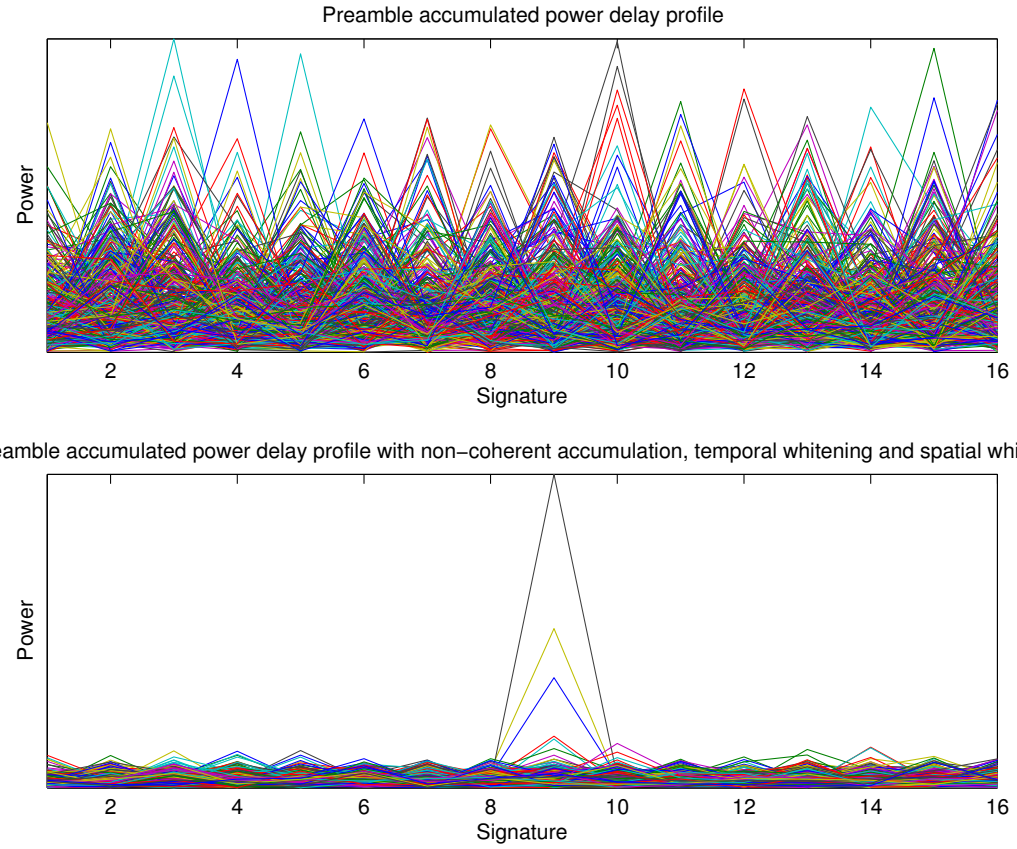
CHALMERS
UNIVERSITY OF TECHNOLOGY



Designing and evaluating an FFT-based RACH preamble detection algorithm

Including temporal whitening and spatial whitening

Master's thesis in Communication Engineering

ANDREAS BRING
KIM ROSBERG

Department of Signals and Systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2015

Master's thesis 2015:06

Designing and evaluating an FFT-based RACH preamble detection algorithm

Including temporal whitening and spatial whitening

Andreas Bring
Kim Rosberg

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Signals and Systems
Division of Communication systems
Communication systems research group
Chalmers University of Technology
Gothenburg, Sweden 2015

Designing and optimizing an FFT-based RACH preamble detection algorithm
Including temporal whitening and spatial whitening
ANDREAS BRING
KIM ROSBERG

© ANDREAS BRING AND KIM ROSBERG, 2015.

Advisor: Rahul Devassy, Chalmers University of Technology, Department of Signals and Systems
Supervisor: Magnus Nilsson, Ericsson
Supervisor: Göran Kronquist, Ericsson
Supervisor: Anders Åström, Ericsson
Examiner: Tommy Svensson, Chalmers University of Technology, Department of Signals and Systems

Master's Thesis 2015:06
Department of Signals and Systems
Division of Communication systems
Communication systems research group
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Preamble correlation peaks in terms of power and signature.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2015

Designing and optimizing an FFT-based RACH preamble detection algorithm
ANDREAS BRING
KIM ROSBERG
Department of Signals and Systems
Chalmers University of Technology

Abstract

An ever increasing demand for higher data rates in mobile telecommunications fuels a need for refinement of algorithms used in modern day mobile telecommunications technologies. The radio interface used in many parts of the world is called Wideband Code Division Multiple Access WCDMA. WCDMA utilizes a shared channel called the Random-Access Channel (RACH) for handling requests for setting up a connection between a User Equipment (UE) and a Radio Base Station (RBS). The physical random-access procedure is initiated when certain requirements are fulfilled and a preamble is sent to initialize the communication [1]. When the preamble is acknowledged by the RBS a random access message is sent, thus finishing the random access procedure. This thesis centers on the development, optimization and testing of an alternative low complexity algorithm for the preamble detection procedure on a RACH in a WCDMA system. Evaluation of the algorithm is done in a WCDMA RACH simulator at Ericsson.

Firstly, a Fast Fourier Transform (FFT)-based baseline algorithm has been developed to match the performance of the currently implemented time-domain baseline algorithm. The FFT-based algorithm demonstrates identical performance to the time-domain algorithm and has therefore been used as the foundation for the consecutive refinements of the algorithm. Secondly, a temporal whitening algorithm has been added to the FFT-based algorithm to improve performance in scenarios where a high rate data user is creating interference by transmitting simultaneously on the WCDMA enhanced uplink channel. The temporal whitening algorithm was designed to temporally whiten the preamble in four separate parts and to use 32 samples of the autocorrelation matrix. Thirdly, a spatial whitening algorithm has also been added to the FFT-based baseline algorithm, to improve performance when several antennas are used in the RBS. The algorithm demonstrates greatly improved performance with a maximum gain in detection probability of approximately 69 % using both temporal and spatial whitening compared to the baseline algorithm in the case of strong interference.

In conclusion, the performance of the preamble detection procedure in presence of interference can be significantly improved by performing receiver-side temporal whitening and spatial whitening. The inclusion of these algorithms does not impair the performance of preamble detection procedure in absence of interference.

Keywords: WCDMA, UE, RBS, RACH, preamble detection, temporal whitening, spatial whitening.

Acknowledgements

First of all, we would like to give our sincerest thanks to Magnus Nilsson, Göran Kronquist and Anders Åström our supervisors at Ericsson AB, who supported us through every part of the thesis work.

Furthermore, we are grateful to Johnny Kemi for giving us the opportunity of carrying out this thesis work and for helping us with administrative matters for the duration of the thesis.

Moreover, we would like to thank our advisor at Chalmers, Rahul Devassy, for proof-reading the report and giving us an unbiased opinion on many matters concerning the report. Finally, we would like to extend our thanks to our examiner Assoc Prof Tommy Svensson for helping us finalize the report and for handling the administrative matters on Chalmers behalf.

Andreas Bring, Kim Rosberg
Gothenburg, June 2015

Contents

List of Figures

List of Tables

Abbreviations

AICH Acquisition Indicator Channel

AWGN Additive White Gaussian Noise

BCH Broadcast Control Channel

CDMA Code Division Multiple Access

DS-CDMA Direct-Sequence Code Division Multiple Access

FDMA Frequency Division Multiple Access

FFT Fast Fourier Transform

IDFT Inverse Discrete Fourier Transform

LOS Line-of-sight

MAI Multiple Access Interference

ODMA Opportunity-Driven Multiple Access

OFDMA Orthogonal Frequency Division Multiple Access

OVSF Orthogonal Variable Spreading Factor

PA Pedestrian A

PN Pseudo Noise

PRACH Physical Random Access Channel

PSD Power Spectral Density

RA Rural Area

RACH Random Access Channel

RBS Radio Base Station

SNR Signal-to-Noise Ratio

TDMA Time-Division Multiple Access

TU Typical Urban

UE User Equipment

VA Vehicular A

WCDMA Wideband Code Division Multiple Access

WTDMA Wideband Time Division Multiple Access

1
Introduction

1.1 Background

The use of electronic devices such as smartphones and tablets has brought new demands for mobile networks, such as being able to send emails, listen to music and stream video regardless of location. Today, there are approximately 1.3 billion smartphone users in the world [2]. The gradual worldwide adoption of these electronic devices causes a greater demand on data rate, network coverage and capacity. The increase in mobile usage puts a strain on mobile networks, which gives rise to a need to refine and enhance base-station hardware as well as software. Moreover, current mobile communication standards impose dedicated and finite spectrum allocation for each service provider, which calls for increased bandwidth efficiency. Consequently, all efforts to improve the new generations of mobile telecommunications technology are limited by finite bandwidth.

In 1992 The International Telecommunication Union declared that frequency bands adjacent to 2 GHz were restricted for the third generation mobile network. This declaration spawned an investigation to find a suitable technology for multiple radio access [3]. The proposals put forward were: Wideband Code Division Multiple Access (WCDMA), Wideband Time Division Multiple Access (WTDMA), Time Division Multiple Access (TDMA), Code Division Multiple Access, CDMA, Orthogonal Frequency Division Multiple Access (OFDMA) and Opportunity Driven Multiple Access (ODMA). Ericsson were one of the contributors to the WCDMA technology, which became used worldwide for numerous third generation mobile systems. The new generation of mobile telecommunications technology would later come to support not only conventional cellular voice, SMS- and MMS-services but also high-speed data transmission which could be used when for example streaming video.

Traffic on a WCDMA channel is separated through the use of codes. In downlink, these codes are unique for every user and allocates a slot on the channel specific to that user. Moreover, the codes are orthogonal to each other, which limits inter-user-interference. Establishing a connection in a WCDMA network is done in the following way; the User Equipment (UE) sends a preamble on the Physical Random Access Channel (PRACH), if the RBS detects the preamble, it sends an acknowledgement message on the Acquisition Indicator Channel (AICH). When the UE receives an acknowledgement, a message containing data and control signaling is sent to the Radio Base Station (RBS), which concludes the communication setup [4]. The Random Access Channel (RACH) is a shared channel and the RACH-procedures

are made to be very fast, with as little overhead as possible. The hardware components in an RBS dictate which types of algorithms can be implemented in preamble detectors. This includes hardware components such as FFT-accelerators. These accelerators introduces the possibility of having an FFT-based preamble detection implementation, which can speed up the detection-procedure. The outcome of which can lead to an improvement in terms of latency and coverage for the system.

1.2 Purpose

The purpose of this master thesis is to develop and evaluate an algorithm for FFT-based preamble detection in a RACH. Moreover, a temporal whitening algorithm and a spatial whitening algorithm are implemented to further improve upon the preamble detection procedure performance, in the presence of a high data rate user causing Multiple Access Interference (MAI). The implementation of the algorithm should lead to more efficient hardware usage, improved coverage and lower latency for a WCDMA system.

1.3 Objective

- Does signal processing, such as temporal whitening and spatial whitening, in the presence of interference further improve the access procedure?
- Does the alternate access procedure provide any gain in terms of reduced latency and coverage area?
- What is the difference in detection probability between a baseline preamble detection algorithm at Ericsson and the FFT-based one?
- Is the computational complexity reduced by utilizing an FFT-based preamble detection algorithm?

1.4 Scope

This thesis is limited to the preamble detection in a RACH as part of a telecommunication system that uses WCDMA. The signal is assumed to be modulated to baseband and gain controlled when it reaches the receiver. Thus the content of the thesis only considers the receiver-side operation on a shared PRACH-channel. The performance of the preamble detection algorithm is evaluated using non-dispersive and dispersive channels and additional interference simulating a high data rate user.

2

Introduction to WCDMA

WCDMA is one among the commonly used multiplexing techniques in a communication system. Multiplexing is the process of allowing simultaneous users to transmit over the same communication channel using signals separated by using time, frequency and/or code. Based on the multiplexing method we have three kinds of channel-access schemes namely time division multiple access (TDMA), frequency division multiple access (FDMA) and code division multiple access (CDMA). The benefit of using CDMA is allowing all users to access the entire frequency spectrum at any given time, thus allowing every user to use the whole bandwidth. In DS-CDMA the information bits are spread over a wider bandwidth by multiplying the data with predefined bits called spreading codes [5][6]. This channel access method, combined with frequency-division duplexing (FDD) for the receiver and transmitter, is the foundation of WCDMA. The channel allocations for a WCDMA system is shown in Fig. 2.1.

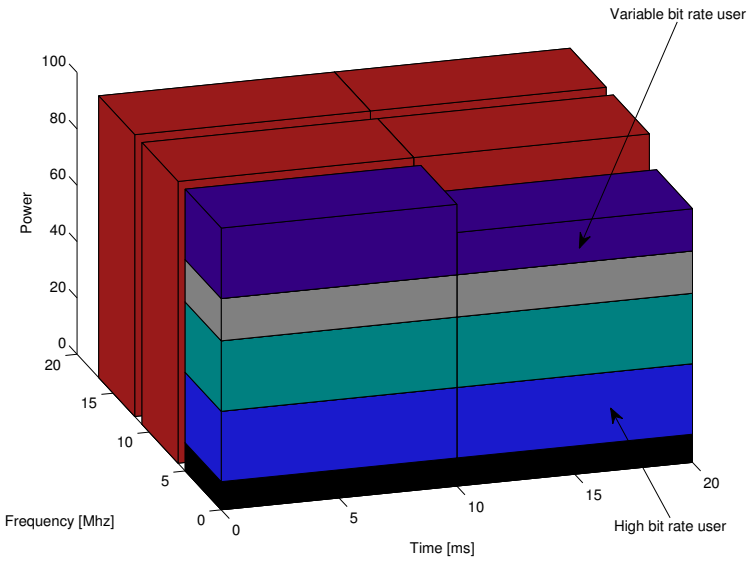


Figure 2.1: Bandwidth allocation in WCDMA in the time-frequency-code space, where different colors represent users with different codes

2.1 Spreading

In WCDMA, enabling several users to communicate on a shared frequency band simultaneously is accomplished by using spreading codes. The spreading procedure applies a unique spreading code on information bits, thus spreading the signal over a bandwidth greater than the original signal. The spread signal can later be despread by applying the same code sequence to the spread signal. Figure 2.2 exemplifies the

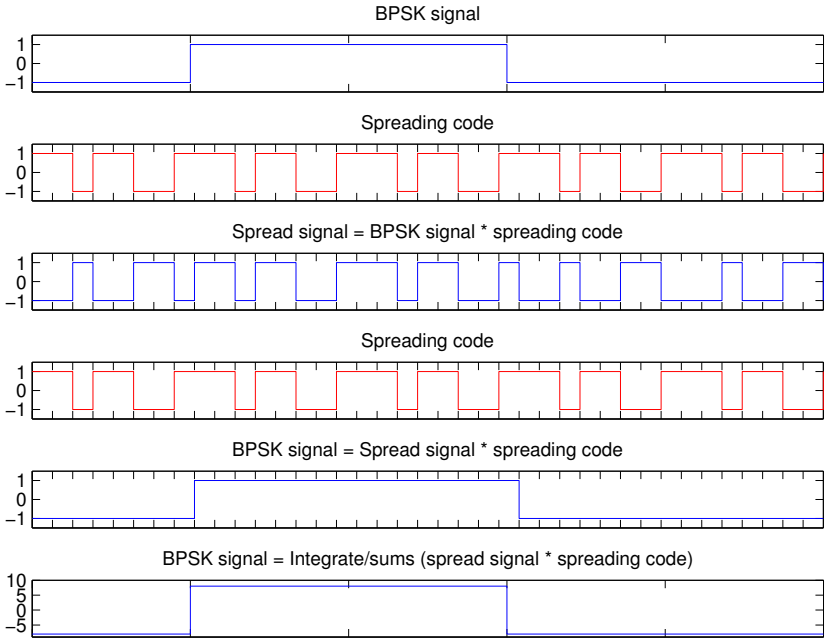


Figure 2.2: Spreading and despreading of a BPSK signal in DS-CDMA with a perfectly synchronized code. The upper half of the figure represents the spreading and the lower the despreading

process of spreading and despreading. In this example a BPSK (± 1) signal is used and the spreading is carried out by multiplication with an eight bit long code, which is known as a channelization code. The resulting spread signal has a spread factor of eight since the code is eight chips long, thus widening the bandwidth of the signal by a factor of eight. The resulting wideband signal is transmitted through a wireless channel and at the receiver side the despreading process can begin. In this example the desired despreading is applied by multiplying the replica of the spreading code, thus resulting in the original signal. After an integration, the amplitude of the signal increases as much as the spreading factor. Hence, the spreading procedure increases the processing gain by a factor of eight in this example. However, in this example perfect synchronization is assumed, which is often not the case in practice [6].

The process of spreading is done with two types of codes called channelization codes and scrambling codes, which are explained in detail in the subsequent section and

is conceptually shown in Fig. 2.3.

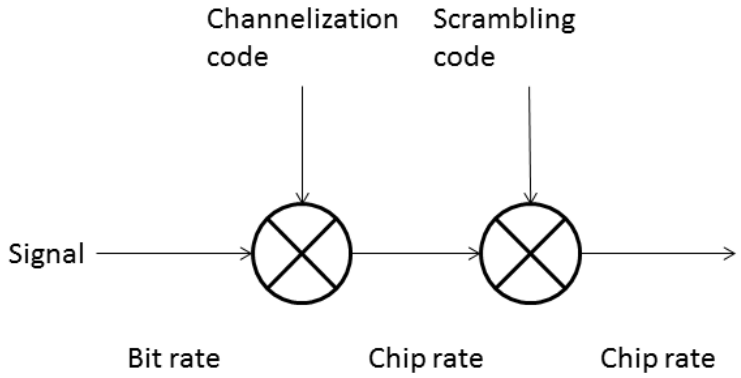


Figure 2.3: WCDMA spreading process.

2.1.1 Channelization Codes

Channelization codes are, in addition to increasing the transmission bandwidth, used to differentiate users within a cell in the downlink and separate the data channels sent from the UE to each cell [7]. The channelization codes are based on the Orthogonal Variable Spreading Factor (OVSF) technique. The use of OVSF enables the spreading factor to be changed whilst maintaining the orthogonality between different channelization codes. The codes are picked from a OVSF tree, which can be derived from a Hadamard matrix. A Hadamard matrix can be constructed using a Sylvester construction and is defined as follows. Given a Hadamard matrix of order 1: $H_1 = [1]$, a matrix of order 2^n can be constructed for all $n \in \mathbb{N}$ such that

$$H_{2^n} = \begin{pmatrix} H_{2^{n-1}} & H_{2^{n-1}} \\ H_{2^{n-1}} & -H_{2^{n-1}} \end{pmatrix} \Rightarrow H_{2^1} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, H_{2^2} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \quad (2.1)$$

and so on [8]. The spreading factor is closely related to the order of the Hadamard matrix, i.e., a matrix of order n yields a spreading factor of n. A fundamental problem with spread spectrum in WCDMA is that each user can cause Multiple Access Interference (MAI) and thereby affect all other users. Consequently, using coding-schemes with low cross-correlation and high auto-correlation properties are of great importance. Hence, orthogonal codes, such as the ones generated in the Hadamard matrix in equation (2.1), are preferable since the codes are characterized by having zero cross-correlation with each other in the case of ideal time synchronization. If two orthogonal codes $A = [+1 + 1 + 1 + 1]$ and $B = [+1 - 1 + 1 - 1]$ are multiplied, the resulting scalar equals

$$A \cdot B = 1 \cdot 1 + 1 \cdot (-1) + 1 \cdot 1 + 1 \cdot (-1) = 0$$

However, multiplying vector A with itself yields

$$A \cdot A = 1 \cdot 1 + 1 \cdot 1 + 1 \cdot 1 + 1 \cdot 1 = 4$$

which is a significant rise in magnitude, thus exemplifying the usefulness of orthogonal codes. However, orthogonality is only fully maintained when the codes are aligned in time. Thus, a shift in time can drastically affect the cross-correlation function, which consequently increases the MAI. This phenomenom may for example occur when transmissions are not synchronized or in the presence of multipath fading. In the case of multipath fading, the crosscorrelation-function may give more than one peak, hence making it harder to detect the point of maximum correlation and in turn tracking the beginning of the code sequence [9] [10].

