Observation window
$\mathbf{n}$	Noise
N	Number of OFDM subcarriers
$\mathbf{N}$	Noise samples in the frequency domain
N_c	Codewords length
$\mathbf{N}_{k,l}$	Channel noise sample
N_s	Search window

n_t	Noise with delay
n_{TCH}	Length of TCH codes
NT	OFDM symbol length
$n(t)$	Impulsive noise term
p	Prime number
p_k	Approximation of $\mathbf{s}_v$
$P(x)$	Basic TCH Base Polynomials
$P_i(x)$	TCH Base Polynomials
q	Current iteration
Q	Total iterations
$\mathbf{r}$	Residual
$\mathbf{R}$	Received sequence
$\mathbf{R}_{k,l}$	Received sequence in the antenna
$\widetilde{\mathbf{R}}$	Sequence without data symbol estimates
$\mathbf{s}$	Training sequence
$\mathbf{S}$	Data symbol
$\mathbf{S}_{k,l}$	Transmitted symbol in the k^{th} subcarrier of the l^{th} OFDM block
$\mathbf{S}_{k,l}^{Pilot}$	Pilot transmitted in the k^{th} subcarrier of the l^{th} OFDM block
$\mathbf{s}_v$	Measurement vector
$s(T)$	OFDM signal
$s(nT)$	OFDM phase and quadrature sampling
$\widehat{\mathbf{S}}$	Symbol estimate
$\widehat{\mathbf{S}}_{k,l}$	Data estimate
$\mathbf{T}$	Delay
T_0	Number of zeros before channel effective response
t_{OFDM}	OFDM block index
t_{TCH}	Correcting capacity of TCH codes
$\widehat{T}_0$	Number of zeros before channel effective response estimate
W	Length of FIR filter
$\mathbf{W}$	Weighting matrix
$\mathbf{W}^q$	Diagonal matrix for an iteration q

w_i^q	Weight for the i^{th} element
x_k	k^{th} data symbol
$\mathbf{X}$	Transmitted sequence
$\mathbf{X}_{k,l}$	Transmitted sequence in the k^{th} subcarrier of the l^{th} OFDM block
$\mathbf{y}$	Received signal
y_t	Received signal with delay
$y(t)$	Observation over time
$\mathbf{Z}$	Toeplitz matrix

Chapter 1

Introduction

This chapter aims to explain the context and the motivation behind the work developed for this dissertation, as well as to present the objectives proposed to its development. An overview of the remainder of the document is provided.

1.1 Motivation and Context

The necessity of developing a code scheme that could have a solid encoding gain, a simpler implementation compared to existing schemes and allowed decoding with a better performance permitted the creation of TCH codes. These codes are error correcting codes whose name originated from the people who created them: M. Tomlinson, F. Cercas and C. Hughes [1]. The codes have the following characteristics:

- Ideal rigid sizes for properties;
- Solid error correcting performance (oriented to transmitting short and sensitive information);
- Simple decoding based on the Fast Fourier Transform (FFT);
- Excellent correlation performance.

These features allow us to efficiently use TCH codes in various applications based on digital transmission systems, such as error correction, carrier synchronism [2], coding [3] [4], channel and phase estimation or spread spectrum systems. The excellent correlation properties of the TCH codes is the most important feature for this thesis since the main goal is to perform and study channel estimation.

Machine-Type Communications (MTC) is a type of communication that allows intelligent machines to produce, process and exchange information directly with each other, with little to no human intervention [5]. The rapid expansion of the Internet of Things (IoT) and with the increase of devices with embedded systems in the market justifies the focus of this study in this type of communication. In the very near future, communications between machines will be dominant since it is expected that, by 2019, over 600 million cellular MTC connections will be implemented worldwide [6]. This means that a wide variety of services will be provided like updating billboards in the Marketing area, synchronizing and raising the efficiency in Industrial production lines or, in the Transportation area, an autonomous vehicle may be able to provide information regarding its current status and the different systems in an automobile might communicate between them in order to offer the best experience to its passengers. MTC will be extremely useful to perform well defined and repetitive tasks not suitable to human beings, given our limited attention span.

The main motivation is to understand if TCH codes are suited to perform channel estimation and also if they are appropriate to use in MTC, which will most likely dominate the future of communications, making it a pertinent type of communication to study. Channel estimation will be crucial in order to obtain a highly viable connection. If we understand the behavior of the channel then we are able to send the signal with less distortion, boosting its capacity and viability.

1.2 Objectives and Investigation Questions

This dissertation is focused on channel estimation using TCH codes in order to understand the performance of these codes when used with this objective. Channel estimation is studied regarding Machine-Type Communications.

The channels are modulated with Orthogonal Frequency-Division Multiplexing (OFDM) and the study present in this dissertation will mainly allow to verify the performance and viability of TCH codes in a type of communication that is in rapid ascension, as mentioned in the "Motivation and Context" section. The performance of channel estimation with TCH codes is obtained via simulations using MATLAB and they are compared with performances of other codes in order to understand how TCH codes keep up with codes already used for this purpose. Different techniques regarding the types of pilots used are also considered with the objective of observing its effect in the performance of TCH codes. Another objective proposed was to write and publish scientific articles related to this thesis, focused mainly on performing accurate and efficient channel estimation using TCH codes and comparing them with different codes.

The dissertation will try to obtain answers to certain questions, for example:

- What is the efficiency of TCH codes when used to estimate the channel and the phase in Machine-Type Communications?
- Will TCH codes be able to perform better than previous codes already established?
- Will it be possible to maintain a simpler implementation when compared with traditional error correcting codes?

1.3 Dissertation layout

This dissertation is organized as follows. Chapter 1 provides a brief introduction, including the motivation of this dissertation and objectives proposed to obtain

with its work. Chapter 2 contains the theoretical research of the areas of study pertinent to this dissertation, including related work and theoretical explanations. Chapter 3 explains the work that was done to perform an efficient channel estimation using TCH codes and displays its results. Chapter 4 presents and analyses the results obtained from joint channel estimation and synchronization. Finally, Chapter 5 concludes the dissertation where the results from previous chapters are summarized, pertinent conclusions are displayed and comments regarding possible future work in this field are also presented.

There is a desire to contribute, regarding the lines of work referenced in this section, in order to continue the work that has been developed and open new doors to future endeavors for TCH codes.

Chapter 2

Fundamental Concepts

This chapter is a literature review of the concepts needed to perform this dissertation. It includes related work and a theoritical explanation of the areas of study.

2.1 TCH Codes

The TCH codes were first demonstrated in [1] through random search with the use of computers. This search allowed to prove the existence of non-linear and non-systematic codes with length n_{TCH}, where:

$$n = 2^m, \;\; m \in \mathbb{N} \tag{2.1}$$

These codes are defined by a conglomerate of Base Polynomials $P_i(x)$, have a number of data bits named k_{TCH} and a correcting capacity t_{TCH}, which can all be described in the following equations:

$$TCH(n_{TCH}, k_{TCH}) = \sum_{i=1}^{h} P_i(x) \tag{2.2}$$

$$k_{TCH} = m + \log_2 h + 1 \tag{2.3}$$

$$P_i(x) \neq P_j(x^r) \mod n_{TCH} \quad i \neq j \ \forall \ t_{TCH} \in \mathbf{N} \tag{2.4}$$

The error correcting capacity of the TCH codes, t_{TCH}, depends on their minimum distance, d_{min}, between the polynomials:

$$d_{min}^{TCH} \geq 2t_{TCH} + 1 \tag{2.5}$$

$$d_{min}^{TCH} \leq H_d[P_i(x), \ \{P_j(x^r)\} \mod n_{TCH}] \leq n_{TCH} - d_{min}^{TCH} \tag{2.6}$$

with H_d standing for Hamming distance between any two polynomials.

In order to generate codes useful to practical applications it was necessary to find an analytical method, well cemented in the coding theory, instead of using the random search previously used. So, TCH codes are built by using the Basic TCH Base Polynomials (also known as B-TCH Polynomials) $P(x)$ of degree n_{TCH}. Equation (2.7) contains the values p, which is a prime number, and a_i, which is defined by the finite field theory:

$$P(x) = \sum_{i=0}^{\left(\frac{p-1}{2}\right)-1} a_i x^{K_i}, \ \ a_i \in GF(q), q = p^k, k \in \mathbb{N} \tag{2.7}$$

where the exponents K_i satisfy Equation (2.8):

$$a^{K_i} = 1 + a^{2i+1}, \ \ i = 0, 1, ..., \frac{p-1}{2} - 1 \tag{2.8}$$

and p is a prime number with:

$$p = n_{TCH} + 1 = 2^m + 1 \tag{2.9}$$

Fermat numbers are prime numbers that obey a certain condition, expressed in the following equation:

$$F_i = 2^{2^i} + 1, \ \ i \in \mathbb{N} \tag{2.10}$$

As of now, there are only five numbers that obey this rule and they are presented in Table 2.1:

i	p	n_{TCH}
0	3	2
1	5	4
2	17	16
3	257	256
4	65537	65536

TABLE 2.1: Fermat numbers for generating TCH codes [7]

This means that it is only possible to generate five pure TCH Polynomials, which correspond to the B-TCH Polynomials mentioned above. There is the possibility of building similar TCH codes that can be extended to other lengths but that comes with the cost of losing the properties and the ideal structure that the B-TCH Polynomials possess.

TCH codes originated by B-TCH polynomials have both good cross and auto-correlation, with the latter assuming always one of these three values: n_{TCH}, the value of the code polynomial, 0 and -4. This translates into a great advantage for higher sized TCH codes, such as TCH codes length $n_{TCH} \geq 256$. For higher n_{TCH} values, the sequences tend to get closer to a Dirac impulse, as depicted in Figure 2.1, showing an auto-correlation function of a B-TCH Polynomial of length 256, with offsets from -128 to 127, so that the peak is displayed in the center.

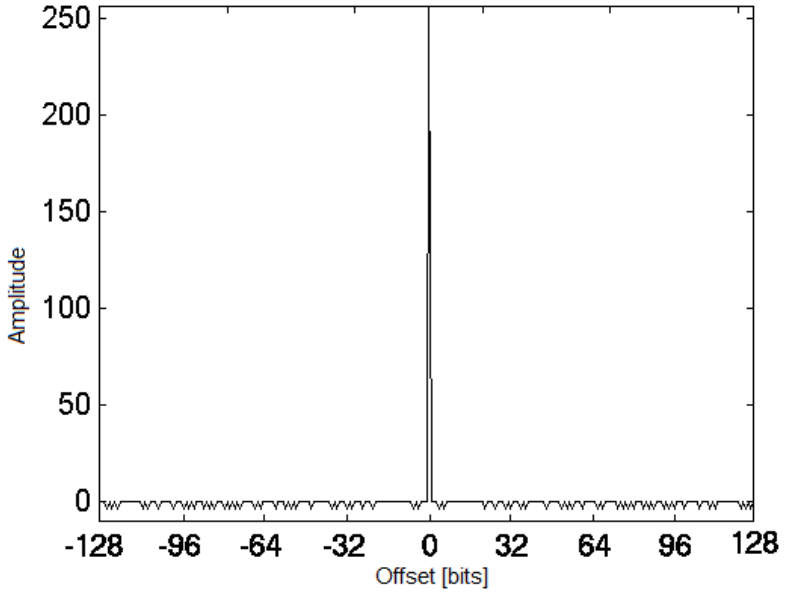

FIGURE 2.1: Auto-correlation of a B-TCH Polynomial with $n = 256$.

In [7], the TCH codes have been used to build a decoder in a simple receiver implementation. It is based in a group of correlators and each of them can be used to compare the received word with $2n_{TCH}$ words from the code. In a $TCH(n,k)$ code, the total number of code words is $2^k = 2n_{TCH}h$ which means that it can be used h correlators to verify the maximum likelihood. By comparing the outputs of the correlators, it is possible to pick the sequence which corresponds to the higher correlation.

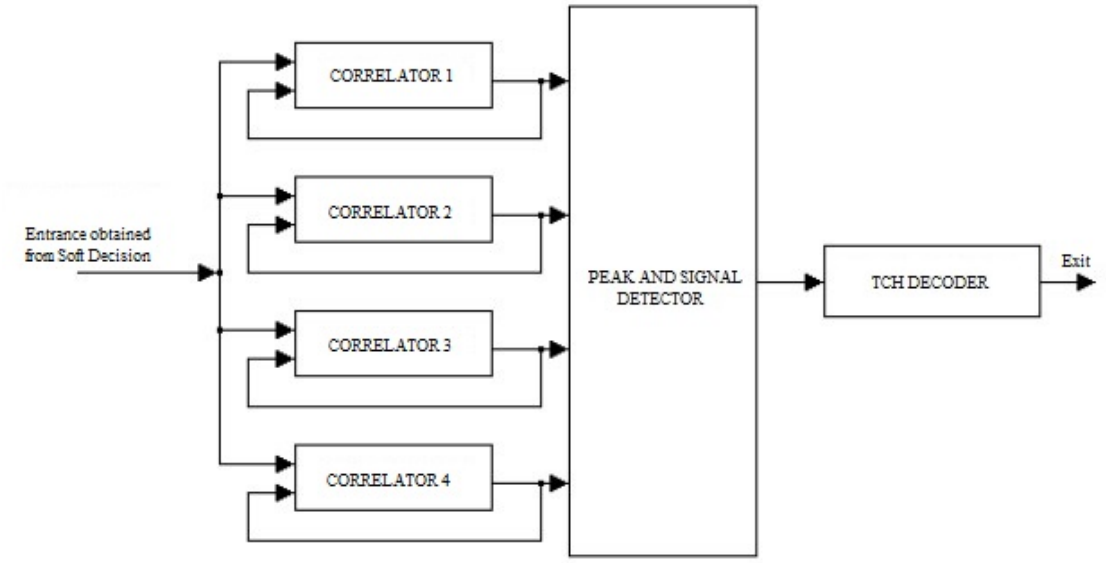

FIGURE 2.2: TCH Receiver's structure [7].

The correlators from Figure 1 can be replaced by two FFT's, a complex multiplication and an IFFT. This way, the real part and the imaginary part are processed, which allows to compare the received word with two TCH sequences simultaneously in both parts. The receiver efficiency doubles and the processing times decrease to half with this implementation.

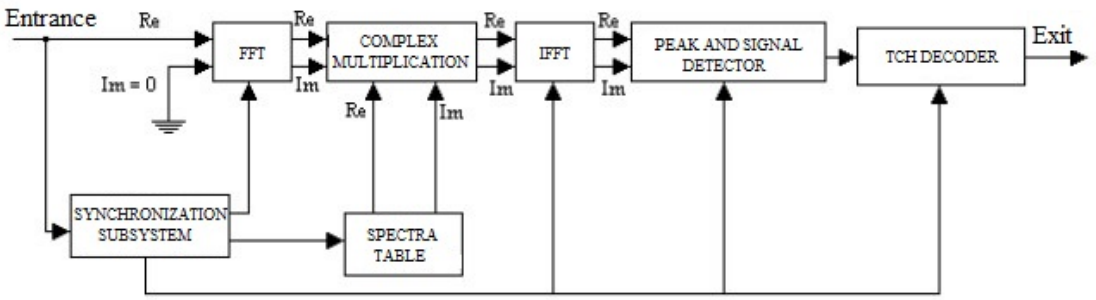

FIGURE 2.3: Simple TCH decoder of maximum likelihood [7].

Studies regarding synchronization and channel and noise estimation have been made in [8] and [9] with the latter using these codes in MIMO systems.

2.2 Channel Estimation

Channel estimation allows the studying of the effects of fading, scattering or attenuation in the channel. The definition of channel comprises everything between the transmitter and the receiver, including where the signal is being transmitted, like open space or cables. It is crucial in order to achieve high data rates and reliability. This means that the process of channel estimation characterizes the physical channel via mathematical models of what is really happening.

The two main estimation methods used are the Least-Square (LS) estimation [10], utilized when the channel and noise distributions are unknown, and the Minimum Mean-Square-Error (MMSE) estimation which improves the LS estimation [11], used when the previous parameters are known. The advantage of LS estimation lies in its simplicity because even without any knowledge of the parameters of the channel, the LS estimators are calculated with very low complexity. Nonetheless, it suffers from a high Mean Squared Error (MSE) which means that LS estimation is mainly used to perform an initial channel estimation which is then complemented by utilizing other methods. MMSE has a better performance when compared to the LS method but the matrix inversions required in this method

make it a highly complex method, meaning it is computationally very heavy. Linear Minimum Mean-Square-Error (LMMSE) [12] [13] can be a solution since this method does not use matrix inversions and it has a lower MSE value than MMSE. In [14], optimal choices of training sequences are investigated regarding multiple transmission antennas and single receive antennas based on LS and MMSE estimators. Pilot Symbol Aided Modulation (PSAM) and MMSE training schemes with orthogonal training are considered in [15].

The most popular way of performing channel estimation consists in using pilots or training sequences. These pilots are carriers with no kind of information, they just carry something (a symbol, for example) that allows us to recognize them and serve as a reference point.

2.2.1 Pilot Symbol Aided Modulation

The pilot symbols used in channel estimation are regularly multiplexed with data in both time and frequency domains [16] [17] [18] and this technique is called Pilot Symbol Aided Modulation (PSAM). In this method, the transmitter inserts known symbols periodically, which means there is no change in pulse shape or peak-to-average-power ratio (PAPR). The receiver is able to estimate the amplitude and phase rotations by interpolating the channel measurements provided by the pilot symbols, aiding the compensation of phase shifting and fading. Figure 2.4 shows the frame structure for an OFDM transmission using PSAM where it is possible to observe an OFDM system with N carriers. The pilot symbols are multiplexed with the data and transmitted with a frequency-domain spacing of ΔN_F and a time-domain spacing of ΔN_T.

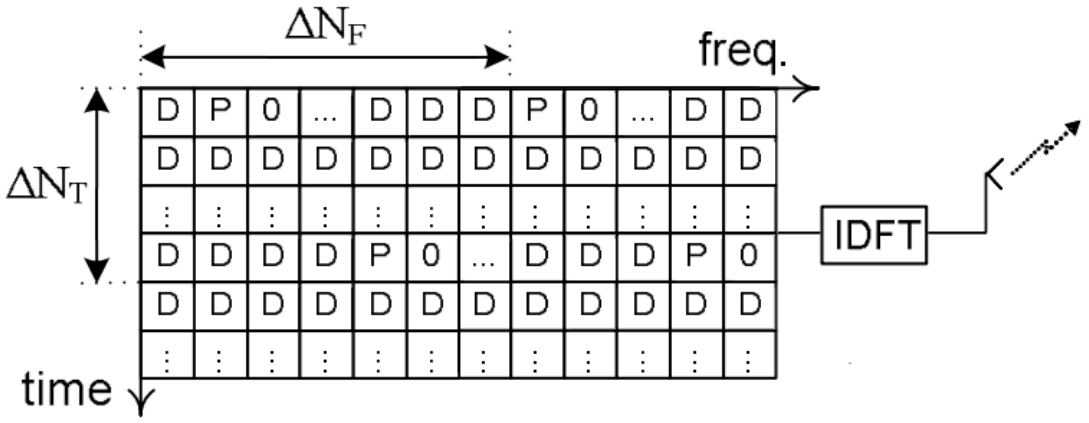

FIGURE 2.4: Frame Structure for an OFDM transmission using PSAM where **D** represents a data symbol and **P** represents a pilot symbol [19].

The following steps are applied by the receiver in order to obtain the channel response estimates [19]:

1. For each pilot symbol, the channel estimate between the transmitter and the receiver is calculated as follows:

$$\widetilde{\mathbf{H}}_{k,l} = \frac{\left(\mathbf{S}_{k,l}^{Pilot}\right)^*}{\left|\mathbf{S}_{k,l}^{Pilot}\right|^2}\mathbf{R}_{k,l} \tag{2.11}$$

   where $\mathbf{S}_{k,l}^{Pilot}$ represents a pilot symbol transmitted in the k^{th} subcarrier of the l^{th} OFDM block and $\mathbf{R}_{k,l}$ is the received sequence in the antenna, which can be expressed as:

$$\mathbf{R}_{k,l} = \mathbf{S}_{k,l}\mathbf{H}_{k,l} + \mathbf{N}_{k,l} \tag{2.12}$$

   with $\mathbf{S}_{k,l}$ being the transmitted symbol in the k^{th} subcarrier of the l^{th} OFDM block, $\mathbf{H}_{k,l}$ representing the overall channel frequency response between the transmitter and the receiver for the k^{th} frequency of the l^{th} time block and with the channel noise sample being depicted by $\mathbf{N}_{k,l}$.

2. We can get channel estimates for the same subcarrier k that do not carry a pilot symbol and are in time-domain positions, l, by applying a finite impulse

response (FIR) filter with length W:

$$\widehat{\mathbf{H}}_{k,l+t_{OFDM}} = \sum_{\lfloor j=-\frac{W-1}{2} \rfloor}^{\lfloor \frac{W}{2} \rfloor} h^{j}_{t_{OFDM}} \widetilde{\mathbf{H}}_{k,l+j\cdot\Delta N_T} \tag{2.13}$$

with t_{OFDM} symbolizing the OFDM block index relative to the last one carrying a pilot and $h^{j}_{t_{OFDM}}$ representing the interpolation coefficients of the estimation filter.

3. The data estimates originated after the first estimation can be used as pilots for channel refinement and the following equation shows how to compute the respective channel estimates:

$$\left(\widehat{\mathbf{H}}_{k,l}\right)^{(q)} = \frac{\mathbf{R}_{k,l}\left(\widehat{\mathbf{S}}_{k,l}\right)^{(q-1)^*}}{\left|\left(\widehat{\mathbf{S}}_{k,l}\right)^{(q-1)}\right|^2} \tag{2.14}$$

where q represents an iteration.

2.2.2 Implicit Pilots

The use of the PSAM approach has the problem of originating an inefficient bandwidth use. To contradict this problem, a different method consisting in using implicit (also called embedded or superimposed) pilots was proposed in [20] and [21]. In this approach, the pilot symbols are superimposed over the data, increasing the pilots' density without sacrificing system capacity. However, the superimposed pilots need to spend more power on the pilot sequence and the levels of mutual interference between the data and the pilots can be high which leads into degrading the performance. Figure 2.5 illustrates a similar scheme as Figure 2.4, only this time it represents how the superimposed pilots are transmitted in an OFDM system.

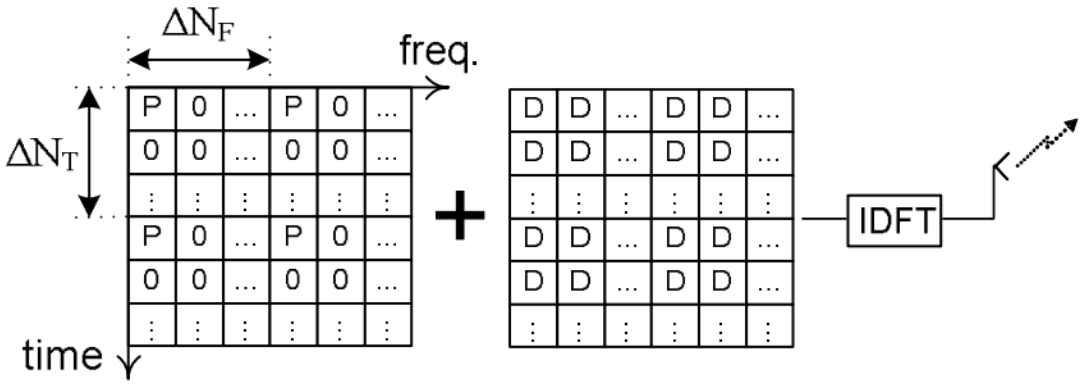

FIGURE 2.5: Frame Structure for an OFDM transmission using implicit pilots where **D** represents a data symbol and **P** represents a pilot symbol [22].

The transmitted and received sequences are expressed in Equations (2.15) and (2.16), respectively:

$$\mathbf{X}_{k,l} = \mathbf{S}_{k,l} + \mathbf{S}_{k,l}^{Pilot} \tag{2.15}$$

$$\mathbf{R}_{k,l} = \mathbf{X}_{k,l}\mathbf{H}_{k,l} + \mathbf{N}_{k,l} \tag{2.16}$$

and the receiver applies the following steps to obtain the channel response estimates in the frequency-domain:

1. The data symbol estimates are removed from the pilots:

$$\left(\widetilde{\mathbf{R}}_{k,l}\right)^{(q)} = \mathbf{R}_{k,l} - \left(\widehat{\mathbf{S}}_{k,l}\right)^{(q-1)} \left(\widehat{\mathbf{H}}_{k,l}\right)^{(q-1)} \tag{2.17}$$

   where $\left(\widehat{\mathbf{S}}_{k,l}\right)^{(q-1)}$ and $\left(\widehat{\mathbf{H}}_{k,l}\right)^{(q-1)}$ represent the data and the channel response estimates from the previous iteration, respectively. This means that this step can only be applied after the first iteration so in the first iteration we define:

$$\left(\widetilde{\mathbf{R}}_{k,l}\right)^{(1)} = \widetilde{\mathbf{R}}_{k,l} \tag{2.18}$$

2. By utilizing a mean with length W, the frequency channel response estimates can be expressed as follows:

$$\left(\widehat{\mathbf{H}}_{k,l}\right)^{(q)} = \frac{1}{W} \sum_{l'=-\lfloor \frac{W}{2} \rfloor}^{l+\lceil \frac{W}{2} \rceil - 1} \frac{\left(\widetilde{\mathbf{R}}_{k,l'}\right)^{(q-1)}}{\mathbf{S}_{k,l'}^{Pilot}} \tag{2.19}$$

3. As shown in the data multiplexed pilots method, the data estimates originated after the first estimation can also be utilized as pilots for channel refinement.

2.3 Orthogonal Frequency-Division Multiplexing

Orthogonal Frequency-Division Multiplexing (OFDM) is a form of multicarrier modulation and its signal is formed by a number of modulated carriers spaced very closely to each other. In [23], the authors presented the first OFDM paper and the authors of [24] proposed the use of Discrete Fourier Transform (DFT) for modulation, demodulation and guard interval, which allowed an easier implementation of OFDM models. A study was conducted in [25] regarding the use of this technology in high speed modems, portable digital communications and high intensity recording software. It also introduced the use of cyclic prefixes. Some of the most recent advances presented in [26] and [27] show how OFDM transmissions through portable communications channels can attenuate the problem of multipath propagation. OFDM has also been included in commercial use due to the developments in technology that have lowered the cost of signal processing [28], [29] and [30].

In conventional multicarrier modulations, when signals are transmitted closely to one another they must have a space between them in order that the receiver can separate them by applying a filter and there must be a guard band between them. In OFDM modulation, the sidebands from each carrier overlap but they can still be received without interference because they are orthogonal to each other. This orthogonality allows a frequency selective channel to be converted into a

non-frequency selective one [31]. Figure 2.6 shows a comparison between the OFDM spectrum and the FDMA spectrum where we can see that OFDM has the advantage of being able to save bandwidth, allowing a more efficient use of the available spectrum for transmission.

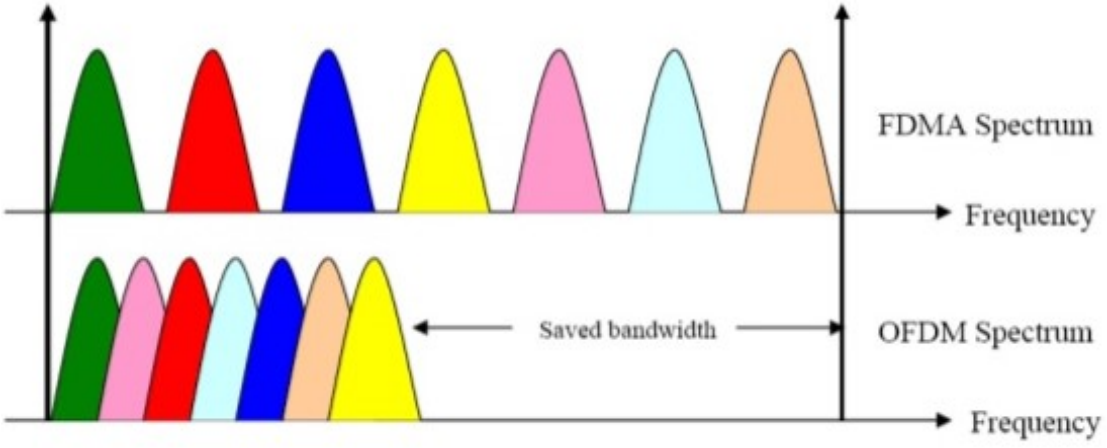

FIGURE 2.6: Comparison of OFDM spectrum with FDMA Spectrum [32].

Besides this ability to save bandwidth, OFDM modulation has a number of advantages, such as:

- The data transmitted on an OFDM signal is spread across its carriers, reducing the data rate and, consequentially, lowering the interference from reflections;
- It reduces the effect of intersymbol interference (ISI) with the help of guard intervals and cyclic prefixes;
- It can successfully and easily adapt to unfavorable channel conditions without complex time-domain equalization;
- OFDM brings robustness to transmissions in multipath fading channels;
- It has low sensitivity to time synchronization errors;
- The FFT can be easily and efficiently implemented which can reduce the computation complexity.

Despite the rather large number of advantages, OFDM also has its downsides:

- The use of cyclic prefixes reduces efficiency of the system;
- It is highly vulnerable to frequency synchronization problems;
- High PAPR requires linear transmitter circuitry which suffers from poor power efficiency.

2.3.1 Mathematical description of OFDM

An OFDM signal consists of N orthogonal subcarriers modulated by N parallel data streams and each of the baseband subcarriers can be expressed as:

$$\phi_k(t) = e^{j2\pi f_k t} \tag{2.20}$$

where f_k represents the frequency of the k^{th} subcarrier. An OFDM baseband symbol that does not possess a cyclic prefix multiplexes N modulated subcarriers:

$$s(t) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} x_k \phi_k(t), \;\; 0 < t < NT \tag{2.21}$$

with x_k as the k^{th} data symbol and NT representing the OFDM symbol's length. The subcarrier frequencies are expressed as f_k and they are equally spaced:

$$f_k = \frac{k}{\sqrt{NT}} \tag{2.22}$$

making subcarriers orthogonal, as described in Equation (2.20), with $0 < t < NT$. The signal depicted in Equation (2.21) separates the data symbols in frequency by overlapping the subcarriers and this is why OFDM modulation uses the available spectrum in such an efficient way.

The signal from Equation (2.21) can be received by utilizing a bank of unmatched filters but, in practice, an alternative demodulation is employed. If we

ignore problems like dispersion and noise, T-spaced in-phase and quadrature sampling of the OFDM symbol generates:

$$s(nT) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} x_k e^{j2\pi \frac{nK}{N}}, \quad 0 \leq n \leq N-1 \tag{2.23}$$

which is the Inverse Discrete Fourier Transform (IDFT) of x_k. This means that the sampled data is demodulated by using a DFT which is one of main properties of OFDM that was proposed in [24], referred earlier in this section.

2.3.2 Cyclic Prefix

The transmission of the signal from Equation (2.21) over a dispersive channel can bring two different problems. One of them is the interference caused by successive OFDM symbols when they are transmitted in a series, known as intersymbol interference. The other problem is when channel dispersion or multipath channel variation destroys the orthogonality between subcarriers, causing intercarrier interference (ICI). To solve both problems, the concept of cyclic prefix was introduced [25].

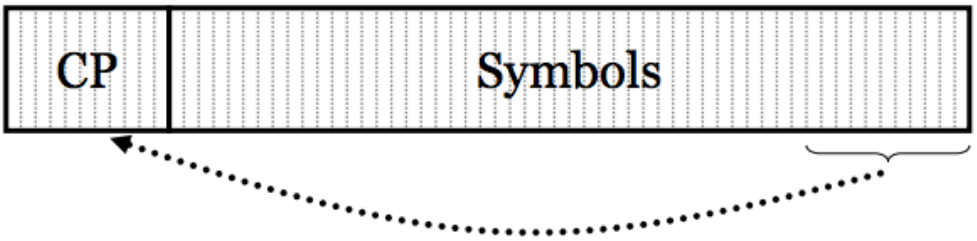

FIGURE 2.7: Cyclic Prefix representation [33].

Figure 2.7 shows that the process consists in copying the last part of an OFDM symbol and inserting it in the beginning, between consecutive OFDM symbols. The cyclic prefix has a duration Δ_{CP} and increases the duration of the OFDM

signal, resulting in:

$$s(T) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} x_k e^{j2\pi f_k t}, \ -\Delta_{CP} \leq t \leq NT \tag{2.24}$$

and the duration Δ_{CP} must be larger than the propagation of the delay to completely eliminate the ISI. But the benefits of eliminating both ISI and ICI come with a cost. More energy is necessary to transmit the signal and this energy requirement depends on the cyclic prefix size (if it is larger then more energy is necessary). The loss of signal-to-noise ratio due to the cyclic prefix is given by:

$$\varepsilon_{loss} = \frac{NT}{NT + \Delta_{CP}} \tag{2.25}$$

2.4 Machine-Type Communications

Machine-Type Communications or Machine to Machine (M2M) communications are automated communications that allow devices to communicate with each other directly or by using a dedicated server, with little or without help of humans and this communication can occur via wired systems or wirelessly. Figure 2.8 illustrates how devices communicate with each other. MTC is known for having a simple implementation, for being cheap and for having a low energy consumption and the main purpose is to make devices enough self-sufficient so they can initiate functions based on the network information. M2M can be used in various applications and some of them are included in Table 2.2.

Functions	**Applications**	**Requirements**
Metering	Power, gas or water metering	Massive number of MTC devices and high coverage
Tracking	Flight radars or asset tracking	High mobility and low power consumption
Security	Surveillance systems	High security and reliability and low latency
Paying	Vending machines	High security

TABLE 2.2: Examples of MTC functions and applications [5]

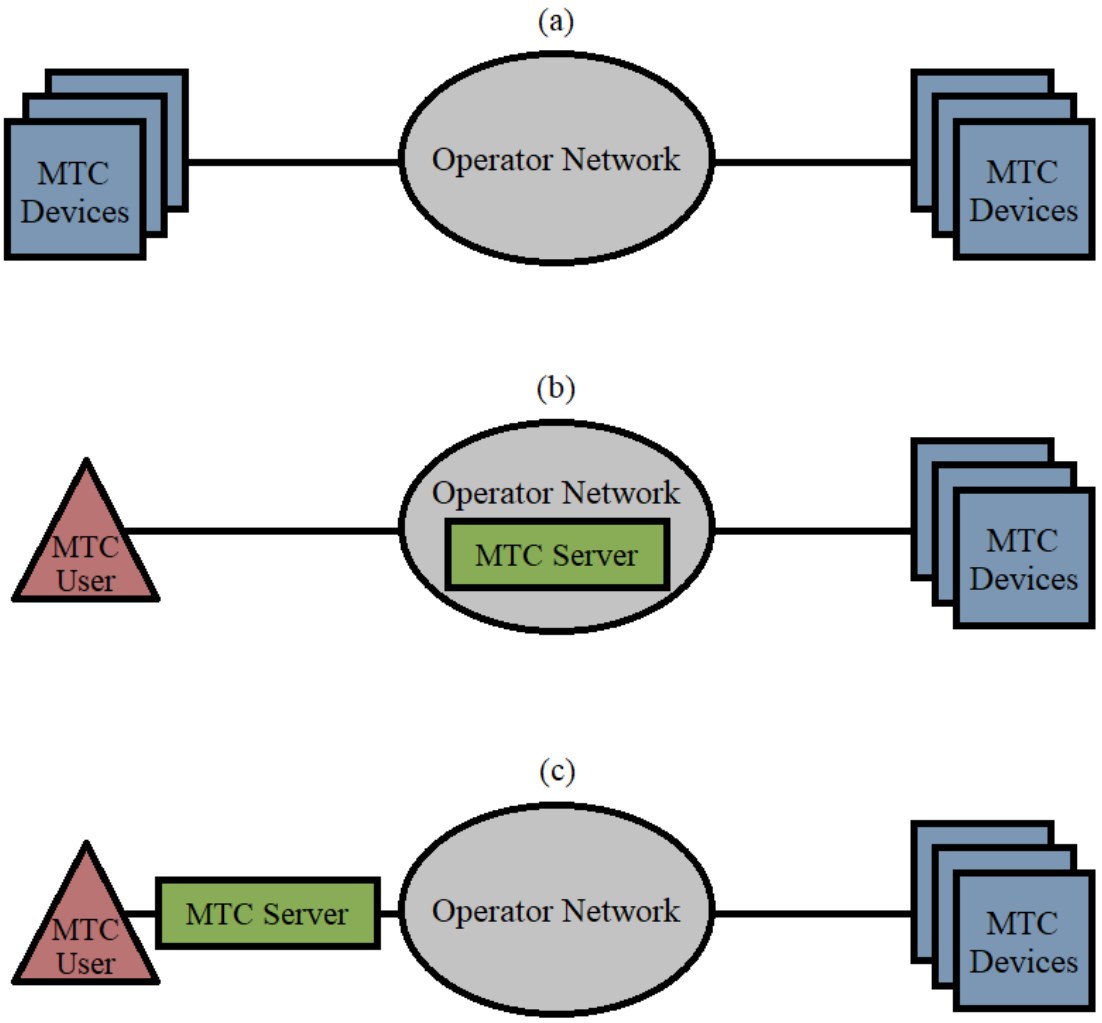

FIGURE 2.8: MTC device communication modes: (a) MTC devices communicate directly with each other, (b) MTC devices communicate with MTC server inside the operator network, (c) MTC devices communicate with MTC server outside the operator network.

This type of communication is supported by various network technologies [5] where point-to-point and multi-hop networks like ad hoc or mesh networks have been used to provide an internet connection to the devices (Internet of Things) [34]. The protocol IEEE 802.15.x provides applications to Personal Area Networks (PAN) [35] and in [36], the 802.11ah protocol supports low-power transmissions with a wider coverage range in Wi-Fi networks. In [37], it is shown how a LTE network can be influenced by MTC and it is concluded that an increasing MTC traffic load does not have an impact on priority services, though it is possible to observe performance degradation for file transfer services.

2.4.1 Features of MTC

In the 3GPP Release 11 [38], various features of M2M were presented and some of the most important are compiled below:

- MTC services can be used by several different devices connected to the same network;
- The transmissions are small which means that devices send small data packets that only include the necessary information;
- If the same message needs to be transmitted to various MTC devices, they can all be managed as a group;
- Time slots are allocated to the transmission and reception of data;
- The devices may send priority messages (can be used as an alarm, for example);
- Transmission can be delayed by the device by sensing high levels of traffic in the network;
- It is possible to restrict the movement of the devices to a certain predefined location;
- MTC devices can be triggered via their location information;
- Monitoring can be implemented if a MTC application needs to monitor the devices.

2.4.2 Cellular Networks in MTC

Cellular networks like UMTS or LTE are extremely suited to this type of communication and are usually utilized in the network domain of M2M applications. Cellular networks [39] allow to utilize its vast geographically coverage area, meaning that operators can provide services like high bandwidth, high mobility and robust

security all over the world. Technologies like femtocell can be used to provide quality of service (QoS) information for important applications (health [40] or security, for example).