2.1.2 Scrambling codes

The second operation of spreading is called scrambling. Scrambling codes are used to differentiate cells in the downlink and to separate users in the uplink. The transmission bandwidth is not affected by the scrambling operation and only serves to separate different sources from each other. Scrambling codes are defined from pseudo noise-codes (PN), since they have better auto-correlation and cross-correlation properties than orthogonal codes [10]. PN-codes, in contrast to orthogonal codes, are not as dependent on time alignment, which mitigates the negative effect of time-of-arrival delays. Another property of scrambling with PN-codes is that a scrambled signal adopts the characteristics of the PN-codes, thus resulting in a signal with random noise characteristics. There are two types of PN-codes used in WCDMA: long codes (Gold codes), and short codes (extended S(2) codes) [9]. Long codes consists of 38400 chips, while short codes only consists of 256 chips [11]. Both long and short codes are used in uplink transmissions. The full derivation of both long and short codes can be found in [9].

In downlink, only long codes are used and the code period is limited to 10 ms. The number of codes is limited to 512, to keep the cell search procedure from becoming too exhaustive for the RBS [7].

2.2 Processing gain

Processing gain is defined as the ratio between the transmission bandwidth and the information bandwidth. In other words, it is the ratio between the bandwidth of the spread signal and the non-spread signal, i.e., the spreading factor. It is defined as follows

$$G_p = \frac{B_{spr}}{B_{info}} \tag{2.2}$$

where B_{spr} is the spread signal bandwidth and B_{info} is the information signal bandwidth. This concept is exemplified in fig 2.2, where the processing gain is equal to eight. The processing gain defines significant system parameters such as the number of allowable users in the system, the degree of multipath effect reduction and the detectability of the signal. The processing gain is inherently dependent on the gain

of increasing the bandwidth of a CDMA-system. This is further explained in the Shannon-Hartley theorem which is defined as

$$C = B \log_2 \left(1 + \frac{S}{N}\right) \tag{2.3}$$

where C is the channel capacity in bits per second, B is the bandwidth of the channel in Hertz, S is the average received signal power over the bandwidth, N is the average noise or interference power over the bandwidth measured in watts and S/N is the SNR. This theorem establishes tradeoff, that is vital to the spread spectrum technique, which states: for a fixed channel capacity, an increase in bandwidth is counteracted by a decrease in Signal-to-noise ratio (SNR) [10]. Thus, the bandwidth increase induced by spreading lowers the SNR-requirement for a fixed channel capacity.

2.3 Random Access Channel

The Random Access Channel RACH is an uplink transport channel which carries control information from the RBS to the network, such as requests to set up a connection. The channel is also used to send packet data from the RBS to the network. The transmissions are made over the Physical Random Access Channel (PRACH)

2.3.1 Random access procedure

Before the random access procedure is initiated, the UE receives the following system information from the Broadcast Control Channel (BCH) transmitted by the RBS [1]

- The cell-specific preamble scrambling code
- The available random access signatures (channelization codes) and RACH sub-channels
- The spreading factor
- The message length (10 or 20 ms)
- Initial preamble power parameter
- The power ramping factor
- The maximum number of retransmissions parameter
- The AICH transmission timing parameter
- The power offset between the preamble and the message
- Transport format parameters

The random access procedure is then started by the UE deriving the available uplink access slots from the set of available RACH sub-channels. There are a total of twelve RACH sub-channels where each sub-channel consists of 15 access slots spaced 5120 chips apart. These access slots are used to coordinate the timing of the preamble transmissions. Hence, the UE selects one access slot based on a random algorithm, which is defined to give every access slot equal probability of being chosen. Subsequently, the UE selects one signature from the received set of available RACH signatures, where all signatures also have been given equal probability of being picked. After determining if the initial preamble power is above the minimum power level defined by the RBS, the preamble is transmitted using the selected access slot, signature, preamble transmission power and provided scrambling code [9]. If no acknowledgement is received on the AICH from the RBS, the UE repeats

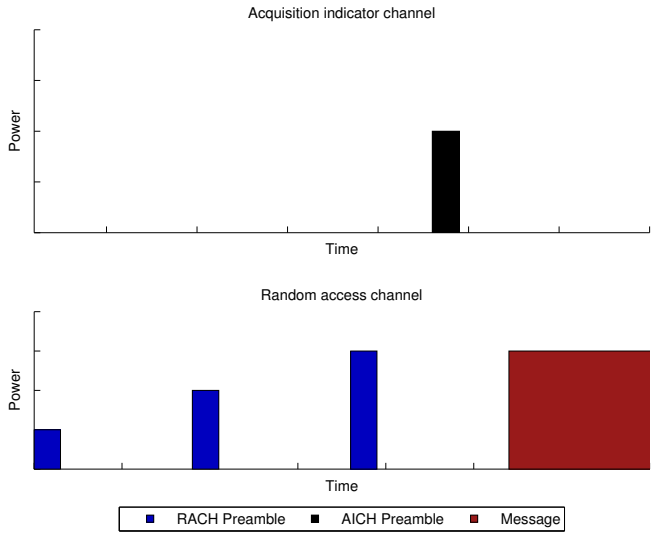


Figure 2.4: RACH procedure: power ramping of RACH preamble, AICH acknowledgement followed by the message.

the procedure. Thus, the UE picks a new signature and the next access slot and increases the transmission power by the ramping factor until the maximum number for retransmissions has been met, or until it receives an acknowledgment on the AICH. If the preamble is detected by the RBS, the UE transmits a 10 ms or 20 ms RACH message requesting a dedicated channel [12]. The procedure can be seen in Fig. 2.4.

2.3.2 RACH preamble code

The RACH preamble code $C_{pre,n,s}$ is a complex valued sequence created from a preamble scrambling code $S_{pre,n}$ and a preamble signature/channelization code $C_{sig,s}$ given by

$$C_{pre,n,s}(n) = S_{pre,n}C_{sig,s}(n)e^{j(\frac{\pi}{4}+\frac{\pi}{2}n)}, \text{ for } n = 0, 1, 2, 3, ..., 4095. \tag{2.4}$$

There exists a total of 8192 scrambling codes and the n:th preamble scrambling code is constructed from a 4096 long code scrambling sequence and is defined as

$$S_{pre,n}(i) = c_{long,n}(i), \text{ for } i = 0, 1, ..., 4095. \tag{2.5}$$

The long scrambling sequence $c_{long,n}$ is defined in the 3GPP standard [9].
The preamble channelization code consists of 256 repetitions of a signature $P_s(n)$ where $n = 0, 1, ..., 15$ and is defined as follows

$$C_{sig,s}(i) = P_s(i \bmod 16), \quad i = 0, 1, 2, ..., 4095 \tag{2.6}$$

where the signatures $P_s(n)$ can be derived from a set of 16 Hadamard codes of length 16 using equation 2.1. Thus, H_{16} can be constructed recursively as:

$$H_{16} = \begin{pmatrix} + & + & + & + & + & + & + & + & + & + & + & + & + & + & + & + \\ + & - & + & - & + & - & + & - & + & - & + & - & + & - & + & - \\ + & + & - & - & + & + & - & - & + & + & - & - & + & + & - & - \\ + & - & - & + & + & - & - & + & + & - & - & + & + & - & - & + \\ + & + & + & + & - & - & - & - & + & + & + & + & - & - & - & - \\ + & - & + & - & - & + & - & + & + & - & + & - & - & + & - & + \\ + & + & - & - & - & - & + & + & + & + & - & - & - & - & + & + \\ + & - & - & + & - & + & + & - & + & - & - & + & - & + & + & - \\ + & + & + & + & + & + & + & + & - & - & - & - & - & - & - & - \\ + & - & + & - & + & - & + & - & - & + & - & + & - & + & - & + \\ + & + & - & - & + & + & - & - & - & - & + & + & - & - & + & + \\ + & - & - & + & + & - & - & + & - & + & + & - & - & + & + & - \\ + & + & + & + & - & - & - & - & - & - & - & - & + & + & + & + \\ + & - & + & - & - & + & - & + & - & + & - & + & + & - & + & - \\ + & + & - & - & - & - & + & + & - & - & + & + & + & + & - & - \\ + & - & - & + & - & + & + & - & - & + & + & - & + & - & - & + \end{pmatrix}. \tag{2.7}$$

The symbols + and - denotes +1 and -1 respectively.

3

Interference suppression

Transmitting over a RACH with interference generated by a high rate data user on WCDMA enhanced uplink generates colored noise. By whitening a signal with respect to the noise, the information signal can be decorrelated from the interference distortions. In this chapter, a conceptual and mathematical description of the temporal whitening and spatial whitening algorithms detailed in chapter 5 is given.

3.1 Spatial whitening

Consider

$$\mathbf{Z} = \begin{pmatrix} \mathbf{z}_1[n] \\ \mathbf{z}_2[n] \\ \vdots \\ \mathbf{z}_a[n] \end{pmatrix} \tag{3.1}$$

where $\mathbf{Z}$ is an array of vectors $\mathbf{z}$ of size a times n, where a is the number of antennas and n is the number of samples of the vectors $\mathbf{z}$, where each vector is a zero mean Gaussian complex random signal. A spatial whitening can be done as

$$\mathbf{Y} = \mathbf{L}_l^{-1}\mathbf{Z} \tag{3.2}$$

where $\mathbf{Y}$ represents the decorrelated array of vectors and $\mathbf{L}_l$ is the lower triangular matrix from a Cholesky decomposition of the covariance matrix as

$$\mathbf{C}_z = \mathbf{L}_l\mathbf{L}^H. \tag{3.3}$$

The noise covariance matrix $\mathbf{C}_z$ can be obtained as

$$\mathbf{C}_z = E[\mathbf{Z}\mathbf{Z}^H]. \tag{3.4}$$

The operation of the spatial whitening filter can be shown by calculating the covariance of the whitened output array of vectors $\mathbf{Y}$

$$\begin{aligned} \mathrm{Cov}(\mathbf{Y}) &= E[\mathbf{Y}\mathbf{Y}^H] \\ &= E[\mathbf{L}_l^{-1}\mathbf{Z}\mathbf{Z}^H\mathbf{L}_l^{-H}] \\ &= \mathbf{L}_l^{-1}\mathbf{C}_z\mathbf{L}_l^{-H} \\ &= \mathbf{I}. \end{aligned} \tag{3.5}$$

Thus, the covariance of $\mathbf{Y}$ is the identity matrix and is therefore an array of white random vectors [13][14].

3.2 Temporal whitening

Let $\mathbf{x}$ be a Gaussian complex random vector $\mathbf{x} = [x[0]x[1]...x[N-1]]^T$ with covariance matrix $\mathbf{C}$ and mean $\mu_x = E[\mathbf{x}] = 0$. The purpose of performing a whitening transform is to make the components of the vector uncorrelated and of variance one, which is achieved by turning the covariance matrix of this vector into the identity matrix. The autocovariance matrix can be calculated as

$$\mathbf{C} = E[(\mathbf{x} - \mu_x)(\mathbf{x} - \mu_x)^H] = E[\mathbf{x}\mathbf{x}^H]. \tag{3.6}$$

Futhermore, since $\mathbf{C}$ is symmetric and positive semidefinite, there exist a square root $\mathbf{C}^{1/2}$ and a matrix such that $\mathbf{C}^{1/2}(\mathbf{C}^{1/2})^H = \mathbf{C}$. Also, as $\mathbf{C}^{1/2}$ is invertible the whitened output vector $\mathbf{y}$ can be given as

$$\mathbf{y} = \mathbf{C}^{-1/2}\mathbf{x} \tag{3.7}$$

where the covariance of the whitened output vector $\mathbf{y}$ can be calculated as

$$\begin{aligned}
\mathrm{Cov}(\mathbf{y}) &= E[\mathbf{y}\mathbf{y}^H] \\
&= \mathbf{C}^{-1/2}E[\mathbf{x}\mathbf{x}^H](\mathbf{C}^{-1/2})^H \\
&= \mathbf{C}^{-1/2}\mathbf{C}(\mathbf{C}^{-1/2})^H \\
&= \mathbf{C}^{-1/2}(\mathbf{C}^{1/2}(\mathbf{C}^{1/2})^H)(\mathbf{C}^{-1/2})^H \\
&= (\mathbf{C}^{-1/2}\mathbf{C}^{1/2})(\mathbf{C}^{-1/2}\mathbf{C}^{1/2})^H \\
&= \mathbf{I}.
\end{aligned} \tag{3.8}$$

Thus, the covariance of vector $\mathbf{y}$ is the identity matrix and consequently the vector has been transformed to an uncorrelated vector with variance one [14].

4
Problem description

This chapter explains the specifics of the random access receiver preamble detection. Including what assumptions are made in terms of transmission conditions and receiver-side information when designing the preamble detection algorithm in this thesis.

4.1 System model

The system model described in this section can be seen in Fig. 4.1. The transmitter (UE) generates a chip sequence preamble code according to the equations in Sec. 2.3.2. The code is subsequently pulse shaped with a Root raised cosine filter and then transmitted using one antenna to the RBS. Since the signal is transmitted in a wireless radio frequency communication environment it is affected by an unknown fading channel and additive noise, see Sec. 4.4 for more details. In this specific

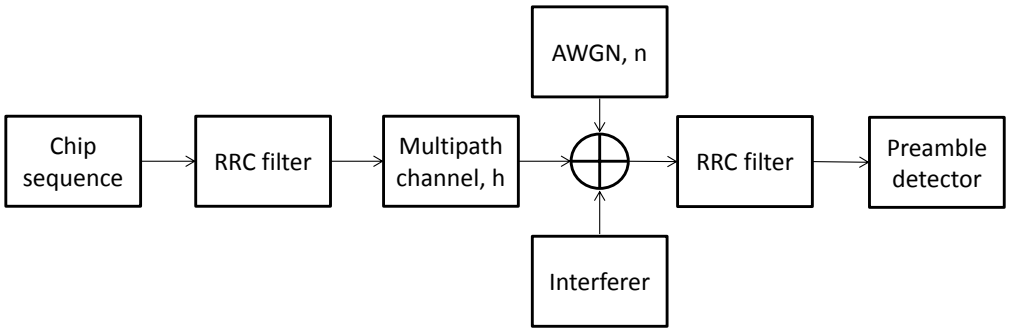


Figure 4.1: Block diagram of the system model

scenario, the receiver also experiences additive noise interference from a high data rate user. The effect of this interference can be mitigated by signal processing in the preamble detection algorithm, see Sec. 4.3. The preamble detector then receives the signal from the transmitter, where the signal gain is normalized and the signal is downsampled, see Sec. 4.2 for further details. This signal is modeled using a discrete time domain model defined as

$$r(n) = \sum_{l=0}^{L-1} h(l)s(n-l) + n_a \tag{4.1}$$

where $h(l)$ is the radio channel impulse response, $s(n)$ is the desired symbol and spreading sequence and n_a is additive noise and interference.

4.2 Preamble detection

The basis for the preamble detection procedure in RACH lies in detection theory. Detection theory is the theory of detecting and making decisions on a distorted signal. A common way of detecting a signal in the presence of distortions is to use the properties of the auto/cross-correlation function. The auto-correlation function is used to calculate the correlation of the signal itself, whereas the cross-correlation function is used to calculate the correlation between two different signals. The cross-correlation function is related to the convolution of two functions, which is defined as the integral of the product of two continuous functions $x(t)$ and $y(t)$ where one is reversed and shifted by a time lag t

$$(x * y)(t) = \int_{-\infty}^{\infty} x(\tau)y(t-\tau)\,d\tau \tag{4.2}$$

The cross-correlation function R_{xy}, in relation to convolution, is equivalent to the convolution of $x^*(-t)$ and $y(t)$ and is expressed as

$$R_{xy}(\tau) = \int_{-\infty}^{\infty} x^*(t)y(t+\tau)\,dt. \tag{4.3}$$

Matched filtering is a filtering method frequently used in signal detection. It builds upon the cross-correlation function, where it matches a certain pattern or pulse shape to a signal in order to detect if it is present in the signal. The matched filter can be defined as the linear filter $h(t)$ that maximizes the output signal-to-noise ratio

$$y(t) = \int_{-\infty}^{\infty} h^*(t-\tau)x(\tau)\,d\tau. \tag{4.4}$$

In order to detect the preamble, the RBS thus needs to find the matched filter coefficients which yield the maximum SNR. The matched filter correlation calculation is done over a search window for each access slot. The length of the search window is defined as the approximate maximum propagation delay in chips between a UE and the RBS. The search window acts as a buffer to ensure the entire preamble is received regardless of propagation delay, since there is no synchronization between the receiver and transmitter and thus the RBS does not know where the preamble begins within the search window.

Hence, each access slot the RBS "slides" the matched filter, with filter coefficients equal to the cell specific scrambling code and all the active signatures from the signature list, over the entire search window to find a correlation peak over a predefined threshold. Thus, after correlating the incoming signal with all signature-specific matched filter coefficients, a decision is made on whether the correlation peaks yield a valid access attempt for the specific signature.

4.3 Interference

An interferer is used in the scenario detailed in fig 4.1 to simulate MAI from a high data rate user on the WCDMA enhanced uplink, when transmitting on a RACH. In the preamble detector, this interference simulates colored Gaussian noise, which impairs the matched filter operation. However, the colored noise can be suppressed by performing receiver-side signal processing. One solution to this is to implement a temporal whitening filter and a spatial whitening filter to mitigate this interference.

4.4 Channel models

Channel models are used to define the physical processes which change a transmitted signal over a medium. Under ideal conditions there would be no noise or distortions, but that is not a realistic interpretation of how a transmission would act under real-life conditions. One of the most commonly used channel models is the so called additive white Gaussian noise (AWGN) channel. It acts under the conditions that the noise is independent, time invariant and uniformly distributed over all frequencies. The distribution of the noise models a zero-mean Gaussian distribution with variance σ^2 and its probability density function (PDF) is defined as

$$N(x) \sim \mathcal{N}(0, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x)^2}{2\sigma^2}}. \tag{4.5}$$

However, the assumption that the channel is only acted upon by additive noise is not realistic in wireless radio frequency communication.

4.4.1 Fading channels

In the case of radio frequency communication a more realistic model to use is a fading channel model. A fading channel is characterized by various effects, such as path loss, shadowing and multipath. Path loss can be described as the loss of power over distance between a transmitter and receiver [15]. Shadowing, or large scale fading, is the effect of the signal being obstructed by different objects, such as buildings and trees. However, the effect of shadowing is not simulated in the scenario for this thesis. Multipath propagation describes the phenomenon of signals reaching a receiving antenna from several paths, due to refraction and reflection. Multipath fading affects the signal in both the time domain and the frequency domain. The following definitions are somewhat simplified to give a conceptual understanding, which is deemed sufficient in the scope of this thesis.

A characteristic of the multipath channel is its time-varying nature. The parameter governing the time varying nature of the frequency dispersion of the channel in time is called coherence time T_c and is defined as the period of time where the fading process is approximately correlated [16]. The frequency dispersion occurs when either the transmitter or the receiver is moving and thus the signal reflects on different surfaces at different times. The fading process is regarded as fast if the coherence time is shorter than the symbol time T_s. Conversely, if $T_c > T_s$ the fading

process is considered slow [17]. The relation between coherence time and Doppler spread B_D can be approximated as

$$T_c \simeq \frac{1}{B_D} \tag{4.6}$$

where the Doppler spread is a measure of the spectral broadening caused by the time varying channel.

In the frequency domain a signal can experience frequency-selective and frequency-flat fading. This occurs when the signal is distorted by the delay spread of the multipath [17]. The parameter governing the time dispersive nature of the channel in frequency is called coherence bandwidth B_c. It is defined as the range of frequencies where the fading process is approximately correlated and thereby flat [16]. If the coherence bandwidth is less than the bandwidth of the transmitted signal B_s the channel is called frequency-selective. Moreover, the channel is called frequency flat for $B_c > B_s$. The coherence bandwidth is related to the root mean square delay spread σ_{RMS}, which is a measure of the difference in time of arrival between the earliest (and strongest) multipath component and the most recent subsequent multipath component. The relation can be approximated as

$$B_c \simeq \frac{1}{\sigma_{RMS}}. \tag{4.7}$$

4.4.2 Fading distributions

The statistical distribution of the multipath fading depends on the propagation scenario. The most prominent fading distribution used for fading channels is the Rayleigh fading channel, which is used for modeling an urban environment with no line of sight (LOS) between transmitter and receiver [18]. The PDF of the SNR per symbol γ of the Rayleigh fading channel is given by

$$p_\gamma(\gamma) = \frac{1}{\bar{\gamma}} e^{-\frac{\gamma}{\bar{\gamma}}}, \text{ for } \gamma \geq 0 \tag{4.8}$$

where $\bar{\gamma}$ is defined as the average SNR per symbol. The Rician fading channel is on the other hand a better model for a scenario with a strong line of sight signal between the transmitter and receiver. The PDF of the SNR per symbol γ of the Rician fading channel in this case is defined as

$$p_\gamma(\gamma) = \frac{(1+n^2)e^{-n^2}}{\bar{\gamma}} e^{-\frac{(1+n^2)\gamma}{\bar{\gamma}}} I_0\left(2n\sqrt{\frac{(1+n^2)\gamma}{\bar{\gamma}}}\right), \text{ for } \gamma \geq 0 \tag{4.9}$$

where n is the Nakagami fading parameter and I_0 is the modified Bessel function of zeroth order. There are additional statistical distributions that can be used for other types of scenarios. But the Rayleigh and Rician fading distributions are the most relevant in the scope of this thesis.

4.4.3 Doppler shift

If a transmitter or a receiver is moving during transmission, the received signal will be affected by a Doppler shift. The cause of the Doppler shift is due to the additional distance the signal has to travel to reach is destination, which also causes a change in phase. The Doppler shift can therefore be defined from the change in distance

$$\Delta\, d = v\Delta t\; cos\; \theta \tag{4.10}$$

where v is the velocity of the receiver toward the transmitter in the direction of motion and θ is the arrival angle of the received signal relative to the direction of motion. This causes as change in phase defined as

$$\Delta\phi = 2\pi v\Delta t\; cos\; \frac{\theta}{\lambda} \tag{4.11}$$

where λ is the signal wavelength. The Doppler shift in frequency is then defined from the relationship between the signal frequency and phase [19]

$$f_D = \frac{1}{2\pi}\frac{\Delta\phi}{\Delta t} = v\; cos\; \frac{\theta}{\lambda}. \tag{4.12}$$

5
Method

In this chapter we explain the methodology used to improve the performance of the preamble detection procedure in a WCDMA system. Firstly, a brief overview is given explaining the operation of a random access receiver. Secondly, a functional overview of the preamble detector is presented. Thirdly, the implementation of the different parts (the blocks in Fig. 5.1) of the preamble detector are explained in detail. Lastly, the setup function and the data acquiring function are explained. The random access receiver consists of three separate parts: a preamble detector, a message searcher and a rake receiver. The first part of the receiver is called a preamble detector and it listens on the PRACH to see if an access attempt has been made - if the access attempt fulfills the requirements [9], it sends an acknowledgement on the AICH channel. The second part is called a message searcher, which listens to the control channel to determine at what times the signal is received, and then applies the despreader accordingly. The last part of the receiver is called a rake receiver and is used to counter-act the multipath fading on the channel. It does so by assigning correlators to different multipath components and then combining them to make use of the varying characteristics of the channel.

5.1 Functional overview of the preamble detector

The operation of the preamble detector is split into several functions, as shown in Fig. 5.1. The combined outcome of these functions lead to a decision of whether or not a preamble has been transmitted. Each function mentioned in this overview are explained in further detail in the upcoming sections.

The chain of operations starts with a spatial whitening, which whitens the spatial interference between the antennas. The input to the spatial whitening function is complex data represented in the time domain. The output spatially whitened data signal is then temporally whitened in multiple parts using a temporal whitening filter. The number of parts is configurable, however, typically four parts are used. The output from the temporal whitening filter is sent to both the Signature Code Matched Filter (SCMF) and the interference estimator. In the SCMF, the spatially and temporally whitened data is transformed to the frequency-domain and correlated with the scrambling code specific to that cell and the signature list.

Thereafter, the data is non-coherently accumulated in the time-domain and then antenna combined using non-coherent accumulation. The amount of parts of the preamble accumulated is configurable and depends on the state of the channel over which the transmission is being made. If the channel is frequency-selective the

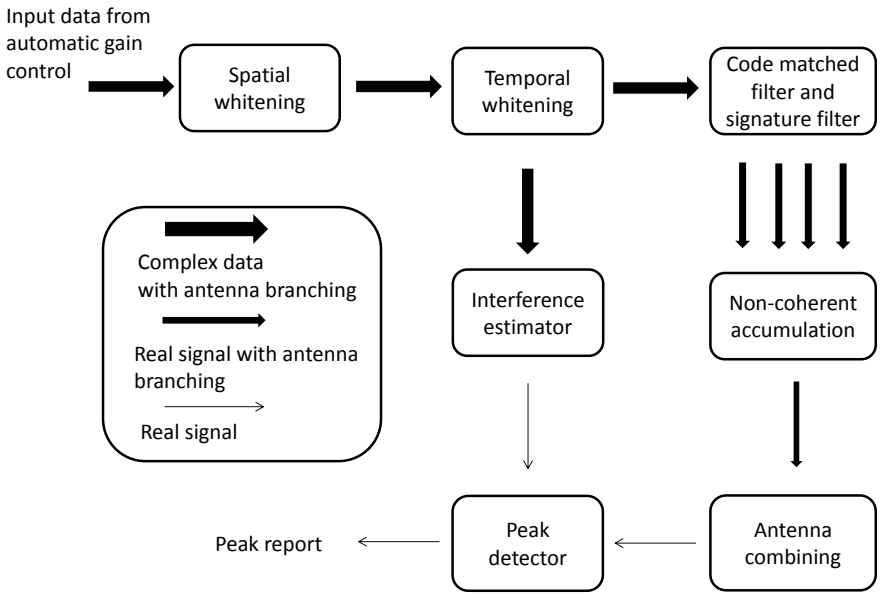


Figure 5.1: Block diagram of the algorithms for the preamble detector

preamble experiences uncorrelated fading at different frequency components, which can be circumvented by coherently accumulating the preamble in several parts. The preamble is typically accumulated in four parts. The final part of the preamble detector is called the peak detector, which determines if the received data has sufficient power and integrity to classify as a detection. This is achieved by firstly receiving a noise estimate from the interference estimator and then setting a threshold according to the noise power. Subsequently, the peak detector compares the peak acquired from the antenna combiner to the threshold and makes a decision based on the correlation power of the data.

5.2 Spatial whitening

The RACH procedure can be subject to interference by high data rate users in the uplink. By spatially whitening the data between the antennas the interference can be suppressed, which could enable successful reception even in the presence of strong interference.

Since the transmitted signal is scrambled with a PN sequence, explained in Sec. 2.1.2, we can treat the received signal as noise. The received signal per antenna is defined as $x_a[n]$ of length $n = 1, 2, ..., 8704$, where a is the number of antennas used. In our case the number of antennas is equal to two, hence a matrix $\mathbf{X}$ can be defined for each received signal per antenna as follows

$$\mathbf{X} = \begin{bmatrix} x[1,1] & x[1,2] & \cdots & x[1,n-1] & x[1,n] \\ x[2,1] & x[2,2] & \cdots & x[2,n-1] & x[2,n] \end{bmatrix}. \tag{5.1}$$

The corresponding estimate of the autocovariance matrix of size 2×2 can thus be formed as

$$\hat{\mathbf{C}}_x = \mathbf{X}\mathbf{X}^H. \tag{5.2}$$

Furthermore, there exists a Cholesky decomposition L of a Hermitian positive-definite matrix $\mathbf{C}_x$, such that

$$\hat{\mathbf{C}}_x = \mathbf{L}^H\mathbf{L}. \tag{5.3}$$

Thus, $\mathbf{C}_x$ can be decomposed, yielding the following lower Cholesky matrix

$$\mathbf{L} = \begin{bmatrix} L[1,1] & 0 \\ L[2,1] & L[2,2] \end{bmatrix} \tag{5.4}$$

where the diagonal entries of L are real and positive. Applying the inverse to the Cholesky matrix and multiplying with the received signal yields

$$\mathbf{X}_{spat} = \mathbf{L}^{-1}\mathbf{X} \tag{5.5}$$

where $\mathbf{X}_{spat}$ is the array of the spatially whitened received signals. However, this is a suboptimal solution since we only use one time-lag of the ACF per antenna for the estimated covariance matrix to make the inversion of the matrix less computationally complex.

5.3 Temporal whitening

The purpose of temporally whitening a signal is to make the signal uncorrelated with variance one, hence making the signal temporally white. More specifically, in terms of problem description the goal is to reduce the temporal interference generated from the simulated high date rate user transmitting simultaneously on the channel WCDMA enhanced uplink.

Since the transmitted signal has low power due to it being spread with a PN-sequence, we assume that the received signal has the characteristics of noise. Furthermore, we assume the received signal is zero mean wide sense stationary Gaussian random process and is output from the spatial whitening function. The temporal whitening is done for every antenna branch separately, thus the autocovariance matrix is given as

$$\mathbf{C} = E[(\mathbf{x}_{spat} - \mu_{x,spat})(\mathbf{x}_{spat} - \mu_{x,spat})^H] = E[\mathbf{x}_{spat}\mathbf{x}_{spat}^H] \tag{5.6}$$

for each antenna and thus $\mathbf{x}_{spat}$ is one vector in the array of vectors $\mathbf{X}_{spat}$, given in equation (5.1). Moreover, using the property of the autocorrelation function (ACF) $r_{xx}[k] = r_{xx}^*[-k]$, the autocovariance matrix can be simplified to [20]

$$\mathbf{C} = \begin{bmatrix} r_{xx}[0] & r_{xx}^*[1] & r_{xx}^*[2] & \cdots & r_{xx}^*[N-1] \\ r_{xx}[1] & r_{xx}[0] & r_{xx}^*[1] & \cdots & r_{xx}^*[N-2] \\ r_{xx}[2] & r_{xx}[1] & r_{xx}[0] & \cdots & r_{xx}^*[N-3] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{xx}[N-1] & r_{xx}[N-2] & r_{xx}[N-3] & \cdots & r_{xx}[0] \end{bmatrix} = \mathbf{R} \tag{5.7}$$

where matrix $\mathbf{C}$ is a positive definite Hermitian matrix. Hence, the autocorrelation matrix $\mathbf{R}$ is also a Hermitian matrix and positive semidefinite [21]. The autocorrelation matrix $\mathbf{R}$ is also seen to be a Hermitian Toeplitz matrix, thus if we let number of data samples tend to infinity $N \to \infty$, the eigenvalues λ_i and eigenvectors $\mathbf{v}_i$ can be found [20][22]. Hence as $N \to \infty$, we can make the following approximations

$$\begin{aligned} \lambda_i &= P_{xx}(f_i) \\ \mathbf{v}_i &= \frac{1}{\sqrt{N}}[1,\ \exp(j2\pi,\ f_i),\ \exp(j4\pi f_i),\ ...,\ \exp(j2\pi(N-1)f_i)]^T \end{aligned} \tag{5.8}$$

for $i = 0, 1, ..., N-1$ and $f_i = \frac{i}{N}$. The eigenvalues are equally spaced samples of the PSD over the frequency interval [0,1] and the eigenvectors are the discrete Fourier transform (DFT) orthonormal vectors. Moreover, since $\mathbf{v}_{N-i} = \mathbf{v}_i^*$ and $P_{xx}(f_{N-i}) = P_{xx}(f_i)$, the complex eigendecomposition can be written as

$$\mathbf{R} = \mathbf{V}^H \Lambda \mathbf{V}. \tag{5.9}$$

With the approximation of equation (5.8) we can now write the square inverse as

$$\mathbf{R}^{-1/2} = \mathbf{V}^H \Lambda^{-1/2} \mathbf{V} = \mathbf{V}^H P_{xx}^{-1/2} \mathbf{V}. \tag{5.10}$$

Using the property derived in equation (3.7) in Sec. 3.2 and equation (5.10)

$$\mathbf{y} = \mathbf{V}^H P_{xx}^{-1/2} \mathbf{V}\mathbf{x} \tag{5.11}$$

$$\mathbf{V}\mathbf{y} = \mathbf{V}\mathbf{V}^H P_{xx}^{-1/2} \mathbf{V}\mathbf{x} \tag{5.12}$$

$$\mathcal{F}\{\mathbf{y}\} = P_{xx}^{-1/2} \mathcal{F}\{\mathbf{x}\}. \tag{5.13}$$

Thus, if the eigenvalues of the autocorrelation matrix is known, then the inverse square root of the power spectral density can be used to whiten the received signal. The full mathematical derivation and heuristic justification of this property is given in [20].

The remainder of this section describes the implementation of the algorithm, which was written in MATLAB and C++ and was designed according to Fig. 5.2. The MATLAB-code can be seen in appendix A.4. In the algorithm implementation, the received signal is firstly zero padded to the next power of two for the length of the preamble, which itself is divided in typically four parts, and the search window in double chip rate as

$$x_{spat,z}[n] = \begin{cases} x_{spat}[n] & \text{for } n = 0, 1, ..., 2175 \\ 0 & \text{for } n = 2176, 2177, ..., l_{fft_t} - 1 \end{cases} \tag{5.14}$$

where l_{fft_t} is defined in 5.9.1. Subsequently, the power spectral density is calculated in discrete time as follows

$$x[k] = \sum_{n=0}^{l_{fft_t}-1} x_{spat,z}[n] \exp\left(-2\pi j k n / l_{fft_t}\right) \tag{5.15}$$

$$S_{xx}[k] = \frac{1}{l_{fft_t}} |x[k]|^2 \tag{5.16}$$

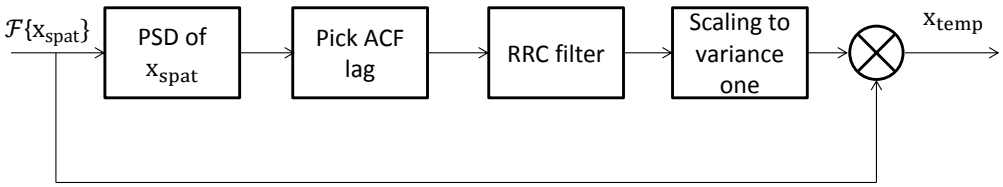


Figure 5.2: Block diagram of the temporal whitening algorithm.

where $x_{spat,z}[n]$ is the zero padded received signal and again l_{fft_t} is the number of samples of the zero padded received signal and is defined in 5.9.1. The zero-padding is necessary in order to use the radix-2 FFT-algorithm and also to circumvent the effect of circular convolution. Although, since twice the length of the received signal is not 2^{14}, circular convolution is not fully prevented. However, the effect of circular convolution had a minimal influence on overall performance, which will be shown in Sec. 7.

The power spectral density of the received signal is calculated for every access slot and antenna. The received signal is, however, typically split into four parts to temporally whiten the parts separately. The separation of the preamble into parts is done since we assume that the channel can be frequency selective. The power spectral density per access slot, antenna and part of the received signal is shown in Fig. 5.3. The PSD seen in the Fig. 5.3 contains a considerable amount of noise,

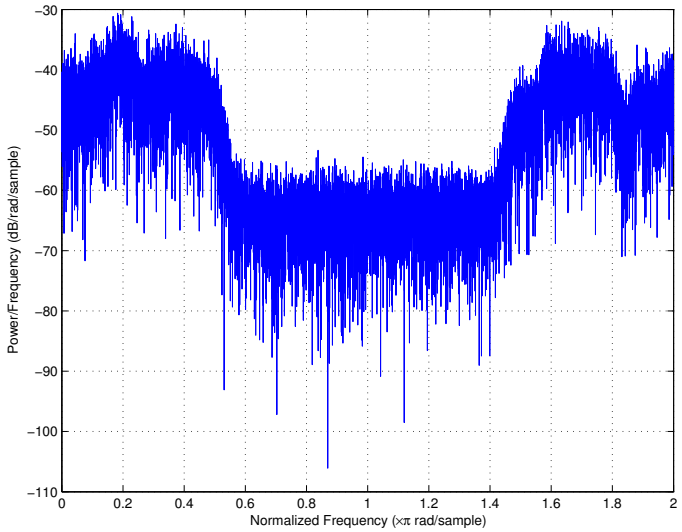


Figure 5.3: Power spectral density $\mathbf{S}_{xx}$ of the received signal in a frequency selective channel.

which is undesirable when designing a filter. Minimizing the noise can be done by reducing the number of samples used in the autocorrelation function. Thus, the signal is transformed back to the time domain by using the Inverse Discrete Fourier Transform (IDFT), yielding the autocorrelation function of the received signal.

$$R_{xx}[n] = \sum_{k=0}^{l_{fft_t}-1} S_{xx}[k] \exp\left(2\pi jkn/l_{fft_t}\right). \tag{5.17}$$

Since the signal has been pulse-shaped with a root-raised cosine, the autocorrelation function exhibits the characteristics of a root-raised cosine. Thus, to reduce the noise, only the samples representing the pulse of the root-raised cosine are selected by using a rectangular windowing function defined as

$$w[n] = \begin{cases} 1 & \text{for } n = l_{fft_t}/2 - 16, l_{fft_t}/2 - 15, ..., l_{fft_t}/2 + 15 \\ 0 & \text{otherwise .} \end{cases} \tag{5.18}$$

The rectangular window is then multiplied with the ACF as follows

$$R_{xx_w}[n] = R_{xx}[n]w[n] \quad \text{for } n = 1, 2, ..., l_{fft_t}. \tag{5.19}$$

Consequently, the decay and ripple in the other time lags are discarded, see fig 5.4.