Figure 2.9 depicts a generic MTC architecture defined by 3GPP [41] that rests in three primary components: MTC device domain, network domain and MTC application domain. The first component refers to all M2M devices in a network that collect and transmit data to other devices in the same network or to a server. The network domain provides communication between MTC devices and also MTC servers, when applicable. Lastly, the MTC application domain consists of MTC servers that serve as destination for the data collected and transmitted by the MTC devices inside the same network. These servers are usually managed by mobile network operators or third party service providers [38]. MTC servers are also responsible of providing users with an interface to access the assigned MTC applications.

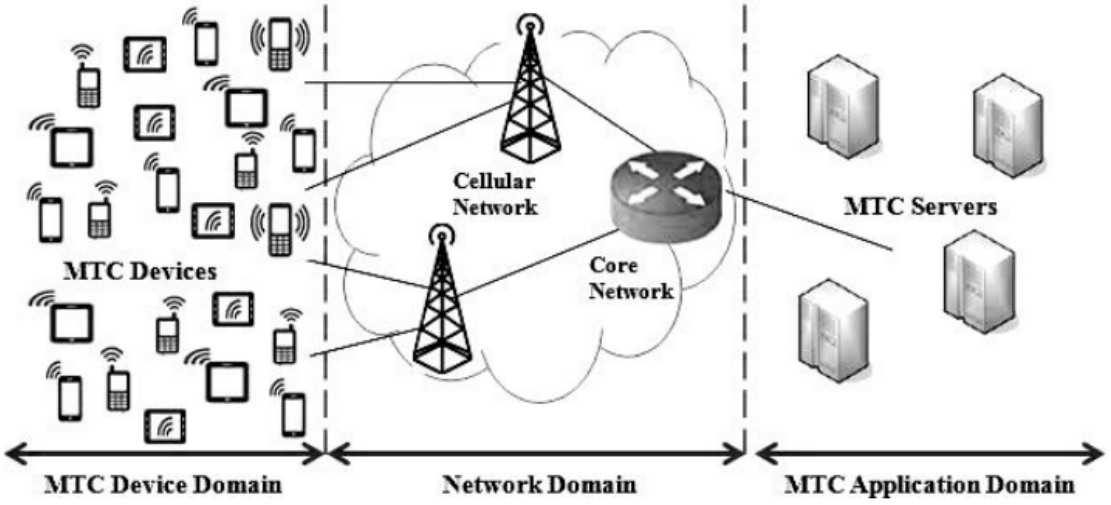

FIGURE 2.9: MTC's generic architecture [41].

2.5 Joint Synchronization and Channel Estimation

Fast and accurate timing synchronization is very important to enable reliable communications in modern wireless systems and has a fundamental role within the context of extreme low-latency MTC, with an example present in [42] where the authors propose a network architecture that absorbs MTC traffic via femtocells. Additionally, accurate channel estimation is crucial in order to enable coherent data detection. Typically, both problems are often handled separately, as seen in [43] and [44] regarding frequency and timing synchronization for OFDM systems and also seen in [45] and [46] where optimal training sequences and channel estimation for wireless OFDM systems were studied, respectively.

Despite the fact that timing synchronization and channel estimation are usually treated separately, there have been a few attempts to address both of these processes' problems together by using a joint approach.

In [47], the authors proposed a simple algorithm to perform joint synchronization and channel estimation algorithm based on Maximum Likelihood (ML) estimation and Generalized Akaike Information Criterion information (GAIC) for the IEEE 801.11 WLAN standard. This algorithm is iterative, allowing the refinement of timing estimates while channel estimation is performed. The channel impulse response will contain leading and tailing zeros if the initial timing estimate obtained is less than the true timing. If the zeros can be estimated then the number of unknown channel coefficients will decrease, leading to a more accurate channel estimation. Results showed the performance is close to the curve depicting exact knowledge of the transmission channel, allowing a simple approach to timing synchronization and channel estimation. [48] also proposed to perform joint timing and frequency synchronization and channel estimation using an ML approach, this time based on a sliding observation vector using a repetitive signal. The objective was to avoid mismatch between results from different individual tasks, so that the problem of errors from one task does not affect the other one.

The work developed in [49] shows the performance of an algorithm based on the shift delay characteristic of the synchronization sequence revealed in the channel estimation process. This characteristic allows to jointly optimize the correct symbol timing offset (STO) with channel estimation via the MMSE criterion. The paper shows that the proposed method reinforces the performance of fine timing synchronization and mean channel estimation error of OFDM systems, allowing nearly ideal STO acquisition accuracy and almost ideal channel impulse response (CIR) MSE performance. The authors of [50] also utilized MMSE to perform channel estimation in tandem with timing synchronization and introduced the notion of effective power delay profile (PDP), encompassing the channel PDP and the probability density function (PDF) of timing synchronization.

More recently, a joint timing and frequency synchronization algorithm was proposed where only one training symbol composed of conjugated symmetric sequence is utilized in a coherent optical orthogonal frequency-division multiplexing (CO-OFDM) system [51]. The timing estimation of this algorithm has the advantage of being robust to poor optical signal-to-noise ratio (OSNR) and chromatic dispersion (CD). The results prove the feasibility and the effectiveness of the proposed algorithm, showing good optical noise and chromatic dispersion tolerance compared with other well established synchronization algorithms and also shows that the frequency estimation error of this algorithm is small.

2.6 Compressed Sensing

Compressed sensing (CS), also known as compressive sensing or compressive sampling, is a signal processing technique that allows to efficiently reconstruct a signal with fewer samples than the sampling theory requires but only if there is knowledge regarding the sparsity of that same signal [52] [53]. Conventionally, one has to sample a signal at a sampling rate equal to or greater than the Nyquist sampling rate so that the signal is completely recovered. However, in applications like imaging, sensor networks or high-speed analog-to-digital compression, the signals

are often sparse meaning that many of its elements are zeros. When sampling an image, for example, only a small fraction of wavelet coefficients may be significant in order to recover the original image while the rest of coefficients are disposed of in various compression algorithms. This process can be wasteful regarding sensing and sampling resources. One of the fundamental theorems of linear algebra says there must be as many equations as there are unknowns, meaning it is not possible to reconstruct a signal from an incomplete set of measurements. But if some of those unknowns are zeros, then it is possible to rebuild the signal and that is what compressed sensing does. It uses the sparsity of a signal in order to make possible the use of a smaller number of measurements to recover the original signal very accurately.

2.6.1 Orthogonal Matching Pursuit

One of the most used reconstruction algorithms is a greedy strategy known as orthogonal matching pursuit (OMP) [54] [55]. Greedy algorithms are iterative signal recovery algorithms capable of calculating the support of the signal and making the locally optimal choice at each iteration, building up to an approximation until the criterion is fulfilled. While not optimally stable, they tend to be extremely fast.

It is an improvement on matching pursuit (MP) but the principle is very similar. In each iteration, an element is chosen from the dictionary elements that best approximates the residual. However, instead of taking the scalar product of the residual and the new dictionary element to get the coefficient weight, the original function is fitted to every dictionary elements that have already been selected by projecting the function orthogonally onto all dictionary atoms that have been selected so far, hence the use of the term orthogonal. This leads to better results than MP with the cost of being computationally heavier.

The work presented in [56] proves that OMP can be used in CS since it shows that OMP successfully reconstructed a sparse signal. Algorithm 1 shows

the application of OMP in CS presented in this paper, where $\mathbf{\Phi}$ represents a measurement matrix, $\mathbf{s}_v$ is a measurement vector, k_s depicts the sparsity level of a signal $\mathbf{a}$, $\widehat{\mathbf{a}}$ is the estimate of $\mathbf{a}$, Λ_k defines a set that contains the positions of the non-zero elements of $\widehat{\mathbf{a}}$, $\mathbf{p}_k$ is an approximation of $\mathbf{s}_v$ and $\mathbf{r}$ represents the residual where:

$$\mathbf{r} = \mathbf{s}_v - \mathbf{p}_k \tag{2.26}$$

Algorithm 1 Orthogonal Matching Pursuit [56]

Input: $\mathbf{\Phi}$, $\mathbf{s}_v$, k_s

1: $\mathbf{r}^{(0)} \leftarrow \mathbf{s}_v$
2: $\Lambda^{(0)} \leftarrow \emptyset$
3: **for** $i = 1, ..., k_s$ **do**
4: $\quad \lambda^{(i)} \leftarrow \arg\max_{j=1,...,n} |\langle \mathbf{r}^{(i-1)}, \Phi_j \rangle|$ // This step is where the algorithm spends most of its running time
5: $\quad \Lambda^{(i)} \leftarrow \Lambda^{(i-1)} \cup \lambda^{(i)}$
6: $\quad \mathbf{\Phi}^{(i)} \leftarrow \left[\mathbf{\Phi}^{(i-1)} \Phi_{\lambda^{(i)}}\right]$
7: $\quad \mathbf{a}^{(i)} \leftarrow \arg\min_{\widehat{\mathbf{a}}} \left\|\mathbf{s}_v - \mathbf{\Phi}^{(i)}\widehat{\mathbf{a}}\right\|_2$ // LS estimation
8: $\quad \mathbf{p}^{(i)} \leftarrow \mathbf{\Phi}^{(i)}\mathbf{a}^{(i)}$
9: $\quad \mathbf{r}^{(i)} \leftarrow \mathbf{s}_v - \mathbf{p}^{(i)}$
10: **end for**
11: $\widehat{\mathbf{a}} \leftarrow \mathbf{a}^{(k)}$

Output: $\widehat{\mathbf{a}}$, $\Lambda^{(k)}$, $\mathbf{p}^{(k)}$, $\mathbf{r}^{(k)}$

OMP does not always offer the same guarantees as a Basis Pursuit algorithm, as seen in [56], but it has the advantage of having a much faster runtime.

2.6.2 ℓ_1-Minimization

ℓ_1-minimization is a Basis Pursuit approach that allows to reconstruct sparse signals. Basis Pursuit was firstly presented in [57] where it was described as an approach that uses convex optimization to find signal representations in over-complete dictionaries. By solving Equation (2.27), it is possible to obtain the decomposition of a signal which minimizes the ℓ_1-norm of the coefficient:

$$\min ||\mathbf{a}||_1 \quad \text{subject to} \quad \mathbf{\Phi a} = \mathbf{s}_v \tag{2.27}$$

where $\mathbf{a}$ is a $N \times 1$ signal, $\mathbf{s}_v$ is a $M \times 1$ measurements vector, $\mathbf{\Phi}$ is a $M \times N$ measurement matrix and $M < N$. $\mathbf{\Phi}$ is given and fixed in advance and it does not depend on the signal.

This approach allows to exactly recover the signal if it is sparse or accurately recover the signal if it is approximately sparse which means that, as long as a signal is sufficiently sparse and the measurement matrix satisfies some conditions independent of the signal, the ℓ_1-minimization will succeed, as proven in [58] and [59].

In [60], the ℓ_1-uncertainty principle was proved . Considering an observation over time:

$$y(t) = f(t) + n(t), \;\; t \in \mathbb{R} \tag{2.28}$$

where $n(t)$ is an impulsive noise term supported on a sparse set T and $f(t)$ is bandlimited and given as:

$$f \in B(\Omega) := \left\{ f : \widehat{f}(\omega) = 0 \;\text{ for }\; |\omega| > \Omega \right\} \tag{2.29}$$

It was observed that by following Equation (2.30) to recover f:

$$\min \left\| y - \widehat{f} \right\|_{\ell_1(\mathbb{R})} \quad \text{subject to } \; \widehat{f} \in B(\Omega) \tag{2.30}$$

it is then possible to exactly recover the signal for whatever values of noise and $\widehat{f} \in B(\Omega)$, with the condition that $|T||\Omega| \leq \frac{\pi}{2}$.

ℓ_1-minimization has several advantages over other CS algorithms, with the most important being that this method provides uniform guarantees, meaning all sparse signals can be reconstructed. This algorithm is also stable and can handle noise and non-exactness inherent to sparse signals. The major drawback is the speed of the process since it relies on linear programming, which has a polynomial runtime.

2.6.3 Iterative Reweighted ℓ_1-Minimization

Despite the good results shown by ℓ_1-minimization regarding the recovery sparse signals from incomplete measurements, a similar method known as iterative reweighted ℓ_1 minimization was proposed in [61]. The main differences between both methods are that this algorithm does not use prior information and, at each step, the ℓ_1 optimization is reweighted using the estimates of the signal obtained in the last minimization step. Algorithm 2 was presented in this paper, where $w_i^q, i = 1, ..., n$ represents the weights for the i^{th} element $\mathbf{a}_i$ of $\mathbf{a}$ in the q^{th} iteration of the iterative reweighted ℓ_1-minimization algorithm and $\mathbf{W}^q$ is the diagonal matrix.

Algorithm 2 Iterative Reweighted ℓ_1 Minimization [61]

1: Set $q = 0$ and $w_i^q = 1, i = 1, ..., n$.
2: Solve the algorithm problem:

$$\mathbf{a}^q = \arg\min \left\| \mathbf{W}^q \mathbf{a} \right\|_1 \quad \text{subject to } \mathbf{\Phi a} = \mathbf{s}_v \tag{2.31}$$

3: With ϵ' representing a tunable positive number, update the weights for each value of i:

$$w_i^{q+1} = \frac{1}{\left| \mathbf{a}_i^q \right| + \epsilon'} \tag{2.32}$$

4: Terminate on convergence or when q reaches the maximum number of iterations, q_{max}. Otherwise, increment q and return to step 2.

Results from [61] showed that the iterative reweighted ℓ_1-minimization algorithm outperforms the standard method, with this algorithm enhancing the sparsity threshold of certain signals, such as sparse signals with Gaussian entries. However, this algorithm fails to improve the sparsity thresholds when the non-zero elements of a signal are flat and it is even more computationally heavy than its standard counterpart. The overall good results of this method pushed the development of more tests that confirmed that the performance was, indeed, very good and encouraged other researchers to build similar approaches that managed to achieve results at the same level and even better, as seen in [62] and [63].

2.6.4 Applications of CS

CS is a method that can be utilized in various applications from different areas. Health is one of those areas, where different CS algorithms have been developed for magnetic resonance imaging (MRI) purposes [64]. MRI is a process that takes a long time to perform so there is an interest in reducing the time needed to collect measurements without decreasing the quality of the image. MR images are sparse in their pixel representation which makes CS a pertinent approach in this area. CS has also been utilized in comparative DNA Microarray [65].

A conventional radar system transmits a pulse and then uses a matched filter to correlate the received signal with the transmitted pulse. The receiver uses a pulse compression system in tandem with an analog-to-digital converter to process the signal. This approach is difficult and expensive, so CS was used to discretize the time-frequency plane into a grid [66]. If the number of targets is small then the grid will be sparsely occupied and the target scene can be recovered [67].

Channel estimation is another application where CS has been utilized quite often [68]. In [69], the number of pilots required was reduced due to the authors performing a channel estimation technique based on CS for multicarrier systems, exploiting the delay-Doppler sparsity.

Recently, CS has been used several times in wireless communications, as seen in [70]. CS is explored in this dissertation for joint timing synchronization and channel estimation.

Chapter 3

Efficient Channel Estimation Using TCH Codes

This chapter focuses on the study of the performance of channel estimation using data multiplexed pilots and implicit pilots based on TCH codes. Results obtained through simulations are also illustrated, paired with the corresponding interpretation and comments.

3.1 System Characterization

3.1.1 Structure of the Transmitter

The transmitter chain depicted in Figure 3.1 is inspired by the works of [22] and it combines QAM constellations with an OFDM transmission that can use data multiplexed or implicit pilots. Each data stream is encoded with turbo codes, interleaved and mapped. The pilot symbols are added to the modulated data symbols and, lastly, the resulting sequence is converted to the time domain by using an IDFT.

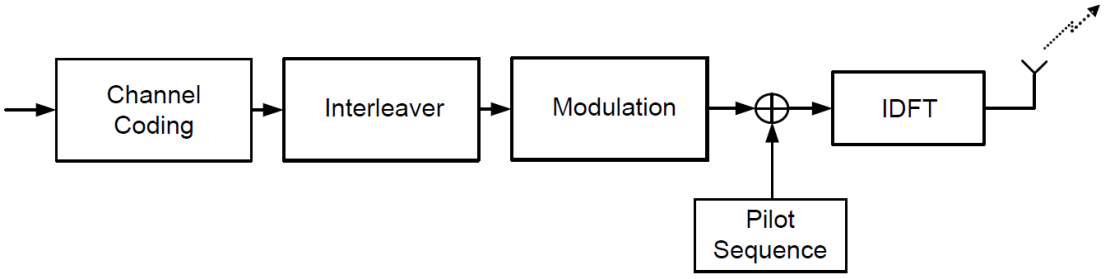

FIGURE 3.1: Structure of the transmitter chain.

3.1.2 Data Multiplexed and Implicit Pilots' Frame Structures

Figure 3.2 illustrates the frame structure considered for an OFDM system with N carriers using data multiplexed pilots, where only the first column of the pilot grid contains pilot symbols and the first column of the data grid is empty.

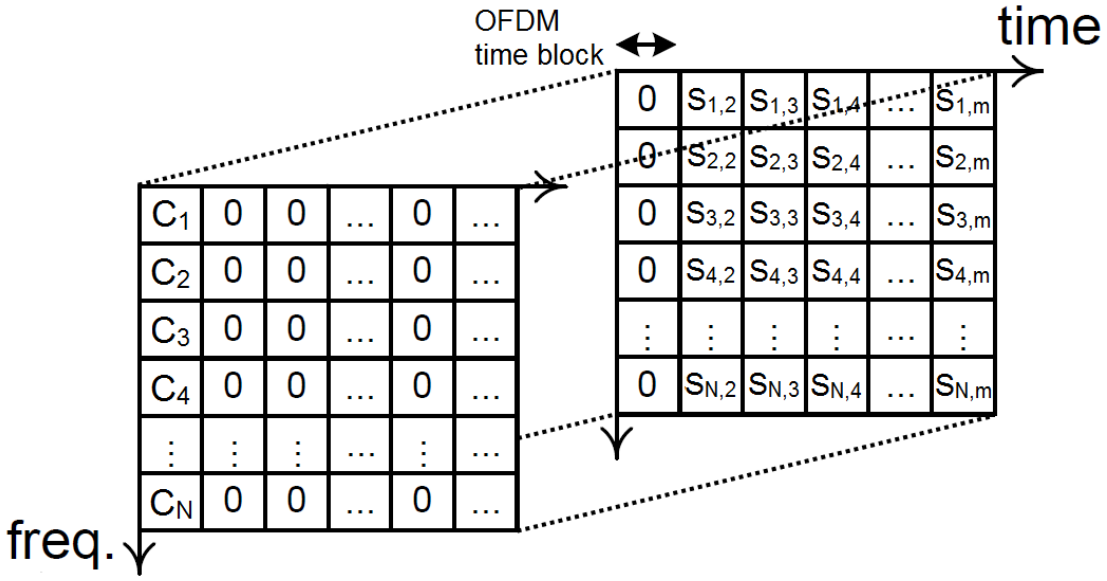

FIGURE 3.2: Frame structure used for an OFDM transmission containing data multiplexed pilots where **C** represents a pilot symbol and **S** represents a data symbol.

Figure 3.3 depicts a frame structure similar to the one shown in Figure 3.2 but this time implicit pilots are used. Now, it is possible to verify that all positions

in both grids are filled since the pilots are superimposed over the data. The grids from both frame structures are built by using an OFDM time block spacing in the time domain. The transmitted sequences are described as follows:

$$\mathbf{X} = \mathbf{S} + \mathbf{C}, \tag{3.1}$$

where $\mathbf{S}$ describes an $N \times 1$ vector where the elements are complex valued modulated symbols drawn from an M-sized complex valued constellation and $\mathbf{C}$ is an $N \times 1$ vector that corresponds to $\mathbf{C} = DFT\{\mathbf{c}\}$ which is the DFT of a TCH codeword. In order to take advantage of the good auto-correlation properties of the TCH codes, we utilize the DFT of these codes. The objective of using these auto-correlation properties is mainly for time synchronization purposes.

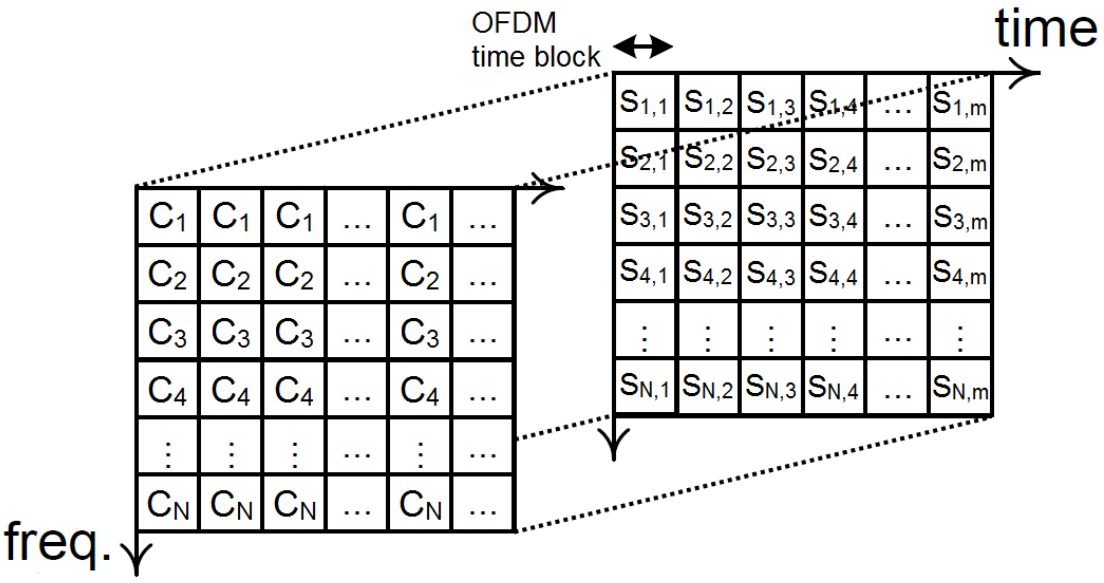

FIGURE 3.3: Frame structure used for an OFDM transmission containing implicit pilots where $\mathbf{C}$ represents a pilot symbol and $\mathbf{S}$ represents a data symbol.

3.1.3 Structure of the Receiver

As mentioned in Section 2.2.2, transmitting superimposed pilots on data creates mutual interference between pilots and data. To reduce this interference and to achieve reliable channel estimation and data detection, a receiver is proposed based

on a similar approach used in [22], capable of performing these tasks via iterative processing. The structure of the referenced receiver is presented in Figure 3.4.

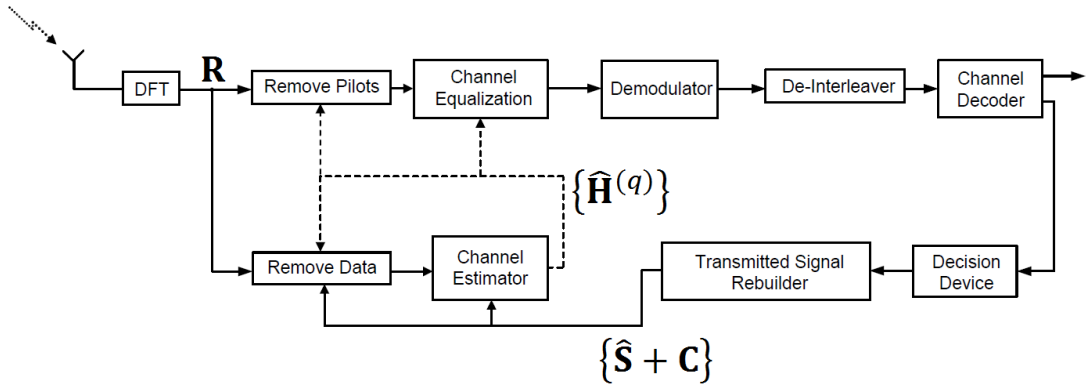

FIGURE 3.4: Structure of the iterative receiver.

Before entering the Channel Equalization block, the pilot symbols are removed from the sequence. Then, the sequence of equalized samples is demodulated into bit streams which are processed to reconstruct an estimate of the transmitted signal. To perform this reconstruction, the sequence also needs to be de-interleaved and decoded after the demodulation. The channel decoder has two outputs where one represents the data estimate sequences and the other represents the code symbols' log-likelihood ratio (LLR) estimate sequences, with the latter passing through the decision device block. The reconstructed sequence, $\widehat{\mathbf{S}}$, can be used to improve the channel estimates in the next iteration.

3.1.4 Channel Estimation

If the overall channel impulse response is shorter than the N_G-sized cyclic prefix, we can describe the frequency domain received sequence as follows:

$$\mathbf{R} = \mathbf{H}(\mathbf{S} + \mathbf{C}) + \mathbf{N}, \tag{3.2}$$

where $\mathbf{H}$ is an $N \times N$ diagonal matrix that stands for the channel frequency response and $\mathbf{N}$ represents an $N \times 1$ vector of noise samples in the frequency

domain. This model directly matches the channel estimation based on implicit pilots method but it can also represent the data multiplexed pilots with a block of pilot symbols by simply establishing $\mathbf{S} = 0$, which can be observed in Figure 3.2.

As mentioned before, the receiver can employ an iterative approach based on [22], meaning it is possible to obtain the frequency channel response. Each of the following steps is executed for each iteration q:

1. Data symbol estimates are removed from pilots. The resulting sequence becomes:

$$\widetilde{\mathbf{R}}^{(q)} = \mathbf{R} - \widehat{\mathbf{S}}^{(q-1)}\widehat{\mathbf{H}}^{(q-1)}, \tag{3.3}$$

   where $\widehat{\mathbf{S}}^{(q-1)}$ and $\widehat{\mathbf{H}}^{(q-1)}$ are the symbol and channel estimates from the previous iteration. When $q = 1$ we simply use $\widetilde{\mathbf{R}}^{(1)} = \mathbf{R}$. The described step is only applied when using superimposed pilots.

2. The channel frequency response estimates is calculated using:

$$\widetilde{\mathbf{H}}^{(q)} = |\mathbf{\Lambda}|^{-2}\mathbf{\Lambda}^{H}\widetilde{\mathbf{R}}^{(q-1)}, \tag{3.4}$$

   where $\mathbf{\Lambda} = diag(\mathbf{C})$, where diag$(\cdot)$ represents a diagonal matrix whose elements are contained in the vector used as argument. $|\mathbf{\Lambda}|$ denotes the element-wise absolute value operation and $(\cdot)^H$ depicts the conjugate transpose of a matrix/vector. After the first iteration, the estimates of data symbols can also be used as pilots for channel estimation refinement. In this case, it is used $\mathbf{\Lambda} = diag\left(\widehat{\mathbf{S}}^{(q-1)}\right)$ for data multiplexed pilots and $\mathbf{\Lambda} = diag\left(\widehat{\mathbf{S}}^{(q-1)} + \mathbf{C}\right)$ for implicit pilots.

3. The channel estimates can be augmented by assuring that the corresponding impulse response has a duration N_G. This is accomplished by utilizing:

$$\widehat{\mathbf{H}}^{(q)} = diag\left(\mathbf{F}\mathbf{T}\mathbf{F}^{H}\widetilde{\mathbf{h}}^{(q)}\right), \tag{3.5}$$

where:

$$\mathbf{T} = \begin{bmatrix} \mathbf{I}_{N_G} \\ \mathbf{0}_{(N-N_G)\times N_G} \end{bmatrix}, \tag{3.6}$$

$\mathbf{0}_{N_{CF}\times(N-N_{CF})}$ represents a size $(N - N_G) \times N_G$ matrix full of zeros, while $\mathbf{I}_{N_G}$ depicts an $N_G \times N_G$ identity matrix. The $N \times N$ scaled discrete Fourier transform (DFT) matrix is represented by $\mathbf{F}$, such that $\mathbf{I}_N = \mathbf{F}^H\mathbf{F}$, and $\widetilde{\mathbf{h}}^{(q)}$ illustrates the $N \times 1$ vector that contains the diagonal of $\widetilde{\mathbf{H}}^{(q)}$.

3.2 Analysis of the Results

The results presented in this section were obtained by performing Monte Carlo simulations in MATLAB with an 8 equal power tap Rayleigh fading channel, using 256 OFDM carriers. Every graph depicts the BER performance, which represents the number of bit errors divided by the total number of transmitted bits during a time interval.

The graph presented in Figure 3.5 shows the results of channel estimation using data multiplexed pilots where blocks with TCH words of length 256 were sent along with pilots. It was used QPSK modulation and the pilot power values, which are relatively measured to the channel data, ranged between 0 to -12 with jumps of -3 dB. Coding was not applied and a conventional receiver was utilized, which means there was only one receiver iteration. A perfect estimation curve is shown for comparison purposes.

The curves that represent higher pilot power values have slightly better results, with 0 and -3 dB having a very similar result to the perfect estimation curve until reaching 20 dB. But after that value, for higher E_s/N_0 values, the curves get further away from the perfect estimation one and all of them converge, reaching a BER value of approximately 2×10^{-4} for a E_s/N_0 value of 50 dB. It is visible that the results are not very good, which was expected since coding was not used.

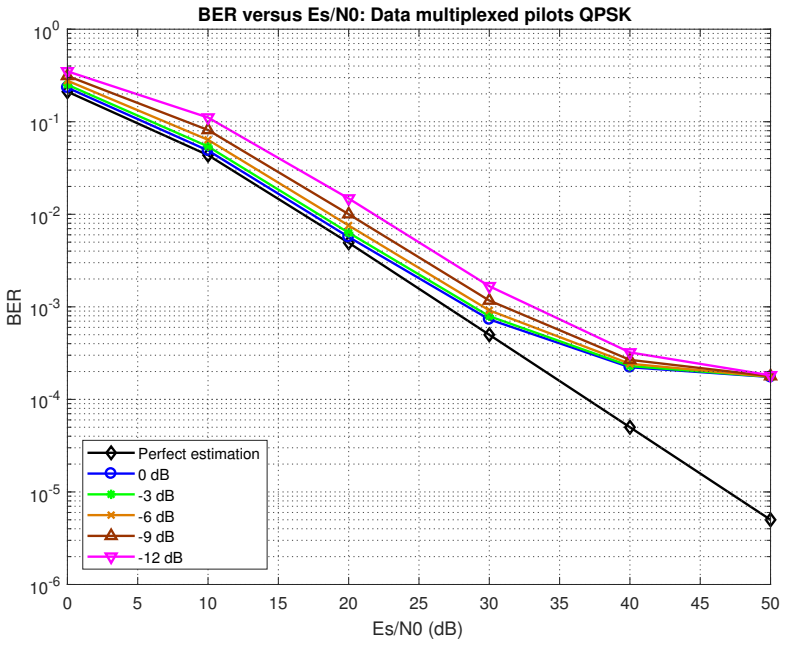

FIGURE 3.5: BER performance of channel estimation using data multiplexed pilots based on TCH codewords and QPSK modulation, considering different pilot power values.

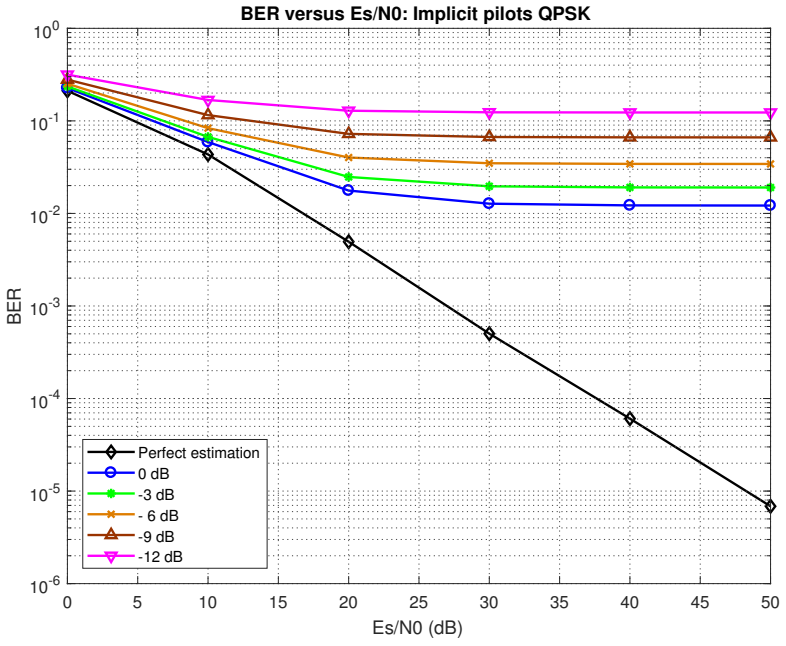

FIGURE 3.6: BER performance of channel estimation using superimposed pilots on TCH codewords and QPSK modulation, considering different pilot power values.

Figure 3.6 illustrates the results of using the same parameters of the previous simulation scenario but this time it is utilized the implicit pilots method instead of using data multiplexed pilots. Clearly, using superimposed pilots is worse than using data multiplexed pilots in this scenario since it is visible that none of the curves can keep up with the perfect estimation curve. All curves stagnate at a E_s/N_0 value around 30 dB and the 0 dB curve, which has the best performance of the five curves, only reaches BER values of approximately 10^{-2}. This bad performance is justifiable not only for not using channel coding but also because an iterative receiver was not applied.

In the results presented in Figures 3.7 and 3.8, the same simulation conditions of figures 3.5 and 3.6 were used, respectively, but this time, 64QAM modulation was used instead of QPSK modulation, in order to understand how the performance is going to be affected when using a different, more complex modulation.

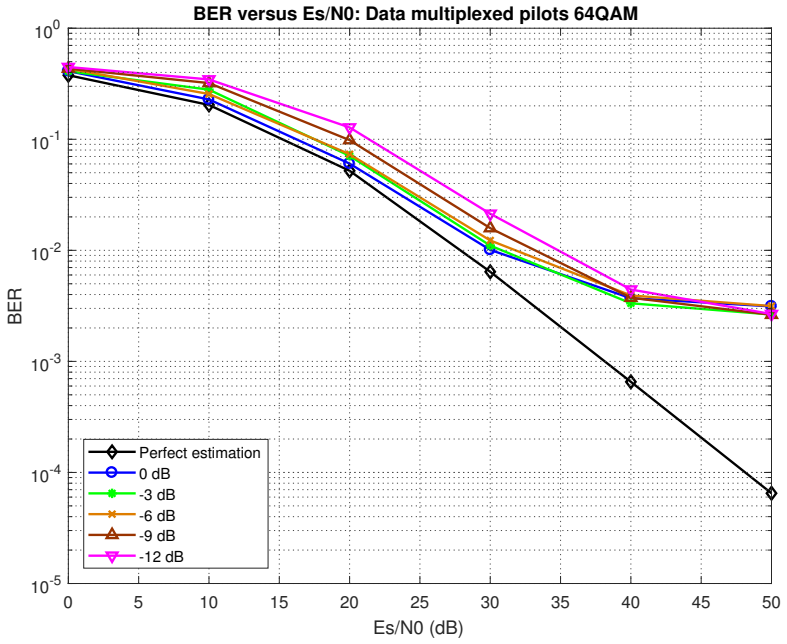

FIGURE 3.7: BER performance of channel estimation using data multiplexed pilots based on TCH codewords and 64QAM modulation, considering different pilot power values.

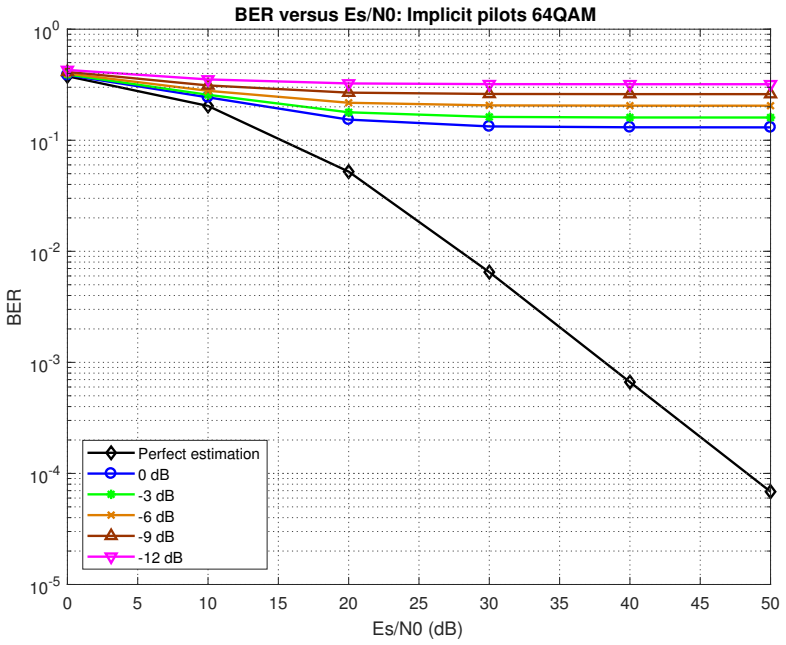

FIGURE 3.8: BER performance of channel estimation using superimposed pilots on TCH codewords and 64QAM modulation, considering different pilot power values.

By comparing the results from both modulations, it is possible to see that 64QAM has a poorer performance. Figure 3.7 shows a similar behavior to Figure 3.5 with 0 dB presenting a very similar performance to the perfect estimation estimation curve until 20 dB. But this time, -3 dB starts to have a slightly better performance at around a E_s/N_0 value of 35 dB. The -12 dB curve also presents a slightly better performance after 45 dB. All of the curves depicting the different pilot power values converge to a BER value of, approximately, 3×10^{-3}.

As for using implicit pilots in 64QAM modulation, it is visible that the performance is far from ideal and it is much worse than when it was used in conjunction with QPSK modulation. Given that the performance of channel estimation with implicit pilots has worse results than the data multiplexed ones, it was decided to utilize an iterative receiver. Figures 3.9 and 3.10 show the results of performing channel estimation using implicit pilots in QPSK and 64QAM modulation, respectively. The pilot power value is fixed at 0 dB and the receiver iterations value is gradually increased from 1 to 8.

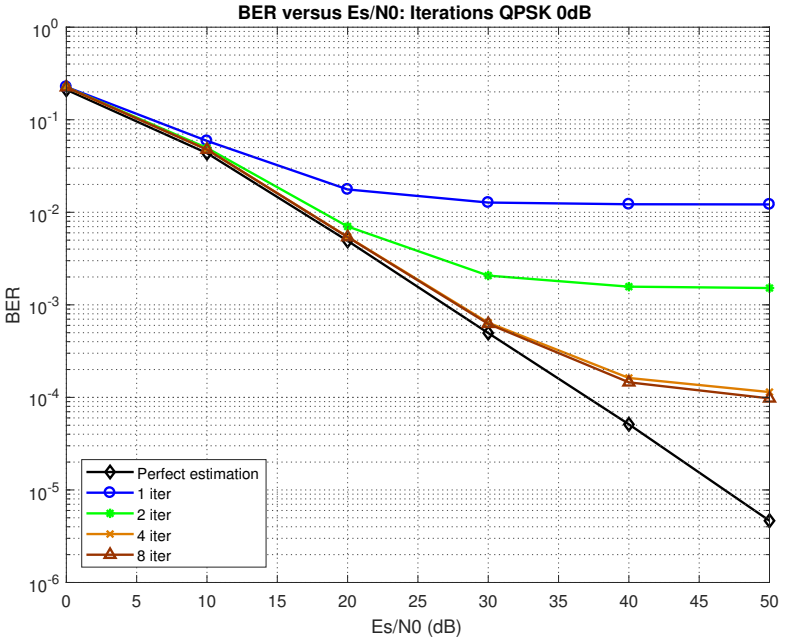

FIGURE 3.9: BER performance of channel estimation using superimposed pilots on TCH codewords and 64QAM modulation, considering different receiver iteration values.