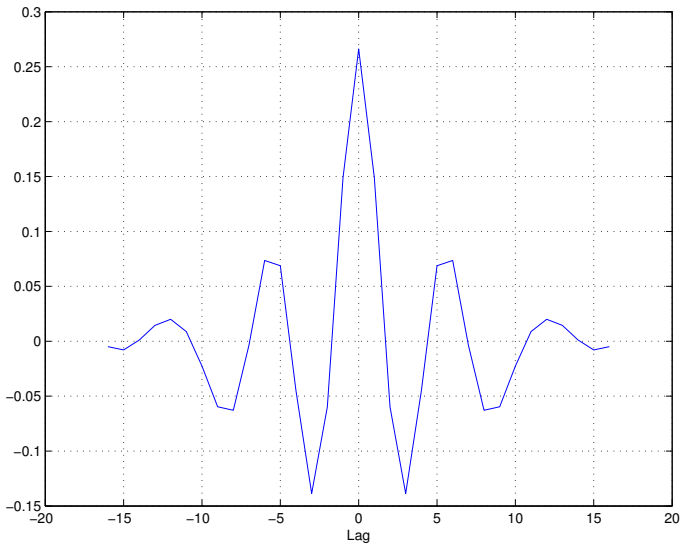


Figure 5.4: 32 lags of the ACF R_{xx_w}

The signal is subsequently transformed back to the frequency domain using the FFT to once again yield the power spectral density

$$S_{hx}[k] = \sum_{n=0}^{l_{fft_t}-1} R_{xx_w}[n] \exp\left(-2\pi jkn/l_{fft_t}\right). \tag{5.20}$$

The difference in noise before and after selecting specific samples can be seen by comparing Fig. 5.5 and Fig. 5.3. Thus, selecting fewer samples of the autocorrelation function generates a smoother spectrum. To further improve the operation of

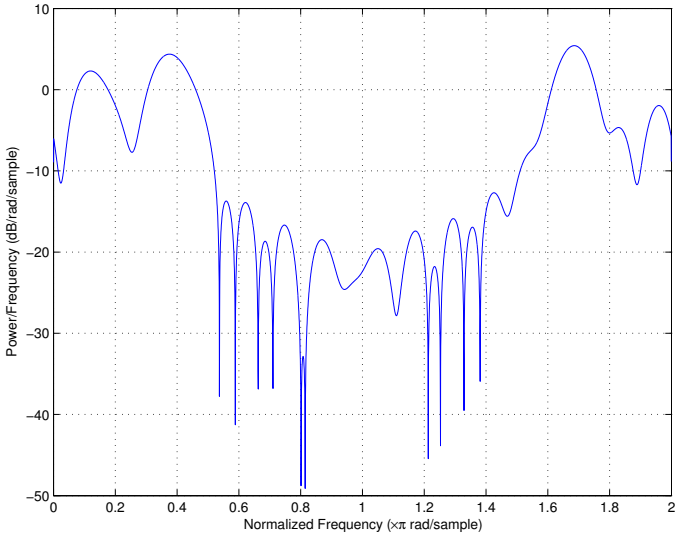


Figure 5.5: Power spectral density S_{hx} derived from performing a FFT of a 32 lag autocorrelation function

the intended temporal whitening filter H_{temp}, the frequencies in the PSD $\mathbf{S}_{hx}$ that is out of interest, due to the fact of oversampling, can be suppressed further. This is especially important since the PSD needs to be inverted to gain the whitening operation and thus the frequencies components with low power approaching zero, as those of no interest, can be amplified to considerably large values. Thus, the frequencies with low power is filtered with a root-raised cosine spectrum shaping filter as shown in Fig. 5.6a.

Firstly, the root-raised filter $\mathbf{H}_{RRC}$ is defined according to Sec. 5.9.1. Secondly, PSD S_{hx} is normalized such that the pass and stop band of the filter cover the intended frequencies of the PSD as

$$S_{hx}[k] = \frac{S_{hx}[k]}{max[|\mathbf{H}_{RRC}|]E[S_{hx}]} \quad \text{for } k = 0, 1, ..., l_{fft_t} - 1. \tag{5.21}$$

Finally, the square root of the PSD is inverted and multiplied element wise with the spectrum shaping filter

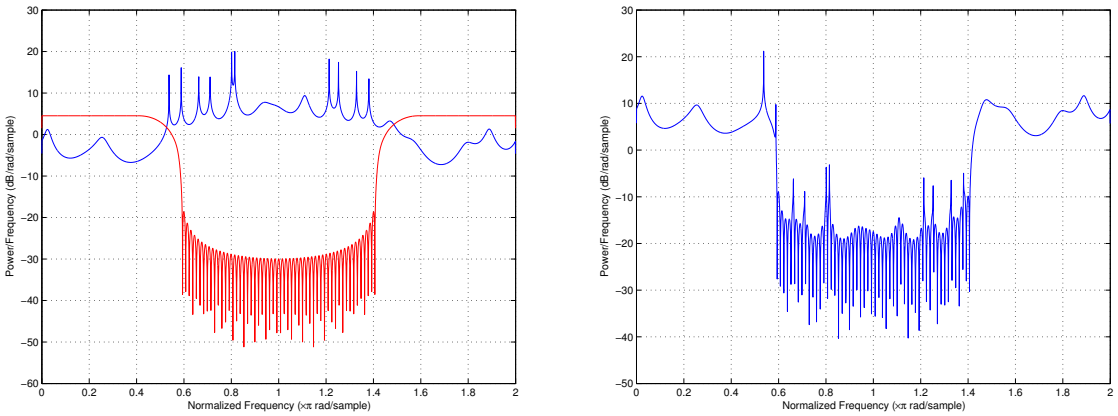
$$H_{temp}[k] = S_{hx}^{-1/2}[k]H_{RRC}[k] \quad \text{for } k = 0, 1, ..., l_{fft_t} - 1. \tag{5.22}$$

The characteristics of both the root raised cosine filter and the inverted and normalized PSD S_{hx} and the filter operation, i.e. the amplification and suppression of the temporal whitening filter, can be seen in Fig. 5.6a. The resulting temporal filter PSD $\mathbf{H}_{temp}$ can be seen in Fig. 5.6b. The final step of creating the whitening filter $\mathbf{H}_{temp}$ is to scale the filter coefficients to produce an output signal with variance one. The scaling is done by calculating the variance of the output signal as

$$\sigma^2 = \frac{1}{l_{fft_t}} \sum_{k=0}^{l_{fft_t}-1} |H_{temp}[k]|^2 S_{xx}[k] \tag{5.23}$$

where l_{fft_t} is calculated in Sec. 5.9.1. The filter coefficients are then scaled as follows

$$\mathbf{H}_{temp} = \frac{\mathbf{H}_{temp}}{\sqrt{\sigma^2}}. \tag{5.24}$$



(a) $\mathbf{H}_{RRC}$ in red and $\mathbf{S}_{hx}^{-1/2}$ in blue

(b) $\mathbf{H}_{temp}$

Figure 5.6: Root raised cosine and temporal whitening filter response.

Equation 5.24 ensures that the output of the whitening filter has variance of one.

Finally, the spatially whitened signal is filtered by applying the temporal whitening filter as

$$x_{temp}[k] = |H_{temp}[k]|x[k] \text{ , for } k = 0, 1, ..., l_{fft_t} - 1 \tag{5.25}$$

where $\mathbf{x}_{temp}$ is the temporally whitened output with variance one. The whitened PSD of $\mathbf{x}_{temp}$ can be seen in fig 5.7. The temporally whitened signal $\mathbf{x}_{temp}$ is then transformed to the time domain using a IDFT to make the parts of the preamble detector function independent of each other.

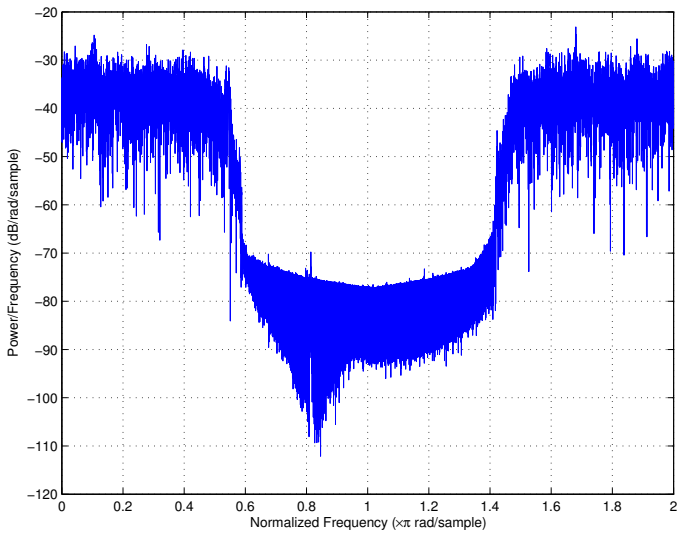


Figure 5.7: PSD of the output signal $\mathbf{x}_{temp}$ from the whitening filter.

5.4 Interference estimator

The preamble peak detector uses a threshold to determine if a valid preamble access attempt was transmitted by the UE. The state of the channel can vary from access slots and antennas, hence an estimate of the interference is calculated to aid in setting the threshold used to determine if a preamble was transmitted. The threshold also needs to be robust to keep a constant false preamble detection rate. The interference estimate is calculated by summing every power component in the frequency spectrum of the PSD. The interference estimate per antenna can be calculated by firstly zero padding the temporally whitened signal

$$x_{temp,z}[n] = \begin{cases} x_{temp}[n] & \text{for } n = 1, 2, ..., 8704 \\ 0 & \text{for } n = 8705, 8706, ..., N \end{cases} \tag{5.26}$$

where N is the length of the next power of two of the length of x_{temp}. The PSD and subsequently the interference power estimate is then calculated as

$$x[k] = \sum_{n=1}^{N} x_{temp,z}[n] \exp\left(-2\pi jkn/N\right) \tag{5.27}$$

$$S_{xx}[k] = \frac{1}{N}|x[k]|^2 \tag{5.28}$$

$$\sigma^2 = \sum_{k=1}^{N} S_{xx}[k] \tag{5.29}$$

which yields the total power across all frequencies, including the power from both the preamble and the noise. However, since a high spreading factor has been applied to the preamble symbols, the preamble has very low power and thus a small effect on the interference power estimate. The estimation is then accumulated for every antenna and an adjustment factor calculated in Sec. 5.9.3 is used as

$$\sigma^2 = adj_{factor} \sum_{a=1}^{n_a} \sigma^2[a] \tag{5.30}$$

where n_a is the number of antennas. The adjustment factor is required since the length of the received signal is equal to the length of a preamble and a search window. Hence, the decision if a valid preamble attempt has been made or not, is based on the power of one preamble. However, interference estimation is not necessary when the temporal whitening function is used, since the output is already normalized.

5.5 Signature and Code-matched filter

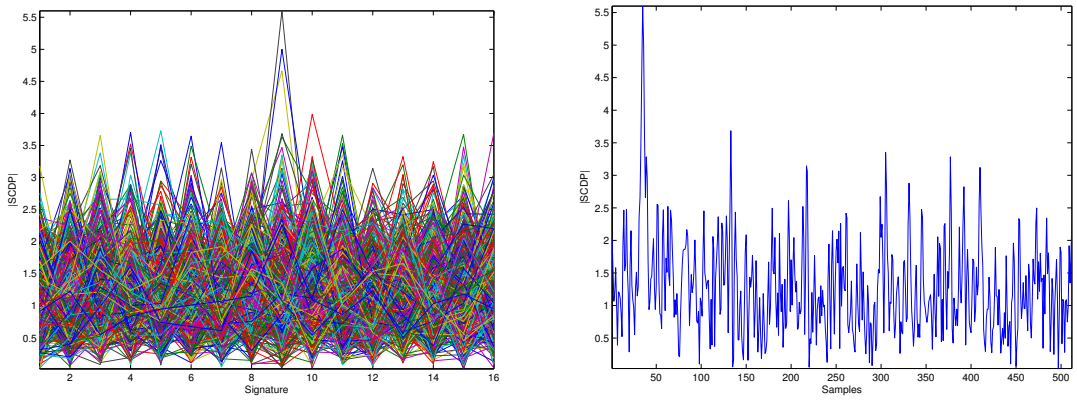
After spatially and temporally whitening the received signal, a signature- and code-matched filter is applied to calculate correlation of different time lags of the received signal and the scrambling code combined with detectable signatures. This is basically two matched filters combined into one, with both the scrambling code and the channelization codes (signatures) as filter coefficients, see Sec. 5.9.4. As mentioned

in Sec. 2.3.2, the preamble is constructed of a cell-specific scrambling code of length 4096 chips and a Hadamard sequence signature. The length of the signatures depends on the applied spreading factor, however, typically a 16-bit Hadamard is used and repeated 256 times. Thus, the SCMF-filter coefficients are constructed from a 16:th order Hadamard matrix multiplied by the cell-specific scrambling code. The correlation is calculated in the frequency domain for each access slot and antenna, by performing an element-wise multiplication between the received signal and the matched filter coefficients. This yields the signature complex delay profiles which, depending on the number of coherent accumulations, is accumulated for one delay profile per antenna.

The output of the signature and code-matched filter is called the signature complex delay profile and calculated as

$$SCDP[s,k] = H_{scmf}[s,k]x_{temp}[k] \text{ for k} = 0,1,...,l_{fft}-1 \tag{5.31}$$

where $SCDP[s,k]$ is calculated per access slot and antenna, s is the signature, l_{fft} is the length of one part to be coherently accumulated and H_{scmf} is the signature and code-matched filter defined in Sec. 5.9.4. The signature complex delay profile



(a) Magnitude of the signature complex delay profile versus the signature number.

(b) Magnitude of the signature complex delay profile for signature nine versus the search window samples

Figure 5.8: Magnitude of the signature complex delay profile in relation to the signature number and in relation to the correlation amplitude in the search window for signature nine.

is then transformed to the time domain using the IDFT. The result can be seen in Fig. 5.8.

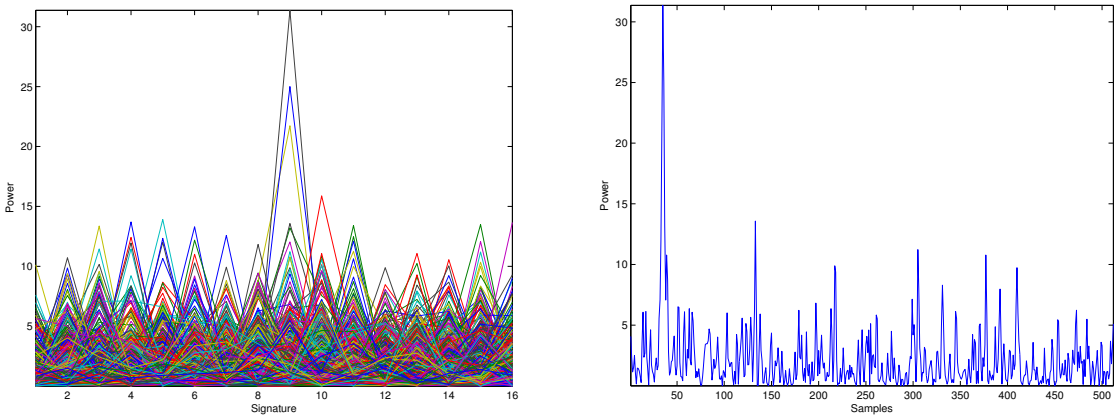
5.5.1 Coherent accumulation

Before the SCMF-filter, the number of coherent accumulations of the received signal is set dependent on the parameter COH, defined in the Sec. 5.9.2. Both the number of coherent and non-coherent accumulations are dependent on the value of COH and the received signal is thereby divided into equal lengths denoted n_{part},

see (5.37). The length of one segment to be coherently accumulated is calculated in equation (5.38). A combination of the accumulated parts is performed since the rate of change of the multipath characteristics in the channel can vary depending on the coherence time of the channel, for more information see Sec. 4.4. If the coherence time of the channel is smaller than the length of the preamble, the signal can experience uncorrelated amplitude fading. However, since a coherent accumulation of the preamble is performed, it can then be combined non-coherently at a later stage.

5.6 Non-coherent accumulation

At this stage, all coherently accumulated preamble parts are combined using non-coherent accumulation, with the intent of decreasing the variance of the signature complex delay profile. In order to use non-coherent accumulation combining the signature complex delay profile is weighted to the SNR, that is by squaring the signal as



(a) Preamble power delay profile versus the signature number.

(b) Preamble power delay profile for signature nine versus the search window samples.

Figure 5.9: Magnitude of the preamble power delay profile in relation to the signature number and in relation to the correlation amplitude in the search window for signature nine.

$$SPDP[s,n] = |SCDP[s,n]|^2 \text{ for n} = 1,2,...,l_{fft} \tag{5.32}$$

where s is the signature. The resulting matrix is called signature power delay profile and is calculated for every access slot and every antenna.The preamble power delay profile can then be calculated from the signature power delay profile as

$$PPDP[s,n] = \sum_{part=0}^{n_{part}} SPDP[part,s,n] \tag{5.33}$$

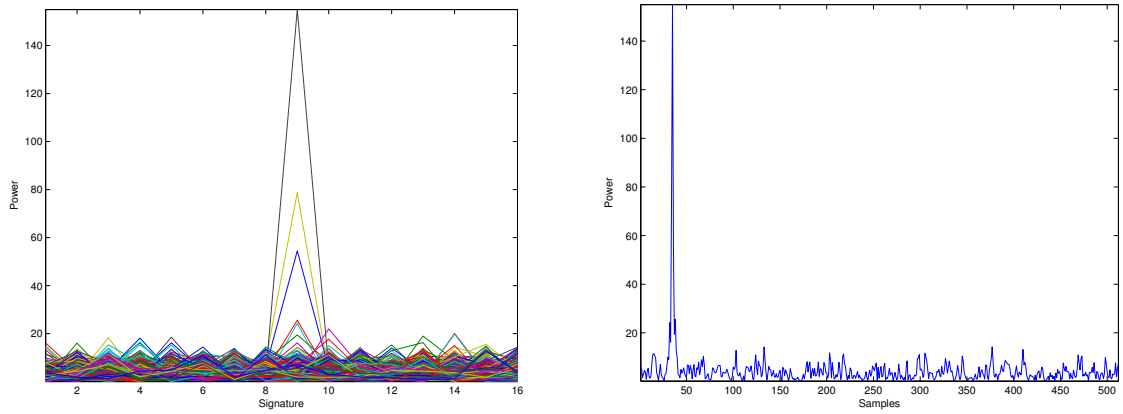
where n_{part} is calculated from equation (5.37). The results can be seen in Fig. 5.9.

5.7 Antenna combining

The purpose of antenna combining is to increase detection sensitivity by using space diversity. To accomplish space diversity gain all diversity antenna branches are diversity combined. Diversity combining is possible since the signature complex delay profile is absolute squared creating the preamble power delay profile, thus the difference in amplitude and phase can be neglected. The preamble accumulated power delay profile is calculated as

$$PAPDP_{a_s}[s] = \sum_{a=1}^{n_a} PPDP_{a_s}[a, s] \tag{5.34}$$

where n_a is the number of antenna branches used, s is the signature and a_s is the current access slot. The results using two antennas are shown in Fig. 5.10.



(a) Preamble accumulated power delay profile versus the signature number.

(b) Preamble accumulated power delay profile for signature nine versus the search window samples.