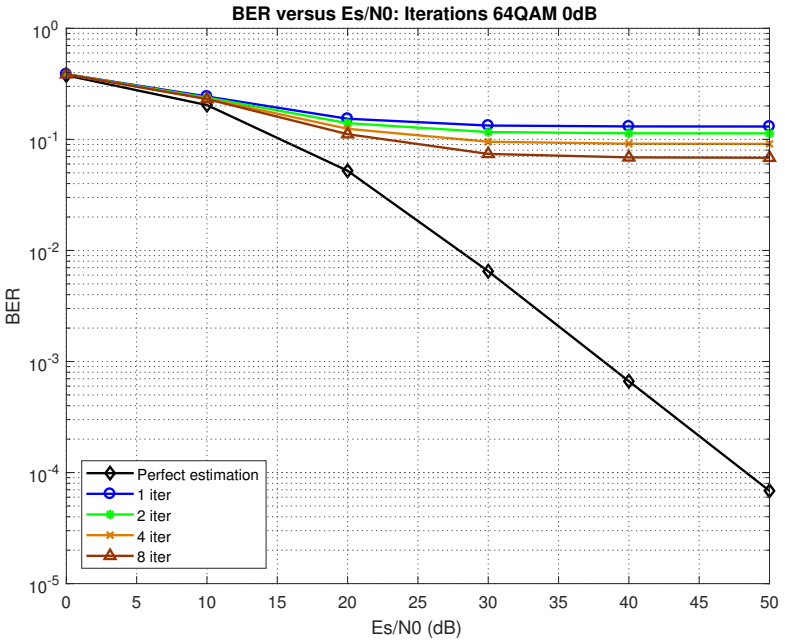

FIGURE 3.10: BER performance of channel estimation using superimposed pilots on TCH codewords and 64QAM modulation, considering different receiver iteration values.

By comparing Figure 3.9 with Figure 3.6, it can be seen that the use of an iterative receiver has allowed to increase the performance of estimating the channel with superimposed pilots on TCH codes. The curve that represents 2 receiver iteration not only shows better BER performance results but the curve also stagnates later, at around 40 dB instead of 30 db for 1 iteration. But the best results come from employing 4 and 8 receiver iterations with both results achieving almost equal results. By utilizing 4 receiver iterations it is also possible to get slightly better results than the ones obtained from performing channel estimation with data multiplexed pilots, as shown in Figure 3.5, making the use of implicit pilots instead of data multiplexed pilots very promising.

Unfortunately, the results of Figure 3.10 do not look very good since the number of iterations had very little impact on the performance of the system, with all of the curves are very far away from the perfect estimation curve. This means that, for 64QAM, it is not enough to utilize an iterative receiver and there is a need to use channel coding in order to make implicit pilots viable.

Figure 3.11 presents again the results for channel estimation using data multiplexed pilots based on TCH codes of length 256 and the modulation is QPSK. But this time it is also used channel coding. The channel encoders are rate 1/2 turbo codes based on two identical convolutional codes with two constituent codes characterized by $G(D) = [1 + D^2 + D^3)/(1 + D + D^3)]$ [71]. At the receiver, 18 turbo decoding iterations were employed while using a conventional receiver.

By analyzing these results, it is possible to see that the curves for high values of pilot power are very close to the curve that depicts perfect estimation, meaning that higher pilot power values, such as 0, -3 and -6 dB, translate into better results, as expected. For higher $E_s/N0$ values, the curves that represent -9 and -12 dB get progressively further away from the perfect estimation curve, resulting in a slightly poorer performance. Nevertheless, any of the pilot power values shown in this graph have a much better result than the ones presented in Figure 3.5, which means that using channel coding improves vastly the performance of channel estimation.

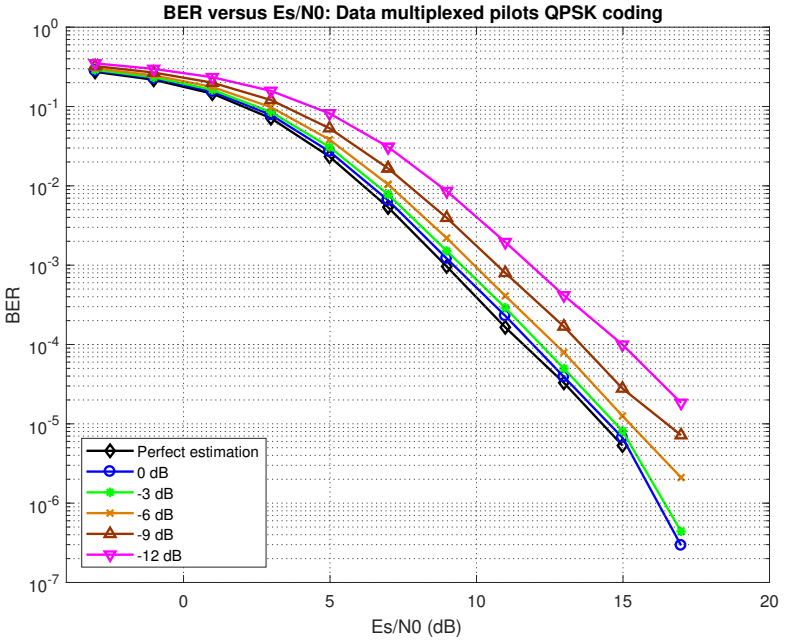

FIGURE 3.11: BER performance of channel estimation using data multiplexed pilots based on TCH codewords, QPSK modulation and channel coding, considering different pilot power values.

Figure 3.12 shows the BER performance results for channel estimation using also QPSK modulation but now, the estimation was performed by using superimposed pilots on TCH codewords. Different values regarding pilot power were considered again and the same turbo codes used in Figure 3.11 were utilized. For this simulation, 6 receiver iterations with 3 turbo decoding iterations each were applied in the iterative scheme.

Once again, the performance is better for higher values of pilot power. But this time, the curves that represent -9 and -12 dB have a larger difference from the perfect estimation curve than the difference observed in Figure 3.11. This results from the interference between the data symbols and the pilots, mutual interference, meaning that higher pilot power levels are needed when performing channel estimation with implicit pilots to obtain a good performance. Now that an iterative receiver and channel coding are being used simultaneously, it is possible to achieve a much better performance than the ones observed in Figures 3.6 and 3.9 and implicit pilots are now able to keep up with data multiplexed pilots.

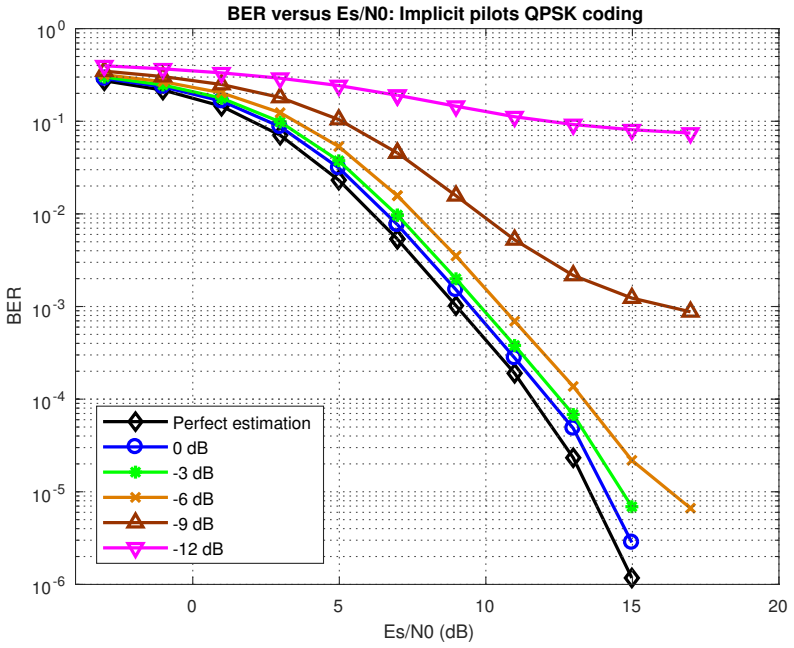

FIGURE 3.12: BER performance of channel estimation using superimposed pilots on TCH codewords, QPSK modulation and channel coding, considering different pilot power values.

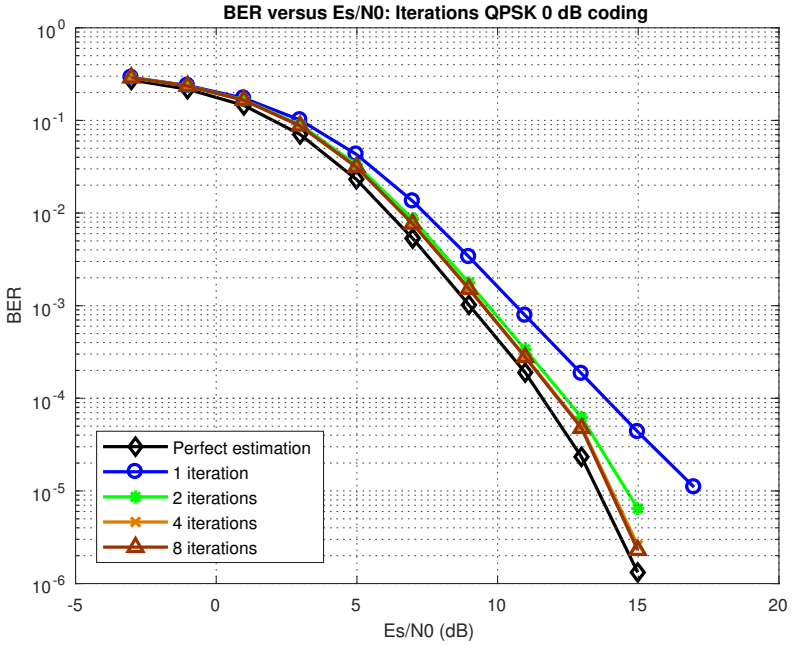

FIGURE 3.13: BER performance of channel estimation using superimposed pilots on TCH codewords, QPSK modulation and channel coding, considering different receiver iteration values.

The results displayed in Figure 3.13 were obtained by using the same simulation parameters used to build Figure 3.12 but this time, the pilot power is fixed at 0 dB and the number of iterations in the receiver is gradually increased.

When the number of iterations used in the receiver is higher, the performance is better, which is visible by comparing the simulated curves with the perfect estimation one. Still, the difference in performance is small after we stop using a conventional receiver and it is almost indistinguishable for the highest simulated values of iterations, 4 and 8. This shows that the performance is not greatly affected by increasing the number of receiver iterations after a certain value is reached, just like it was already shown in Figure 3.9 though now that there is channel coding, the effect of applying iterations in the receiver is not as big as it was when there was no channel coding at all.

In Figure 3.14, it is considered the performance of channel estimation using either data multiplexed pilots or implicit pilots and also if the data sent is based on TCH codewords or conventional pilots.

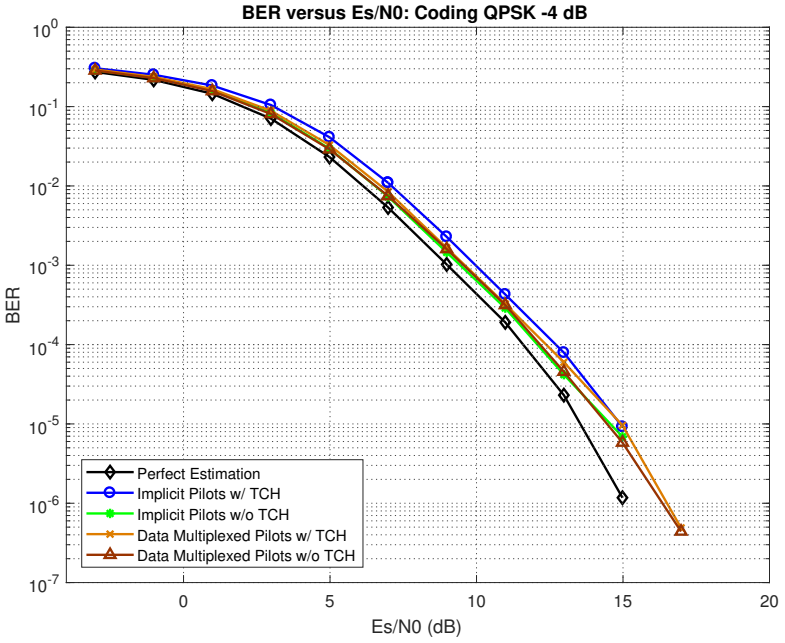

FIGURE 3.14: BER performance of channel estimation utilizing QPSK modulation and coding while considering different pilot approaches and based on TCH codewords or conventional pilots.

It was used QPSK modulation and the channel coding and receiver structures used in Figures 3.11 and 3.12 were used to simulate the curves regarding data multiplexed pilots and implicit pilots, respectively. Only a single value of pilot power, -4 dB, was considered, given the results from previous graphs.

All of the simulated cases possess an almost identical and very good performance, which is justified by observing the proximity between all of the curves and also because no BER floor is visible. The difference between the performance of data multiplexed pilots and implicit pilots is very small and even though data multiplexed pilots have a slightly better performance, implicit pilots have the advantage of avoiding spectral degradation and have a more efficient bandwidth use. Conventional pilots, labeled in Figure 3.14 as "w/o TCH", have a slightly better performance than the ones based on TCH code words but the difference is really small and TCH codewords have the benefit of possessing good synchronization properties, meaning there is the possibility of performing joint channel estimation and synchronization with TCH codes.

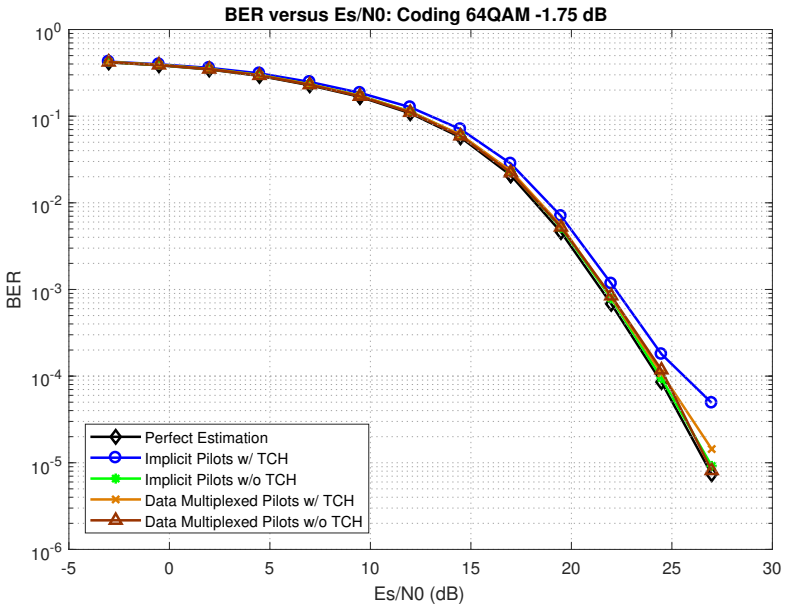

FIGURE 3.15: BER performance of channel estimation utilizing 64QAM modulation and coding while considering different pilot approaches and based on TCH codewords or conventional pilots.

Finally, Figure 3.15 was built using the same simulation conditions of Figure 3.14 with the exception of the modulation, which this time is 64-QAM instead of QPSK, and the pilot power, which is now a fixed value of -1.75 dB. This fixed value is higher because amplitude modulation methods are more susceptible to noise.

As expected, the results show that with 64-QAM modulation we need higher BER values than the ones obtained from the QPSK simulations. All the curves are closer to the perfect estimation curve than the ones presented in Figure 3.14, with the "Implicit Pilots w/TCH" showing a small detour starting around 25 dB. But the declivity is smaller this time, which is observable in the graph.

After analyzing the results of using TCH codes for channel estimation using two different pilot approaches, data multiplexed pilots and implicit pilots, it is possible to conclude that both approaches are reliable. Using TCH codes for channel estimation is justified based on the fact that they have very similar performance levels when compared with conventional pilots and also because TCH codes have great synchronization properties, meaning that it is possible to simultaneously use them in the system for synchronization purposes, making TCH codes a better choice for channel estimation. Regarding the modulations used, QPSK has a better performance than 64-QAM but both present really good performances, showing that TCH codes can be successfully used with both modulations. The work developed in this chapter was utilized to write a paper that has been published [72] and is presented in Appendix A.

Chapter 4

Joint Timing Synchronization and Channel Estimation Using TCH Codes and Compressed Sensing

In this chapter the performance of TCH codes in timing synchronization is evaluated as well as channel estimation using a sparse based reconstruction approach, namely compressed sensing. Joint channel estimation and timing synchronization problem is formulated in a form that can fit CS framework and then modified versions of sparse reconstruction techniques are proposed that can find reliable solutions for it. This approach allows the channel impulse response (CIR) to be obtained simultaneously to the symbol timing offset. Through numerical simulations the performance of TCH codes versus other different codes is compared.

4.1 Signal Model and Problem Formulation

For timing synchronization and channel estimation purposes, a training sequence $\mathbf{s} = [s_0 \;\; s_1 \;\; ... \;\; s_{N_c-1}]^T$ of length N_c is considered, where $(.)^T$ denotes the transpose of a vector/matrix. This sequence is transmitted through the channel, represented by a finite response filter $\mathbf{h} = [h_0 \;\; h_1 \;\; ... \;\; h_{L-1}]^T$, where L is the maximum expected

channel length and $h_1 \in \mathbb{C}$. Not all h_1 are necessarily nonzero and the effective channel length, which is unknown at the receiver, can be much smaller than L. If the signal arrives at the receiver with an unknown delay (in samples) of T, then it can be written as the linear convolution of the delayed training sequence and the channel as:

$$y_t = \sum_{l=0}^{L-1} h_l s_{t-T-l} + n_t, \;\; 0 \leq t \leq M_o - 1 \tag{4.1}$$

where M_o is the number of observation samples and n_t represents the noise. The received signal can be rewritten by using a matrix/vector notation:

$$\mathbf{y} = \mathbf{Z}\mathbf{h}^{ext} + \mathbf{n} \tag{4.2}$$

where $\mathbf{Z}$ is an $M_o \times N_s$ Toeplitz matrix defined as:

$$\mathbf{Z} = \begin{bmatrix} \ddots & 0 & & 0 \\ \vdots & s_0 & \ddots & \vdots \\ \ddots & \vdots & \ddots & 0 \\ 0 & s_{N_c-1} & & s_0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & & \ddots & s_{N_c-1} \\ 0 & & & 0 \end{bmatrix} \tag{4.3}$$

with N_s representing the size of the search window, which is not necessarily equal to the observation window M_o, and $\mathbf{h}^{ext}$ being the channel impulse response augmented with several zeros, namely T_0 zeros before the effective channel response and $N_s - L - T_0$ zeros after:

$$\mathbf{h}^{ext} = \begin{bmatrix} \underbrace{0 \ldots 0}_{T_0} & \mathbf{h}^T & \underbrace{0 \ldots 0}_{N_s - L - T_0} \end{bmatrix}^T \tag{4.4}$$

Using this representation and assuming the usual scenario where $\left\|\mathbf{h}\right\|_0 \leq L$ and the size of the search window is large, then $\left\|\mathbf{h}^{ext}\right\|_0 \ll N_s$ and the channel estimation and synchronization problem can be formulated in a form that is closely

related to sparse signal reconstruction problems and CS framework, both expressed in Equations 4.5 and 4.6:

$$\min_{\mathbf{h}_{ext}} f\left(\mathbf{h}^{ext}\right) \triangleq \left\|\mathbf{y} - \mathbf{Z}\mathbf{h}^{ext}\right\|_2^2 \tag{4.5}$$

$$\text{subject to} \quad \max\{|i-j+1| : h_i^{ext} \neq 0, h_j^{ext} \neq 0\} \leq L \tag{4.6}$$

where $||.||_p$ is the ℓ_p-norm of a vector and $||.||_0$ is its cardinality.

4.2 Joint Timing and Channel Estimation

In order to solve Equations 4.5 and 4.6, an extensive search with combinatorial computational complexity is required, meaning other alternative approaches that are computational feasible for problems of practical sizes must be used. Equations 4.5 and 4.6 are related to cardinality constrained and minimization problems in CS framework. Relaxing Equation 4.6 into the simpler constraint $\left\|\mathbf{h}^{ext}\right\|_0 \leq L$ allows the application of CS reconstruction techniques to recover $\mathbf{h}^{ext}$, T_0 and $\mathbf{h}$. In this section some modified versions of sparse reconstruction techniques are proposed that can cope with the problem presented in Equations 4.5 and 4.6.

4.2.1 Constrained Length OMP

The first approach proposed is based on a greedy strategy known as OMP [56], already referenced in section 2.6.1. The algorithm is modified in order to cope with the problem shown in Equation 4.6. Algorithm 3 summarizes its main steps. In each iteration one new element for the support set Ω is selected. Lines 4-7 and 11 show the proposed modifications, which limit the addition of a new candidate position to the support of $\widehat{\mathbf{h}}^{ext}$ only if it does not result in the violation of Equation 4.6. The proposed algorithm allows the use of a conventional LS method or MMSE method, in line 9, in order to compute the channel estimates $\widehat{\mathbf{h}}_{\Omega}^{ext}$ at the support positions. This is required for the residual value, $\mathbf{r}$.

Algorithm 3 Constrained Length OMP

Input: $\mathbf{y}$, $\mathbf{Z}$, L, N_s

$\widehat{\mathbf{h}}^{ext} = 0$, $\mathbf{r} = \mathbf{y}$, $\Omega = \emptyset$, $\Lambda = \emptyset$, $\widehat{T}_0 = N - 1$, $t_{max} = 0$
for $l = 0, ..., L-1$ **do**
  $\widetilde{\mathbf{h}}^{ext} \leftarrow \mathbf{Z}^H \mathbf{r}$
  repeat
    $t_{best} \leftarrow \arg\max_{t \notin \Lambda} \left| \widetilde{h}_t^{ext} \right|$
    $\Lambda \leftarrow \Lambda \cup \{t_{best}\}$
  until $t_{max} - L + 1 \leq t_{best} \leq L - 1 + \widehat{T}_0$
  $\Omega \leftarrow \Omega \cup \{t_{best}\}$
  $\widehat{\mathbf{h}}_{\Omega}^{ext} \leftarrow \left(\mathbf{Z}_{\Omega}^H \mathbf{Z}_{\Omega}\right)^{-1} \mathbf{Z}_{\Omega}^H \mathbf{y}$ // LS estimation
  $\widehat{\mathbf{h}}_{\Omega}^{ext} \leftarrow \left(\mathbf{Z}_{\Omega}^H \mathbf{Z}_{\Omega} + 2\sigma^2 \mathbf{I}_l\right)^{-1} \mathbf{Z}_{\Omega}^H \mathbf{y}$ // MMSE estimation
  $\mathbf{r} \leftarrow \mathbf{y} - \mathbf{Z}\widehat{\mathbf{h}}^{ext}$
  $t_{max} \leftarrow \max(\Omega), \widehat{T}_0 \leftarrow \min(\Omega)$
end for
$\widehat{\mathbf{h}} \leftarrow \widehat{\mathbf{h}}^{ext}_{\widehat{T}_0 : \widehat{T}_0 + L - 1}$ // small magnitude elements are considered as 0

Output: $\widehat{T}_0$, $\widehat{\mathbf{h}}$

4.2.2 Reweighted ℓ_1-regularized Optimization

The second approach proposed to solve the reconstruction problem follows a strategy based on convex relaxation where the problem is replaced by a ℓ_1 minimization problem, allowing the use of convex optimization techniques. These techniques are usually applied to real valued models, which means Equation 4.2 has to be rewritten as:

$$\widehat{\mathbf{y}} = \widehat{\mathbf{Z}}\widehat{\mathbf{h}}^{ext} + \widehat{\mathbf{n}} \tag{4.7}$$

with:

$$\widehat{\mathbf{Z}} = \begin{bmatrix} \mathrm{Re}\{\mathbf{Z}\} & -\mathrm{Im}\{\mathbf{Z}\} \\ \mathrm{Im}\{\mathbf{Z}\} & \mathrm{Re}\{\mathbf{Z}\} \end{bmatrix} \tag{4.8}$$

$$\widehat{\mathbf{y}} = \begin{bmatrix} \mathrm{Re}\{\mathbf{y}\} \\ \mathrm{Im}\{\mathbf{y}\} \end{bmatrix} \tag{4.9}$$

$$\widehat{\mathbf{h}}^{ext} = \begin{bmatrix} \mathrm{Re}\{\mathbf{h}^{ext}\} \\ \mathrm{Im}\{\mathbf{h}^{ext}\} \end{bmatrix} \tag{4.10}$$

$$\widehat{\mathbf{n}} = \begin{bmatrix} \text{Re}\{\mathbf{n}\} \\ \text{Im}\{\mathbf{n}\} \end{bmatrix} \tag{4.11}$$

Using the representation expressed in Equation 4.7, it is possible to estimate $\widehat{\mathbf{h}}^{ext}$ by minimizing the Euclidean distance and adding an ℓ_1-norm weighted regularization term:

$$\min_{\mathbf{h}_{ext}} \ \frac{1}{2} \left\| \widehat{\mathbf{y}} - \widehat{\mathbf{Z}}\widehat{\mathbf{h}}^{ext} \right\|_2^2 + \lambda \left\| \mathbf{W}\widehat{\mathbf{h}}^{ext} \right\|_1 \tag{4.12}$$

where λ symbolizes a positive penalizing parameter and $\mathbf{W} = \text{diag}([w_0 \ \ldots \ w_{2N-1}])$ represents a weighting matrix. By adopting an approach similar to basis pursuit denoising (BPDN), as seen in [73], Equation 4.12 can be rewritten as follows:

$$\min_{\mathbf{u},\,\mathbf{v}} \ \frac{1}{2} \left\| \widehat{\mathbf{y}} - \begin{bmatrix} \widehat{\mathbf{Z}} & -\widehat{\mathbf{Z}} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} \right\|_2^2 + \lambda \begin{bmatrix} \mathbf{W} & 0 \\ 0 & \mathbf{W} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} \tag{4.13}$$

$$\text{subject to } \ \mathbf{u} \geq 0, \mathbf{v} \geq 0 \tag{4.14}$$

which is a convex quadratic program. In this formulation:

$$\widehat{\mathbf{h}}^{ext} = \mathbf{u} - \mathbf{v} \tag{4.15}$$

$$\left| \widehat{h}_i^{ext} \right| = u_i + v_i, \ \ i = 0, ..., 2N-1 \tag{4.16}$$

Following an idea similar to the one shown in [61], the use of a reweighting procedure is proposed, with Algorithm 4 illustrating its steps. It consists in solving the quadratic problem presented in Equations 4.13 and 4.14 Q times by using weights that are defined according to the solution of the previous iteration, which can be seen in line 5. The solution of the quadratic problem stated in line 4 can be obtained by utilizing an interior-point method, seen in [74], [75] and [76]. Lines 9-14 are used to force the solution to be feasible, according to the maximum channel length constraint seen in Equation 4.6. If these lines are not applied, the algorithm will just generate a generic sparse solution. If only one iteration is employed ($Q = 1$) then $\mathbf{W}$ is the identity matrix and the approach becomes closely related to BPDN. As for the penalizing parameter, λ, [73] suggests the

following choice:

$$\lambda = \sigma\sqrt{2\log(N_s)} \tag{4.17}$$

Algorithm 4 Constrained Length ℓ_1-Regularized Optimization

Input: $\widehat{\mathbf{y}}$, $\widehat{\mathbf{Z}}$, L, N_s, ε, Q

1: $\widehat{\mathbf{h}}^{ext} = 0$, $\Omega = \emptyset$, $\widehat{T}_0 = N - 1$, $t_{max} = 0$
2: $w_i = 1, \;\; i = 0, ..., 2N_s - 1$
3: **for** $q = 0, ..., Q - 1$ **do**
4: Solve the quadratic problem (Equations 4.13 and 4.14) and obtain $\mathbf{u}$, $\mathbf{v}$
5: $w_i = \frac{1}{u_i + v_i + \varepsilon}, \;\; i = 0, ..., 2N_s - 1$
6: **end for**
7: $\widehat{\mathbf{h}}^{ext} \leftarrow \mathbf{u}_{0:N_s-1} - \mathbf{v}_{0:N_s-1} + i\left(\mathbf{u}_{N_s:2N_s-1} - \mathbf{v}_{N_s:2N_s-1}\right)$
8: $\Lambda \leftarrow \text{supp}\left(\widehat{\mathbf{h}}^{ext}\right)$ // small magnitude elements are considered as 0 and supp($\mathbf{x}$) returns the set of indices of nonzero elements in $\mathbf{x}$
9: **while** $\Lambda \neq \emptyset$ **and** $t_{max} - \widehat{T}_0 < L - 1$
10: $t_{best} \leftarrow \arg\max_{t \in \Lambda} \left|\widehat{h}_t^{ext}\right|$
11: **if** $t_{max} - L + 1 \leq t_{best} \leq L - 1 + \widehat{T}_0$ **do** $\Omega \leftarrow \Omega \cup \{t_{best}\}$
12: $\Lambda \leftarrow \Lambda \backslash \{t_{best}\}$
13: $t_{max} \leftarrow \max(\Omega), \;\; \widehat{T}_0 \leftarrow \min(\Omega)$
14: **end while**
15: $\widehat{\mathbf{h}} \leftarrow \widehat{\mathbf{h}}^{ext}_{\widehat{T}_0:\widehat{T}_0+L-1}$ // small magnitude elements are considered as 0

Output: $\widehat{T}_0$, $\widehat{\mathbf{h}}$

4.2.3 ℓ_2-regularized Optimization

While not so often used for obtaining sparse solutions, the reconstruction problem can also be addressed through ℓ_2-norm regularization, as explained in [73] and [76]. It can be expressed as follows:

$$\min_{\mathbf{h}_{ext}} \;\; \frac{1}{2}\left\|\mathbf{y} - \mathbf{Z}\mathbf{h}^{ext}\right\|_2^2 + \lambda\left\|\mathbf{h}^{ext}\right\|_2^2 \tag{4.18}$$

In this case, it is easy to obtain a closed form solution, expressed in the next Equation:

$$\widehat{\mathbf{h}}^{ext} = \left(\mathbf{Z}^H\mathbf{Z} + \lambda\mathbf{I}_{N_s}\right)^{-1}\mathbf{Z}^H\mathbf{y} \tag{4.19}$$

Regarding the penalizing parameter, λ, a similar value to the one used in BPDN can be adopted, as seen in Equation 4.17. In order to constrain the final estimate according to Equation 4.6, the procedure matching lines 8 to 15 of Algorithm 4 can be directly applied.

4.3 Analysis of the Results

In this section, the performance of the proposed sparse reconstruction based approaches is evaluated for joint timing synchronization and channel estimation using TCH codes by performing Monte Carlo simulations with an 8 equal power tap Rayleigh fading channel. The training sequences use Zadoff-Chu sequences and also pseudorandom codewords in order to compare the performance of TCH codes against these two different sequence types.

The analysis is divided into four parts: firstly results that compare the effect of using different codeword lengths, N_c, are presented. Next, the comparison of using different channel lengths, L, is performed. Thirdly, the effects of using an observation window that is equal to or smaller than the simulation window, $M_o \leq N_s$, is shown. Finally, the effects of using cardinality and also what happens if polishing is not utilized will be observed.

For comparison purposes with the proposed sparse reconstruction techniques, the matched filter (MF) and the Oracle Least Squares approaches will also be used. MF corresponds to the use of a conventional direct correlation approach for estimating the tap positions and respective coefficients while the latter represents an ideal case where the exact tap positions are known (i.e. perfect synchronization is assumed) and simple LS channel estimation is applied.

4.3.1 Effects of Different Codeword Lengths

In this subsection, the considered training sequences have lengths of $N_c = 16$ and $N_c = 256$. The results for $N_c = 64$ are presented in Appendix B. A search window of $N_s = 128$ samples was used for $N_c = 16$, while a length of $N_s = 512$ samples was adopted when $N_c = 256$. The maximum expected channel length is $L = 10$. The observation window, M_o, employed is considered to be the best case scenario, with $M_o = N_s + N_c$.

Figures 4.1 to 4.3 illustrate the channel estimation MSE performance for various sparse reconstruction methods using TCH codes, Zadoff-Chu sequences and pseudorandom codewords, respectively. By observing these Figures, it is clear that MF and ℓ_2-regularized have the weakest performance of all methods employed, for both pilot sizes. As expected, simulations with $N_c = 256$ present much better results than the ones where $N_c = 16$.

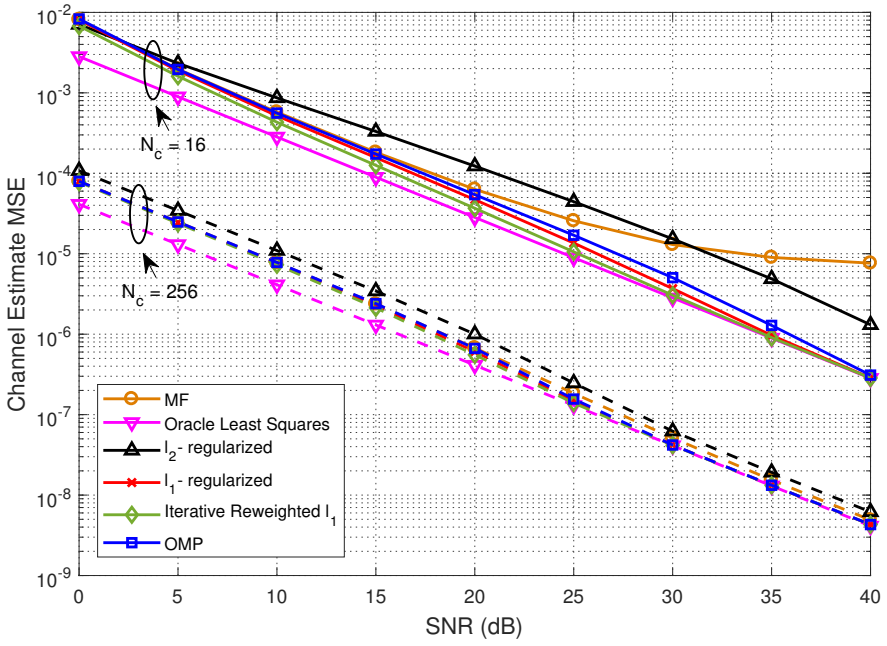

FIGURE 4.1: MSE performance for different joint timing synchronization and channel estimation methods, considering TCH codes and different pilot size values.

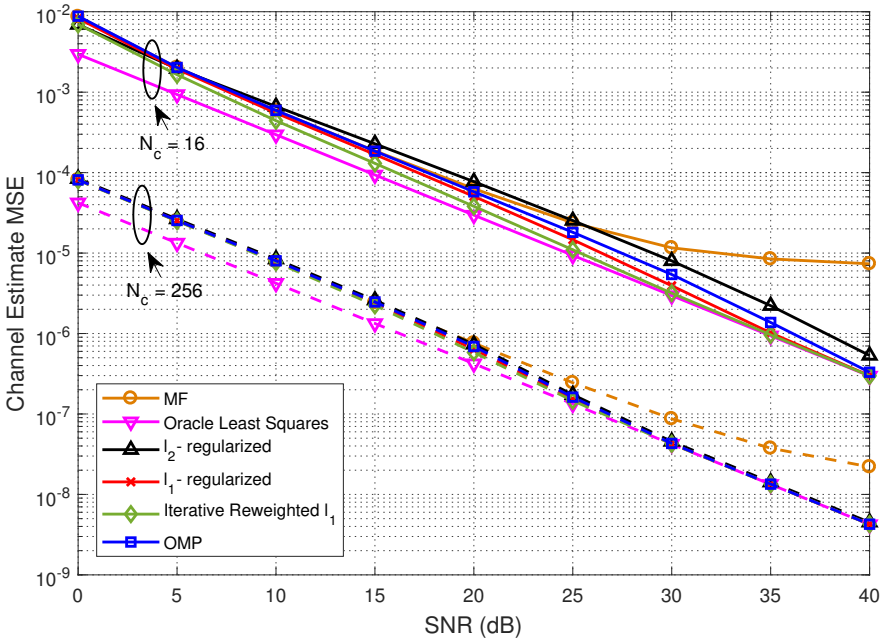

FIGURE 4.2: MSE performance for different joint timing synchronization and channel estimation methods, considering Zadoff-Chu sequences and different pilot size values.

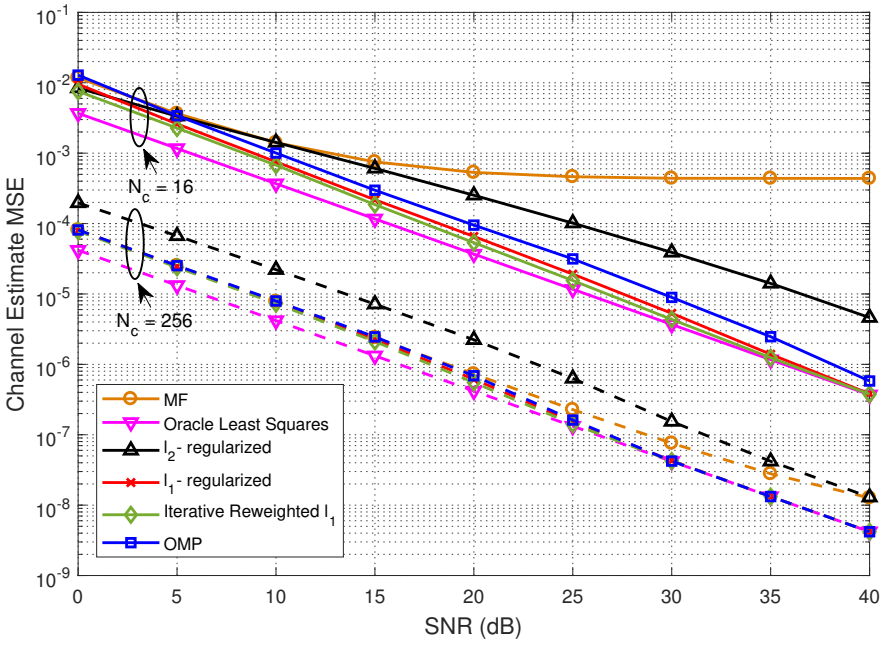

FIGURE 4.3: MSE performance for different joint timing synchronization and channel estimation methods, considering pseudorandom codewords and different pilot size values.