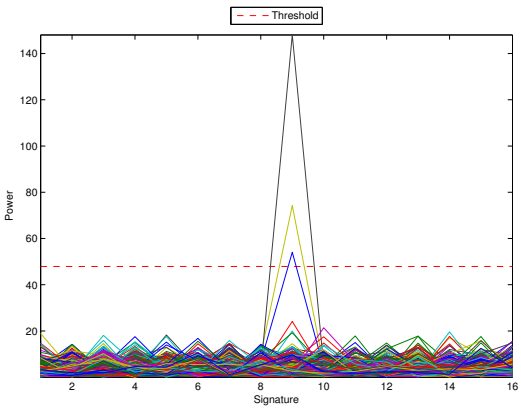
Figure 5.10: Magnitude of the preamble accumulated power delay profile in relation to the signature number and in relation to the correlation amplitude in the search window for signature nine.

5.8 Peak detector

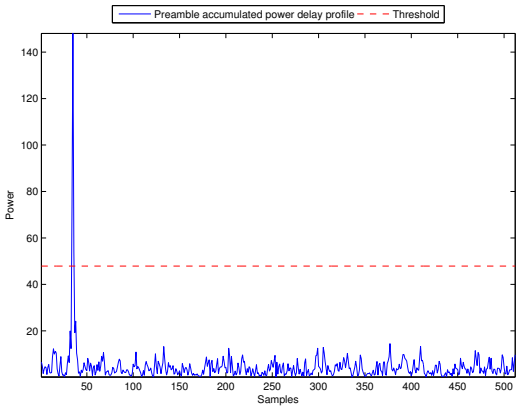
The purpose of the peak detector function is to determine if a valid access attempt was carried out. The interference estimation calculated for every access slot by equation (5.29) is combined with predefined threshold parameter calculated in equation (5.39) as

$$th_{a_s} = \sigma_{a_s}^2 th_l \tag{5.35}$$

where a_s represents the current access slot. If any value in the preamble accumulated power delay profile for a given signature, calculated in equation 5.34, is greater than or equal to th_{a_s} then a detection is registered for that specific signature. The decision is illustrated in Fig. 5.11.



(a) Preamble accumulated power delay profile versus the signature number, with the threshold visualized.



(b) Preamble accumulated power delay profile for signature nine versus the search window samples, with the threshold visualized.

Figure 5.11: A visualization of a successful preamble detection, for a given threshold

5.9 Setup

The preamble detector have initial default parameters, such as the number of parts to temporally whiten the received signal, the number of autocorrelation lags to select for the temporal whitening filter, threshold value for detections and more. The MATLAB implementation of the preamble peak detector can be seen in Sec. A.1. The initial default parameters for the C++ implemented code can be changed in the .par file and can be seen in Sec. A.9. The purpose of the setup function is to minimize computations for the preamble peak detector by initially calculating static parameters used in the algorithm. The MATLAB implementation of the setup function can be seen in appendix A.2.

5.9.1 Root raised cosine

The root raised cosine used in the temporal whitening function is created with the use of a C++ library called IT++, where the default number of created samples is 129 and the roll-off factor/beta value is set to 0.18 [23]. A roll-off factor of 0.18 is used to ensure that pulse shape of the root raised becomes wider than the root raised pulse used on the transmission side, to not exclude any information. In the temporal whitening function the root raised cosine is used to band limit the temporal whitening filter, hence the root raised cosine is zero padded to the length of the received signal in the frequency domain and is calculated as

$$l_{fft\_pp} = 2^{\text{round}[\log_2(l_{rs}/n_{p\_parts})]} \tag{5.36}$$

where l_{rs} is the length of the received signal and n_{p_parts} is the number of parts to temporally whiten the received signal. The purpose of using the nearest power

of two is to some extent limit the effect of circular convolution by padding zeros and thus be able to use the radix-2 FFT algorithm. The root raised cosine is then transformed to the frequency domain using the FFT.

5.9.2 Non-coherent and coherent accumulation parameters

The signature code matched filter function performs a coherent accumulation of the received signal, where the number of non-coherent accumulations, after defining COH, is calculated as

$$n_{parts} = \begin{cases} 1 & \text{for } COH = 4 \\ 4 & \text{for } COH = 2 \end{cases} \tag{5.37}$$

where the variable COH defines the number of parts to coherently and non-coherently accumulate. The length of one part to be coherently accumulated is calculated as

$$l_{fft} = 2^{10+COH}. \tag{5.38}$$

Both parameters are used both in the signature and the code matched filter function.

5.9.3 Interference estimator parameters

The predefined default threshold is represented in decibel as th_{db} and can be transformed to linear scale as

$$th_l = 10^{\frac{th_{db}}{20}}. \tag{5.39}$$

An adjustment factor is also used in the interference estimator and can be calculated as

$$adj_{factor} = \frac{1}{2l_{fft}} \frac{l_{rs} - l_{psw}}{l_{rs}} \frac{1}{n_{parts}} \tag{5.40}$$

where the first multiplier is needed when estimating the power spectral density and represents the number of samples for the FFT. The second multiplier is used since the length of the received signal is equal to the length of a preamble and a search window. Thus the length of the preamble is divided by the length of the received signal. The last multiplier is used to adjust the fact that coherent accumulation is used and that the threshold value should not change dependent on the number of accumulated parts in the SCMF.

5.9.4 Filter coefficients for signature and code matched filter

The initial information needed for the signature and code matched filter function to be able to calculate the matched filter coefficients is the number of coherent accumulations, the cell-specific scrambling code and the maximum number of signatures denoted n_{sign}. Firstly, the RBS determines the scrambling code used for the cell in

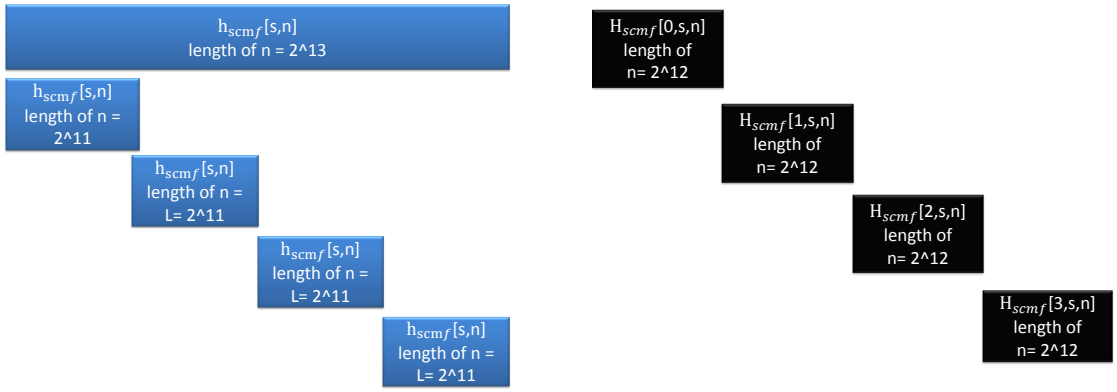
question. Then the scrambling code is rotated by $\frac{-\pi}{4}$ and complex conjugated as follows

$$scr_{code}[n] = \exp\left(-\sqrt{-1}\frac{\pi}{4}\right)\left(s_{pre}[n]\exp\left(\sqrt{-1}\left(\frac{\pi}{2}+\frac{\pi k}{4}\right)\right)\right)^* = s_{pre}[n]\exp\left(-\sqrt{-1}\left(\frac{\pi}{2}+\frac{k\pi}{2}\right)\right) \tag{5.41}$$

where the scrambling code $S_{pre,n}$ is defined in equation (2.5). Secondly, a Hadamard matrix of order equal to the maximum number of signatures, in this case 16, is generated according to equation (2.1). This matrix of order 16 serves as the filter coefficients for the signature matched filter. Subsequently, the Hadamard matrix is spread/repeated to fit the scrambling code, where typically a 256 repetition of the 16-bit Hadamard sequence is constructed as defined in equation (2.6). The resulting matrix is a 16 by 4096 matrix, where every row represents a channelization code/signature. The cell-specific scrambling code is then multiplied with the newly constructed channelization code matrix to construct the matched filter coefficient matrix in the time domain as

$$h_{scmf}[s, 2n+1] = s_{code}[n]c_{sig}[s, n], \quad \text{for } n = 0, 1, ..., n_{max}-1 \text{ and } s = 0, 1, ..., n_{sign}-1 \tag{5.42}$$

where s represent the signatures, n_{sign} is the maximum number of signatures, n is the number of samples and n_{max} is the length of the scrambling code. Since the received signal is represented in double chip rate, the matched filter coefficients are upsampled to double chip rate, which is done by predefining matrix h_{scmf} with zeroes, where the length of the rows are double the length of the rows of the channelization code matrix. Furthermore, depending on the number of accumulated parts, defined in equation (5.37), the matched filter coefficients are divided into a number of parts. For example, with variable COH set to four, the filter coefficients are not divided since equation (5.38) states that $n_{part} = 1$. However, when COH is equal to two, equation (5.38) states that $n_{part} = 4$, thus the filter coefficients are divided into four equal parts with length $n_L = 2^{12}$ (defined in equation (5.38)) as shown in Fig. 5.12a. Lastly, the rows in the matched filter coefficients matrix are flipped from left



(a) Time domain **(b)** Frequency domain

Figure 5.12: Matched filter coefficients structure for four part accumulation of preamble signal.

to right and transformed to the frequency domain with the FFT. The purpose of

flipping the coefficients is to fulfill the requirements for the matched filter operation, see equation (4.4). The number of elements from every row included in the Fourier transform are dependent on the number of parts to be accumulated and an example with four parts is shown in Fig. 5.12. The matched filter coefficients are denoted H_{scmf} and are structured as a three dimensional matrix where one matrix represents one part and one row represents one signature.

5.10 Obtaining data

The data used in the MATLAB version are obtained from ascii files created by lstools from the Ericsson simulator and read into MATLAB with the function fetch data and can be seen in A.8.

6
Computational complexity

To increase the performance of the preamble detector more complex algorithms, such as interference cancellation, could be used. However, the RACH has a tight processing time and more advance algorithms would make it difficult to use.

Also, by using a time domain solution compared to a frequency domain would also yield an increase in complexity. For instance the FFT element-wise multiplication was used instead of convolution three times in the temporal whitening and one time in the signature code matched filter. The received signal is defined as $\mathbf{x}$ with a length of N samples and the filter coefficients are defined as $\mathbf{h}$ with a length of N samples. The convolution of the received signal with the filter coefficients would result in $2N-1$ samples, where the computation of each of the outputs requires approximately N^2 computations, hence the overall computational complexity is $O(N^2)$. Since we instead calculated the convolution in the frequency domain as

$$z = \mathcal{F}^{-1}\{\mathcal{F}\{\mathbf{x}\}\mathcal{F}\{\mathbf{h}\}\} \tag{6.1}$$

where each transform requires $O(N \log_2 N)$ operations, the multiplication requires $O(N)$ and the inverse Fourier transform requires $O(N \log_2 N)$ operations. This results in an overall computation of $O(N \log_2 N)$ and therefore fewer computations are needed compared to the convolution. However, the Fourier transform assumes that the signal is periodic and thus the results is periodic. To circumvent the issue of circular convolution zero padding is used. If the received signal $\mathbf{x}$ is zero padded to length M and the filter coefficients $\mathbf{h}$ is still of length N, then the resulting convolution will be $N+M-1$ samples, thus there will be no "wrap-around" [26]. In this algorithm the zero padding results in approximately doubling the length of the received signal and doubling the length of the filter coefficients used in the temporal whitening and in the signature code matched filter.

7
Results

This chapter includes the results and discussions from the three different algorithms showcased in chapter 5. Firstly, all simulation parameters relevant to the preamble detection procedure are listed, including a description of the simulation environments used. Secondly, the results from the frequency domain baseline algorithm compared to the time domain baseline algorithm are presented. Thereafter, results from the baseline algorithm using the temporal whitening algorithm in the frequency domain are shown. Lastly, the results from the baseline algorithm using both the temporal and spatial whitening algorithms are presented.

All algorithms are compared to the time domain baseline algorithm to give a frame of reference to the performance. In addition, the simulations are set up to showcase the performance of the algorithm under varying circumstances, as to ensure the robustness of the algorithm.

7.1 Simulation parameters and environments

The simulations done in this project are only in the scope of the preamble detection procedure in the RACH. The simulations are carried out at the receiver-side and thus entail the performance of the algorithm in probability of detection and probability of false detection. In this thesis the parameters with the most impact on performance are: the energy of the signal sent by the UE and the energy of the interference (in this case MAI), transmission environment (channel model) for the UE and the interfering transmitter, as well as the velocity of both the UE and interfering transmitter. The energy of a signal in a radio frequency communication system is most commonly described in terms for SNR as the energy ratio between the signal bits and the noise $\frac{E_b}{N_0}$. However, in a WCDMA-system the bits are spread with codes and the noise is measured in interference, since the most limiting factor in a coded system is the interference between users. Thus, the ratio is instead called chip-to-interference ratio. All simulations are run such that the detection probability spans from 0% to 100% versus the corresponding chip-to-interference ratio. The following parameters are used for all simulation environments.

- The maximum number of frames sent and successfully detected is set to 10 000.
- The velocity of the UE is 0 km/h for the AWGN channel and 3 km/h for the Pedestrian channel.
- The carrier frequency is 2 GHz.

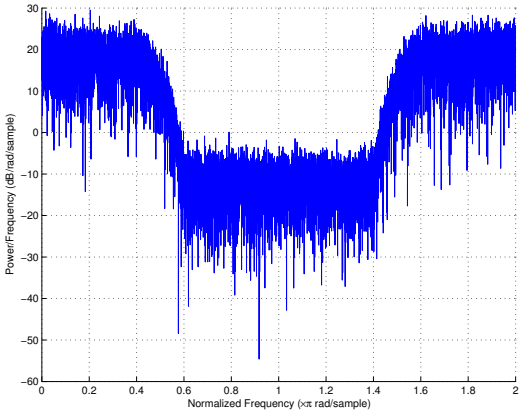
- At chip rate, the SF is equal to 256.
- Oversampling is set to 2.
- The preamble search window is set to 256 chips. Thus the cell size for the RBS is 10 km.
- Only signature nine is used by the transmitter.
- Two antennas are used on the receiver side.

Five different simulation environments are utilized in these simulations. An AWGN channel is used to showcase the performance for a basic noise model. For macrocellular simulations over a fading channel environment Pedestrian A (PA), Vehicular A (VA), Typical Urban (TU) and Rural Area (RA) channels are used [25] [24].

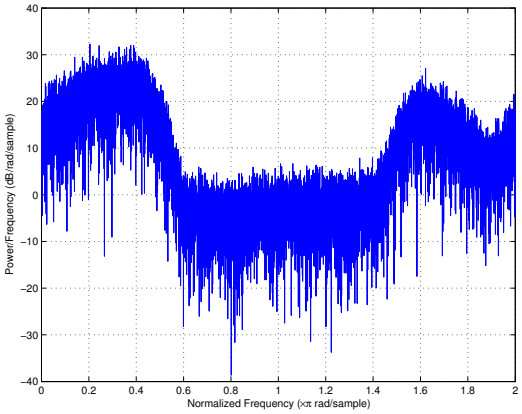
The PA channel is utilized to simulate an environment where the UE is located on a street or inside a building and is connected to a RBS located outdoors. The VA channel simulates an environment where the UE is driving in either a rural or urban area. The velocity of the vehicle also introduces a Doppler shift, see Sec. 4.4.3 for more details. The TU channel is used to simulate an urban transmission environment with high delay spread. The urban transmission environment leads to a high amount of multipath reflections and refractions and is therefore an interesting simulation environment to evaluate. The last channel used is the RA channel which simulates a rural transmission environment with typically one direct path (LOS). In a rural area, there are typically not many obstructions, thus the low delay spread. Table 7.1 details the fading profile of the channels, as well as the velocities of the UE used in conjunction with the channels.

Channel	σ_{RMS} **(ns)**	B_c **(MHz)**	**Doppler spectrum and n.o. taps**	**Interferer velocity (km/h)**
PA	45	22.2	3 Rayleigh	30
VA	370	2.7	5 Rayleigh	150
RA	140	7.14	1 Rician (LOS) & 9 Rayleigh	0
TU	500	2	20 Rayleigh	0
Case 3	≈250	4	3 Rayleigh	0

Table 7.1: Description of the channels in terms of RMS delay spread, coherence bandwidth, doppler spectrum and interferer velocity.



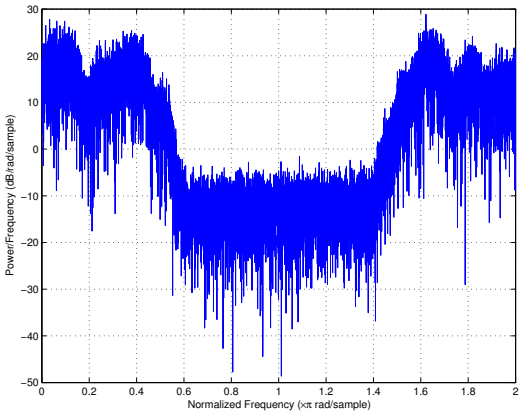
(a) The PA channel



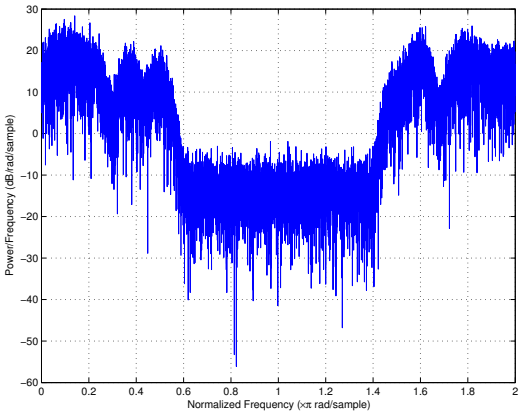
(b) The RA channel

Figure 7.1: PSD of the received signal when the interferer transmits through one of the channels defined in table 7.1.

The power spectral density of the received signal, with an interferer transmitting through different channels, is presented in Fig. 7.1 and 7.2. The received signal experiences frequency flat fading when the interferer transmits through the PA and RA channels. However, in opposite to the PA and RA channels, the received signal experiences frequency-selective fading when the interferer transmits through the VA and TU channels.



(a) The VA channel



(b) The TU channel

Figure 7.2: PSD of the received signal when the interferer transmits through one of the channels defined in table 7.1.

The following cases given in table 7.2 are used in the simulations. All cases are defined for channels given in table 7.1.

Cases	**UE channel**	**Interferer channel**
Case 1	AWGN	No interferer
Case 2	Case 3	No interferer
Case 3	AWGN	PA
Case 4	AWGN	RA
Case 5	AWGN	TU
Case 6	AWGN	VA
Case 7	PA	PA
Case 8	PA	RA
Case 9	PA	TU
Case 10	PA	VA
Case 11	PA	No interferer

Table 7.2: Description of the cases used for the simulations

7.2 FFT-based baseline algorithm versus time domain baseline algorithm

The following results show the performance of the FFT-based baseline algorithm compared to the performance of a baseline time domain algorithm at Ericsson. The algorithms are evaluated for transmission over two different channels and two different values of COH.