In Figure 4.1, it is observable that, for the most part of this simulation, ℓ_2-regularized has the worst performance for both pilot sizes. The three sparse based techniques have very similar performance, with a small loss for OMP when $N_c = 16$. Iterative Reweighted ℓ_1 achieves slightly better results than its counterparts and for higher values of SNR, performances of sparse based approaches are identical to the ideal Oracle Least Squares.

As for Figure 4.2, MF possesses the overall worst performance. When using Zadoff-Chu sequences, ℓ_2-regularized can keep up with the performance of OMP, ℓ_1-regularized and Iterative Reweighted ℓ_1, which can be observed in higher SNR values for $N_c = 256$.

Regarding Figure 4.3, MF has, by far, the weakest performance when $N_c = 16$, presenting a floor at around 4×10^{-4} and ℓ_2-regularized has the worst results for $N_c = 256$. Nonetheless, the performance of the three sparse based approaches is almost optimal.

Comparing these three Figures, all codes have a similar performance, with Zadoff-Chu sequences achieving slightly better results for a pilot size of 16 and TCH codes showing that they have the best performance for a pilot size of 256, with the worst methods, MF and ℓ_2-regularized, attaining similar results to the overall best method, Iterative Reweighted ℓ_1.

Figures 4.4 to 4.6 analyze the synchronization performance by measuring the total error rate detection rate of the same approaches that have been previously used. This error is measured as the ratio of the number of incorrect tap positions divided by N_s. MF is once again outperformed by the other methods and its results are far from being good since the graphs illustrate high irreducible floors for every tested scenario.

Figure 4.4 depicts that, apart from MF, ℓ_2-regularized is the worst performing method for both pilot sizes. This happens due to the fact that ℓ_2-norm based reconstruction tends to produce solutions that are not rigorously sparse. Iterative Reweighted ℓ_1 achieved once again the best performance.

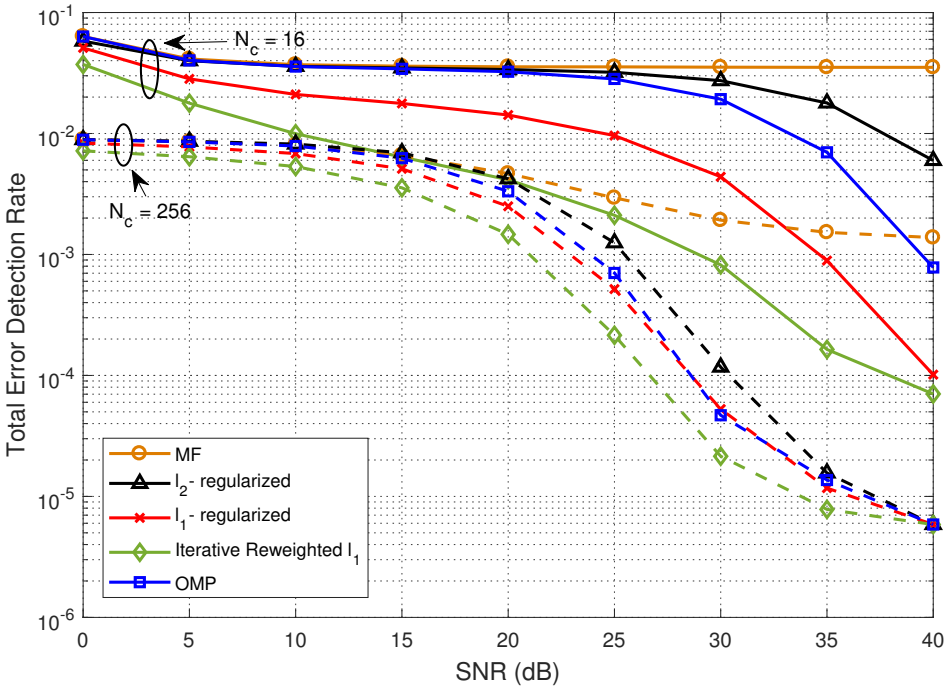

FIGURE 4.4: Probability of incorrect multipath positions (timing) for different approaches, considering TCH codes and different pilot size values.

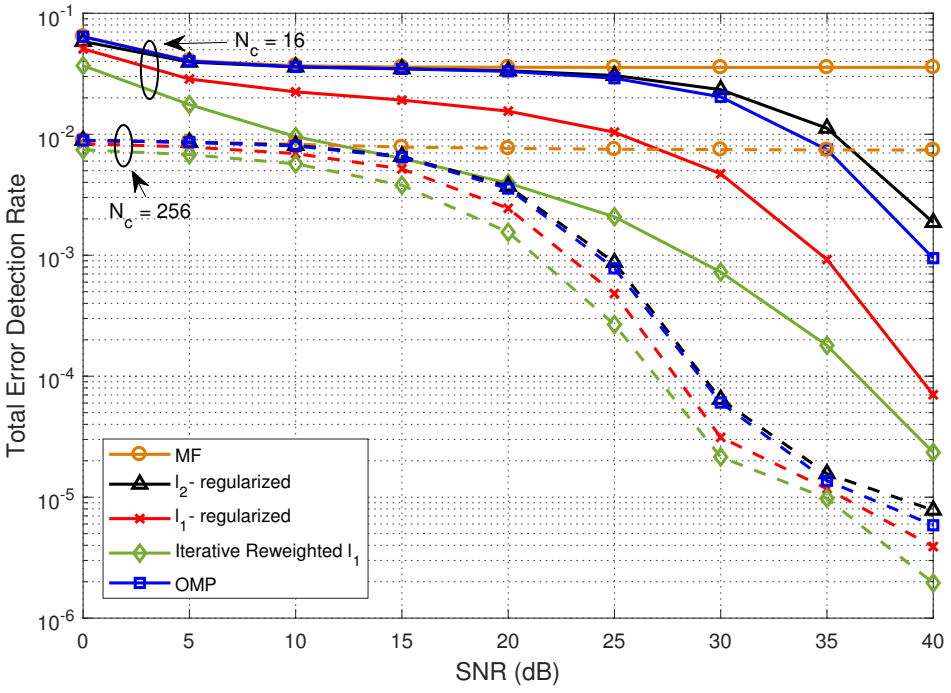

FIGURE 4.5: Probability of incorrect multipath positions (timing) for different approaches, considering Zadoff-Chu sequences and different pilot size values.

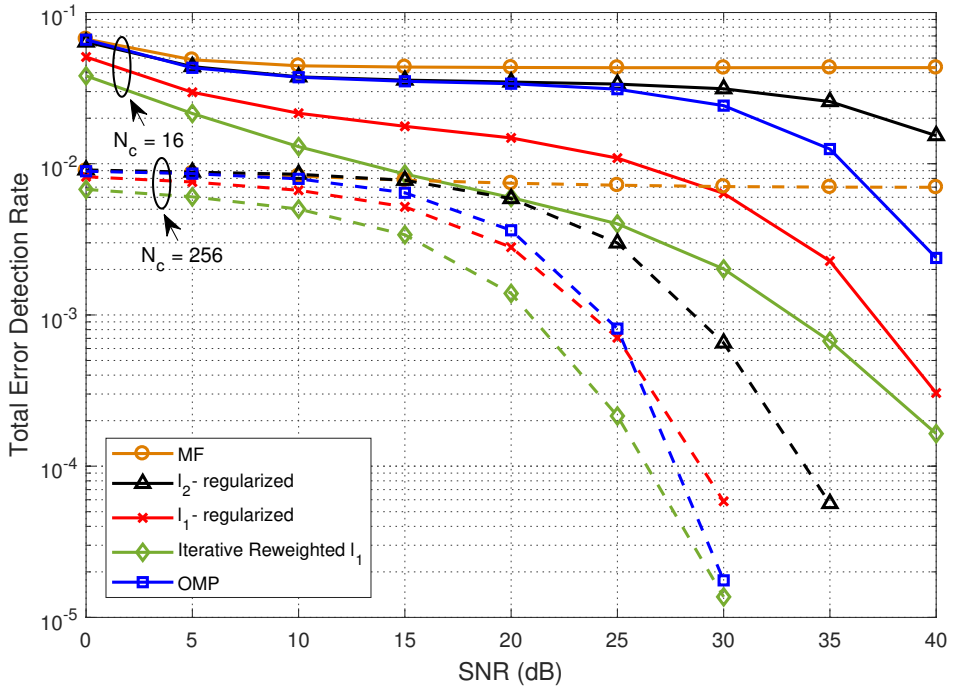

FIGURE 4.6: Probability of incorrect multipath positions (timing) for different approaches, considering pseudorandom codewords and different pilot size values.

It is observable in Figure 4.5 that ℓ_2-regularized and OMP have similar performances for both pilot sizes and their results are the worst, excluding MF. Nonetheless, every sparse based approach presents good results for $N_c = 256$, with ℓ_1-regularized only being outperformed by Iterative Reweighted ℓ_1.

Figure 4.6 has similar results to the previous Figures. The biggest difference is that OMP starts to have better performance than ℓ_1-regularized at around 25 dB. Yet again, ℓ_2-regularized joins MF with the worst performance while Iterative Reweighted ℓ_1 has the best one.

Every pilot type has similar results, with pseudorandom codewords having the slightly lowest results of them all. Zadoff-Chu sequences manage to outperform TCH codes but only by a small difference. The graphs from this subsection showed that TCH codes can successfully be used to perform joint channel estimation and synchronization achieving similar results than the other methods and even being able to outperform them in some cases.

4.3.2 Effects of Different Channel Lengths

The results presented in this subsection illustrate that performance is affected by varying the maximum expected channel length, L. As seen in the previous subsection, $N_c = 256$ has the best performance, as expected, so the next graphs only contain results for when $N_c = 256$, which means the search window is $N_s = 512$. Results for $N_c = 16$ and $N_c = 64$ are contained in Appendix B. The same observation window used in subsection 4.3.2 is employed.

Figures 4.7 to 4.9 compare the effect of maximum expected channel length of 5 and 20 for MSE channel estimation performance of the different methods. $L = 5$ curves are illustrated as straight lines while $L = 20$ curves are represented by dashed lines.

As expected, for lower values of maximum expected channel length, the performance is better. This happens because there are less replicas of the channel when L is lower, making channel estimation and synchronization easier to perform.

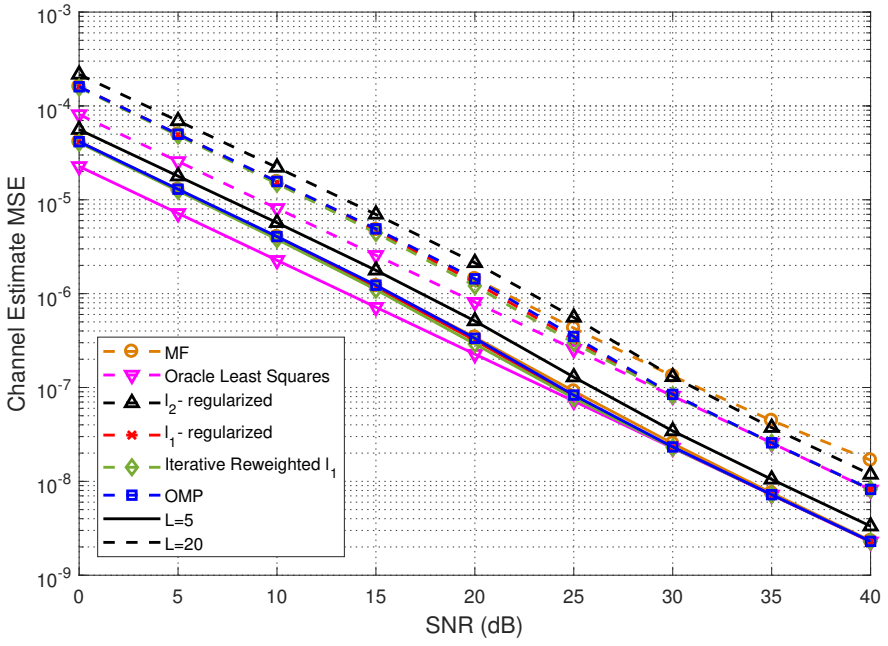

FIGURE 4.7: MSE performance for different joint timing synchronization and channel estimation methods, considering TCH codes and different channel length values.

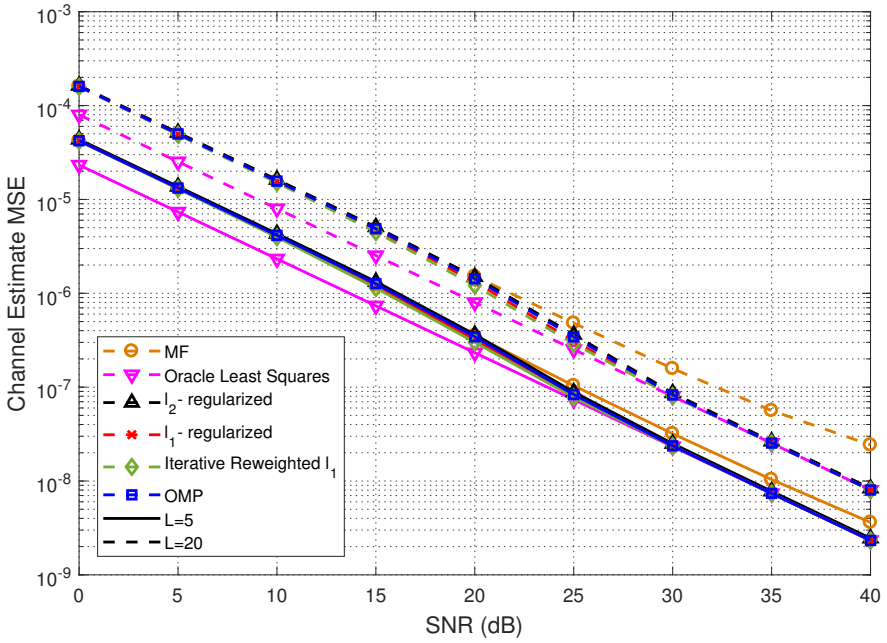

FIGURE 4.8: MSE performance for different joint timing synchronization and channel estimation methods, considering Zadoff-Chu sequences and different channel length values.

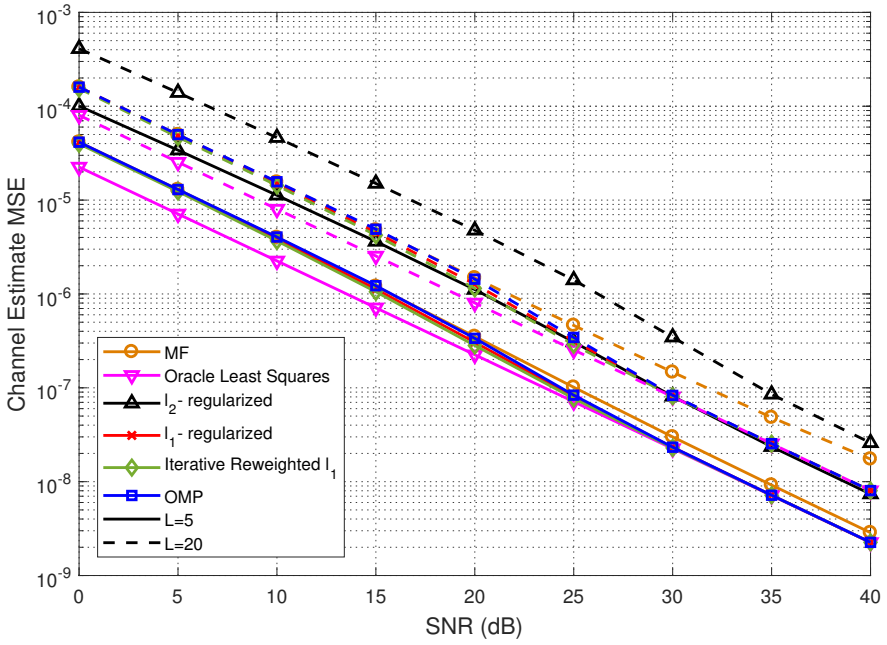

FIGURE 4.9: MSE performance for different joint timing synchronization and channel estimation methods, considering pseudorandom codewords and different channel length values.

In general, MF and ℓ_2-regularized have the worst performance but they are still very similar to the other three methods for lower SNR values. OMP, ℓ_1-regularized and Iterative Reweighted ℓ_1 have almost identical results and, starting at 25 dB, their results are almost the same as the ideal Oracle Least Squares. There is little difference between the types of pilots, with Zadoff-Chu sequences having a marginal advantage in terms of performance over the other two.

The next graphs, Figures 4.10 to 4.12 present the results regarding synchronization. Similarly to the results exposed in Figures 4.4 to 4.6, MF has the worst results, exhibiting high irreducible error floors in every figure except the one referring to TCH codes.

Iterative Reweighted ℓ_1 is, yet again, the overall best performing method, with ℓ_1-regularized and OMP having slightly weakest performances, easily observable in the graph regarding pseudorandom codes, which is weaker than TCH codes and Zadoff-Chu sequences. The latter has a slightly better performance than TCH codes, showing that Zadoff-Chu sequences have a small advantage over TCH codes regarding synchronization results.

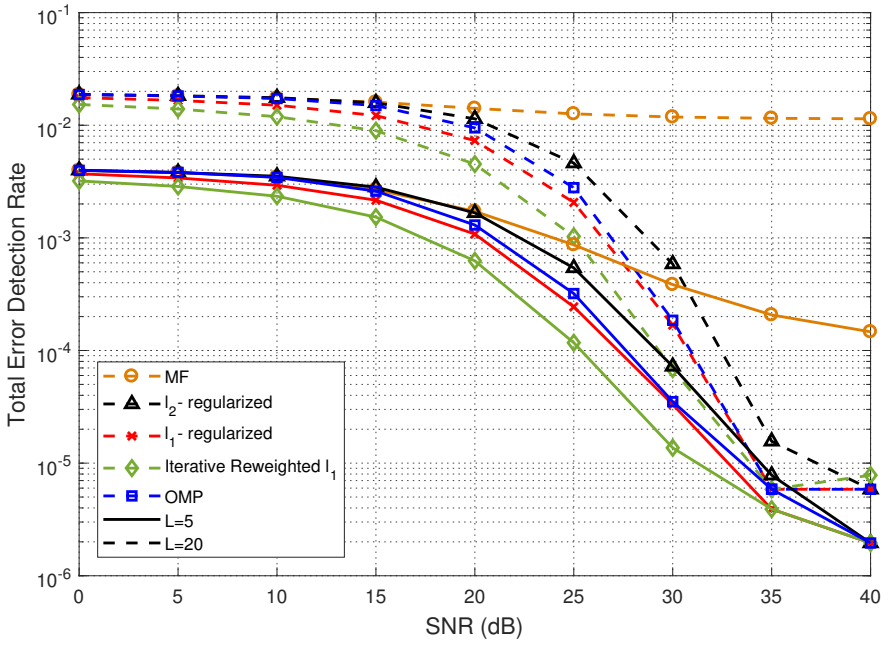

FIGURE 4.10: Probability of incorrect multipath positions (timing) for different approaches, considering TCH codes and different channel length values.

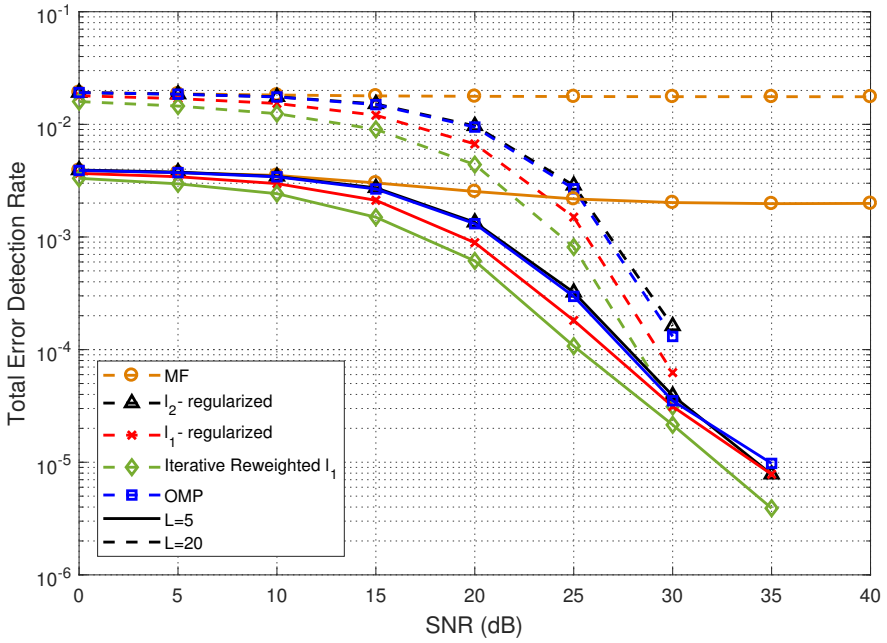

FIGURE 4.11: Probability of incorrect multipath positions (timing) for different approaches, considering Zadoff-Chu sequences and different channel length values.

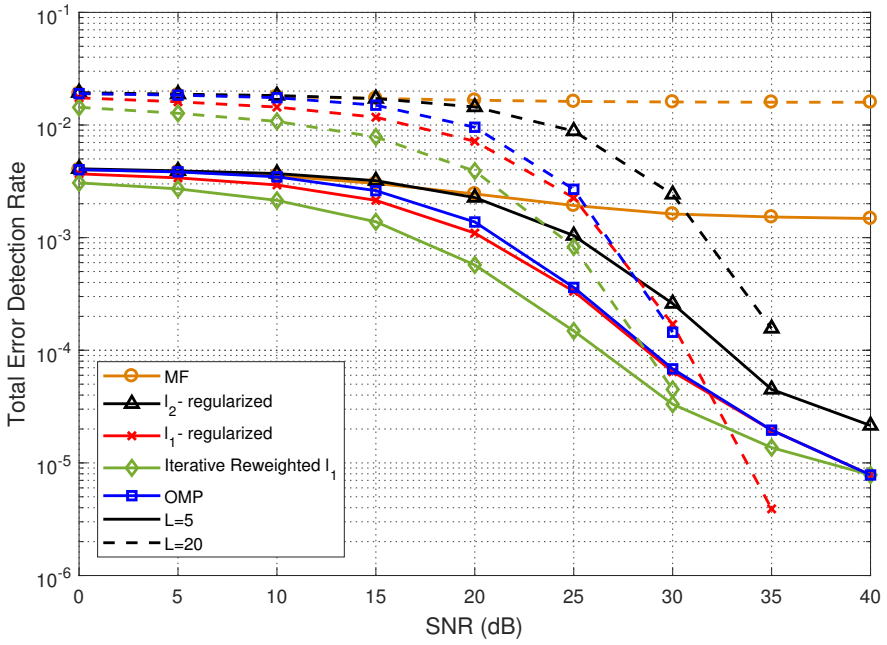

FIGURE 4.12: Probability of incorrect multipath positions (timing) for different approaches, considering pseudorandom codewords and different channel length values.

Performance is better when $L = 5$ and that is noticeable for lower values of SNR. However, when these values are higher, the performance of $L = 20$ starts to get much closer to the performance of $L = 5$, meaning it is viable to use the best performing methods and any of these codes even when $L = 20$, which was the maximum value tested in this dissertation.

4.3.3 Effects of Different Observation Window Sizes

In previous subsections an observation window of $M_o = N_s + N_c$ was used, which is considered to be the best case scenario. The following results show the effect of reducing the observation window, M_o, in order to understand how performance is affected. Since simulations for $N_c = 256$ are computationally very heavy due to the size of the pilots and the size of the search window, the simulations were performed only for a pilot size of 64, which means the search window has a size of 128. Only one channel length value, $L = 10$, was considered.

Figures 4.13 to 4.15 illustrate MSE channel estimation performance of the different methods, for TCH codes, Zadoff-Chu sequences and pseudorandom codewords, respectively. For these three figures, an observation window that has the same size as the search window, $M_o = N_s$, was employed.

As expected, results show that the curves are no longer close to the Oracle Least Squares curve. In any of the three graphs, MF has the worst performance by far while the ℓ_1-regularized and Iterative Reweighted ℓ_1 approaches have the overall best performance. In Figure 4.13, OMP and ℓ_2-regularized have similar performances, slightly worse than the other two sparse based approaches. Figure 4.14 shows that OMP has a very similar performance to the best two approaches and Figure 4.15 illustrates that ℓ_2-regularized has an advantage over OMP. Comparing these three figures, it is possible to observe that TCH codes managed to attain better results than its counterparts.

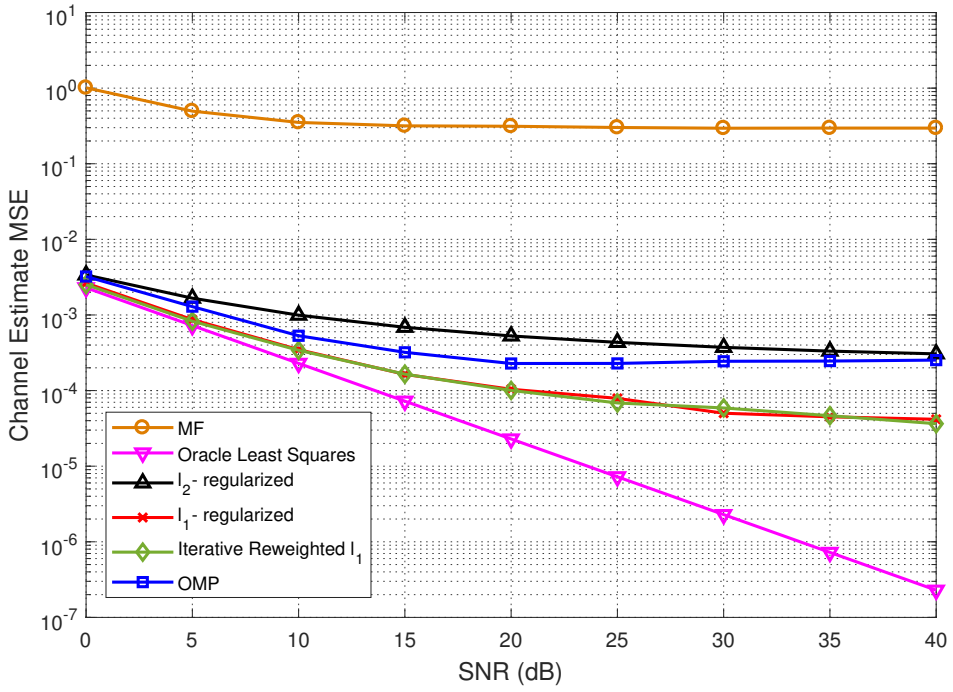

FIGURE 4.13: MSE performance for different joint timing synchronization and channel estimation methods, considering TCH codes and $M_o = N_s$.

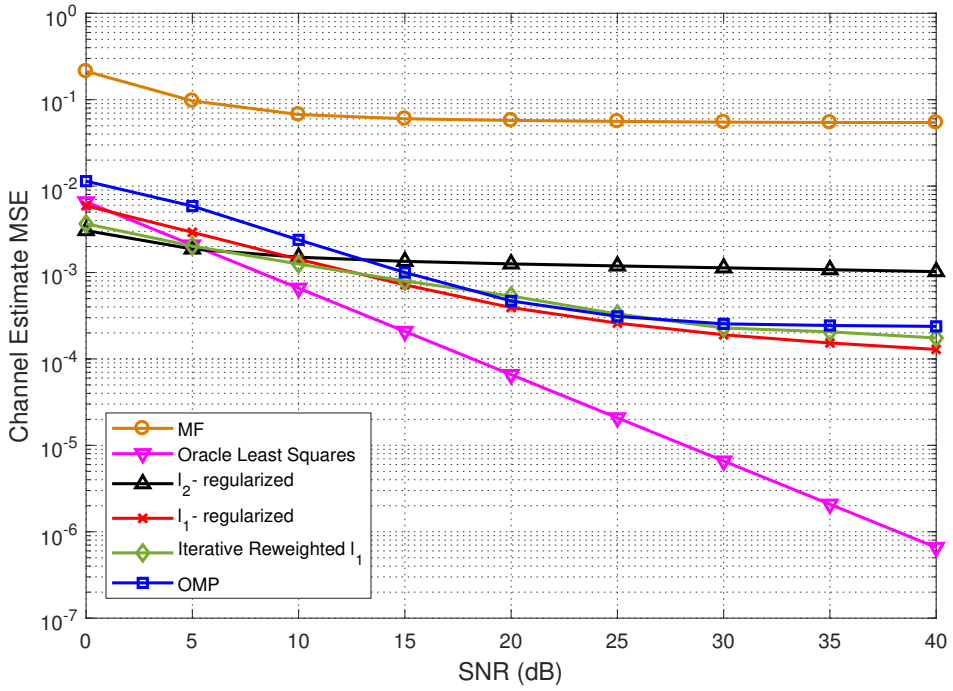

FIGURE 4.14: MSE performance for different joint timing synchronization and channel estimation methods, considering Zadoff-Chu sequences and $M_o = N_s$.

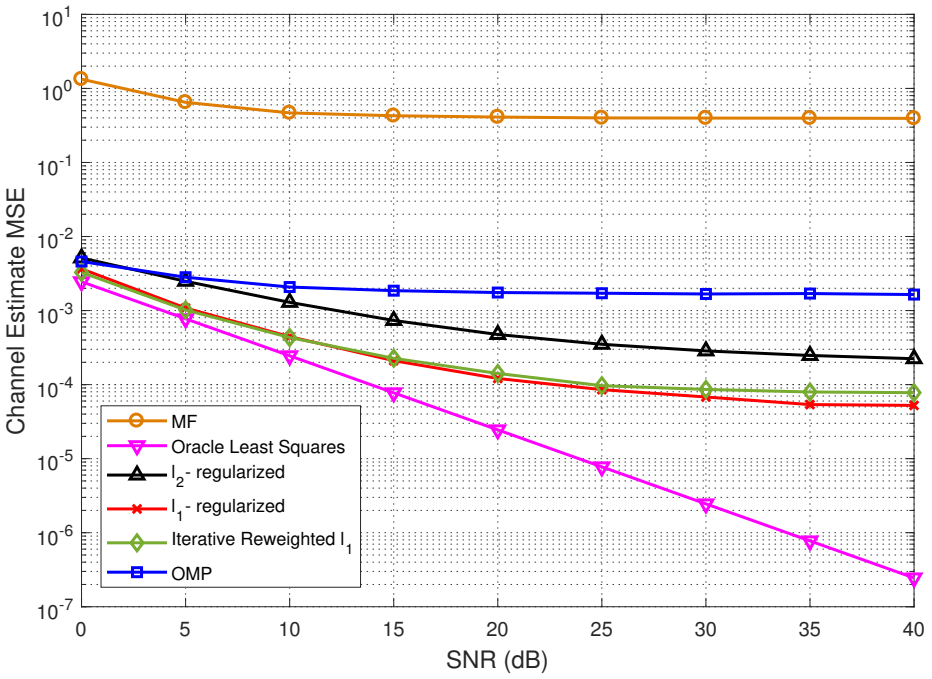

FIGURE 4.15: MSE performance for different joint timing synchronization and channel estimation methods, considering pseudorandom codewords and $M_o = N_s$.

Figures 4.16 to 4.18 analyze the synchronization performance just like in previous subsections. The observation window utilized is also equal to the search window.

MF is, once again, the worst performing method presenting an error floor in every graph. In Figures 4.16 and 4.17, OMP has a better performance than ℓ_2-regularized for higher SNR's while the opposite happens in Figure 4.18. Iterative Reweighted ℓ_1 has the best performance, with TCH codes getting the best results, around 5×10^{-4}, while Zadoff-Chu sequences and pseudorandom codewords only attained 4×10^{-3} and 10^{-3}, respectively.

Despite the obvious reduction in quality on performance for both channel estimation and synchronization, it is still possible to obtain some good results when using the sturdier approaches, namely Iterative Reweighted ℓ_1 and ℓ_1-regularized, that managed to be less affected by the reduction of the observation window. TCH codes seem to be more viable than the other two methods.

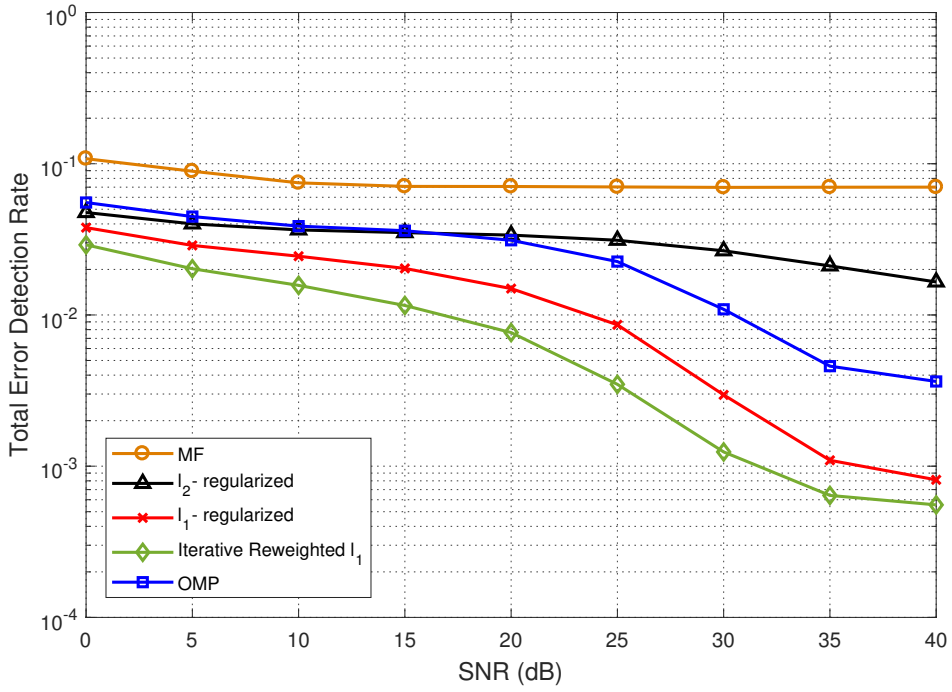

FIGURE 4.16: Probability of incorrect multipath positions (timing) for different approaches, considering TCH codes and $M_o = N_s$.

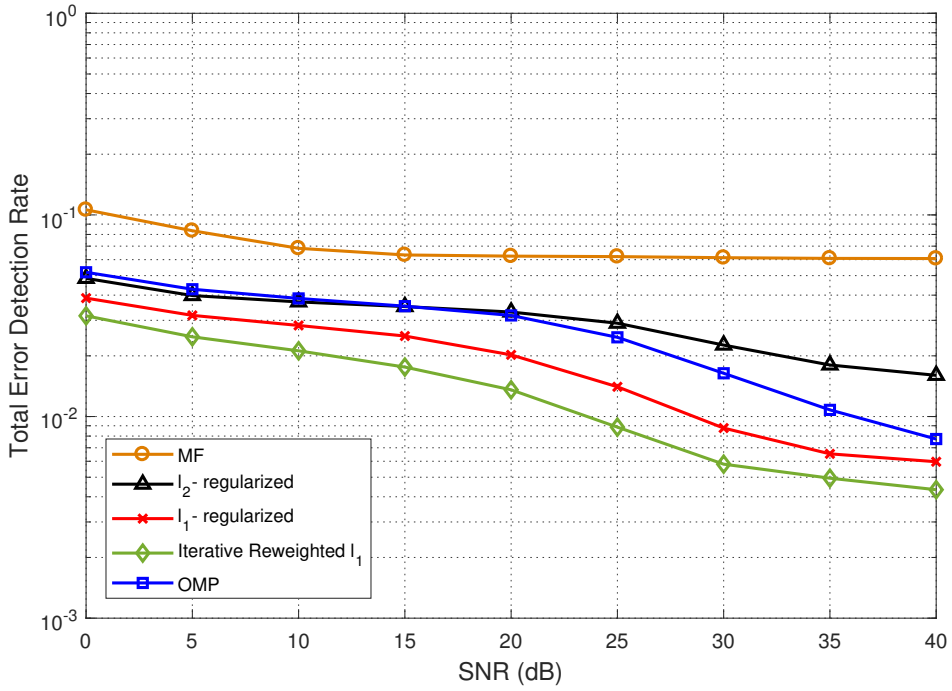

FIGURE 4.17: Probability of incorrect multipath positions (timing) for different approaches, considering Zadoff-Chu sequences and $M_o = N_s$.

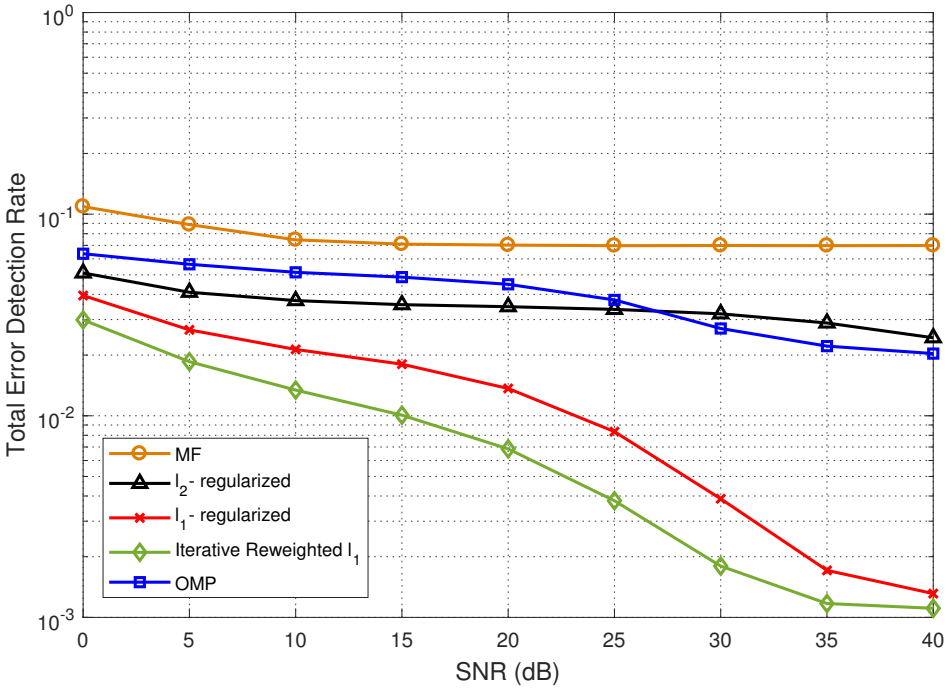

FIGURE 4.18: Probability of incorrect multipath positions (timing) for different approaches, considering pseudorandom codewords and $M_o = N_s$.

In Figures 4.19 to 4.24, the observation window has been reduced even further, now being $M_o = 3N_s/4$, which means that the observation window is smaller than the search window. In order to obtain acceptable results, a cyclic prefix to attenuate the effect of decreasing the observation window is now used, with its length being half of the pilot size. The next three graphs depict the results of MSE channel estimation performance.

Despite using an observation window smaller than the search window, the results are similar to when both windows had the same length. This happens because with was utilized a cyclic prefix or else the results would have been much worse. Comparing Figure 4.19 to Figure 4.13, OMP and ℓ_2-regularized have now a worse performance, while the other three methods had a positive increase in their results. OMP is now identical ℓ_2-regularized while ℓ_1-regularized's curve is almost equal to the one belonging to Iterative Reweighted ℓ_1.

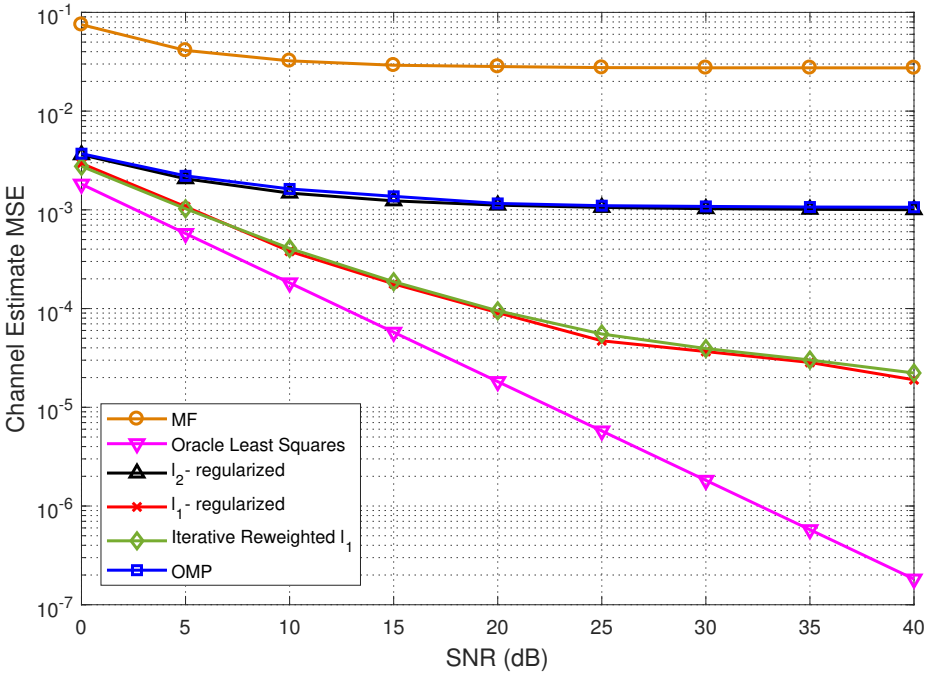

FIGURE 4.19: MSE performance for different joint timing synchronization and channel estimation methods, considering TCH codes and $M_o = 3N_s/4$.

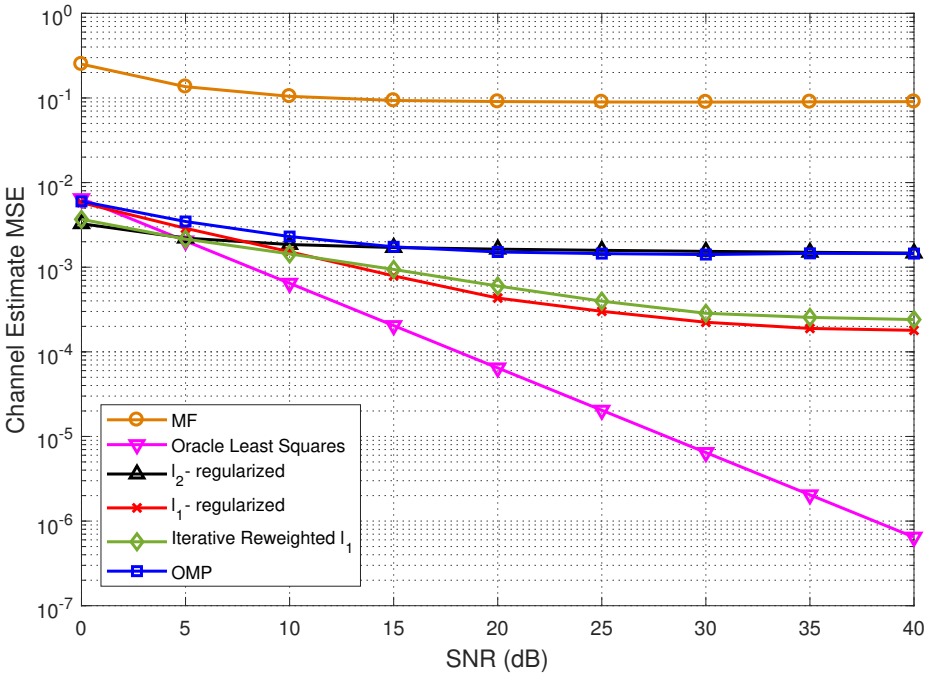

FIGURE 4.20: MSE performance for different joint timing synchronization and channel estimation methods, considering Zadoff-Chu sequences and $M_o = 3N_s/4$.

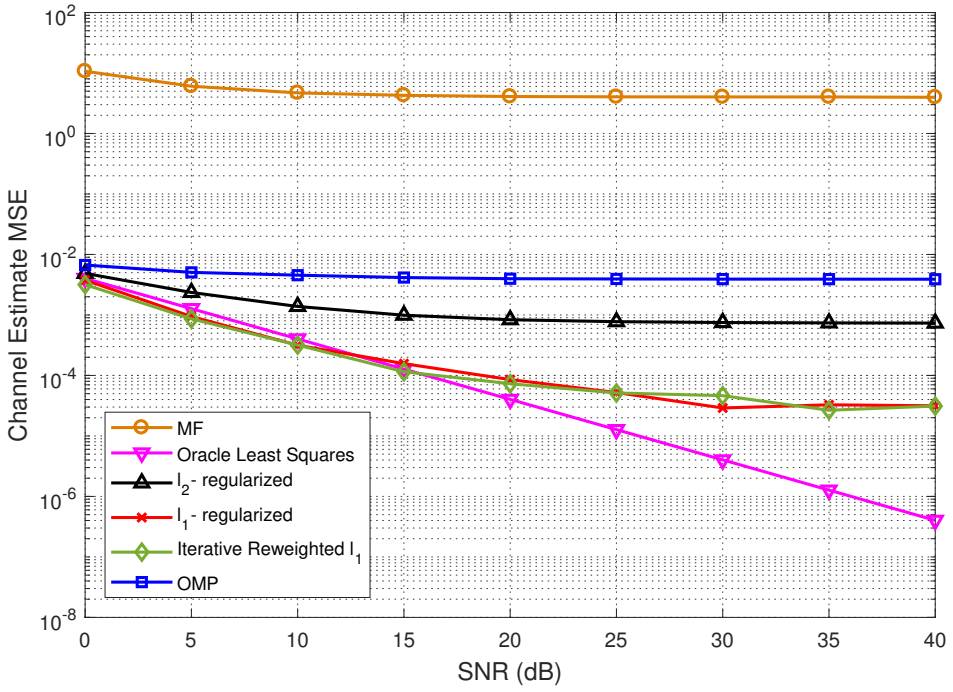

FIGURE 4.21: MSE performance for different joint timing synchronization and channel estimation methods, considering pseudorandom codewords and $M_o = 3N_s/4$.

As for figure 4.20, every method got slightly worse, with OMP showing the biggest decrease in performance out of the five methods. Just like in Figure 4.19, OMP's performance is paired with ℓ_2-regularized while ℓ_1-regularized has almost the same results as Iterative Reweighted ℓ_1.

Even though Figure 4.21 is very similar to 4.15, it is possible to observe that every method got worse results with the exception of ℓ_1-regularized and Iterative Reweighted ℓ_1. OMP is not as similar to ℓ_2-regularized as it was in other different pilot approaches.

ℓ_1-regularized and Iterative Reweighted ℓ_1 methods seem to withstand the decrease of the length of the observation window better than its counterparts and TCH codes managed, once again, to outperform slightly the other two different codes. Zadoff-Chu sequences and pseudorandom codewords showed very similar performances between them when $M_o = 3N_s/4$.

The following three figures depict the analysis of synchronization performance. MF continues to show the worst performance of all. For Figure 4.22, performance stayed the same for the best two approaches, ℓ_1-regularized and Iterative Reweighted ℓ_1. OMP has lightly weaker results than ℓ_2-regularized until around 25 dB.

Figure 4.23 has the same evolution as Figure 4.22, with ℓ_1-regularized and Iterative Reweighted ℓ_1 being able to present practically no degradation, while the other three methods got worse. OMP was also the most affected by the reduction of the observation window.

Figure 4.24 is actually showing a small improvement in ℓ_1-regularized and Iterative Reweighted ℓ_1 methods. OMP is now worse than ℓ_2-regularized during every measurement of the graph. TCH codes show a slightly better performance than the other two sequences, being able to cope better with the window reduction. Zadoff-Chu sequences, which used to be the most competitive sequences against TCH codes, are now displaying the poorest results out of the three codewords.

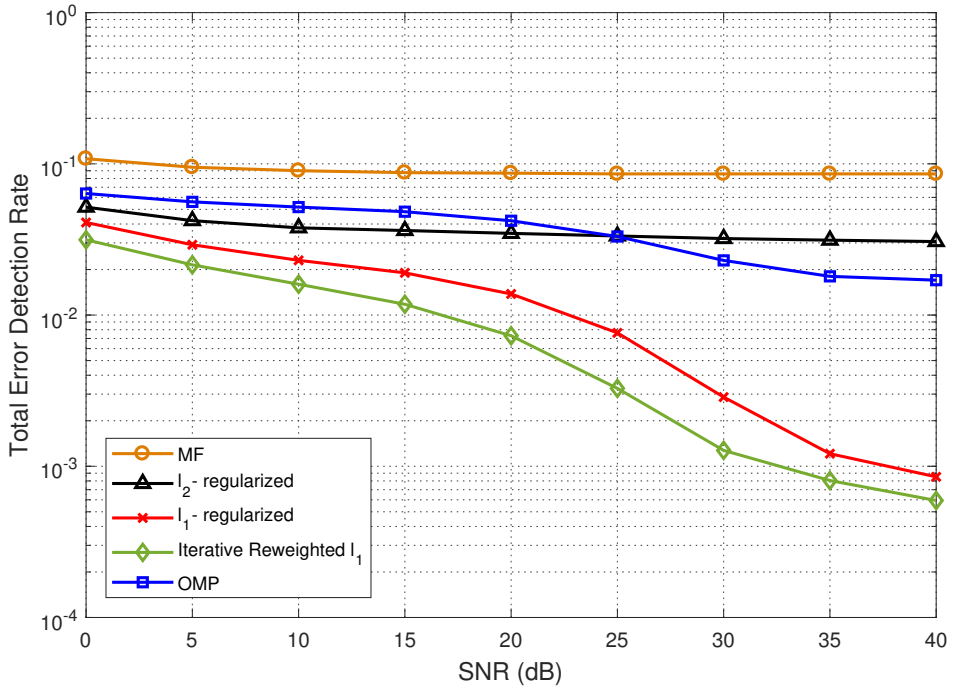

FIGURE 4.22: Probability of incorrect multipath positions (timing) for different approaches, considering TCH codes and $M_o = 3N_s/4$.

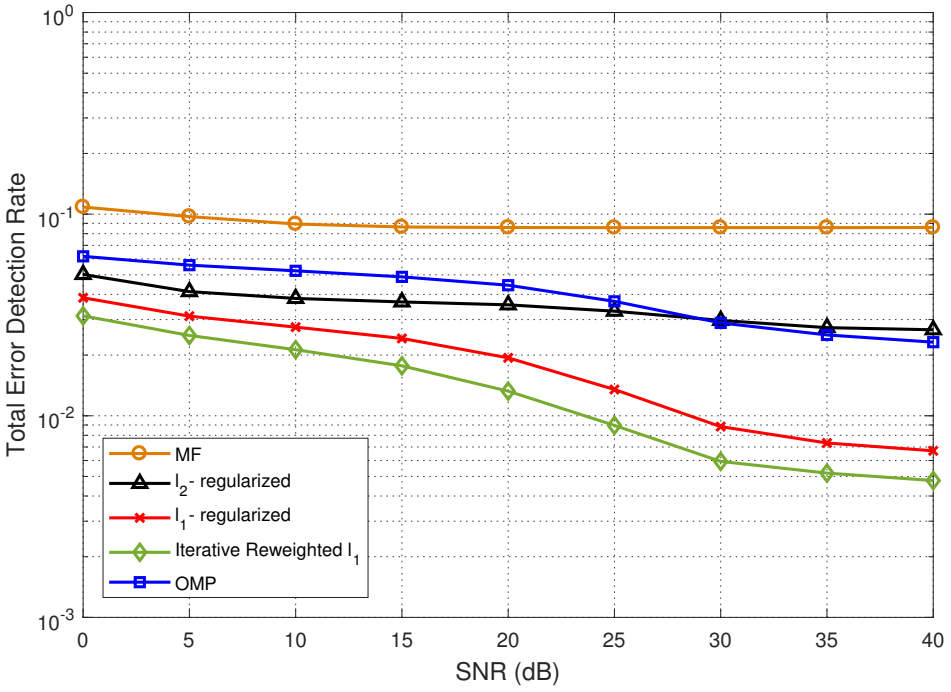

FIGURE 4.23: Probability of incorrect multipath positions (timing) for different approaches, considering Zadoff-Chu sequences and $M_o = 3N_s/4$.

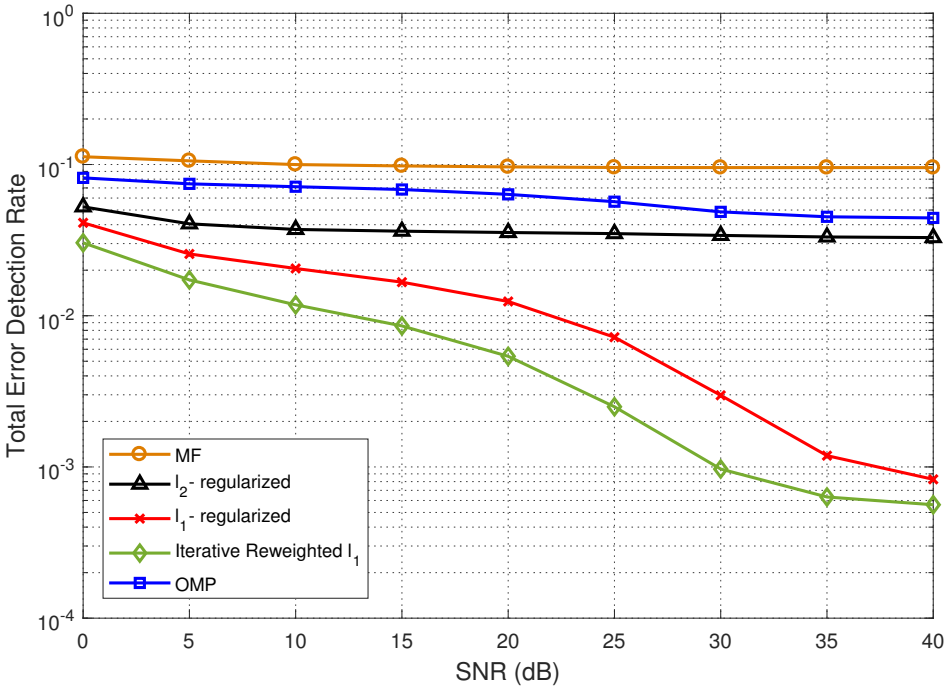

FIGURE 4.24: Probability of incorrect multipath positions (timing) for different approaches, considering pseudorandom codewords and $M_o = 3N_s/4$.

The last figures of this subsection display the results of simulations with an even smaller observation window, $M_o = 2N_s/3$.

It is observable that, in Figures 4.25 to 4.27, channel estimation has been greatly affected by using this observation window size. MF is by far the worst performing one, reaching MSE values of 10^3 when using TCH codes.

ℓ_1-regularized and Iterative Reweighted ℓ_1 still possess the best results but their curves have strong floors now and their performance is not very far from OMP and ℓ_2-regularized, which means that not even the best performing methods can withstand such a small observation window.

Figures 4.28 to 4.30 depict the synchronization results for $M_o = 2N_s/3$ and further confirm that this window size is no longer viable to utilize in order to perform joint channel estimation and synchronization. TCH codes were the pilot types that coped better with the window reduction while ℓ_1-regularized and Iterative Reweighted ℓ_1 were best sparse reconstruction methods to utilize. Further results regarding $M_o = N_s/2$ are presented in Appendix B.

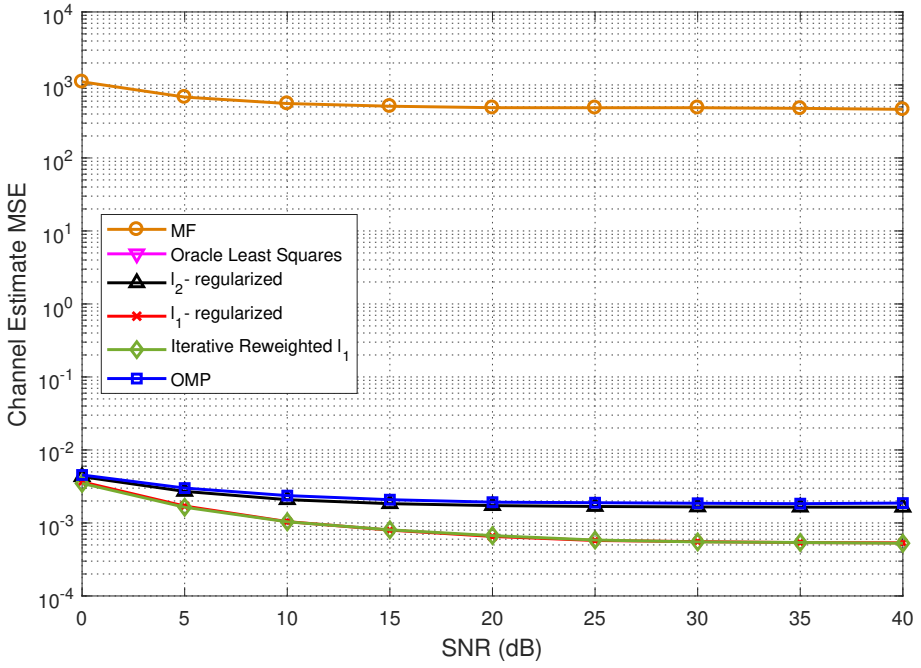

FIGURE 4.25: MSE performance for different joint timing synchronization and channel estimation methods, considering TCH codes and $M_o = 2N_s/3$.

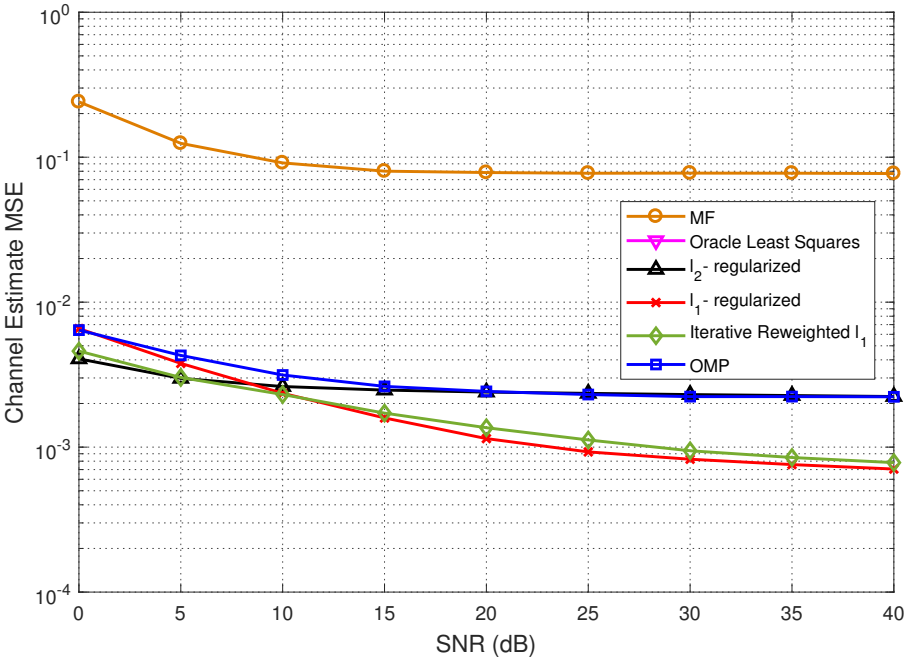

FIGURE 4.26: MSE performance for different joint timing synchronization and channel estimation methods, considering Zadoff-Chu sequences and $M_o = 2N_s/3$.

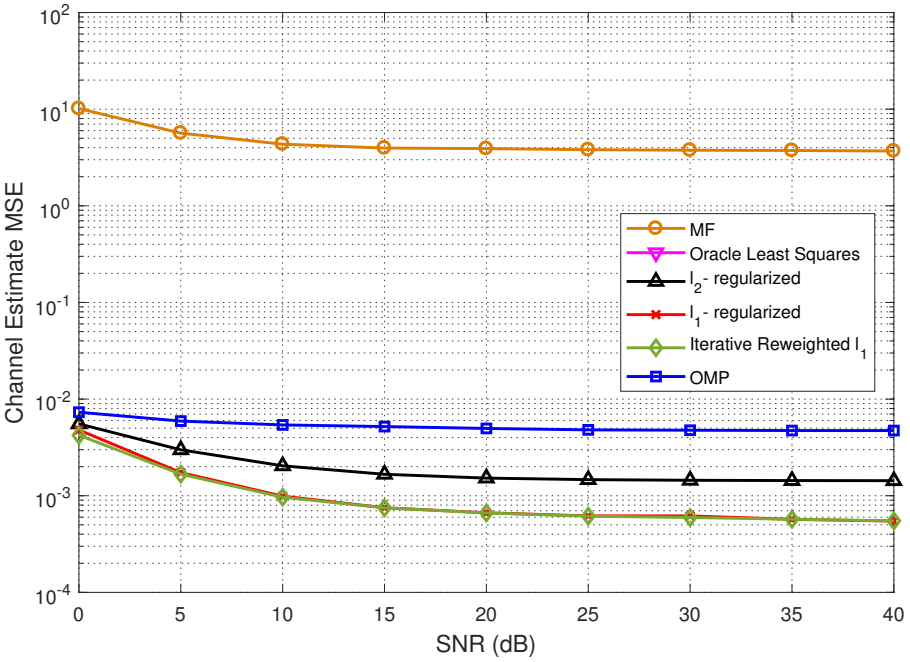

FIGURE 4.27: MSE performance for different joint timing synchronization and channel estimation methods, considering pseudorandom codewords and $M_o = 2N_s/3$.

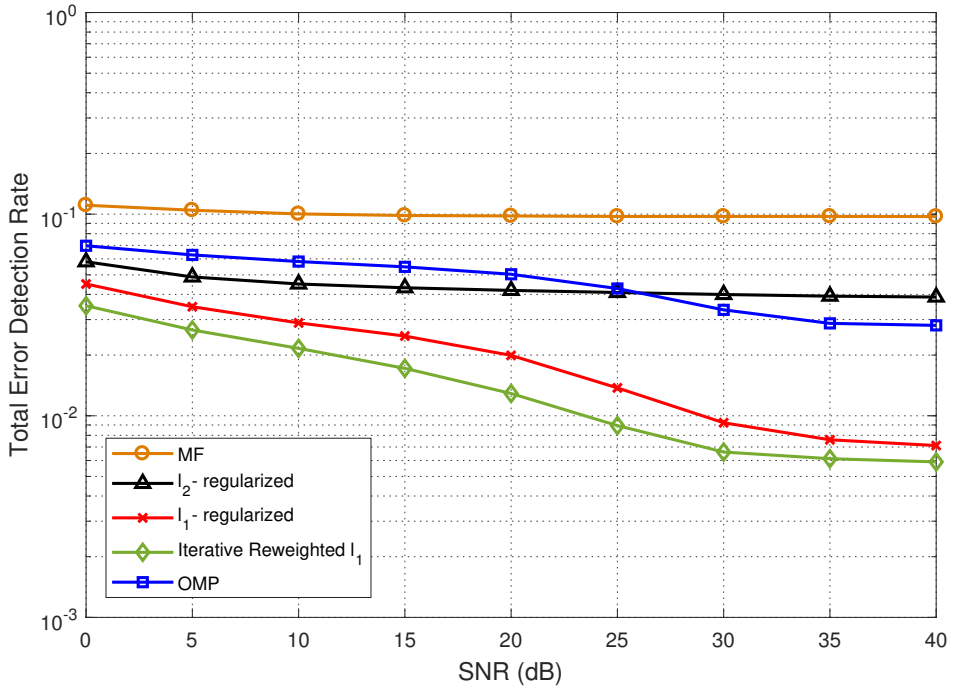

FIGURE 4.28: Probability of incorrect multipath positions (timing) for different approaches, considering TCH codes and $M_o = 2N_s/3$.

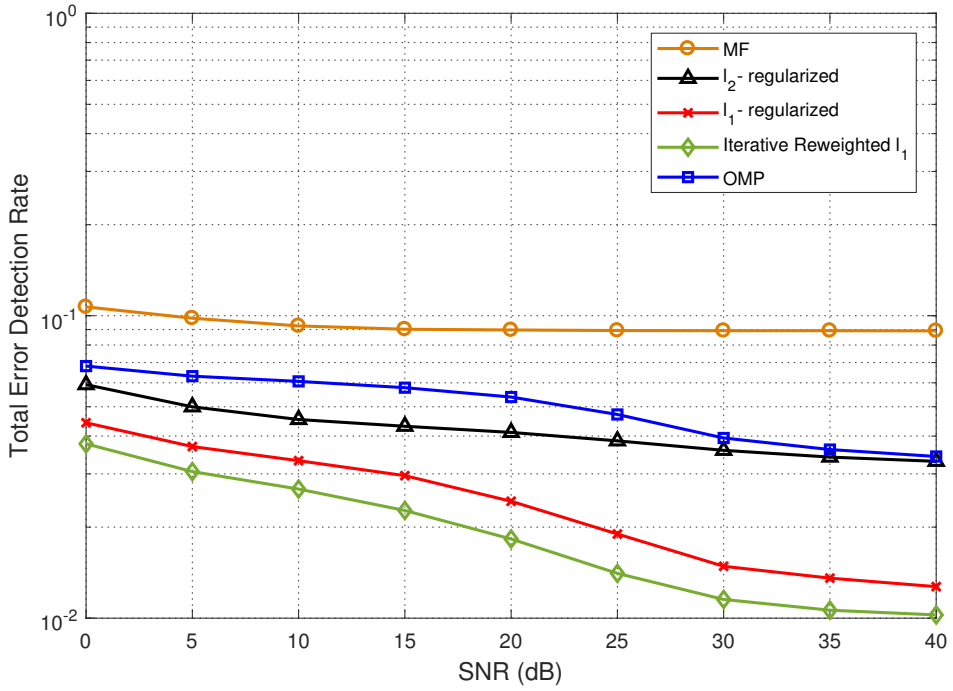

FIGURE 4.29: Probability of incorrect multipath positions (timing) for different approaches, considering Zadoff-Chu sequences and $M_o = 2N_s/3$.

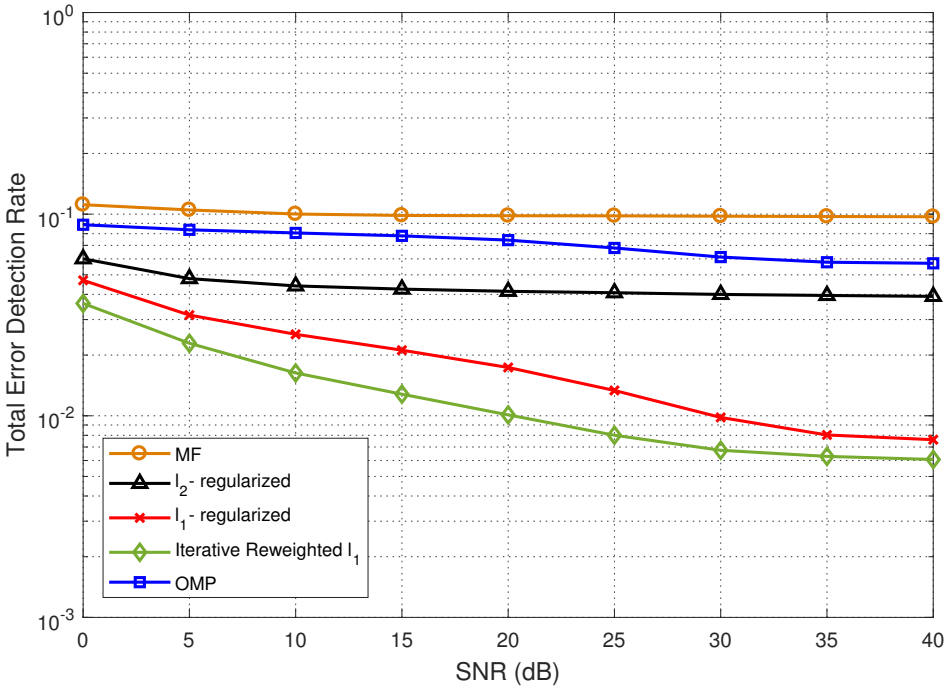

FIGURE 4.30: Probability of incorrect multipath positions (timing) for different approaches, considering pseudorandom codewords and $M_o = 2N_s/3$.

4.3.4 Effects of Cardinality and Polishing

Cardinality is used as a constraint that says that all tap positions should be located inside a window with the same size of the maximum expected channel length, L, and is is used by ℓ_2-regularized, ℓ_1-regularized and Iterative Reweighted ℓ_1. Polishing is used by ℓ_1-regularized and Iterative Reweighted ℓ_1 and it is utilized with the purpose of performing a final LS estimation using estimated support.

In previous subsections, cardinality was never applied, while polishing was always utilized. In this final subsection, the effects of using simple cardinality and not utilizing polishing are analyzed. A channel length of $L = 10$, with a pilot size of $N_C = 64$ and a simulation window of $N_s = 128$ samples are applied. The observation window is $M_o = N_s + N_c$.

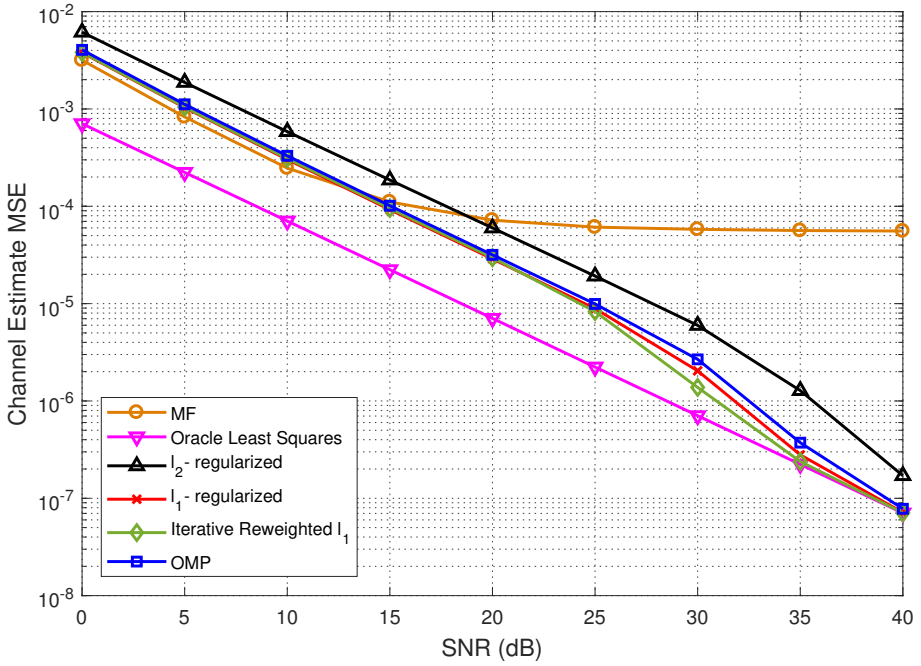

FIGURE 4.31: MSE performance for different joint timing synchronization and channel estimation methods, considering TCH codes and using cardinality.

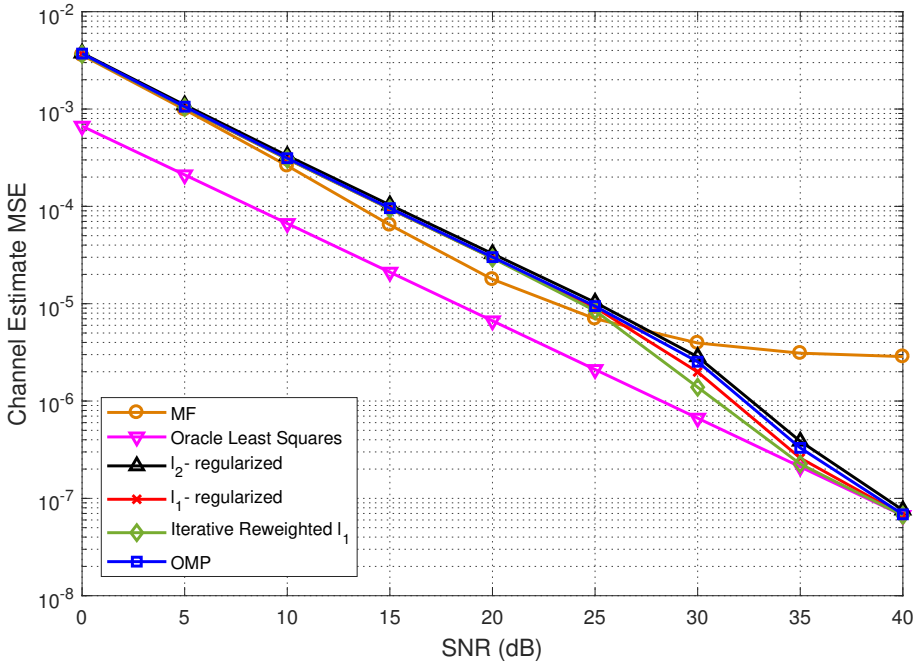

FIGURE 4.32: MSE performance for different joint timing synchronization and channel estimation methods, considering Zadoff-Chu sequences and using cardinality.

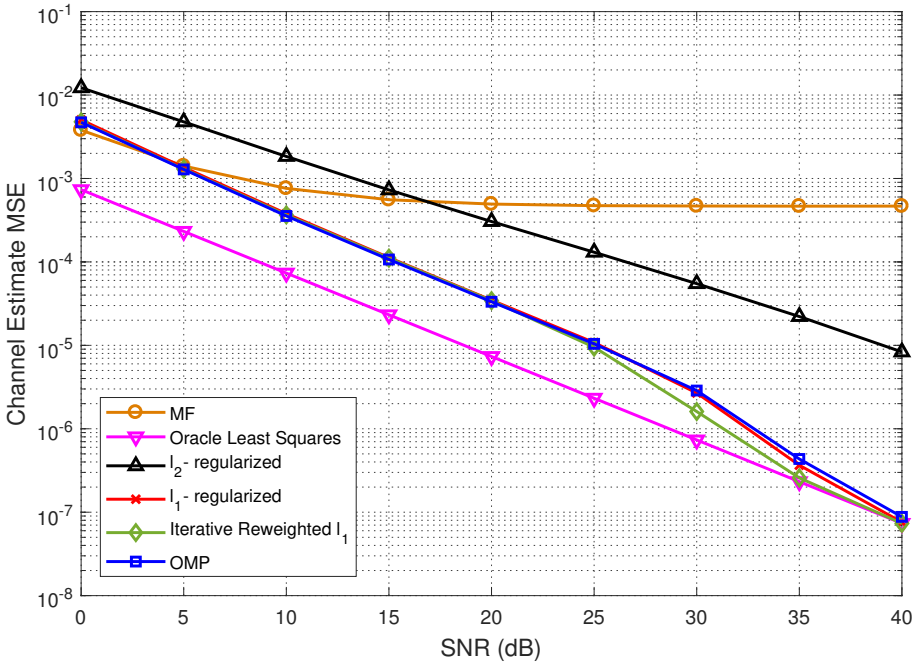

FIGURE 4.33: MSE performance for different joint timing synchronization and channel estimation methods, considering pseudorandom codewords and using cardinality.

Figures 4.31 to 4.33 depict the channel estimation MSE performance for the different methods while using cardinality. It is observable that the curves are far from the Oracle Least Squares ideal curve, especially for lower SNR values, which means that is where cardinality affects the most. MF and ℓ_2-regularized have the overall worst performance out of all methods, notably when using TCH codes and pseudorandom codewords. As for the other three methods, they all have very similar performances, with Iterative Reweighted ℓ_1 having again the best results, and starting at 30 dB, these curves get closer to the ideal curve.

Figures 4.34 to 4.36 illustrate the effect of cardinality in synchronization performance. MF is consistently bad in all of the graphs, presenting a high irreducible error floor. Now, it can be seen that most curves start to drop much later, at around 25 dB, having an almost equal performance until that point. As usual, Iterative Reweighted ℓ_1 has the best performance and Zadoff-Chu sequences have the upper hand against its counterparts, though by a small margin.

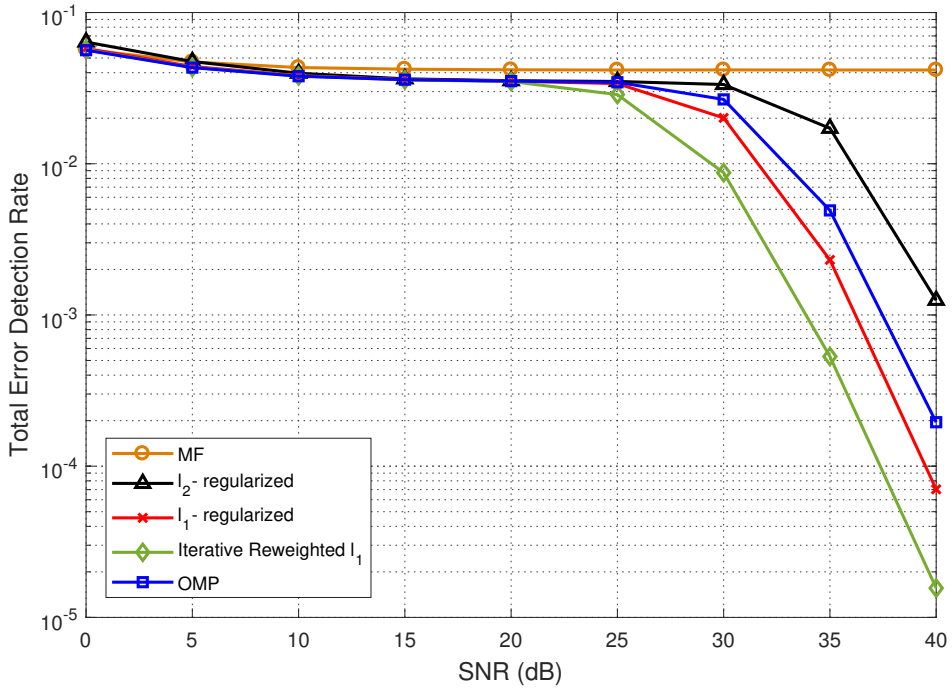

FIGURE 4.34: Probability of incorrect multipath positions (timing) for different approaches, considering TCH codes and using cardinality.