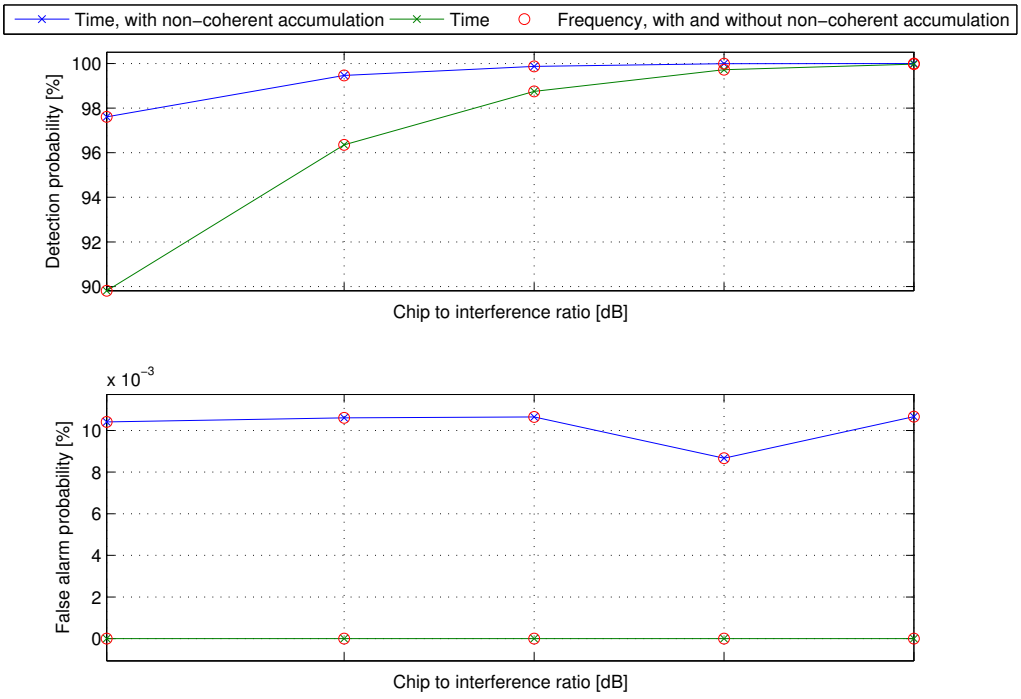


Figure 7.3: Case 1: Time based algorithm versus FFT-based algorithm

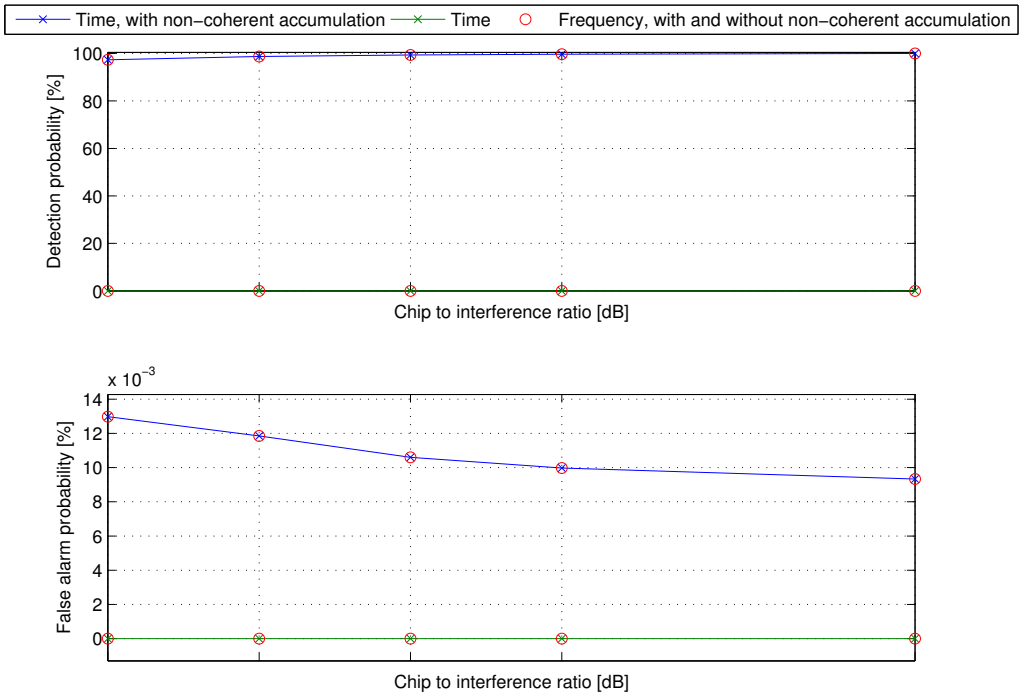


Figure 7.4: Case 1 with 1040 Hz frequency error: Time domain algorithm versus FFT-based algorithm.

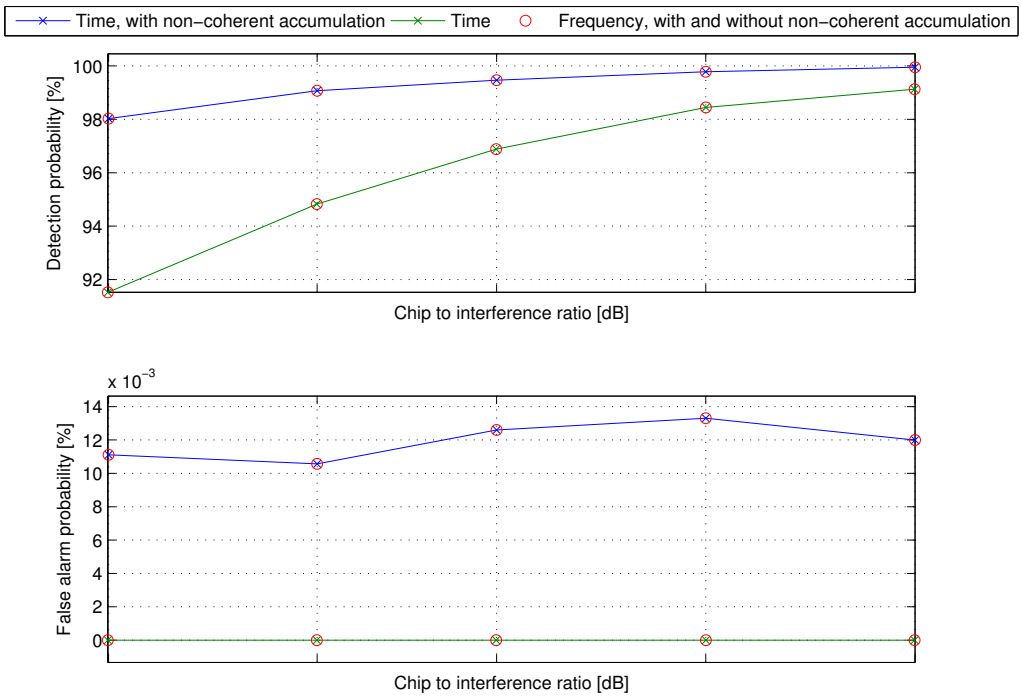


Figure 7.5: Case 2: Time domain algorithm versus FFT-based algorithm.

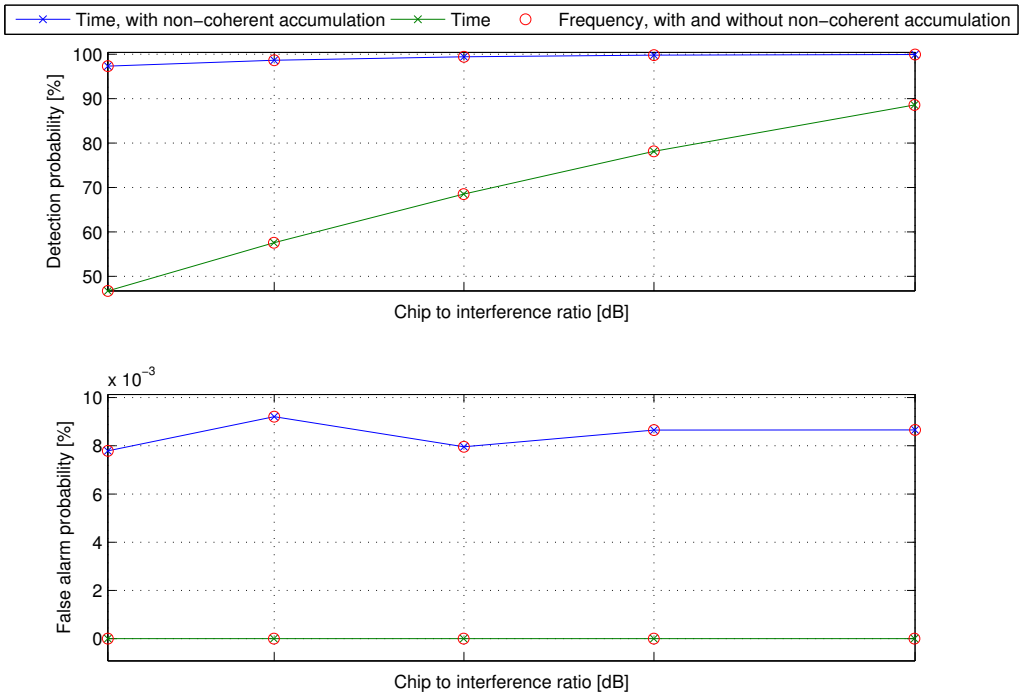


Figure 7.6: Case 2 with 500 Hz frequency error: Time domain algorithm versus FFT-based algorithm.

7.2.1 Discussion

The FFT-based baseline algorithm demonstrated identical performance to the equivalent time-domain algorithm. This was our prediction, since the only difference between the two implementations are that the convolutions are either done in the time domain or the frequency domain. Thus, because of the equivalence of time domain convolution and frequency domain elementwise multiplication, the performance should be equal. The algorithms were tested across several different scenarios and consistently gave the same results, thus we adopted our FFT-based algorithm in the future as a reference to the time-domain algorithm.

When we introduced a frequency error in the simulations the performance of the algorithms degraded significantly, see Fig. 7.4 and 7.6. This was, however, circumvented by using a coherent accumulation of the preamble which allowed the channel, through which the preamble is sent, to change during the preamble interval. Thus, the performance of the algorithm was greatly improved in both the AWGN and Case 3 channel with frequency error, when using coherent accumulation.

7.3 FFT-based algorithm with temporal whitening

The following results show the performance of the FFT-based algorithm with temporal whitening compared to the performance of the time-domain baseline algorithm. The algorithms are evaluated when the UE transmits over two different channels, with an interferer transmitting simultaneously over four different channels. The default preamble detection threshold for the FFT-based algorithm with temporal whitening is slightly lowered to produce approximately the same amount of false detections as the baseline algorithm.

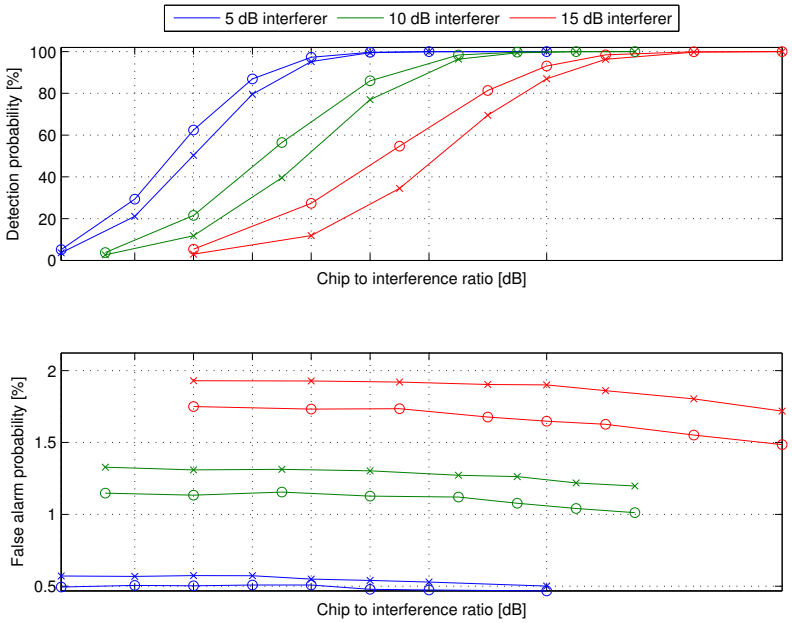


Figure 7.7: Case 3: FFT-based algorithm with temporal whitening (lines with circles) versus the baseline algorithm (lines with crosses)

Interferer[dB]	Baseline[%]	Temporal whitening[%]	Difference[%]	Ratio
5	50.3	62.4	12.11	1.24
10	39.58	56.45	16.86	1.43
15	34.48	54.71	20.24	1.59

Table 7.3: Case 3: Maximum difference in detection probability between the algorithms for different interference power levels.

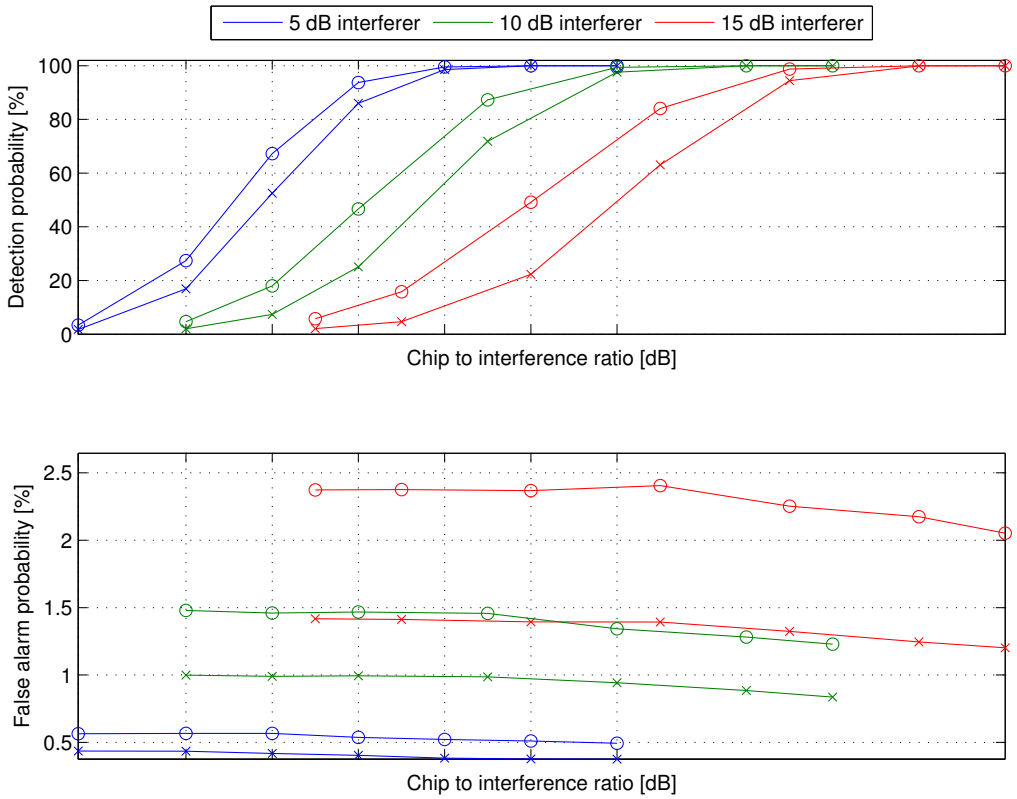


Figure 7.8: Case 4: FFT-based algorithm with temporal whitening (lines with circles) versus the baseline algorithm (lines with crosses)

Interferer[dB]	Baseline[%]	Temporal whitening[%]	Difference[%]	Ratio
5	52.51	67.25	14.74	1.28
10	25.1	46.64	21.54	1.86
15	22.28	49.15	26.86	2.21

Table 7.4: Case 4: Maximum difference in detection probability between the algorithms for different interference power levels.

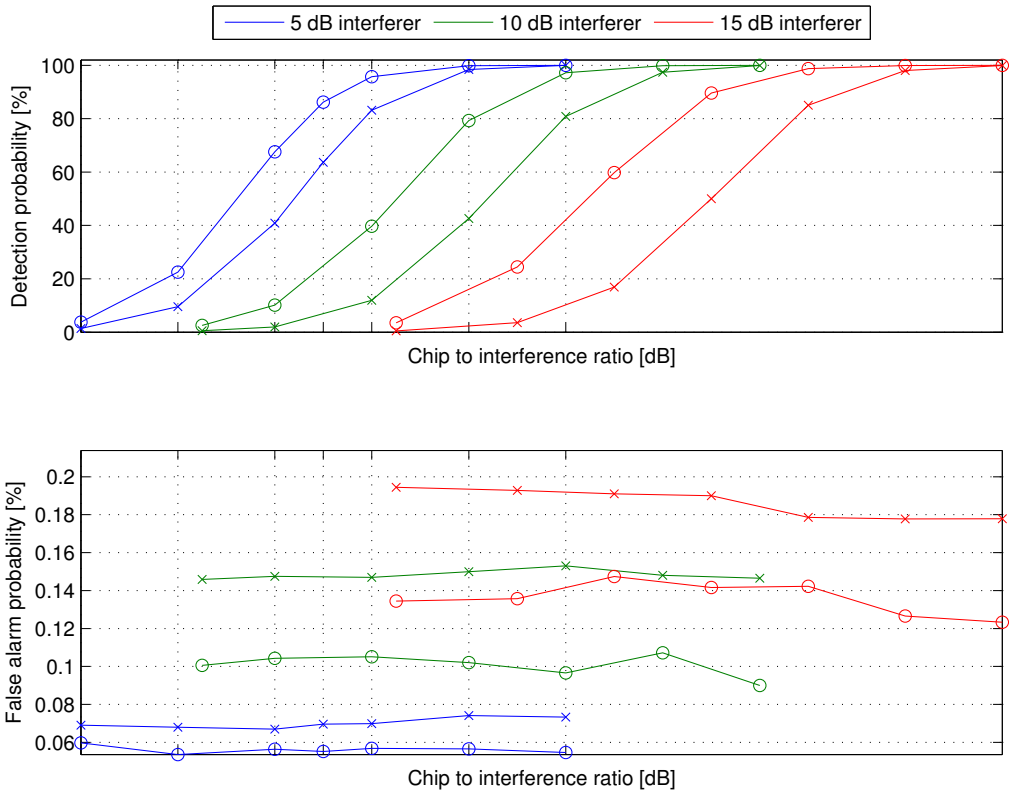


Figure 7.9: Case 5: FFT-based algorithm with temporal whitening (lines with circles) versus the baseline algorithm (lines with crosses)

Interferer[dB]	Baseline[%]	Temporal whitening[%]	Difference[%]	Ratio
5	40.83	67.59	26.77	1.66
10	42.61	79.3	36.69	1.86
15	16.89	59.78	42.89	3.54

Table 7.5: Case 5: Maximum difference in detection probability between the algorithms for different interference power levels.