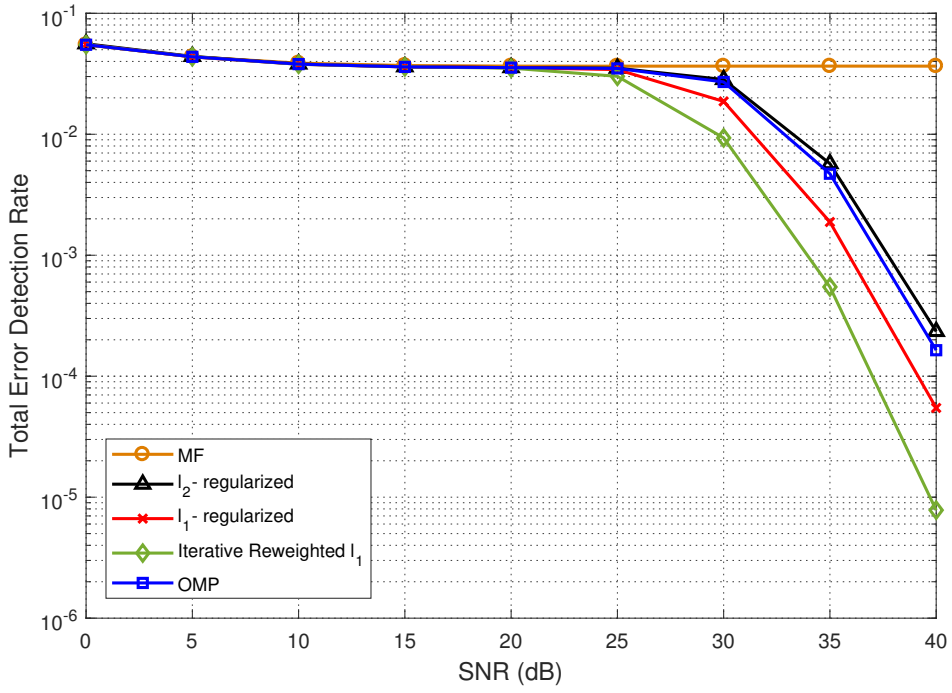

FIGURE 4.35: Probability of incorrect multipath positions (timing) for different approaches, considering Zadoff-Chu sequences and using cardinality.

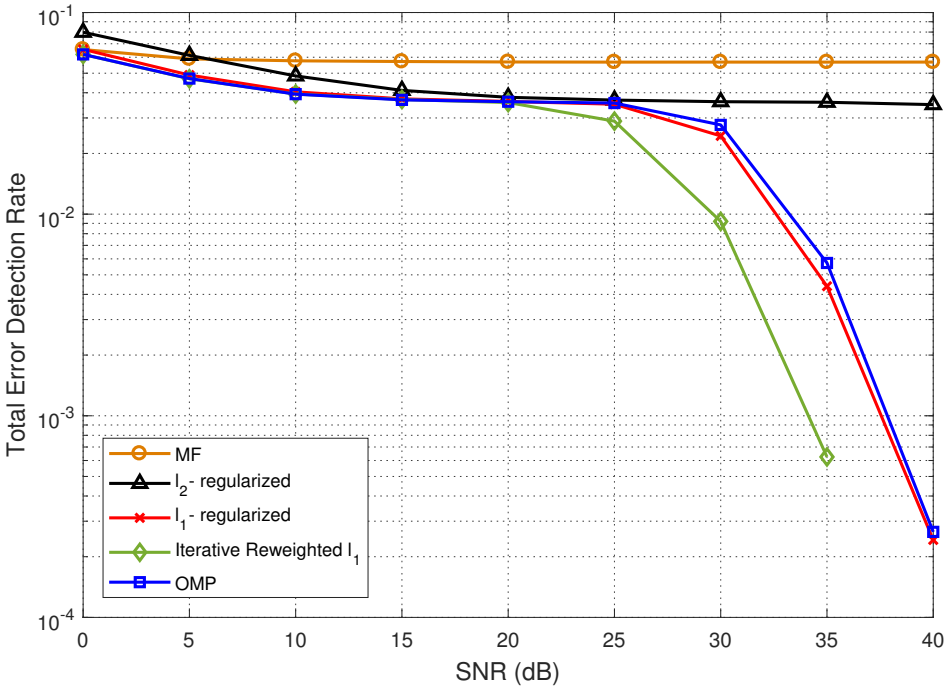

FIGURE 4.36: Probability of incorrect multipath positions (timing) for different approaches, considering pseudorandom codewords and using cardinality.

The next graphs, presented in Figures 4.37 to 4.42, are the last of this section and they show what happens when polishing is not used. The most notable difference regarding previous cases is that ℓ_1-regularized and with Iterative Reweighted ℓ_1 are no longer the two best performing methods in terms of channel estimation, with OMP showing better performance than these two.

By not performing a final LS estimation using estimated support, the performance of ℓ_1-regularized and with Iterative Reweighted ℓ_1 and even MF decreased, while the other two remained unaffected due to not possessing the option of using polishing. But this decrease in quality is very small and is mostly only observable for higher SNR values.

As for the type of pilots used, the most affected were the pseudorandom codewords, which also show the weakest results. TCH codes and Zadoff-Chu sequences have similar curves, with the latter marginally outperforming TCH codes. Zadoff-Chu sequences were also he least affected by the lack of polishing.

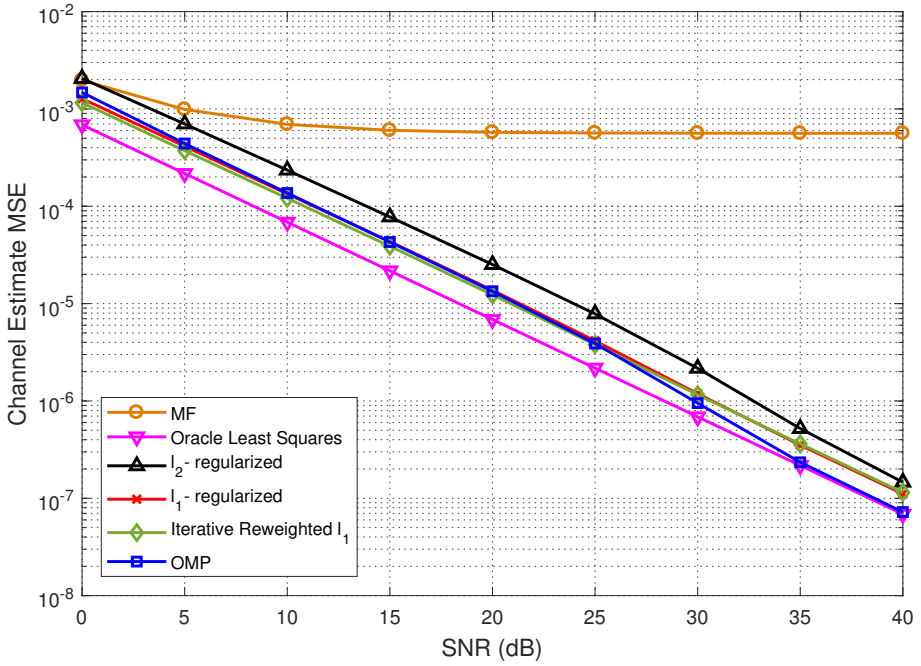

FIGURE 4.37: MSE performance for different joint timing synchronization and channel estimation methods, considering TCH codes and not using polishing.

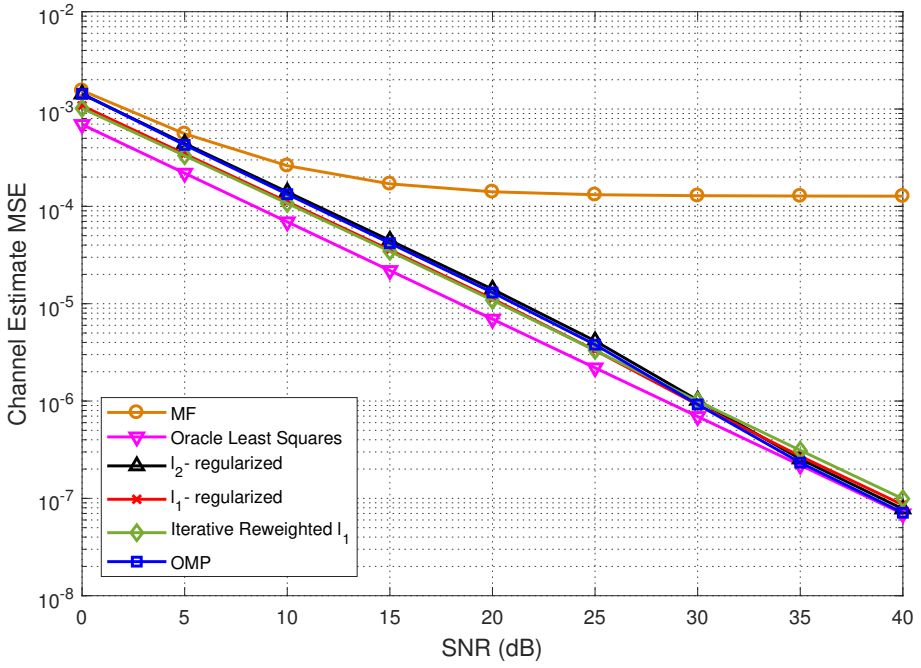

FIGURE 4.38: MSE performance for different joint timing synchronization and channel estimation methods, considering Zadoff-Chu sequences and not using polishing.

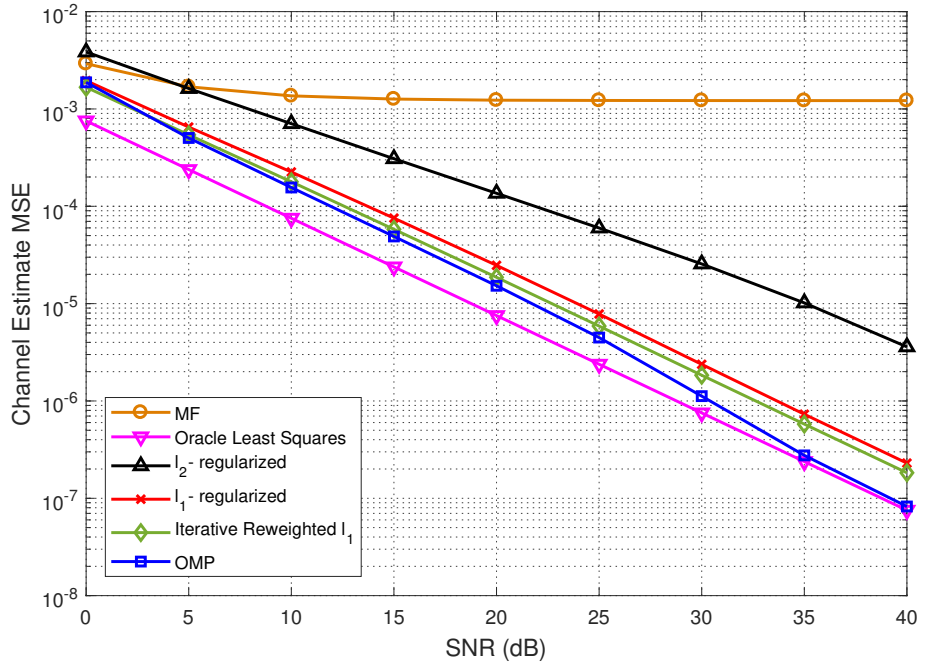

FIGURE 4.39: MSE performance for different joint timing synchronization and channel estimation methods, considering pseudorandom codewords and not using polishing.

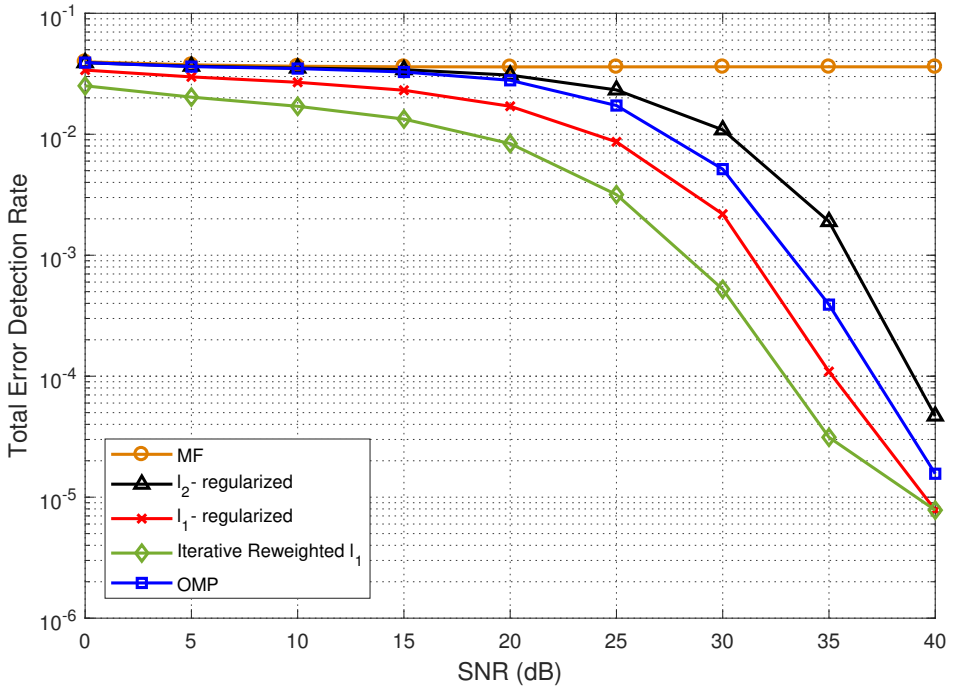

FIGURE 4.40: Probability of incorrect multipath positions (timing) for different approaches, considering TCH codes and not using polishing.

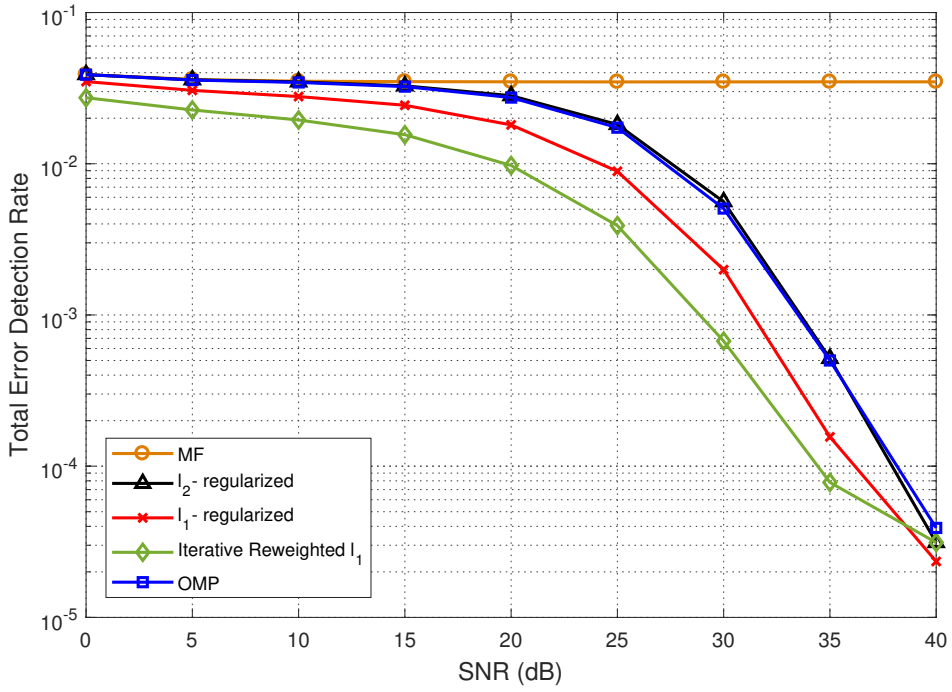

FIGURE 4.41: Probability of incorrect multipath positions (timing) for different approaches, considering Zadoff-Chu sequences and not using polishing.

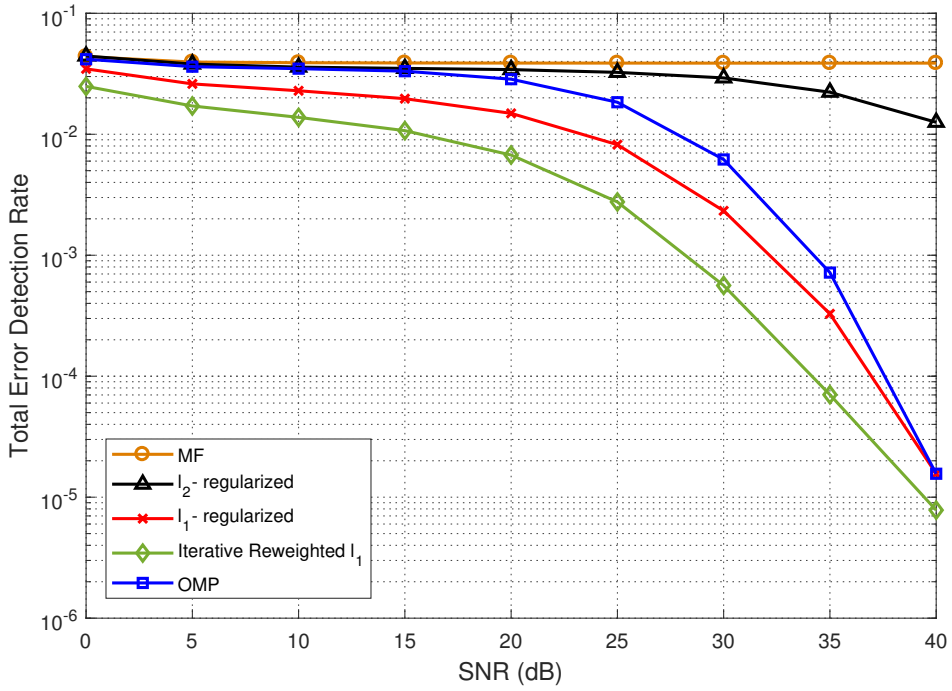

FIGURE 4.42: Probability of incorrect multipath positions (timing) for different approaches, considering pseudorandom codewords and not using polishing.

As for the performance in synchronization, the graphs were somewhat affected by not using polishing even though the difference is minimal. The most notable cases are ℓ_1-regularized equaling Iterative Reweighted ℓ_1 at 40 dB in the TCH graph and ℓ_1-regularized surpassing Iterative Reweighted ℓ_1 at 40 dB in the Zadoff-Chu graph.

After analyzing the results presented in these different cases, it can be concluded that Compressed Sensing can be successfully used in joint timing synchronization and channel estimation. TCH codes with $N_c = 256$ have an overall better performance in the channel estimation part, while Zadoff-Chu sequences seem to be slightly superior for synchronization when using a perfect observation window, M_o. When using a smaller window, TCH codes have the best performance in general, being able to keep up with the reduction of the observation window. Regarding the proposed methods, ℓ_1-regularized and Iterative Reweighted ℓ_1 are, without a doubt, the best performing methods and even though they are computationally heavier than OMP, they are the best choice since they can also withstand smaller observation windows better.

Chapter 5

Conclusions

This chapter sums up the work developed in this dissertation, summarizes the obtained results and gives ideas on how this work can be continued in the future.

5.1 Conclusions

The main objective proposed for this dissertation was to study the performance of channel estimation using TCH codes, regarding Machine-Type Communications. At first, the focus was solely on performing channel estimation but, after successfully performing it, it was decided to test TCH codes on joint channel estimation and timing synchronization using a technique that has not been very used for this purpose, compressive sensing.

Firstly, it was performed an initial study in order to get a better understanding of TCH codes, channel estimation with a focus on two different pilot types, data multiplexed pilots, where the pilot symbols are multiplexed with the data, and implicit pilots, where the pilots are superimposed over the data and are sent simultaneously, OFDM modulation and Machine-Type Communications, considered to be the future of communications between smart devices. After this study came the implementation of TCH codes and the different types of pilots on a simulator using MATLAB. The results obtained from the simulations displayed TCH codes

as a successful choice to use in channel estimation with both pilot approaches and for both modulations tested in this work. This work allowed to write and publish a conference paper on this matter.

Given the great synchronization properties of TCH codes, joint synchronization and channel estimation was the next step in order to test these codes. Compressed sensing was chosen due to its good results in the channel estimation field and also because, to the best of my knowledge, it hadn't been used for joint synchronization and channel estimation purposes. After studying the algorithms with best performance, they were implemented also in MATLAB alongside TCH codes and two other codes, Zadoff-Chu sequences and pseudorandom codewords. After testing variations in the size of the pilots, alterations in the channel length and reducing the observation window, the results revealed that TCH codes can be successfully utilized to execute channel estimation and synchronization simultaneously and, additionally, revealed that CS can be used to perform these tasks together, with Iterative Reweighted ℓ_1 presenting the best performance both in terms of MSE of the channel estimates and multipath timing error rate.

Ultimately, it can be concluded that TCH codes can be successfully utilized in channel estimation, they have similar and sometimes better performance than its counterparts and they are appropriate to use in MTC due to TCH codes performing well in jopint synchronization and channel estimation, which means these processes are executed faster which is important in MTC application related to security, for example.

5.2 Future Work

An immediate future task is to write a letter and also papers regarding the use of CS in joint channel estimation and synchronization. The simulations in this dissertation could lead to the implementation of TCH codes in hardware, like a Software Defined Radio (SDR) board, for example. The use of TCH codes in channel estimation, synchronization and coding simultaneously could be studied.

Appendices

Appendix A

Efficient Channel Estimation Using TCH Codes: Paper

This appendix contains a paper that was written based on the results discussed and presented in Chapter 3 [72]. This paper was co-written with my colleague Sílvia Catarino, Professor Francisco Cercas and Professor Nuno Souto, from ISCTE-IUL, and professor Rui Dinis, from FCT-UNL. It has been accepted in the 8^{th} International Workshop on Recent Advances in Broadband Access Networks that will take place in Munich, Germany between November 6-8.

Efficient Channel Estimation Using TCH Codes

Bruno Lopes[1], Sílvia Catarino[1], Francisco Cercas[1,3], Nuno Souto[1,3], Rui Dinis[2,3]
[1]ISCTE-IUL, Av. das Forças Armadas, Lisboa, Portugal,
bmcls@iscte.pt, scsco@iscte.pt, francisco.cercas@iscte.pt, nuno.souto@iscte.pt
[2]FCT-UNL, Monte da Caparica, Portugal, rdinis@fct.unl.pt
[3]Instituto de Telecomunicações, Av. Rovisco Pais, 1, Lisboa, Portugal

***Abstract*— In this paper, we consider the use of TCH codes to perform channel estimation in an OFDM system, using either data multiplexed pilots or superimposed pilots over the data. TCH codes possess several properties that allow us to use them efficiently in various applications which includes channel estimation, as we address in this paper. With this objective, several performance results were obtained through simulations which allowed the evaluation of the impact of different pilot power levels and modulations, as well as the comparison of TCH against other conventional pilots. In order to cope with the interference between pilots and data, an iterative receiver with interference suppression was employed for the superimposed pilots method.**

***Index Terms*— TCH codes, channel estimation, OFDM, data multiplexed pilots, implicit pilots.**

I. Introduction

In present-day communications, there is a need to mitigate the effects of multipath fading. This is possible by using Orthogonal Frequency Division Multiplexing (OFDM) since it has multicarrier modulation. The usable bandwidth in an OFDM system is divided into orthogonal sub-channels, allowing that a frequency selective channel to be converted into a non-frequency selective one [1] [2] [3]. This paper outlines a method to estimate the channel in an OFDM system by utilizing TCH codes with data multiplexed or implicit pilots.

TCH codes are error correcting codes first demonstrated in [4] and they are described as having a sturdy error correcting performance (oriented to transmit short and sensible information), as being able to use Fast Fourier Transform (FFT) to execute simple decoding, ideal rigid sizes and an excellent correlation performance. All of these features grant us an opportunity to utilize TCH codes in distinct applications based on digital transmission systems like error correction, spread spectrum systems or channel and phase estimation. The admirable correlation properties of these codes is what allows us to perform and study the channel estimation and to use them for the first time with this objective.

The main purpose of channel estimation is to compensate the effects of attenuation, fading and scattering suffered by the signal in the channel and the most prominent way of performing it consists in transmitting training sequences or pilots that characterizes the distortion that the channel causes, regarding attenuation and phase shift. The Least-Square (LS) estimation [5] and its improvement, the Minimum Mean-Square-Error (MMSE) estimation [6], are the two most used methods of channel estimation. The first is used when the distributions of channel and noise are unknown while the latter is utilized if the previous parameters are identified.

The pilot symbols used in channel estimation are regularly multiplexed with the data in both time and frequency domains [7] [8] [9]. However, this approach may originate an inefficient bandwidth use and to contradict this problem, a different method consisting in using implicit pilots was proposed in [10] and [11]. In this approach, the pilot symbols are superimposed over the data, increasing the pilots' density without sacrificing system capacity, though more power has to be spent on the pilot sequence.

The main objective of this paper, is to study the performance of channel estimation using data multiplexed and implicit pilots based on TCH codes. We analyze these methods by comparing results that illustrate the BER performance obtained through simulations. Following a similar approach to the one adopted in [12], an iterative receiver capable of performing joint detection and channel estimation is used in order to mitigate the mutual interference in the data and in the pilot symbols caused by the use of embedded pilots.

The paper is organized as follows. Section II portrays TCH codes and how they are built. Secondly, in Section III, the system characterization is presented, which contains information about the OFDM transmission frame structures and also the transmitter and the receiver we used. Section IV describes the channel estimation process and the results obtained are presented in Section V. Lastly, Section VI contains the appropriate conclusions taking into account the obtained results.

II. TCH Codes

TCH codes [4] [13] are a class of binary, non-systematic, non-linear and cyclic block codes, with length $n = 2^m$, where m represents any positive integer. These codes have successfully been utilized in various applications, including synchronization [14], coding [15][16] and Ultra-Wideband (UWB) systems.

A TCH block code can be identified as $TCH(n, k, t)$, where n represents the code length, k identifies the number of information bits in a code word and t is its error correcting capacity. They can be defined by Equations (1) and (2) that

[1]This work was supported by Fundação para a Ciência e Tecnologia and Instituto de Telecomunicações under project UID/EEA/50008/2013.

illustrate a set h generator base polynomials $P_i(x)$:

$$TCH(n,k) = \sum_{i=1}^{h} P_i(x), \tag{1}$$

$$P_i(x) \neq P_j(x^r) \mod n \quad i \neq j \forall t \in \mathbf{N}, \tag{2}$$

The error correcting capacity of the TCH codes depends on their minimum distance, represented by d_{min} in Equations (3) and (4), between those polynomials:

$$d_{min} \geqslant 2t + 1, \tag{3}$$

$$d_{min} \leqslant H_d[P_i(x), \{P_j(x^r)\} \mod n] \leqslant n - d_{min}, \tag{4}$$

Where H_d stands for Hamming distance. They are also balanced and, by using Basic TCH Polynomials (B-TCH Polynomials) of degree n, we can generate these codes:

$$P(x) = \sum_{i=0}^{(\frac{p-1}{2})-1} a_i x^{K_i}, \tag{5}$$

where the exponents K_i satisfy Equation 6:

$$a^{K_i} = 1 + a^{2i+1}, i = 0, 1, ..., (\frac{p-1}{2}) - 1, \tag{6}$$

p is a prime number with $p = n + 1 = 2^m + 1$. Prime numbers that obey this condition are called *Fermat* numbers:

$$F_i = 2^{2^i} + 1, \tag{7}$$

Only five numbers are known to obey the Fermat number rule, which means that we can only generate pure TCH polynomials, also designated as B-TCH Polynomials, for code lengths $n = 2, 4, 16, 256 \ and \ 65536$. Even though it is possible to build similar TCH codes that can be extended for other code lengths, this comes at the cost of losing the properties and also the ideal structure of the B-TCH polynomials. TCH codes originated by B-TCH polynomials have both good cross and auto-correlation, with the latter assuming only three-values: n, the value of the code polynomial, 0 and -4. This translates into a great advantage for higher sized TCH codes, such as TCH codes length $n \geqslant 256$. For higher n values, the sequences tend to get closer to a Dirac impulse, as depicted in Figure 1, showing an auto-correlation function of a B-TCH Polynomial of length 256, which are used in this paper, with offsets from -128 to 127, so that the peak is displayed in the center.

III. System Characterization

We describe a low-pass OFDM signal in (8) where X_k depicts the data symbols, N is the total number of sub-carriers and the symbol length is represented by T

$$\upsilon(t) = \sum_{k=0}^{N-1} X_k e^{\frac{2\pi kt}{T}}, 0 \leqslant t \leqslant T, \tag{8}$$

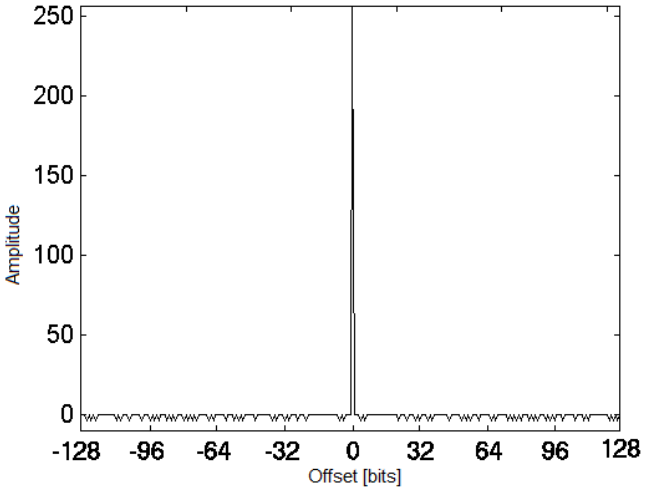

Fig. 1: Auto-correlation of a B-TCH Polynomial with $n = 256$.

There is a need to prevent intersymbol interference, so we insert a guard interval of length T_g immediately before the OFDM block. A cyclic prefix is transmitted during the guard interval so that the signals in the intervals $-T_g \leqslant t < 0$ and $T - T_g \leqslant t < T$ remain equal. The following equation represents an OFDM signal with a cyclic prefix

$$\upsilon(t) = \sum_{k=0}^{N-1} X_k e^{\frac{2\pi kt}{T}}, T_g \leqslant t \leqslant T, \tag{9}$$

A transmitter chain inspired in [12] is shown in Figure 2 and it combines QAM constellations with an OFDM transmission that can use data multiplexed or implicit pilots.

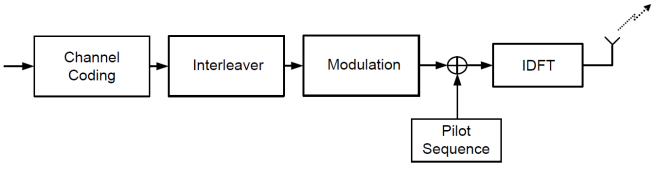

Fig. 2: Transmitter chain.

Figure 3 shows the frame structure that we consider for an OFDM system with N carriers using data multiplexed pilots, where only the first column of the pilot grid contains pilot symbols and the first column of the data grid is empty. Figure 4 illustrates a similar frame structure but this time implicit pilots are used. This means that all the positions in both grids are filled since the pilots are superimposed over the data. In both frame structures, the grids are built by utilizing an OFDM time block spacing in the time domain. We characterize the transmitted sequences as follows

$$\mathbf{X} = \mathbf{S} + \mathbf{C}, \tag{10}$$

where $\mathbf{S}$ describes an $N \times 1$ vector where the elements are complex valued modulated symbols drawn from an M-sized complex valued constellation and $\mathbf{C}$ is an $N \times 1$ vector

that corresponds to $\mathbf{C} = DFT\{\mathbf{c}\}$ which is the DFT of a TCH codeword. In order to take advantage of the good auto-correlation properties of the TCH codes, we utilize the DFT of these codes. The objective of using these auto-correlation properties is mainly for time synchronization purposes.

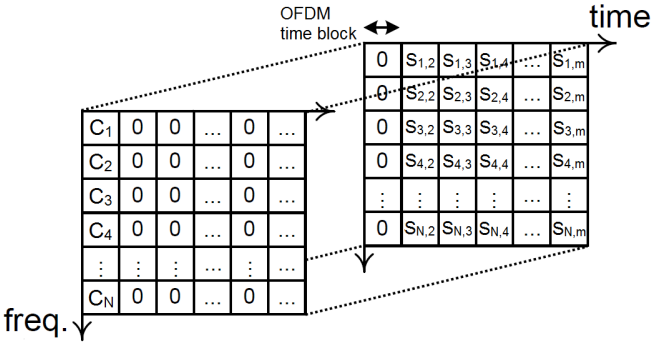

Fig. 3: Frame structure used for an OFDM transmission containing data multiplexed pilots where **C** represents a pilot symbol and **S** represents a data symbol.

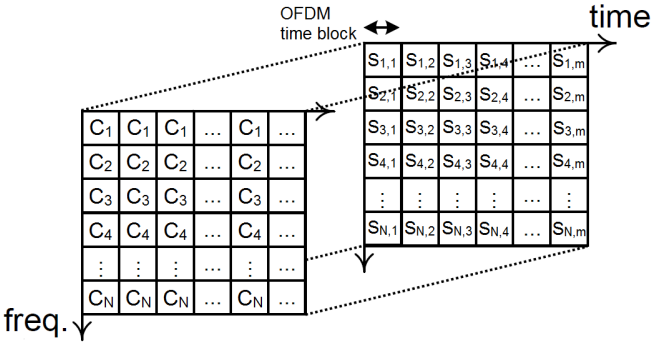

Fig. 4: Frame structure used for an OFDM transmission containing implicit pilots where **C** represents a pilot symbol and **S** represents a data symbol.

As we mentioned in Section I, transmitting superimposed pilots on data creates mutual interference. In order to reduce it, and also to attain reliable channel estimation and data detection, we propose a receiver based on a similar scheme [12], capable of performing these tasks via iterative processing. The structure of the referenced receiver is shown in Figure 5.

Firstly, the pilot symbols are removed from the sequence. Then, they enter the Channel Equalization block and after that, the sequences of equalized samples are demodulated into bit streams. These bit streams are processed so that an estimate of the transmitted signal, $\widehat{\mathbf{S}}$, can be reconstructed. In the following iteration, the reconstructed sequence can be utilized so that the channel estimates are enhanced.

IV. Channel Estimation

Many techniques regarding channel estimation on OFDM systems [17] can be used, each of them with its characteristics and many differences between them, like whether they use time or frequency domain samples, the complexity of the technique in question, performance and *a priori* information utilized. The latter can be made of sub-carrier's correlation in the frequency or time domains and the quantity of *a priori* information affects the estimation quality, so the more information of this type there is, the better the estimation.

If the overall channel impulse response is shorter than the N_G-sized cyclic prefix, we can describe the frequency domain received sequence as follows

$$\mathbf{R} = \mathbf{H}(\mathbf{S} + \mathbf{C}) + \mathbf{N}, \tag{11}$$

where $\mathbf{H}$ is an $N \times N$ diagonal matrix that stands for the channel frequency response and $\mathbf{N}$ represents an $N \times 1$ vector of noise samples in the frequency domain. Both $\mathbf{S}$ and $\mathbf{C}$ have been described in equation (10). This model directly matches the channel estimation based on implicit pilots method but, by establishing $\mathbf{S} = 0$, the model can likewise represent the data multiplexed pilots with a block of pilot symbols.

The receiver can employ an iterative approach based on [12] and therefore it is possible to obtain the frequency channel response. Each of the following steps is executed for each iteration q:

1) Data symbol estimates are removed from pilots. The resulting sequence becomes
$$\widetilde{\mathbf{R}}^{(q)} = \mathbf{R} - \widehat{\mathbf{S}}^{(q-1)}\widehat{\mathbf{H}}^{(q-1)}, \tag{12}$$
where $\widehat{\mathbf{S}}^{(q-1)}$ and $\widehat{\mathbf{H}}^{(q-1)}$ are the symbol and channel estimates from the previous iteration. When $q = 1$ we simply use $\widetilde{\mathbf{R}}^{(1)} = \mathbf{R}$. The described step is only applied when using superimposed pilots.
2) The channel frequency response estimates is calculated using
$$\widetilde{\mathbf{H}}^{(q)} = |\mathbf{\Lambda}|^{-2}\mathbf{\Lambda}^H\widetilde{\mathbf{R}}^{(q-1)}, \tag{13}$$
where $\mathbf{\Lambda} = diag(\mathbf{C})$, where diag$(\cdot)$ represents a diagonal matrix whose elements are contained in the vector used as argument. $|\mathbf{\Lambda}|$ denotes the element-wise absolute value operation and $(\cdot)^H$ depicts the conjugate transpose of a matrix/vector. After the first iteration, the estimates of data symbols can also be used as pilots for channel estimation refinement. In this case we use $\mathbf{\Lambda} = diag(\widehat{\mathbf{S}}^{(q-1)})$ for data multiplexed pilots and $\mathbf{\Lambda} = diag(\widehat{\mathbf{S}}^{(q-1)} + \mathbf{C})$ for implicit pilots.
3) We can augment the channel estimates by assuring that the corresponding impulse response has a duration N_G. This is accomplished by utilizing
$$\widehat{\mathbf{H}}^{(q)} = diag(\mathbf{FTF}^H\widetilde{\mathbf{h}}^{(q)}), \tag{14}$$
where
$$\mathbf{T} = \begin{bmatrix} \mathbf{I}_{N_G} \\ \mathbf{0}_{(N-N_G)\times N_G} \end{bmatrix}, \tag{15}$$
$\mathbf{0}_{N_{CF}\times(N-N_{CF})}$ represents a size $(N-N_G)N_G$ matrix full of zeros, while $\mathbf{I}_{N_G}$ depicts an $N_G \times N_G$ identity matrix. The $N \times N$ scaled discrete Fourier transform

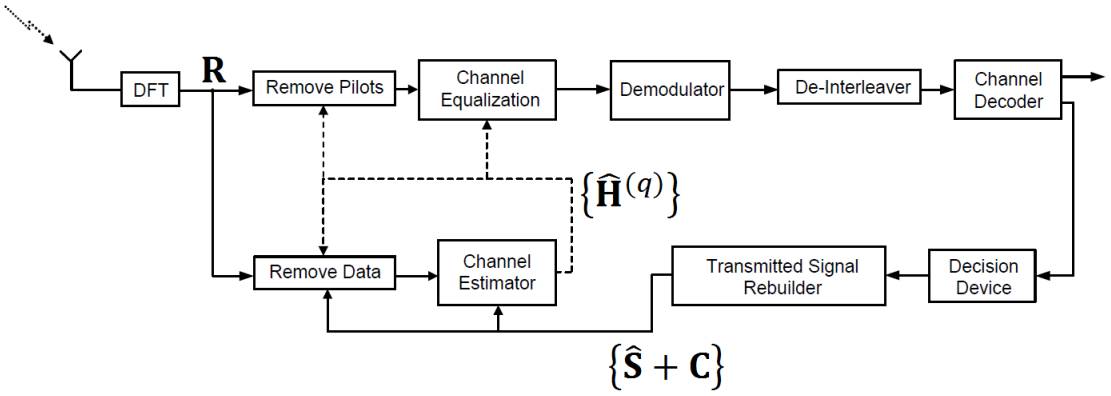

Fig. 5: Structure of the iterative receiver.