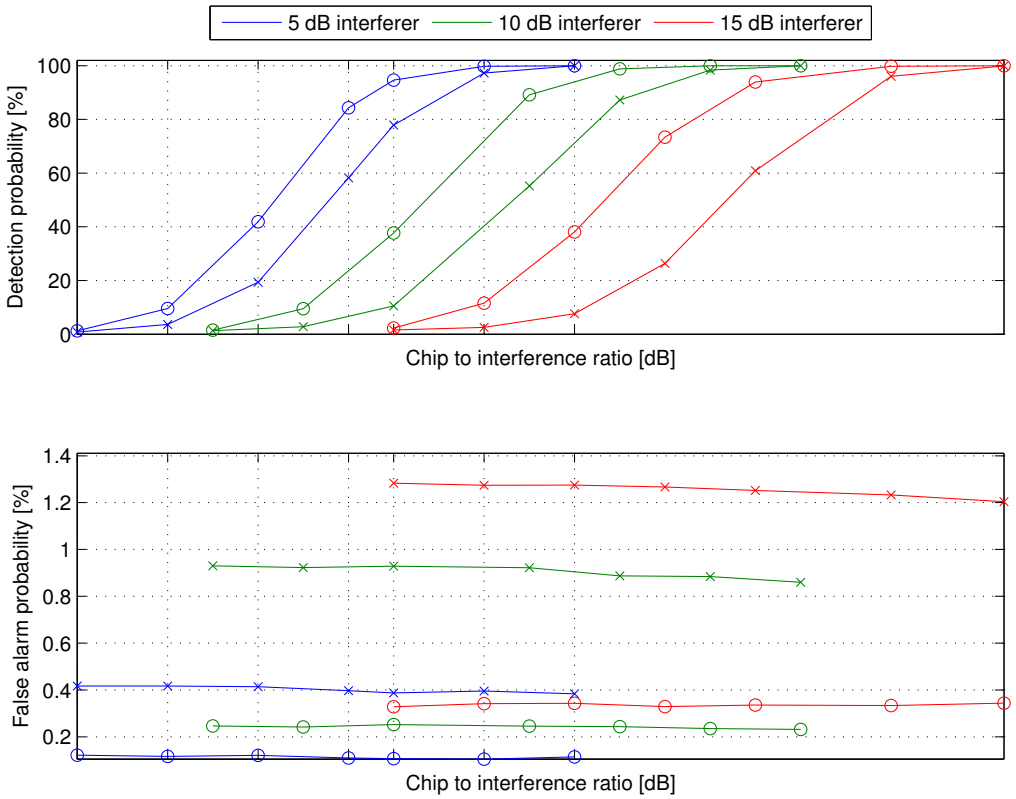


Figure 7.10: Case 6: FFT-based algorithm with temporal whitening (lines with circles) versus the baseline algorithm (lines with crosses)

Interferer[dB]	Baseline[%]	Temporal whitening[%]	Difference[%]	Ratio
5	58.24	84.35	26.11	1.45
10	55.19	89.16	33.96	1.62
15	26.37	73.41	47.03	2.78

Table 7.6: Case 6: Maximum difference in detection probability between the algorithms for different interference power levels.

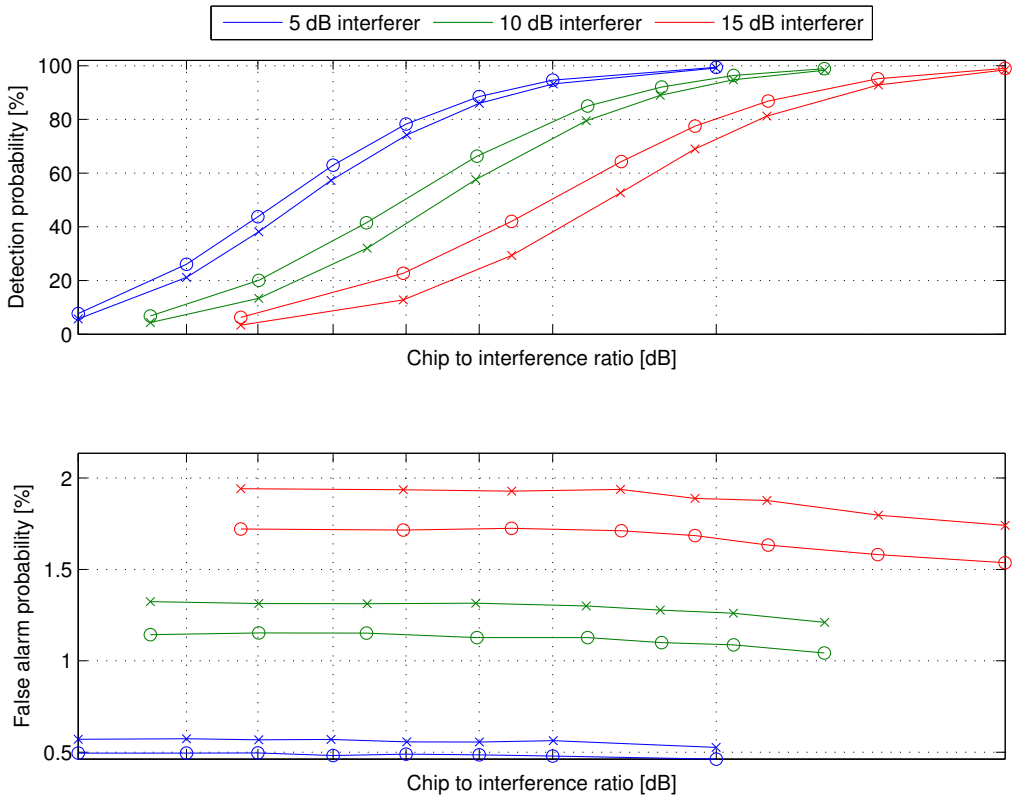


Figure 7.11: Case 7: FFT-based algorithm with temporal whitening (lines with circles) versus the baseline algorithm (lines with crosses)

Interferer[dB]	Baseline[%]	Temporal whitening[%]	Difference[%]	Ratio
5	57.29	62.97	5.68	1.1
10	32.04	41.5	9.46	1.3
15	29.31	42.02	12.71	1.43

Table 7.7: Case 7: Maximum difference in detection probability between the algorithms for different interference power levels.

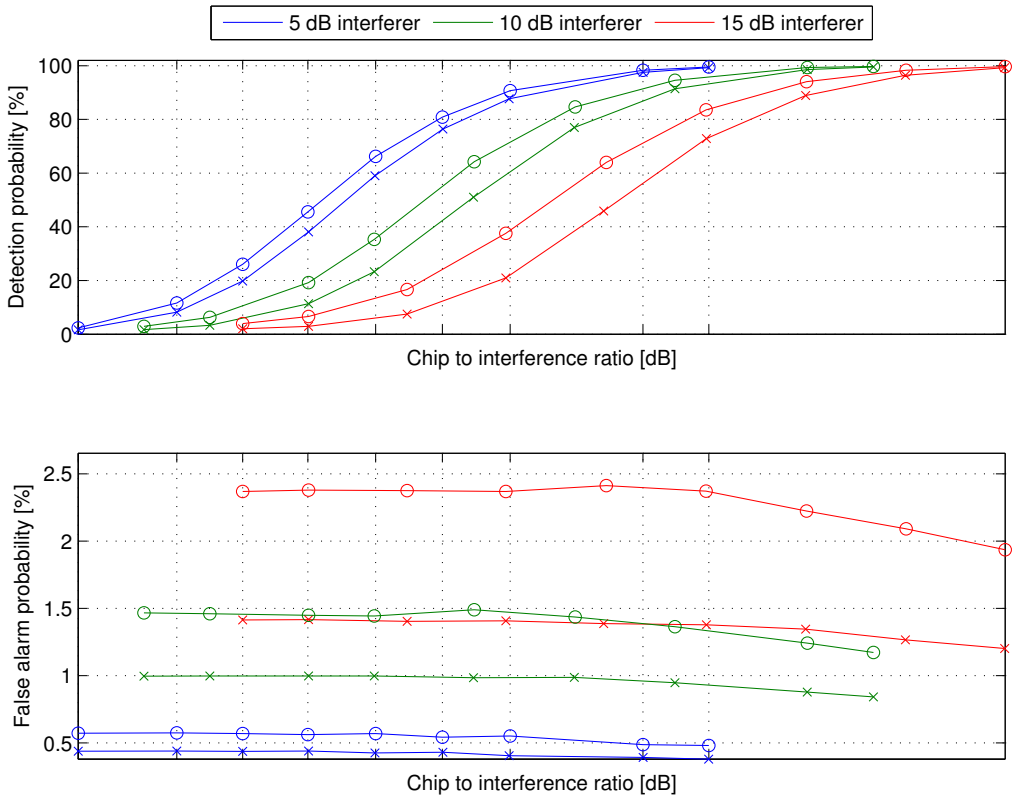


Figure 7.12: Case 8: FFT-based algorithm with temporal whitening (lines with circles) versus the baseline algorithm (lines with crosses)

Interferer[dB]	Baseline[%]	Temporal whitening[%]	Difference[%]	Ratio
5	38.07	45.59	7.52	1.2
10	51.02	64.25	13.23	1.26
15	45.86	64	18.15	1.4

Table 7.8: Case 8: Maximum difference in detection probability between the algorithms for different interference power levels.

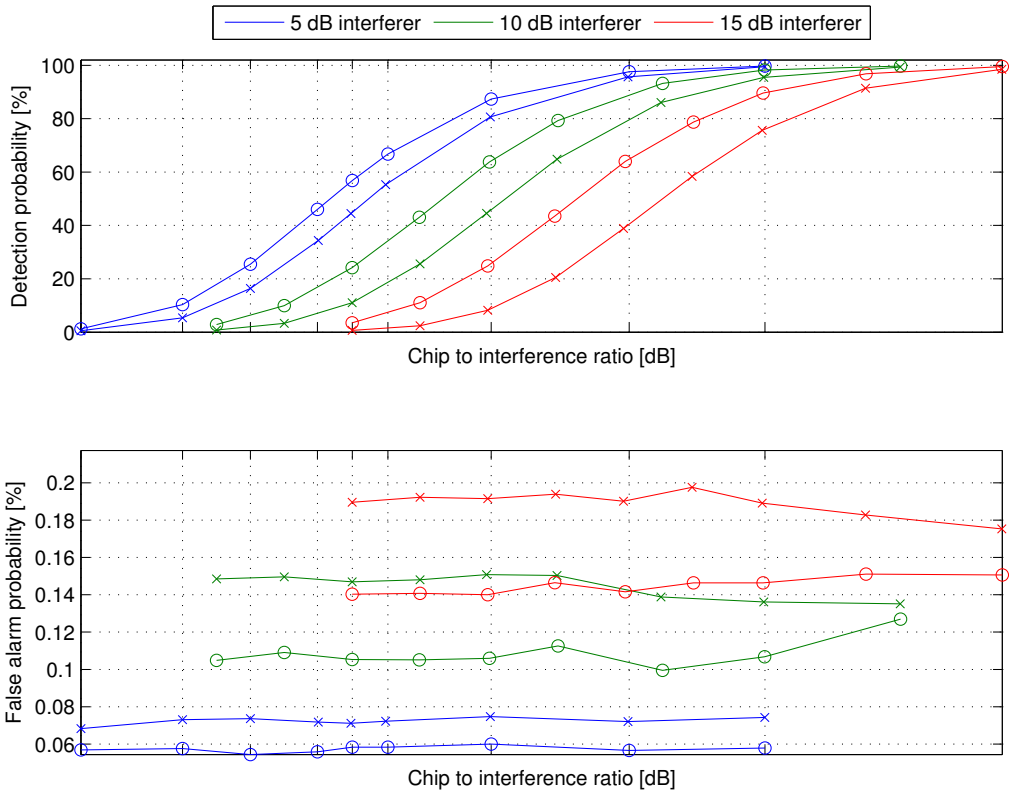


Figure 7.13: Case 9: FFT-based algorithm with temporal whitening (lines with circles) versus the baseline algorithm (lines with crosses)

Interferer[dB]	Baseline[%]	Temporal whitening[%]	Difference[%]	Ratio
5	44.44	56.8	12.36	1.28
10	44.52	63.8	19.28	1.43
15	38.88	63.97	25.09	1.65

Table 7.9: Case 9: Maximum difference in detection probability between the algorithms for different interference power levels.

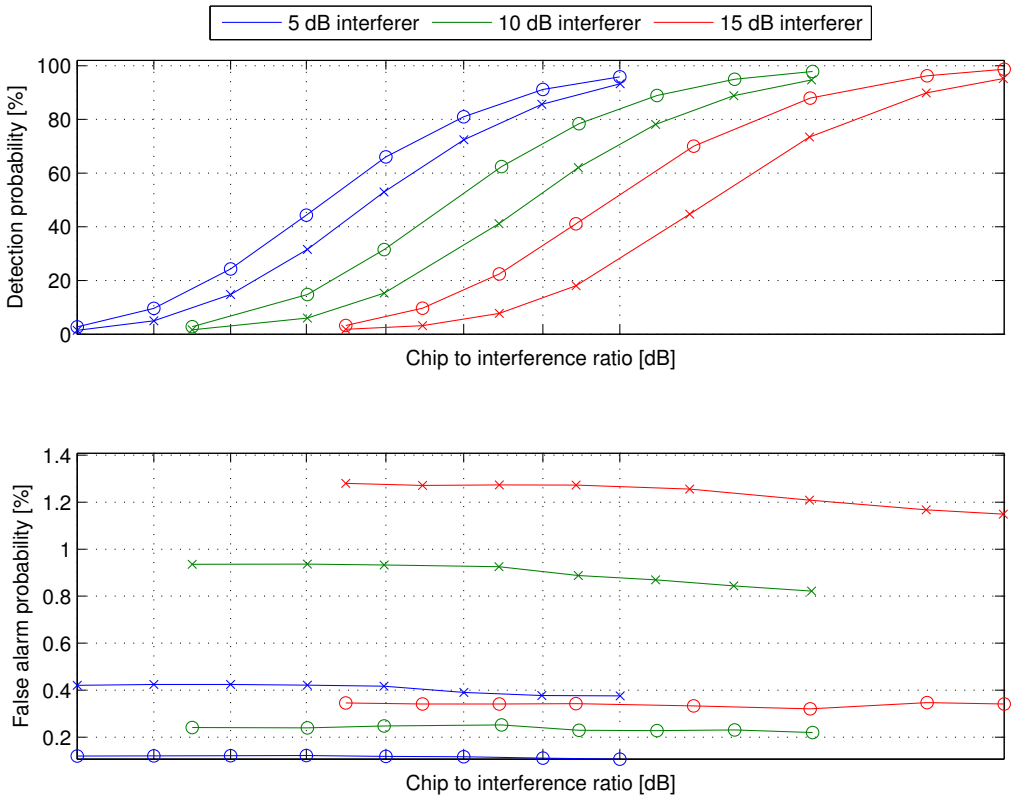


Figure 7.14: Case 10: FFT-based algorithm with temporal whitening (lines with circles) versus the baseline algorithm (lines with crosses)

Interferer[dB]	Baseline[%]	Temporal whitening[%]	Difference[%]	Ratio
5	52.97	66.05	13.08	1.25
10	41.16	62.41	21.24	1.52
15	44.7	70.02	25.31	1.57

Table 7.10: Case 10: Maximum difference in detection probability between the algorithms for different interference power levels.

7.3.1 Discussion

The FFT-based algorithm with temporal whitening improved the performance of the algorithm in relation to the baseline algorithm. The more power with which the interferer was transmitting, the greater the performance gain. The smallest performance gain was seen when the interferer was transmitting through a PA and RA channel, see for example Fig. 7.11 and 7.12. This is most likely due to the fact that the channels are less dispersive than the TU and VA channel, which can be seen in Fig. 7.2. We assume that the coherence bandwidth for both the RA and PA channel is greater than the bandwidth of the preamble, thus the channel is "flat" for the duration of the preamble and the effect of whitening the spectrum is minimal. The

temporal whitening procedure essentially equalizes the spectrum. Thus, by looking at the spectrum of the received signal affected by an interferer transmitting through a PA or RA channel, see Fig. 7.1, it is easy to see that there is less to equalize than if the interferer was transmitting through a VA or TU channel. Hence, the baseline algorithm performs better in the RA and PA channels relative to the algorithm with the temporal whitening.

The largest performance gain was seen when the interferer was transmitting through a VA or TU channel, especially in the case where the interferer was transmitting through a VA channel, where the detection probability increased by a maximum of approximately 47% relative to the baseline algorithm, see Fig. 7.10. Again, by looking at the received spectrum when the interferer transmits through a VA or TU channel, it is clear that the interferer introduces faster fading of the spectrum and thus the need of whitening the spectrum increases.

7.4 FFT-based algorithm with temporal whitening and extended zero padding for FFT.

The following results show the performance of the frequency-domain baseline algorithm with temporal whitening and extended zero padding for the FFT, compared to the performance of the temporal whitening algorithm with less zero padding. The results are very similar for several other channels and can be seen in appendix B.1.

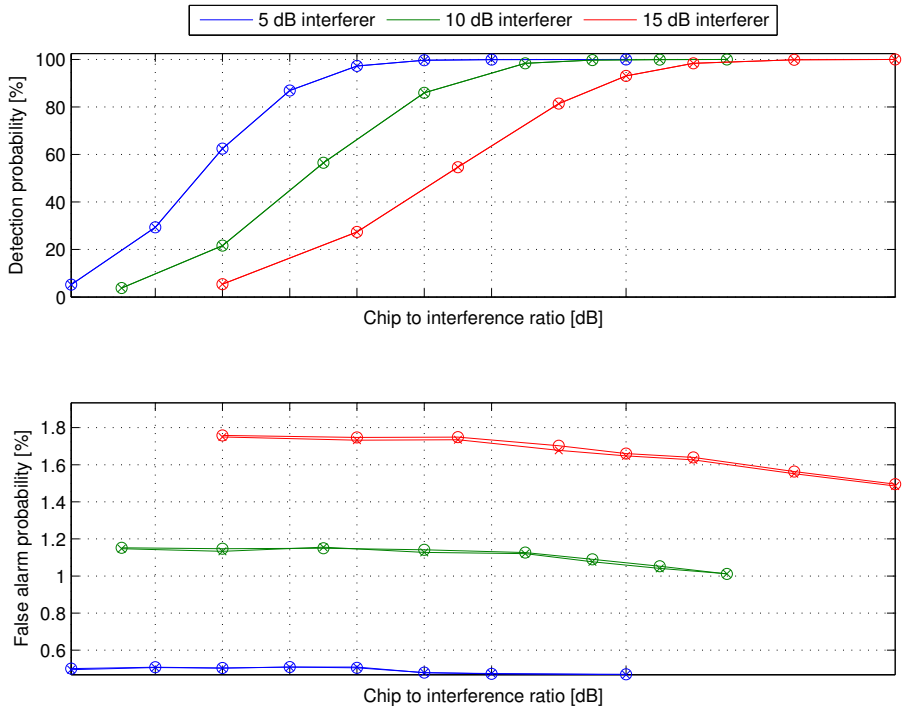


Figure 7.15: Case 3: FFT-based algorithm with temporal whitening and extended zero padding (lines with circles) versus the temporal whitening algorithm with less zero padding (lines with crosses)

Interferer[dB]	Baseline[%]	Temporal whitening[%]	Difference[%]	Ratio
5	62.4	62.46	0.06	1
10	56.45	56.54	0.09	1
15	81.35	81.45	0.1	1

Table 7.11: Case 3: Maximum difference in detection probability between the algorithms for different interference power levels.

7.4.1 Discussion

During the design of the temporal whitening algorithm we were confronted with the possibility of circular convolution being a problem. However, if we increased

the amount of zeroes padded to the signal before using the FFT, the effect circular convolution could be neglected. Thus, we ran the algorithm without extra zeroes padded to the signal versus the one with extra zeroes and the results showed very similar performance to the initial temporal whitening algorithm, see Fig. 7.15. This lead us to disregard the possibility of circular convolution in the scope of the algorithm in the future.