(DFT) matrix is represented by $\mathbf{F}$, such that $\mathbf{I}_N = \mathbf{F}^H\mathbf{F}$, and $\widetilde{\mathbf{h}}^{(q)}$ illustrates the $N \times 1$ vector that contains the diagonal of $\widetilde{\mathbf{H}}^{(q)}$.

V. Results

The BER performance results of all of the graphs present in this section were attained by performing Monte Carlo simulations with an 8 equal power tap Rayleigh fading channel, using 256 OFDM carriers. For Figure 6, we used QPSK modulation and we varied the pilot power values, which are relatively measured to the channel data, from 0 to -12 with jumps of -3 dB. We performed the channel estimation by utilizing data multiplexed pilots and, along with them, were sent blocks with TCH words of length 256. The channel encoders were rate $1/2$ turbo codes based on two identical convolutional codes with two constituent codes characterized by $G(D) = [1 + D^2 + D^3)/(1 + D + D^3)]$[18]. 18 turbo decoding iterations were applied at the receiver and this receiver was conventional, meaning there was only one receiver iteration. A perfect estimation curve is shown for comparison purposes.

By analyzing these results, we verify that higher valued pilot powers, from 0 to -6 dB, translate into better results since the curves representing high values of pilot power are almost adjacent to the curve that portrays perfect estimation. For higher $E_s/N0$ values, the performance is slightly inferior, illustrated by the -9 and -12 dB curves that get further away from the perfect estimation curve.

In Figure 7, we maintained the same simulation conditions used in Figure 6 with the exception of two parameters: superimposed pilots with TCH codewords were used to perform the estimation, instead of data multiplexed pilots, and in the receiver we applied 3 turbo decoding iterations for each of the 6 receiver iterations in the iterative scheme.

Once again, the performance is better when the pilot power value is higher. The curves depicting -9 and -12 dB have a larger discrepancy from the perfect estimation curve than the same curves observed in Figure 6 because of interference between the data symbols and pilots, inherent to a scheme that uses implicit pilots. These results indicate that, to obtain

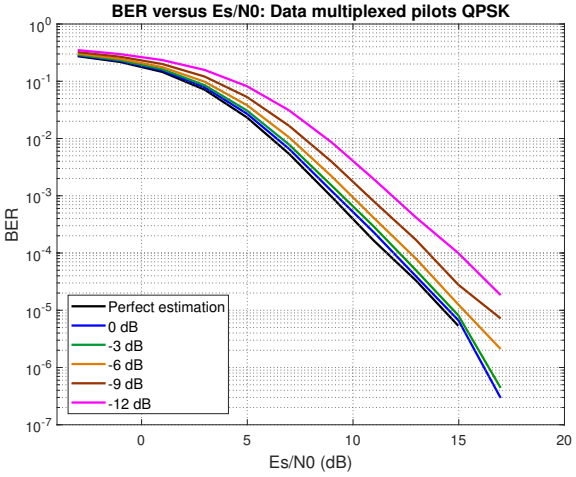

Fig. 6: BER performance of channel estimation utilizing QPSK modulation and data multiplexed pilots based on TCH codes for different pilot powers.

a good performance and to compensate the interference when using implicit pilots, it is necessary to use higher pilot power levels.

The results displayed in Figure 8 were obtained by using the same simulation parameters used to build Figure 7 but this time, the pilot power is fixed at 0 dB and the number of iterations in the receiver is gradually increased.

When the number of iterations used in the receiver is higher, the performance is better, which is visible by comparing the simulated curves with the perfect estimation one. Still, the difference in performance is small after we stop using a conventional receiver and it is almost indistinguishable for the highest simulated values of iterations, 4 and 8, meaning that the performance is not affected greatly by increasing the number of receiver iterations after a certain value is reached.

The results presented in Figure 9 were obtained by using either data multiplexed pilots or implicit pilots, while once again considering QPSK modulation and also by changing

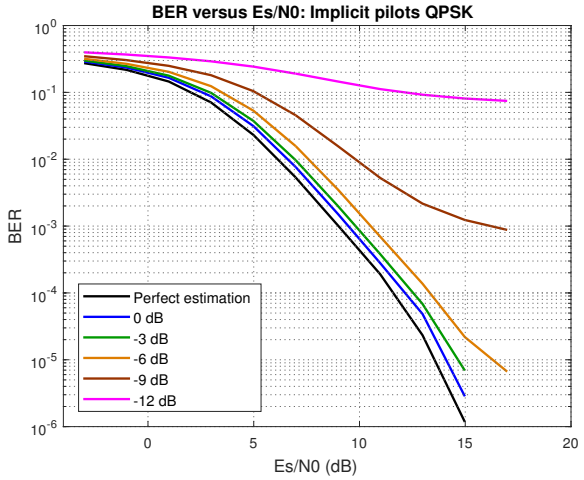

Fig. 7: BER performance of channel estimation utilizing QPSK modulation and superimposed pilots on TCH codes for different pilot powers.

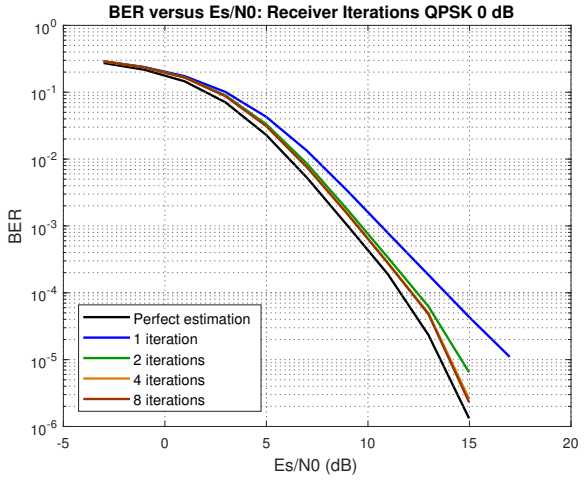

Fig. 8: BER performance of channel estimation utilizing QPSK modulation and superimposed pilots on TCH codes for different receiver iteration values.

the data sent, meaning it was either based on TCH codewords or conventional data. The channel coding and receiver structures used in Figures 6 and 7 were used to simulate the curves that represent data multiplexed pilots and implicit pilots in Figure 9, respectively. Based on the results demonstrated in Figure 6 and Figure 7, we considered single pilot power value, which is -4 dB. A curve regarding perfect estimation is also shown for reference.

All of the simulated cases possess an almost identical and very good performance, which is justified by observing the proximity between all of the curves and also because no BER floor is visible. The difference between the performance of data multiplexed pilots and implicit pilots is very small and even though data multiplexed pilots have a better performance, we can avoid spectral degradation by

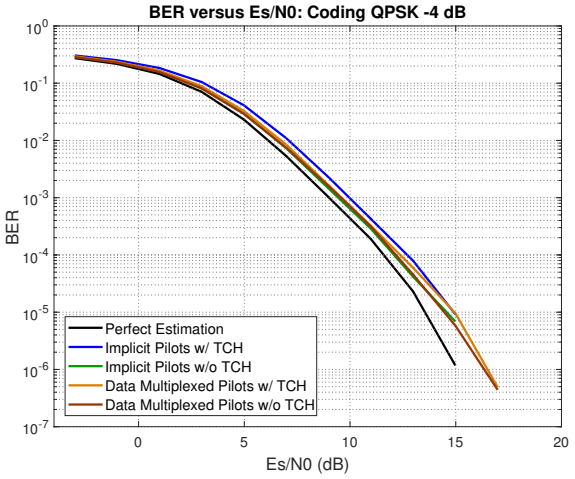

Fig. 9: BER performance of channel estimation utilizing QPSK modulation while considering different pilot approaches and based on TCH code words or conventional pilots.

using implicit pilots. Conventional pilots, labeled in Figure 9 as "w/o TCH", have a slightly better performance than the ones based on TCH code words but the difference is really small and TCH codewords have the benefit of possessing good synchronization properties.

Finally, Figure 10 was built using the same simulation conditions of Figure 9 with the exception of the modulation, which this time is 64-QAM instead of QPSK, and the pilot power, which is now a fixed value of -1.75 dB.

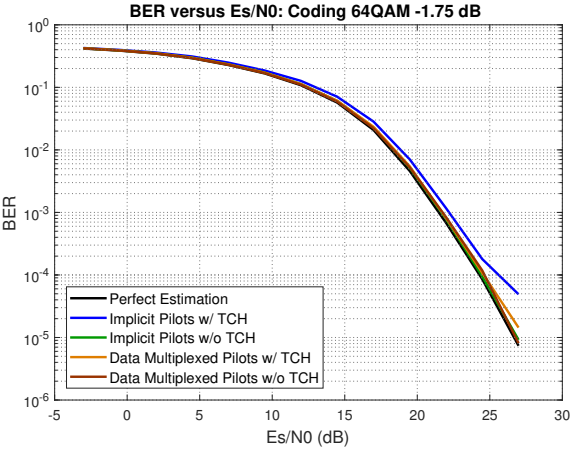

Fig. 10: BER performance of channel estimation utilizing 64-QAM modulation while considering different pilot approaches and based on TCH code words or conventional pilots.

The results show that with 64-QAM modulation we need higher BER values than the ones obtained from the QPSK simulations, as expected. Amplitude modulation methods are more susceptible to noise and that is the reason why we used a higher pilot power value than the one used in Figure 9.

But in this simulation, all the curves are closer to the perfect estimation curve, with the "Implicit Pilots w/TCH" showing a small detour starting around 25 dB.

VI. Conclusions

After studying and analyzing the use of TCH codes for channel estimation using two different pilot approaches, data multiplexed pilots and implicit pilots, we conclude that both approaches are reliable with each of them having its own advantages which are referenced in Section V. Using TCH codes for channel estimation is justified based on the fact that not only they have very similar performance levels when compared with conventional pilots but also that TCH codes have great synchronization properties, meaning that it is possible to simultaneously use them in the system for synchronization purposes, making TCH codes a better choice for channel estimation. Regarding the modulations used, QPSK has a better performance than 64-QAM but both present really good performances, showing that TCH codes can be successfully used with both modulations, as we expected.

References

[1] Cimini, L., "Analysis and simulation of a digital mobile channel using orthogonal frequency division multiplexing," *IEEE transactions on communications*, vol. 33, no. 7, pp. 665–675, 1985.

[2] Bingham, J. A., "Multicarrier modulation for data transmission: An idea whose time has come," *IEEE Communications magazine*, vol. 28, no. 5, pp. 5–14, 1990.

[3] Bingham, J. A., *ADSL, VDSL, and multicarrier modulation*. Wiley New York, 2000.

[4] Cercas, F., Tomlinson, M., and Albuquerque, A., "TCH: A New Family of Cyclic Codes Length 2^m," in *Information Theory, 1993. Proceedings. 1993 IEEE International Symposium on*. IEEE, 1993, pp. 198–198.

[5] Biguesh, M. and Gershman, A. B., "On channel estimation and optimal training for mimo systems," in *Sensor Array and Multichannel Signal Processing Workshop Proceedings, 2004*. IEEE, 2004, pp. 387–391.

[6] Biguesh, M. and Gershman, A. B., "Training-based mimo channel estimation: a study of estimator tradeoffs and optimal training signals," *IEEE transactions on signal processing*, vol. 54, no. 3, pp. 884–893, 2006.

[7] Li, Y., "Pilot-symbol-aided channel estimation for ofdm in wireless systems," *IEEE transactions on vehicular technology*, vol. 49, no. 4, pp. 1207–1215, 2000.

[8] Lam, C.-T., Falconer, D. D., Danilo-Lemoine, F., and Dinis, R., "Channel estimation for sc-fde systems using frequency domain multiplexed pilots," in *Vehicular Technology Conference, 2006. VTC-2006 Fall. 2006 IEEE 64th*. IEEE, 2006, pp. 1–5.

[9] Zeng, Y. and Ng, T. S., "Pilot cyclic prefixed single carrier communication: Channel estimation and equalization," *IEEE Signal Processing Letters*, vol. 12, no. 1, pp. 56–59, 2005.

[10] Chen, N. and Zhou, G. T., "Superimposed training for ofdm: a peak-to-average power ratio analysis," *IEEE Transactions on Signal Processing*, vol. 54, no. 6, pp. 2277–2287, 2006.

[11] Tugnait, J. K. and Meng, X., "On superimposed training for channel estimation: performance analysis, training power allocation, and frame synchronization," *IEEE Transactions on Signal Processing*, vol. 54, no. 2, pp. 752–765, 2006.

[12] Souto, N., Dinis, R., and Silva, J. C., "Efficient channel estimation for ofdm systems with hierarchical constellations," in *Communications and Information Technologies, 2007. ISCIT'07. International Symposium on*. IEEE, 2007, pp. 998–1002.

[13] Cercas, F. A. B., "A new family of codes for simple receiver implementation," *Ph. D. Thesis, Technical University of Lisbon, Instituto Superior Técnico, Lisbon*, 1996.

[14] Souto, N., Silva, J., Cercas, F., Almeida, A., and Correia, A., "Synchronization with TCH codes," in *Proc Wireless Personal Multimedia Communications Symp.-WPMC*, 2005, pp. 998–1002.

[15] Del Re, E. and Pierucci, L., *Satellite Personal Communications for Future-generation Systems: Final Report: COSY 252 Action*. Springer Science & Business Media, 2012.

[16] Aretz, K., Haardt, M., Konhäuser, W., and Mohr, W., "The future of wireless communications beyond the third generation," *Computer Networks*, vol. 37, no. 1, pp. 83–92, 2001.

[17] Liu, Y., Tan, Z., Hu, H., Cimini, L. J., and Li, G. Y., "Channel estimation for ofdm," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 1891–1908, 2014.

[18] 3GPP, "Multiplexing and channel coding (FDD)," 3rd Generation Partnership Project (3GPP), TS 25.212-v13.1.0, Jun. 2016.

Appendix B

Additional Results

This appendix presents additional results that complement the results shown in Chapter 4.

B.1 Effects of Different Codeword Lengths: $N_c = 64$

Figures B.1 to B.6 display the MSE performance of channel estimation and the synchronization performance for $N_c = 64$. Results show the same trend as the graphs presented in subsection 4.3.2: MF and ℓ_1-regularized are the worst performing methods, Iterative Reweighted ℓ_1 has the best results, pseudorandom sequences are the slightly poorer pilots while TCH codes and Zadoff-Chu have stronger performances. $N_c = 64$ is the middle ground between $N_c = 16$ and $N_c = 256$, possessing a better performance than the former and is less computationally heavy than the latter.

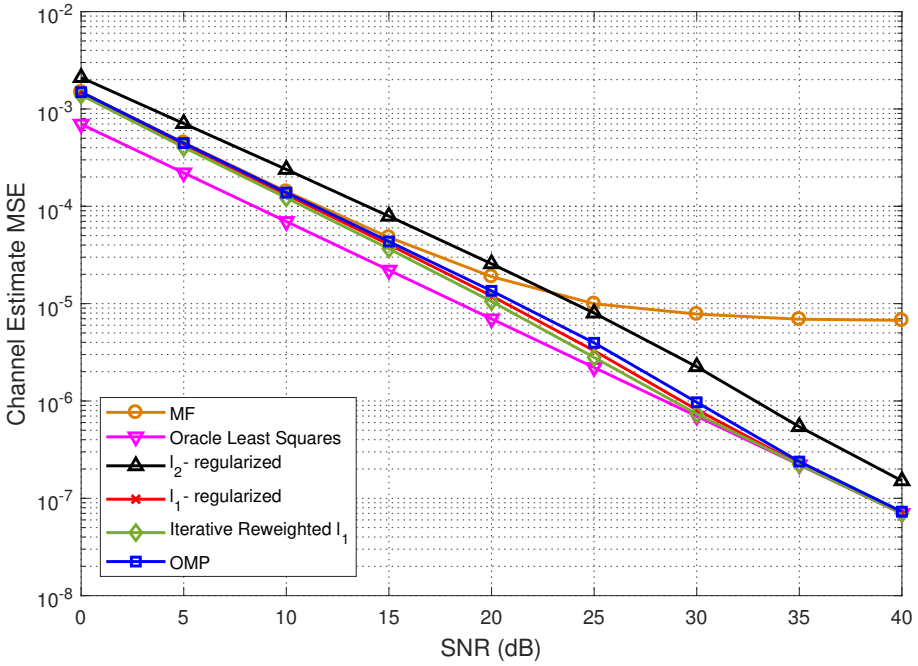

FIGURE B.1: MSE performance for different joint timing synchronization and channel estimation methods, considering TCH codes and $N_c = 64$.

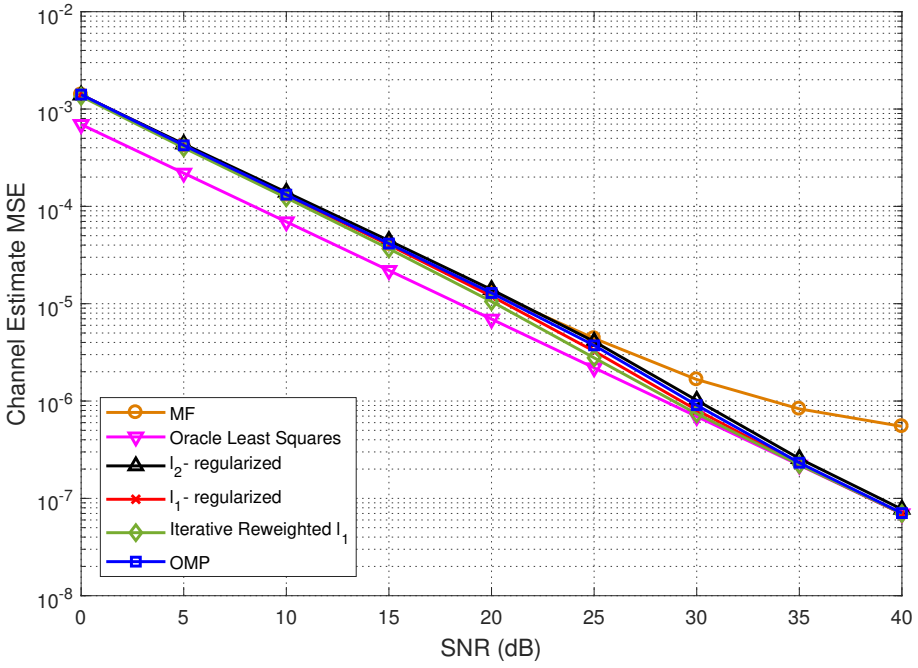

FIGURE B.2: MSE performance for different joint timing synchronization and channel estimation methods, considering Zadoff-Chu sequences and $N_c = 64$.

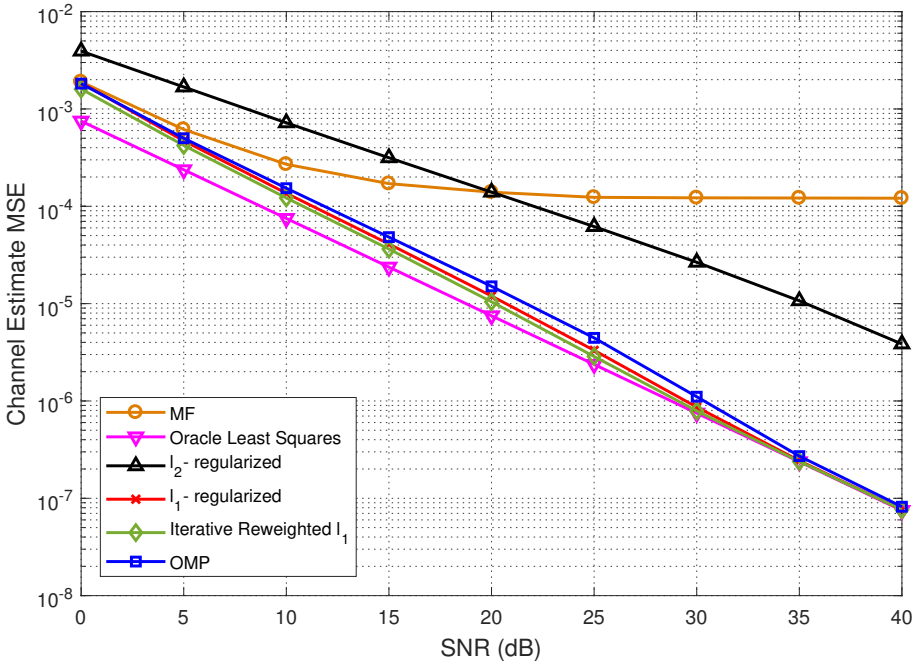

FIGURE B.3: MSE performance for different joint timing synchronization and channel estimation methods, considering pseudorandom codewords and $N_c = 64$.

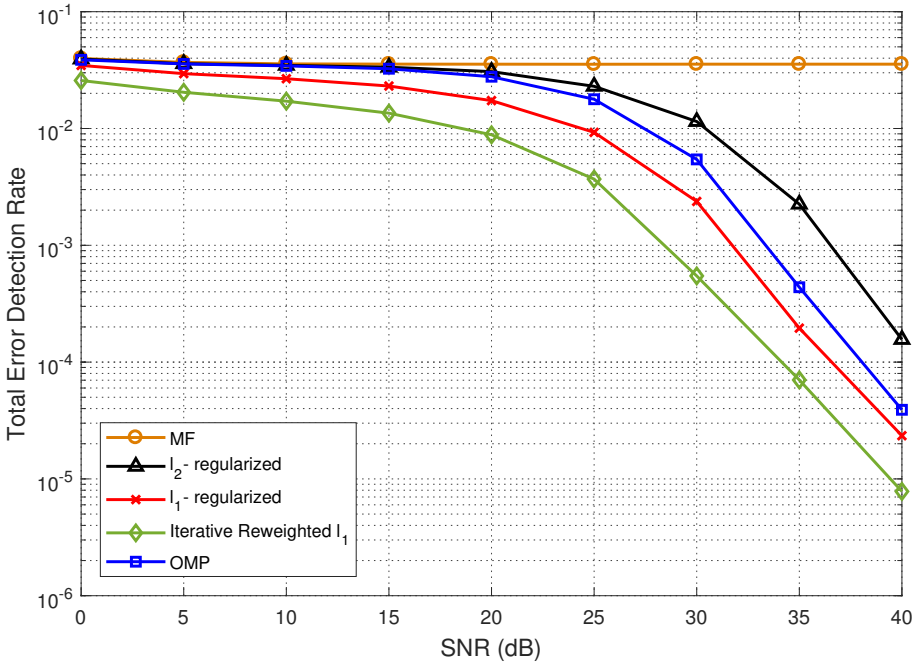

FIGURE B.4: Probability of incorrect multipath positions (timing) for different approaches, considering TCH codes and $N_c = 64$.

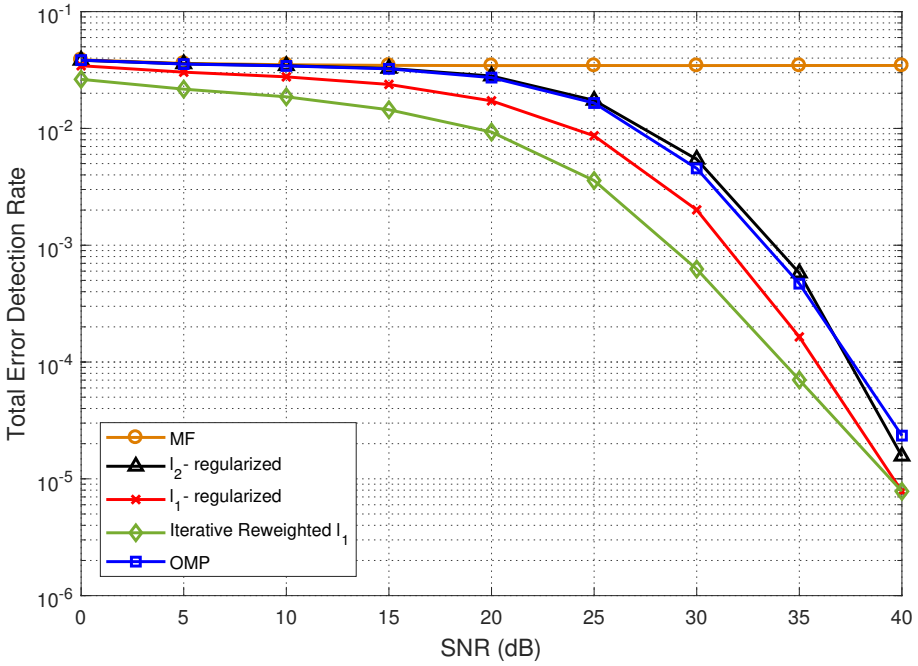

FIGURE B.5: Probability of incorrect multipath positions (timing) for different approaches, considering Zadoff-Chu sequences and $N_c = 64$.

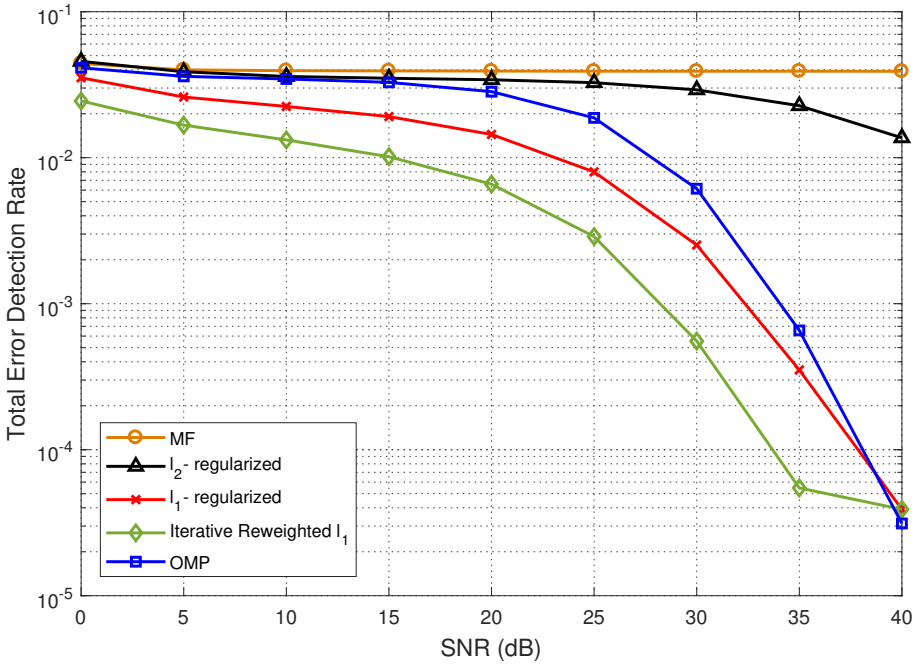

FIGURE B.6: Probability of incorrect multipath positions (timing) for different approaches, considering pseudorandom codewords and $N_c = 64$.

B.2 Effects of Different Channel Lengths: $N_c = 16$ and $N_c = 64$

This section shows the effect of using different channel lengths, L, for different pilot sizes. Figures B.7 to B.12 show MSE performance and total error rate for $N_c = 16$ and Figures B.13 to B.18 illustrate the same but for $N_c = 64$. Results from both pilot sizes show the same trend as in subsection 4.3.2. As expected, these results are slightly worse due to using smaller pilot sizes.

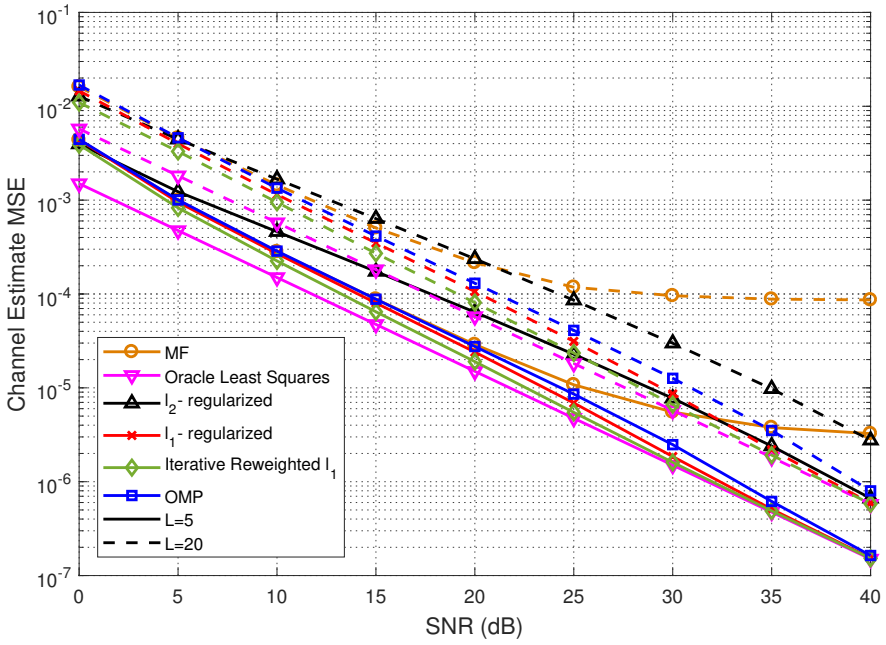

FIGURE B.7: MSE performance for different joint timing synchronization and channel estimation methods, considering TCH codes, different channel length values and $N_c = 16$.

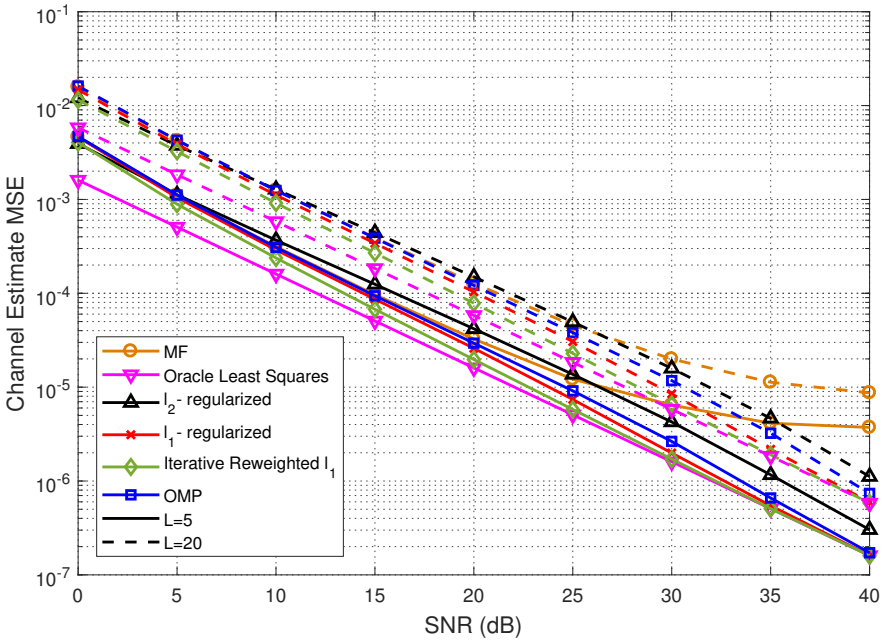

FIGURE B.8: MSE performance for different joint timing synchronization and channel estimation methods, considering Zadoff-Chu sequences, different channel length values and $N_c = 16$.

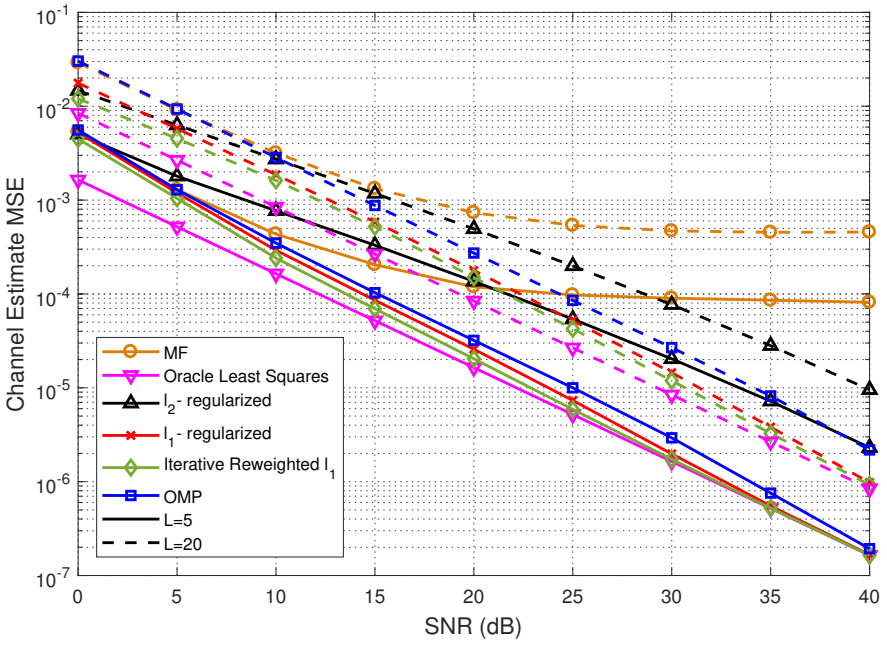

FIGURE B.9: MSE performance for different joint timing synchronization and channel estimation methods, considering pseudorandom codewords, different channel length values and $N_c = 16$.

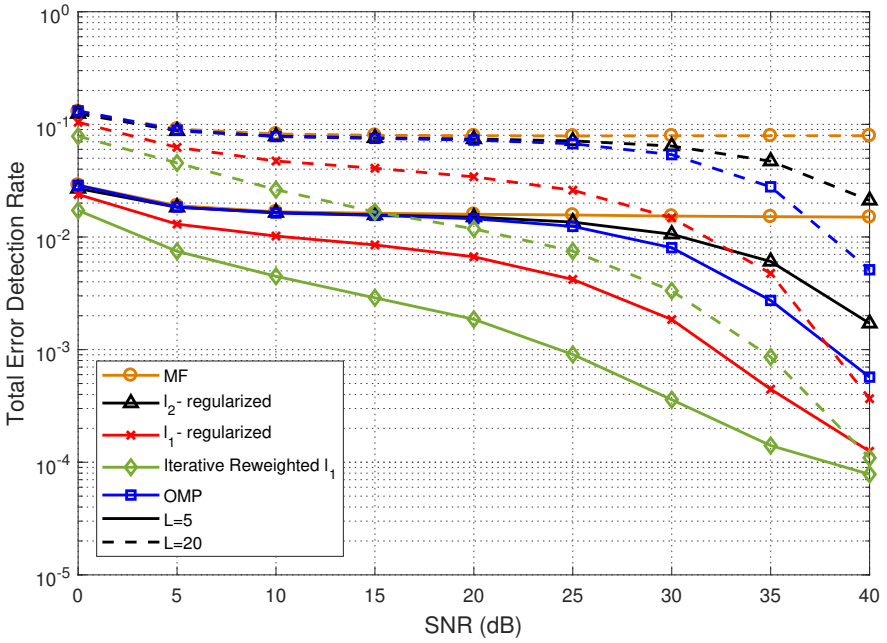

FIGURE B.10: Probability of incorrect multipath positions (timing) for different approaches, considering TCH codes, different channel length values and $N_c = 16$.

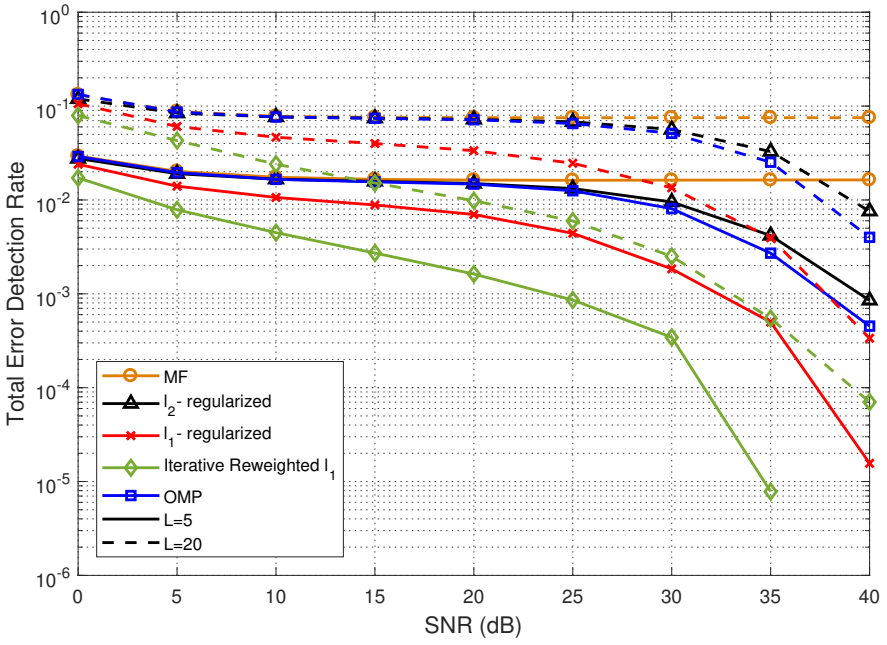

FIGURE B.11: Probability of incorrect multipath positions (timing) for different approaches, considering Zadoff-Chu sequences, different channel length values and $N_c = 16$.

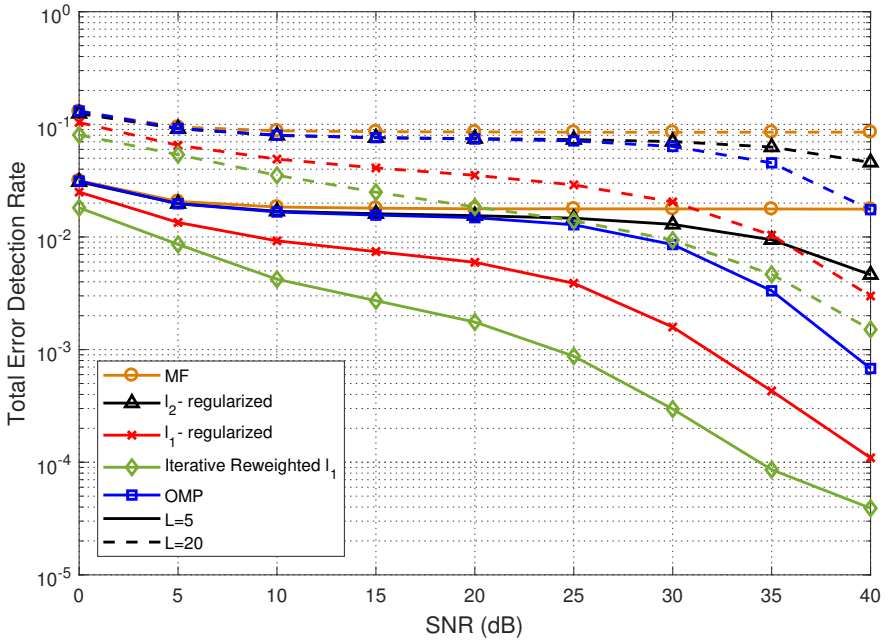

FIGURE B.12: Probability of incorrect multipath positions (timing) for different approaches, considering pseudorandom codewords, different channel length values and $N_c = 16$.

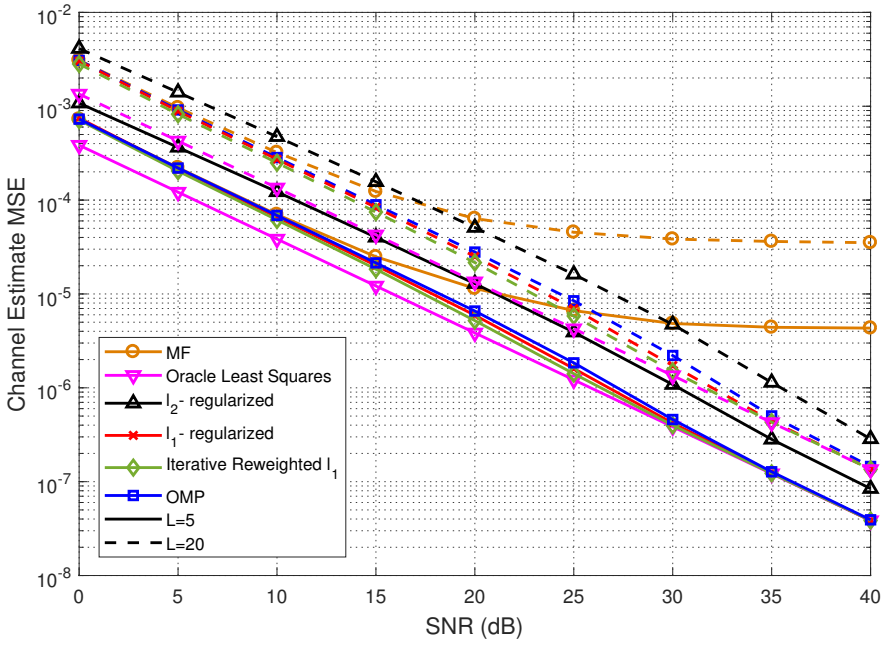

FIGURE B.13: MSE performance for different joint timing synchronization and channel estimation methods, considering TCH codes, different channel length values and $N_c = 64$.

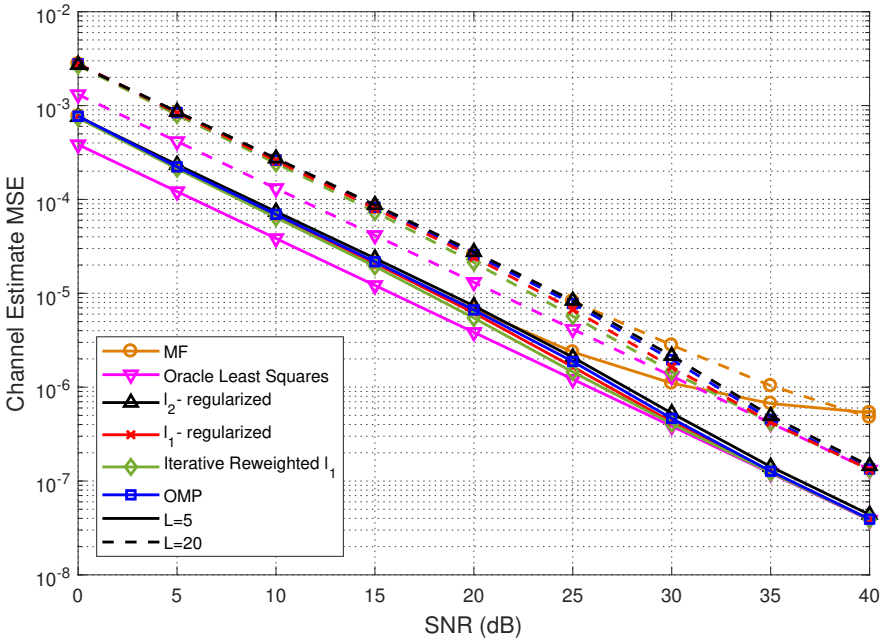

FIGURE B.14: MSE performance for different joint timing synchronization and channel estimation methods, considering Zadoff-Chu sequences, different channel length values and $N_c = 64$.

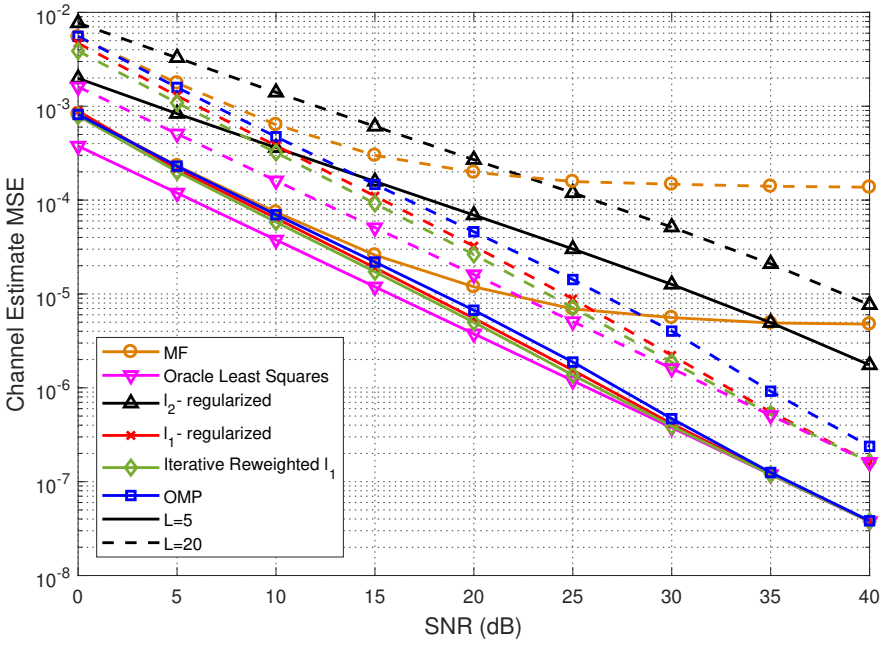

FIGURE B.15: MSE performance for different joint timing synchronization and channel estimation methods, considering pseudorandom codewords, different channel length values and $N_c = 64$.

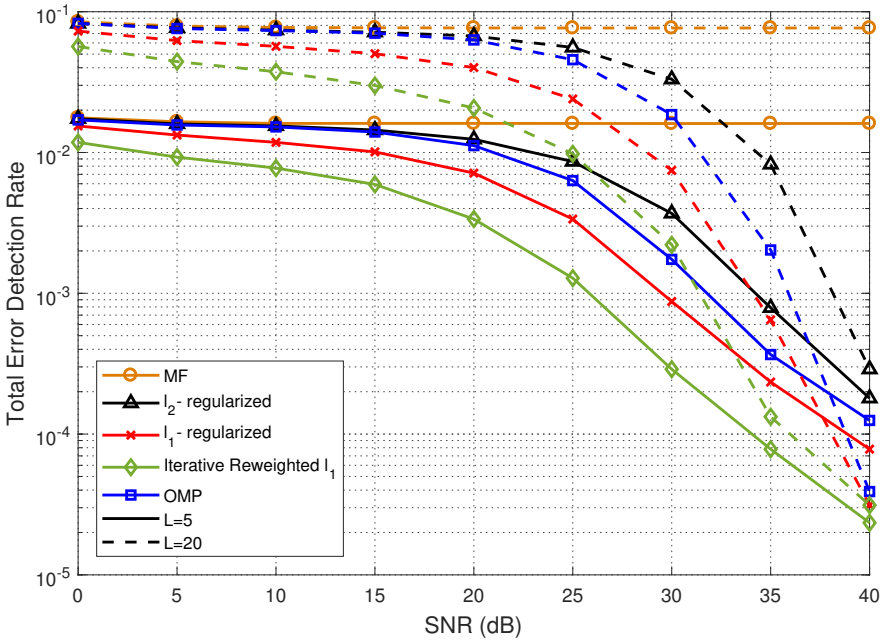

FIGURE B.16: Probability of incorrect multipath positions (timing) for different approaches, considering TCH codes, different channel length values and $N_c = 64$.

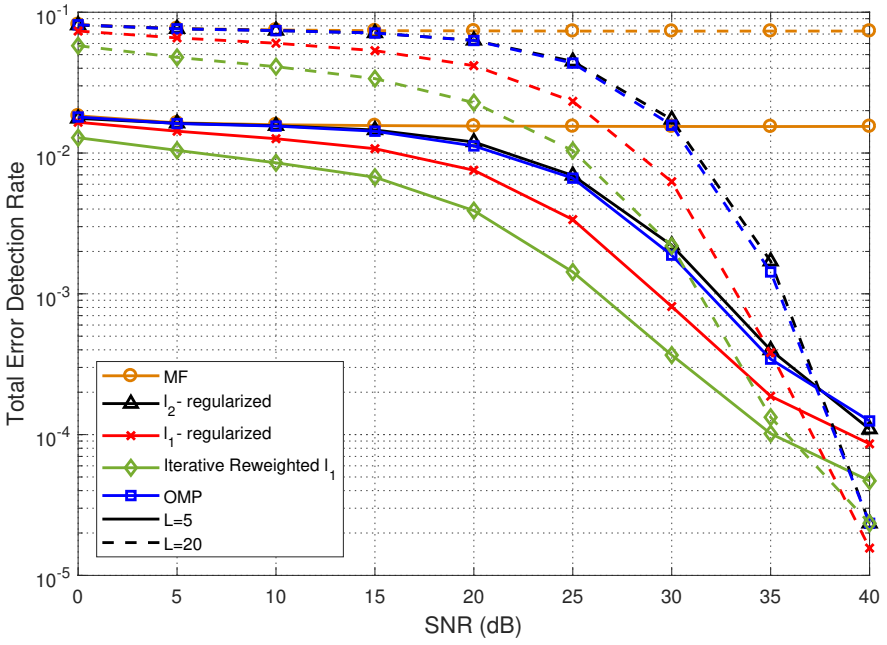

FIGURE B.17: Probability of incorrect multipath positions (timing) for different approaches, considering Zadoff-Chu sequences, different channel length values and $N_c = 64$.

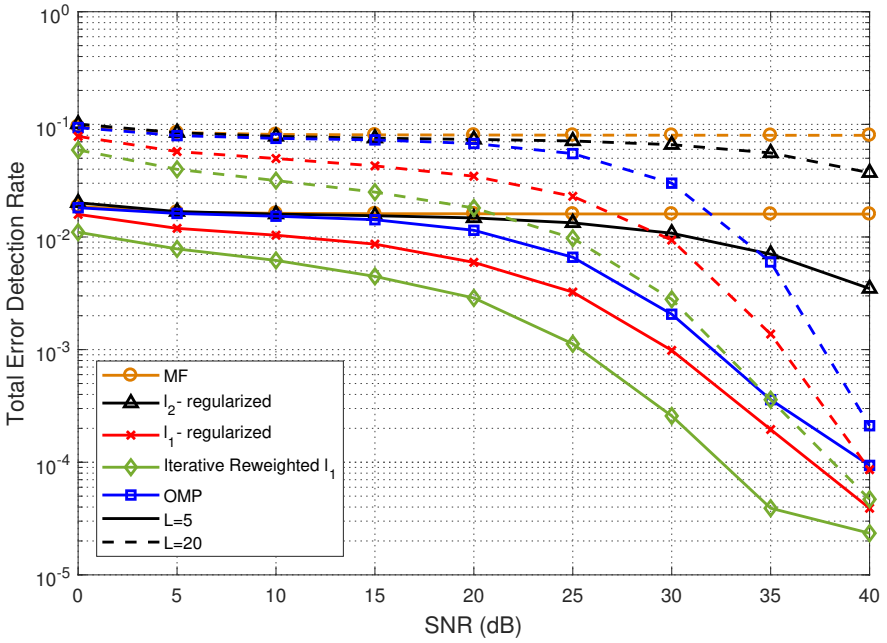

FIGURE B.18: Probability of incorrect multipath positions (timing) for different approaches, considering pseudorandom codewords, different channel length values and $N_c = 64$.

B.3 Effects of Different Observation Window Sizes: $M_o = N_s/2$

This last section goes over the usage of an observation window of size $M_o = N_s/2$. This was the smallest window used in this dissertation and shows how even Iterative Reweighted ℓ_1 and ℓ_1- regularized, the two methods that showed the best performance throughout all these simulations, cannot cope with such a small window and present very similar results to OMP and ℓ_2- regularized.

Figures B.19 to B.21 illustrate channel estimation MSE performance while Figures B.22 to B.24 depict synchronization performance.

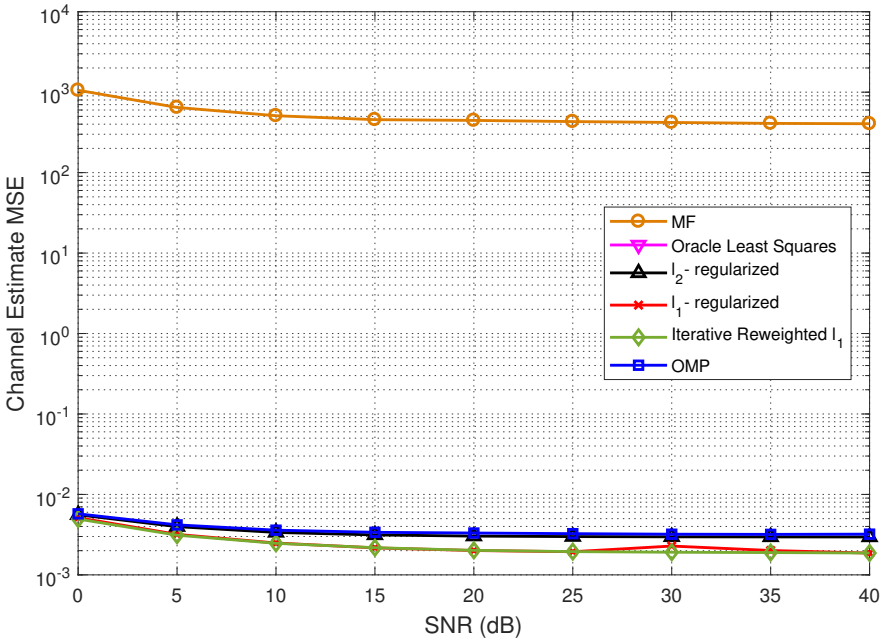

FIGURE B.19: MSE performance for different joint timing synchronization and channel estimation methods, considering TCH codes and $M_o = N_s/2$.

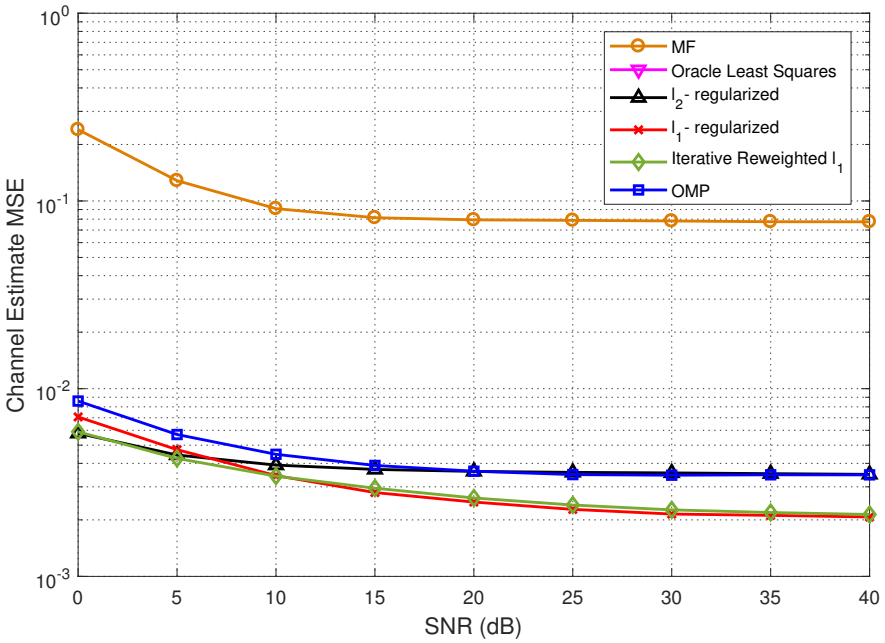

FIGURE B.20: MSE performance for different joint timing synchronization and channel estimation methods, considering Zadoff-Chu sequences and $M_o = N_s/2$.

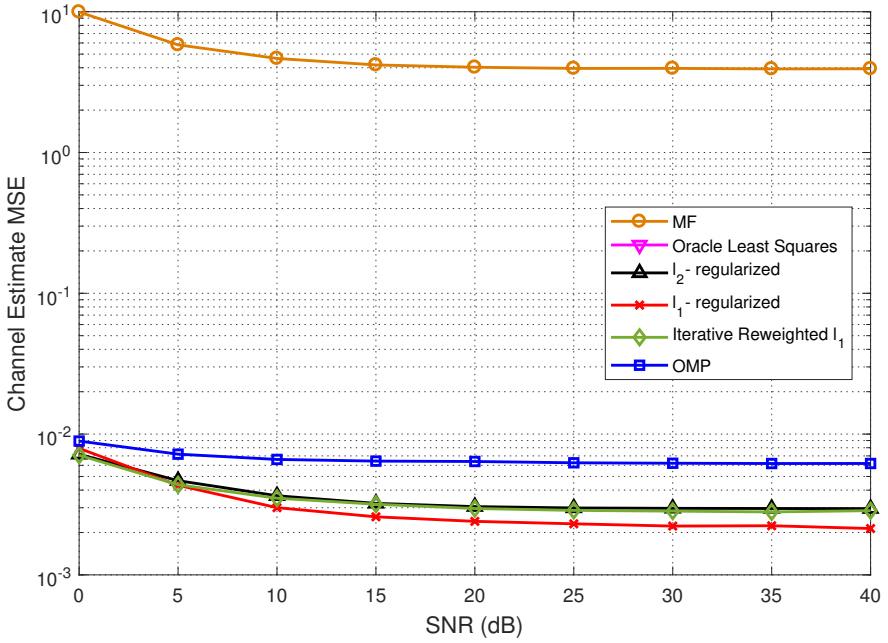

FIGURE B.21: MSE performance for different joint timing synchronization and channel estimation methods, considering pseudorandom codewords and $M_o = N_s/2$.

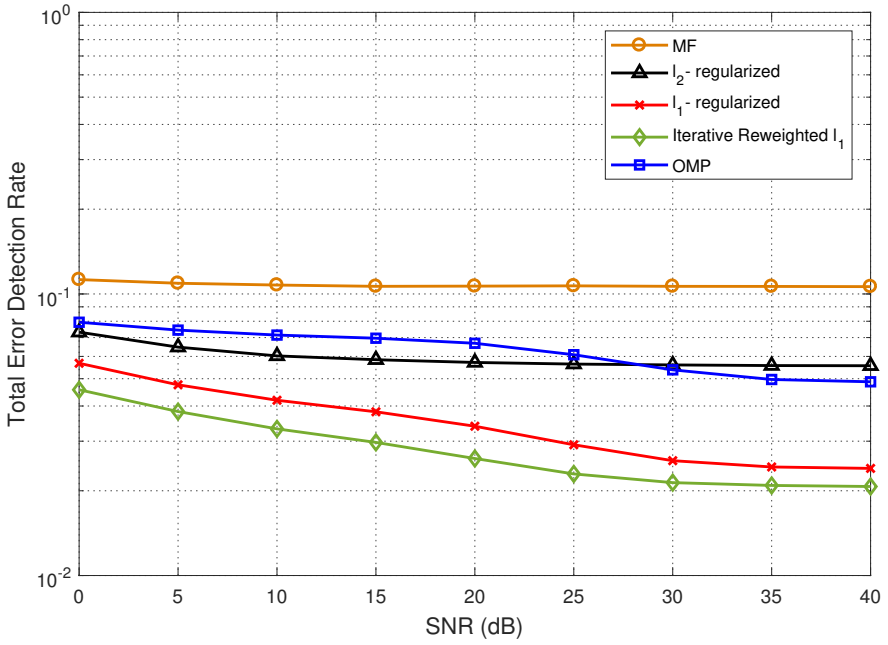

FIGURE B.22: Probability of incorrect multipath positions (timing) for different approaches, considering TCH codes and $M_o = N_s/2$.

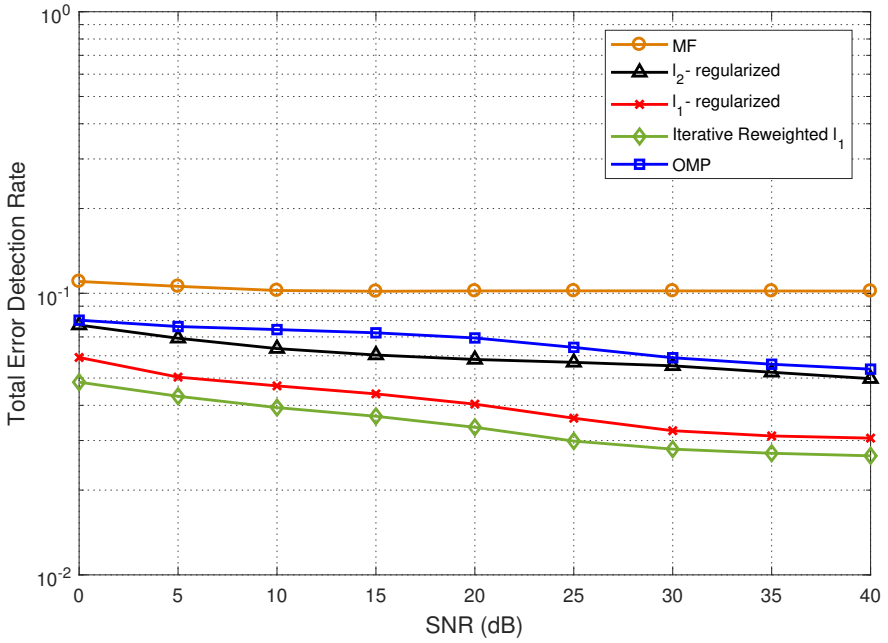

FIGURE B.23: Probability of incorrect multipath positions (timing) for different approaches, considering Zadoff-Chu sequences and $M_o = N_s/2$.

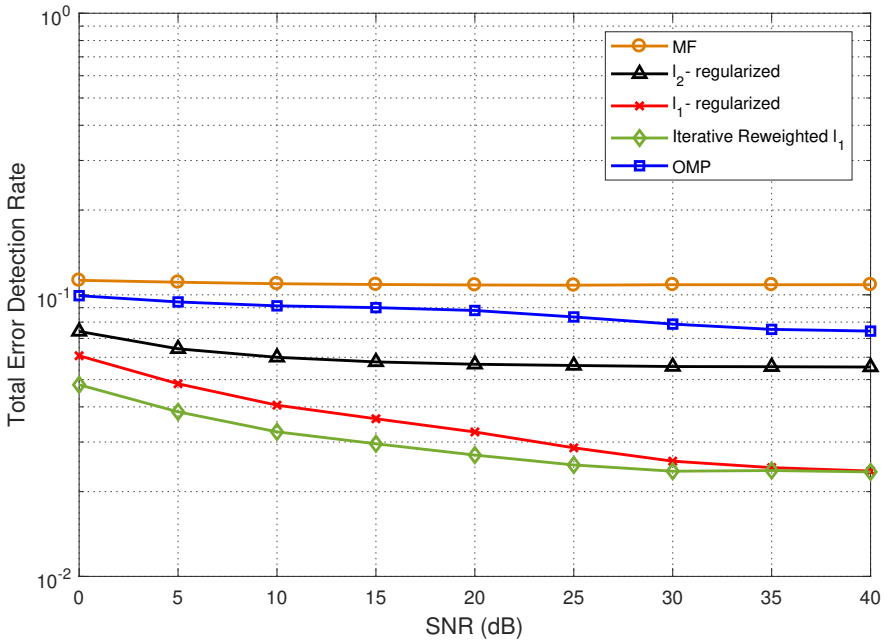

FIGURE B.24: Probability of incorrect multipath positions (timing) for different approaches, considering pseudorandom codewords and $M_o = N_s/2$.

Bibliography

[1] F Cercas. TCH. A new family of cyclic codes length 2m. *Proceedings of the 1993 IEEE International Symposium on Information Theory*, 1993.

[2] N S Souto, J C Silva, F Cercas, A Almeida, and A Correia. Synchronization with TCH Codes. In *Proc Wireless Personal Multimedia Communications Symp.-WPMC*, pages 998–1002, 2005.

[3] Enrico Del Re and Laura Pierucci. *Satellite Personal Communications for Future-generation Systems: Final Report: COSY 252 Action*. Springer Science & Business Media, 2012.

[4] Kurt Aretz, Martin Haardt, Walter Konhäuser, and Werner Mohr. The Future of Wireless Communications Beyond the Third Generation. *Computer Networks*, 37(1):83–92, 2001.

[5] Hamidreza Shariatmadari, Rapeepat Ratasuk, Sassan Iraji, Andrés Laya, Tarik Taleb, Riku Jäntti, and Amitava Ghosh. Machine-type communications: Current status and future perspectives toward 5G systems. *IEEE Communications Magazine*, 53(9):10–17, 2015.

[6] Tobias Ryberg. The global wireless M2M market. *Berg Insight*, 2009.

[7] Francisco António Bucho Cercas. *A new family of codes for simple receiver implementation*. PhD thesis, 1996.

[8] João Carlos Silva, Rui Dinis, Nuno Souto, and Paulo Montezuma. On the use of TCH sequences for Synchronization, Noise and Channel Estimation. *rd*

International Conference on Signal Processing and Communication Systems, pages 1–5, 2009.

[9] João Carlos Silva, Hugo Silva, Rui Dinis, Eunice Gomes, and Nuno Souto. On the use of TCH sequences for synchronization and channel estimation in MIMO systems. *4th International Conference on Signal Processing and Communication Systems, ICSPCS'2010 - Proceedings*, 10, 2010.

[10] Y Li. Optimum Training Sequences for OFDM Systems with Multiple Transmit Antennas. *Proc. of Global Telecommunications Conference*, 3:1478–1482, 2000.

[11] Anna Scaglione and Azadeh Vosoughi. Turbo estimation of channel and symbols in precoded MIMO systems. *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference*, 4:413–416, 2004.

[12] Savitri Galih, Riafeni Karlina, Ade Irawan, Trio Adiono, Adit Kurniawan, and others. Low complexity partial sampled MMSE channel estimation for downlink OFDMA IEEE 802.16 e system. In *Intelligent Signal Processing and Communication Systems, 2009. ISPACS 2009. International Symposium on*, pages 162–166. IEEE, 2009.

[13] Baoguo Yang, Zhigang Cao, and Khaled Ben Letaief. Analysis of low-complexity windowed DFT-based MMSE channel estimator for OFDM systems. *IEEE Transactions on Communications*, 49(11):1977–1987, 2001.

[14] Mehrzad Biguesh and Alex B. Gershman. Downlink channel estimation in cellular systems with antenna arrays at base stations using channel probing with feedback. *Eurasip Journal on Applied Signal Processing*, 2004(9):1330–1339, 2004.

[15] Xiaoli Ma, G.B. Giannakis, and S. Ohno. Optimal training for block transmissions over doubly selective wireless fading channels. *IEEE Transactions on Signal Processing*, 51(5):1351–1366, 2003.

[16] Ye Li. Pilot-symbol-aided channel estimation for OFDM in wireless systems. *IEEE transactions on vehicular technology*, 49(4):1207–1215, 2000.

[17] Chan-Tong Lam, David D Falconer, Florence Danilo-Lemoine, and Rui Dinis. Channel estimation for SC-FDE systems using frequency domain multiplexed pilots. In *Vehicular Technology Conference, 2006. VTC-2006 Fall. 2006 IEEE 64th*, pages 1–5. IEEE, 2006.

[18] Yonghong Zeng and Tung Sang Ng. Pilot cyclic prefixed single carrier communication: Channel estimation and equalization. *IEEE Signal Processing Letters*, 12(1):56–59, 2005.

[19] Mario Marques da Silva, Americo Correia, Rui Dinis, Nuno Souto, and Joao Carlos Silva. *Transmission techniques for emergent multicast and broadcast systems*. CRC Press, 2010.

[20] Ning Chen and Guo Tong Zhou. Superimposed training for OFDM: a peak-to-average power ratio analysis. *IEEE Transactions on Signal Processing*, 54(6):2277–2287, 2006.

[21] Jitendra K Tugnait and Xiaohong Meng. On superimposed training for channel estimation: performance analysis, training power allocation, and frame synchronization. *IEEE Transactions on Signal Processing*, 54(2):752–765, 2006.

[22] Nuno Souto, Rui Dinis, and João Carlos Silva. Efficient channel estimation for OFDM systems with hierarchical constellations. In *Communications and Information Technologies, 2007. ISCIT'07. International Symposium on*, pages 998–1002. IEEE, 2007.

[23] Robert W. Chang. Synthesis of Band-Limited Orthogonal Signals for Multichannel Data Transmission. *Bell System Technical Journal*, 45(10):1775–1796, 1966.

[24] S. B. Weinstein and Paul M. Ebert. Data Transmission by Frequency-Division Multiplexing Using the Discrete Fourier Transform. *IEEE Transactions on Communication Technology*, 19(5):628–634, 1971.

[25] Abraham Peled and Antonio Ruiz. Frequency domain data transmission using reduced computational complexity algorithms. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'80.*, volume 5, pages 964–967. IEEE, 1980.

[26] B Saltzberg. Performance of an efficient parallel data transmission system. *IEEE Transactions on Communication Technology*, 15(6):805–811, 1967.

[27] Chin Keong Ho, Behrouz Farhang-Boroujeny, and F Chin. Added pilot semi-blind channel estimation scheme for OFDM in fading channels. In *Global Telecommunications Conference, 2001. GLOBECOM'01. IEEE*, volume 5, pages 3075–3079. IEEE, 2001.

[28] Richard Van Nee, Geert Awater, Masahiro Morikura, Hitoshi Takanashi, Mark Webster, and Karen W. Halford. New high-rate wireless LAN standards. *IEEE Communications Magazine*, 37(12):82–88, 1999.

[29] Richard Van Nee and Ramjee Prasad. *OFDM for Wireless Multimedia Communications*. Artech House, Inc., Norwood, MA, USA, 1st edition, 2000.

[30] Richard Van Nee. A New OFDM Standard for High Rate Wireless Lan in the 5 GHz Band. *Signal Processing*, pages 258–262, 1999.

[31] Leonard Cimini. Analysis and simulation of a digital mobile channel using orthogonal frequency division multiplexing. *IEEE transactions on communications*, 33(7):665–675, 1985.

[32] Shobhakar Dhakal. Performance Analysis of OFDM over AWGN and Rayleigh Channels. [Online]. Available: http://www.slideshare.net/SarojDhakal1/ofdm-performance-analysis – [visited in January 2017].

[33] Hyung G.Myung. Cyclic Prefix representation. Online. https://commons.wikimedia.org/wiki/File:Cyclic_prefix.png - [visited in September 2017].

[34] Kwang Cheng Chen and Shao Yu Lien. Machine-to-machine communications: Technologies and challenges. *Ad Hoc Networks*, 18:3–23, 2014.

[35] Kan Zheng, Suling Ou, Jesus Alonso-Zarate, Mischa Dohler, Fei Liu, and Hua Zhu. Challenges of massive access in highly dense LTE-advanced networks with machine-to-machine communications. *IEEE Wireless Communications*, 21(3):12–18, 2014.

[36] Weiping Sun, Okhwan Lee, Yeonchul Shin, Seongwon Kim, Changmok Yang, Hyoil Kim, and Sunghyun Choi. Wi-Fi could be much more. *Communications Magazine, IEEE*, 52(11):22–29, 2014.

[37] Thomas Pötsch, Safdar Nawaz Khan Khan Marwat, Yasir Zaki, and Carmelita Gorg. Influence of future M2M communication on the LTE system. In *Wireless and Mobile Networking Conference (WMNC), 2013 6th Joint IFIP*, pages 1–4. IEEE, 2013.

[38] 3GPP. Service requirements for Machine-Type Communications (MTC). *3rd Generation Partnership Project*, TS 22.368(v11.6.0), 2012.

[39] Abdelmohsen Ali, Walaa Hamouda, and Murat Uysal. Next generation M2M cellular networks: challenges and practical considerations. *IEEE Communications Magazine*, 53(9):18–24, 2015.

[40] Edward Mutafungwa. Applying MTC and femtocell technologies to the continua health reference architecture. In *International Conference on Grid and Pervasive Computing*, pages 105–114. Springer, 2011.

[41] Tarik Taleb and Andreas Kunz. Machine type communications in 3GPP networks: potential, challenges, and solutions. *IEEE Communications Magazine*, 50(3), 2012.

[42] Massimo Condoluci, Mischa Dohler, Giuseppe Araniti, Antonella Molinaro, and Kan Zheng. Toward 5G densenets: Architectural Advances for Effective Machine-Type Communications over Femtocells. *IEEE Communications Magazine*, 53(1):134–141, 2015.

[43] T M Schmidl and D C Cox. Robust Frequency and Timing Synchroniation for OFDM. *IEEE Trans. on Communications*, 45(6):1613–1621, 1997.

[44] Malik Muhammad Usman Gul, Xiaoli Ma, and Sungeun Lee. Timing and frequency synchronization for OFDM downlink transmissions using Zadoff-Chu sequences. *IEEE Transactions on Wireless Communications*, 14(3):1716–1729, 2015.

[45] Jonathan H. Manton. Optimal training sequences and pilot tones for OFDM systems. *IEEE Communications Letters*, 5(4):151–153, 2001.

[46] Mehmet Kemal Ozdemir and Huseyin Arslan. Channel Estimation for Wireless OFDM Systems. *IEEE Communications Surveys & Tutorials*, 9(2):18–48, 2007.

[47] E.G. Larsson and G.B. Giannakis. Joint symbol timing and channel estimation for OFDM based WLANs. *IEEE Communications Letters*, 5(8):325–327, 2001.

[48] V.K. Bhargava and K.B. Letaief. A combined timing and frequency synchronization and channel estimation for OFDM. *IEEE Transactions on Communications*, 54(3):416–422, 2006.

[49] Yanyan Zhang, Jianhua Zhang, and Minghua Xia. Joint timing synchronization and channel estimation for OFDM systems via MMSE criterion. *IEEE Vehicular Technology Conference*, pages 1–4, 2008.

[50] V. Srivastava, Chin Keong Ho Chin Keong Ho, Patrick Ho Wang Fung Patrick Ho Wang Fung, and Sumei Sun Sumei Sun. Robust MMSE channel estimation

in OFDM systems with practical timing synchronization. *2004 IEEE Wireless Communications and Networking Conference (IEEE Cat. No.04TH8733)*, 2(April):0–5, 2004.

[51] Xinwei Du, Jing Zhang, Yan Li, Changyuan Yu, and Pooi-Yuen Kam. Efficient joint timing and frequency synchronization algorithm for coherent optical OFDM systems. *Optics express*, 24(17):19969–19977, 2016.

[52] David L Donoho. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006.

[53] Emmanuel J Candès, Justin Romberg, and Terence Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on information theory*, 52(2):489–509, 2006.

[54] Yagyensh Chandra Pati, Ramin Rezaiifar, and Perinkulam Sambamurthy Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on*, pages 40–44. IEEE, 1993.

[55] Stéphane G Mallat and Zhifeng Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on signal processing*, 41(12):3397–3415, 1993.

[56] Joel A Tropp and Anna C Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on information theory*, 53(12):4655–4666, 2007.

[57] S S Chen and David L Donoho. Examples of basis pursuit. *Proceedings of Wavelet Applications in Signal and Image Processing III (San Diego, CA)*, 1995.

[58] Emmanuel J Candes and Terence Tao. Decoding by linear programming. *IEEE transactions on information theory*, 51(12):4203–4215, 2005.

[59] David L Donoho. High-dimensional centrally symmetric polytopes with neighborliness proportional to dimension. *Discrete & Computational Geometry*, 35(4):617–652, 2006.

[60] Benjamin Franklin Logan. *Properties of high-pass signals.* PhD thesis, Columbia University, 1965.

[61] Emmanuel J Candes, Michael B Wakin, and Stephen P Boyd. Enhancing sparsity by reweighted 1 minimization. *Journal of Fourier analysis and applications*, 14(5):877–905, 2008.

[62] Karthik Mohan and Maryam Fazel. Iterative reweighted algorithms for matrix rank minimization. *Journal of Machine Learning Research*, 13(Nov):3441–3473, 2012.

[63] Dennis Sundman. *Compressed Sensing Algorithms and Applications Licentiate Thesis in Telecommunications.* 2012.

[64] Michael Lustig, Juan M Santos, Jin-Hyung Lee, David L Donoho, and John M Pauly. Application of compressed sensing for rapid MR imaging. *SPARS,(Rennes, France)*, 2005.

[65] Farzad Parvaresh, Haris Vikalo, Sidhant Misra, and Babak Hassibi. Recovering sparse signals using sparse measurement matrices in compressed DNA microarrays. *IEEE Journal of Selected Topics in Signal Processing*, 2(3):275–285, 2008.

[66] Richard Baraniuk and Philippe Steeghs. Compressive radar imaging. In *Radar Conference, 2007 IEEE*, pages 128–133. IEEE, 2007.

[67] Matthew A Herman and Thomas Strohmer. High-resolution radar via compressed sensing. *IEEE transactions on signal processing*, 57(6):2275–2284, 2009.

[68] Christian R Berger, Zhaohui Wang, Jianzhong Huang, and Shengli Zhou. Application of compressive sensing to sparse channel estimation. *IEEE Communications Magazine*, 48(11), 2010.

[69] Georg Taubock and Franz Hlawatsch. A compressed sensing technique for OFDM channel estimation in mobile environments: Exploiting channel sparsity for reducing pilots. In *Acoustics, speech and signal processing, 2008. ICASSP 2008. IEEE international conference on*, pages 2885–2888. IEEE, 2008.

[70] Jun Won Choi, Byonghyo Shim, Yacong Ding, Bhaskar Rao, and Dong In Kim. Compressed sensing for wireless communications: Useful tips and tricks. *IEEE Communications Surveys & Tutorials*, 2017.

[71] 3GPP. Multiplexing and Channel Coding (FDD). *3rd Generation Partnership Project*, TS 25.212(v13.1.0), 2016.

[72] Bruno Lopes, Sílvia Catarino, Francisco Cercas, Nuno Souto, and Rui Dinis. Efficient Channel Estimation Using TCH Codes. In *International Congress on Ultra Modern Telecommunications and Control Systems*, pages 1–5, 2017.

[73] Scott Shaobing Chen, David L Donoho, and Michael A Saunders. Atomic decomposition by basis pursuit. *SIAM review*, 43(1):129–159, 2001.

[74] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[75] Joel A Tropp and Stephen J Wright. Computational methods for sparse solution of linear inverse problems. *Proceedings of the IEEE*, 98(6):948–958, 2010.

[76] Zheng Zhang, Yong Xu, Jian Yang, Xuelong Li, and David Zhang. A survey of sparse representation: algorithms and applications. *IEEE access*, 3:490–530, 2015.