7.5 FFT-based algorithm with temporal whitening with different numbers of ACF lags

The following results show the performance of the frequency-domain baseline algorithm with temporal whitening with different numbers of autocorrelation function lags.

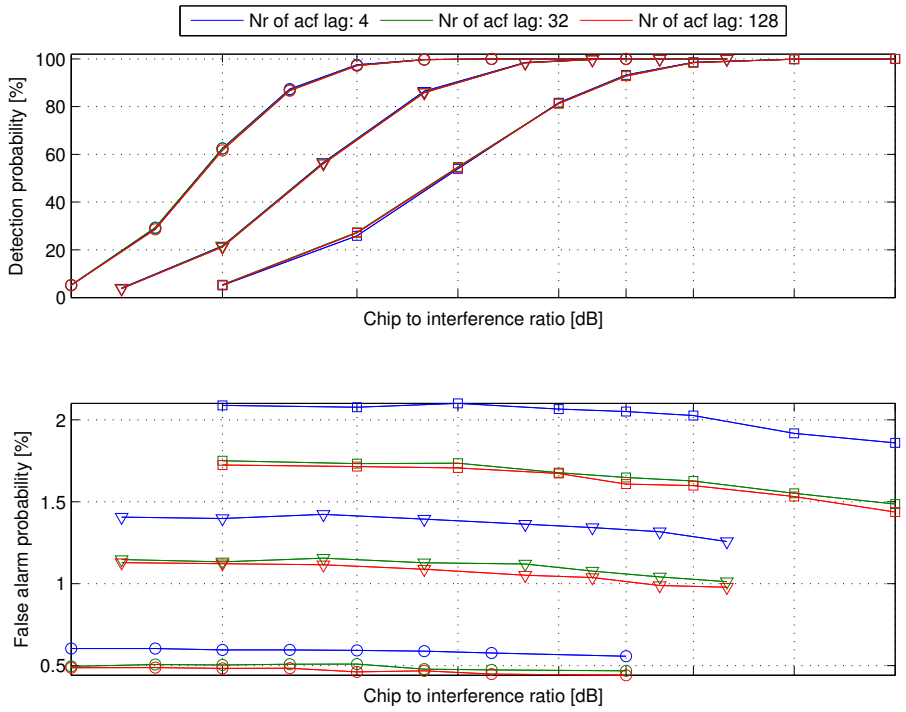


Figure 7.16: Case 3: FFT-based algorithm with temporal whitening with different numbers of ACF lags (lines of different colors). Circles, triangles and squares represent 5 dB, 10 dB and 15 dB interference power respectively

Interferer[dB]	Nr of acf lag 32[%]	Nr of acf	Difference[%]	Gain[dB]
5	86.92	4 lag at 87.36 %	0.44	1.01
10	86	4 lag at 86.48 %	0.48	1.01
15	81.35	4 lag at 81.65 %	0.29	1

Table 7.12: Case 3: Maximum difference in detection probability in relation to the algorithm with 32 ACF lags for different interference power levels.

7.5.1 Discussion

The number of ACF samples chosen when generating the temporal whitening filter was determined by running the same cases as previously with a different number of

samples of the ACF. When we chose less than 32 samples of the ACF, the shape of the root raised cosine could not be properly represented which lead to a loss of the characteristics of the PSD. Conversely, when more than 32 samples of the ACF were chosen, the characteristics of the PSD became too noisy. Thus, we chose the number of samples that showed the best performance overall, which was 32 samples.

7.6 FFT-based algorithm with temporal whitening of the signal in parts

The following results show the performance of the frequency-domain baseline algorithm with temporal whitening of the signal in different parts.

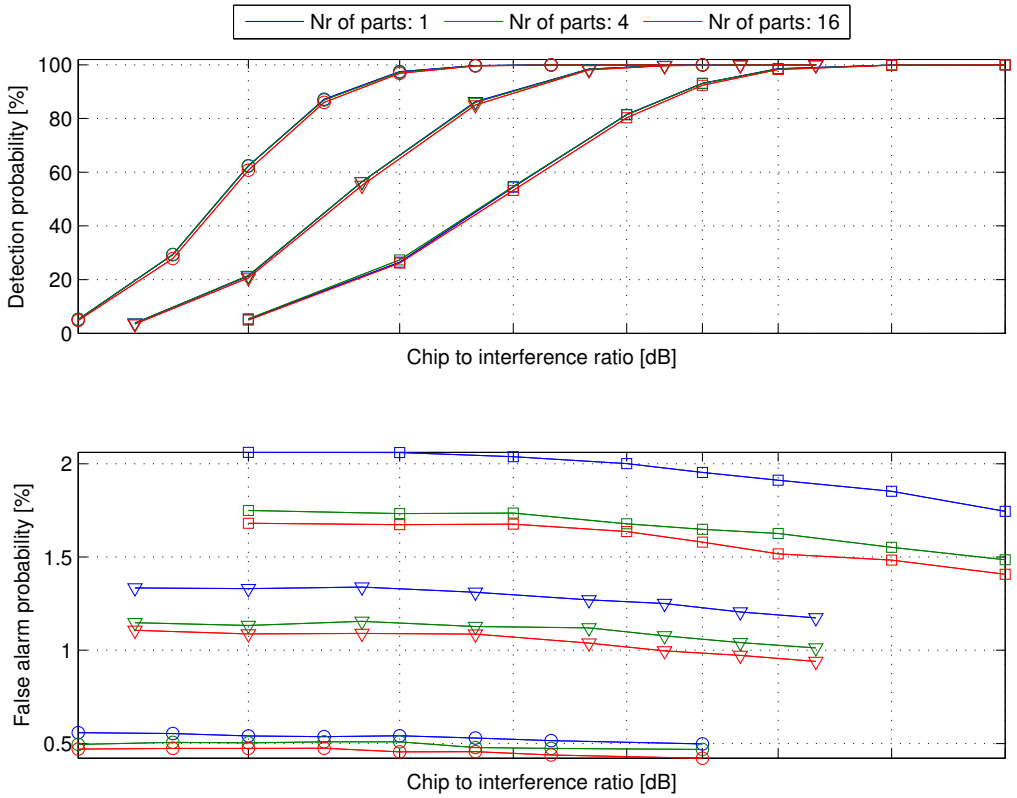


Figure 7.17: Case 3: FFT-based algorithm with temporal whitening of the signal in parts (lines of different colors). Circles, triangles and squares represent 5 dB, 10 dB and 15 dB interference power respectively

Interferer[dB]	Nr of parts 4[%]	Nr of parts	Difference[%]	Ratio
5	86.92	1 lag at 87.15	0.23	1
10	86	1 lag at 86.28	0.28	1
15	98.41	1 lag at 98.56	0.16	1

Table 7.13: Case 3: Maximum difference in detection probability for the algorithm with four parts in relation to other amounts of whitening parts for different interference power levels.

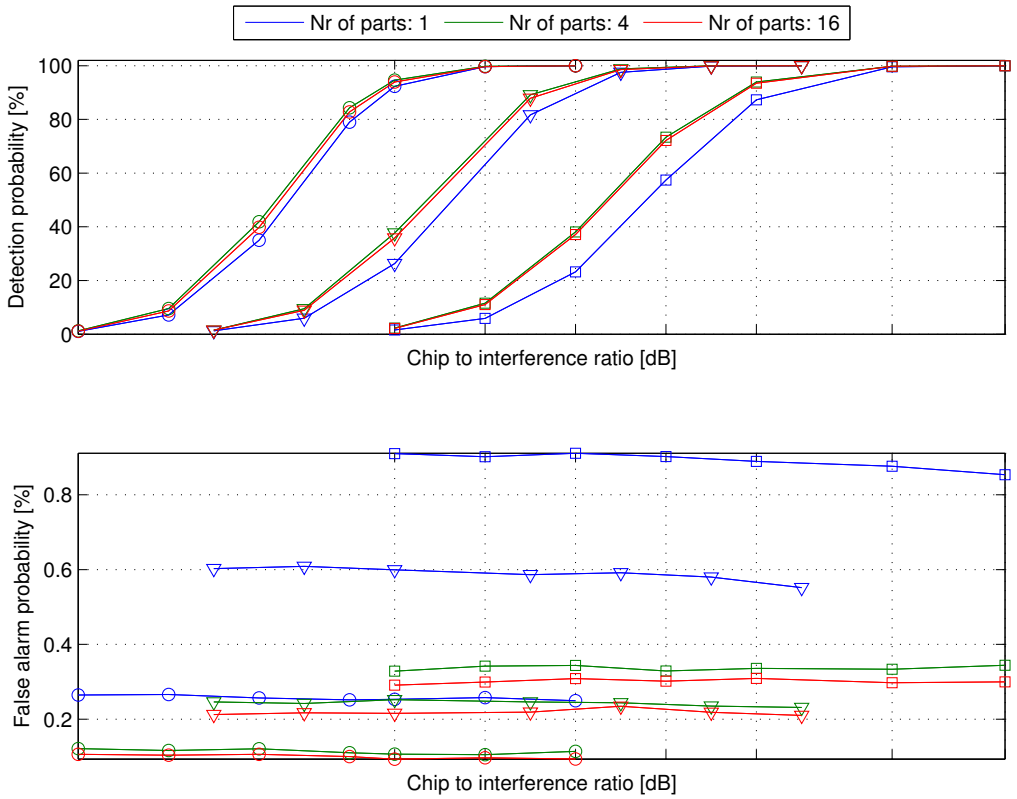


Figure 7.18: Case 6: FFT-based algorithm with temporal whitening of the signal in parts (lines of different colors). Circles, triangles and squares represent 5 dB, 10 dB and 15 dB interference power respectively

Interferer[dB]	Nr of parts 4[%]	Nr of parts	Difference[%]	Ratio
5	100	1 lag at 100	0	1
10	100	1 lag at 99.99	-0.01	1
15	99.99	1 lag at 99.99	0	1

Table 7.14: Case 6: Maximum difference in detection probability for the algorithm with four parts in relation to other amount of parts for different interference power levels.

7.6.1 Discussion

Temporally whitening the signal in parts improved the performance of the algorithm when the interferer was traveling at a high velocity. This is most likely due to the fact that the signal loses its WSS property when a Doppler shift is introduced. The basis for this procedure is the same as for the coherent accumulation, i.e., to let the channel, through which the preamble is sent, change during the preamble interval. The number of parts to split the preamble into and then temporally whiten was

determined by running multiple cases where we changed the number of parts. The cases concluded that four parts produced the best results overall.

However, in the case of the interferer transmitting though a PA channel, see Fig. 7.17, there was a marginal gain when temporally whitening the entire preamble. Our assumption is that this can be explained by that a bigger data set gives a better estimate of the spectrum when then channel is considered to be flat. Finally, since the algorithm exhibited the best performance overall when temporally whitening the preamble in four parts, we included it in future implementations of the algorithm.

7.7 FFT-based algorithm with spatial whitening

The following results show the performance of the FFT-based algorithm with spatial whitening compared to the performance of the time-domain baseline algorithm. The algorithms are evaluated when the UE transmits over two different channels, with an interferer transmitting simultaneously over four different channels. The default preamble detection threshold for the FFT-based algorithm with spatial whitening is once again lowered to produce the same amount of false detections as the baseline algorithm.

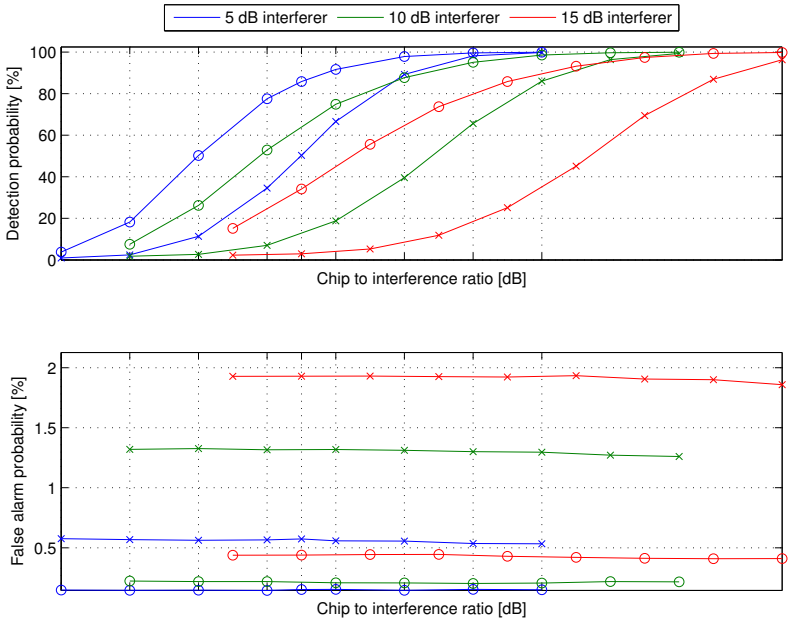


Figure 7.19: Case 3: FFT-based algorithm with spatial whitening (lines with circles) versus the baseline algorithm (lines with crosses)

Interferer[dB]	Baseline[%]	Spatial whitening[%]	Difference[%]	Ratio
5	34.56	77.59	43.02	2.24
10	18.8	74.88	56.08	3.98
15	11.9	73.77	61.88	6.2

Table 7.15: Case 3: Maximum difference in detection probability between the algorithms for different interference power levels.

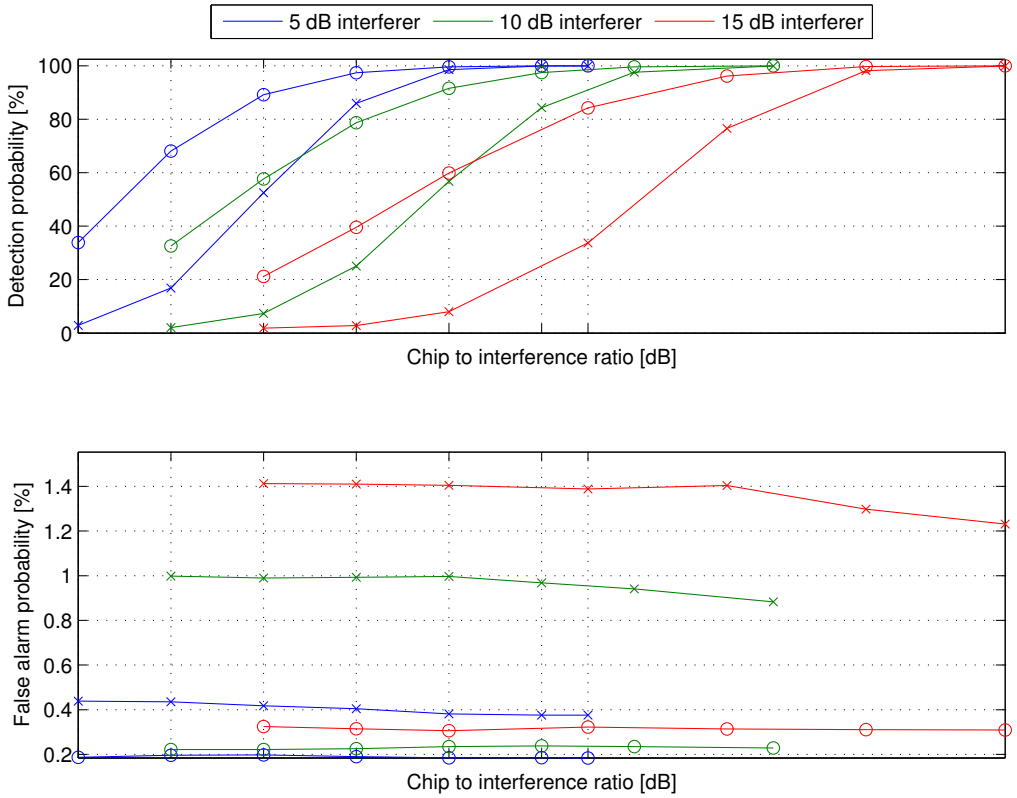


Figure 7.20: Case 4: FFT-based algorithm with spatial whitening (lines with circles) versus the baseline algorithm (lines with crosses)

Interferer[dB]	Baseline[%]	Spatial whitening[%]	Difference[%]	Ratio
5	16.87	68.09	51.22	4.04
10	25.1	78.75	53.65	3.14
15	7.99	59.87	51.88	7.49

Table 7.16: Case 4: Maximum difference in detection probability between the algorithms for different interference power levels.

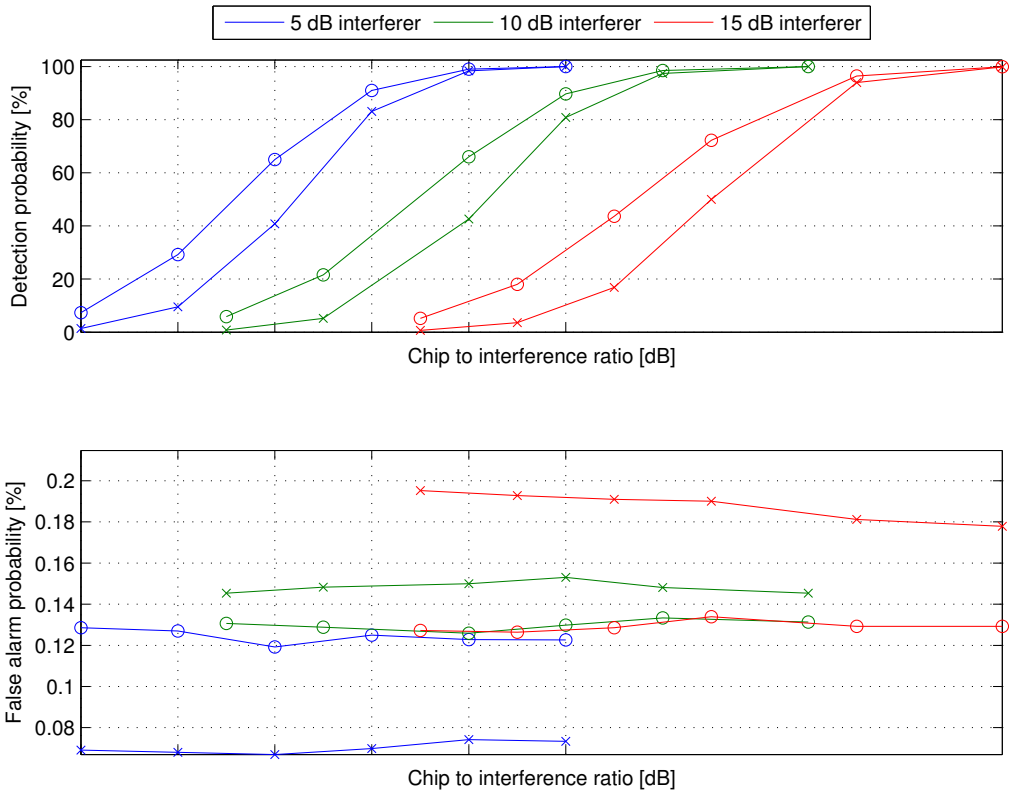


Figure 7.21: Case 5: FFT-based algorithm with spatial whitening (lines with circles) versus the baseline algorithm (lines with crosses)

Interferer[dB]	Baseline[%]	Spatial whitening[%]	Difference[%]	Ratio
5	40.83	65.01	24.18	1.59
10	42.61	66.04	23.43	1.55
15	16.89	43.63	26.74	2.58

Table 7.17: Case 5: Maximum difference in detection probability between the algorithms for different interference power levels.

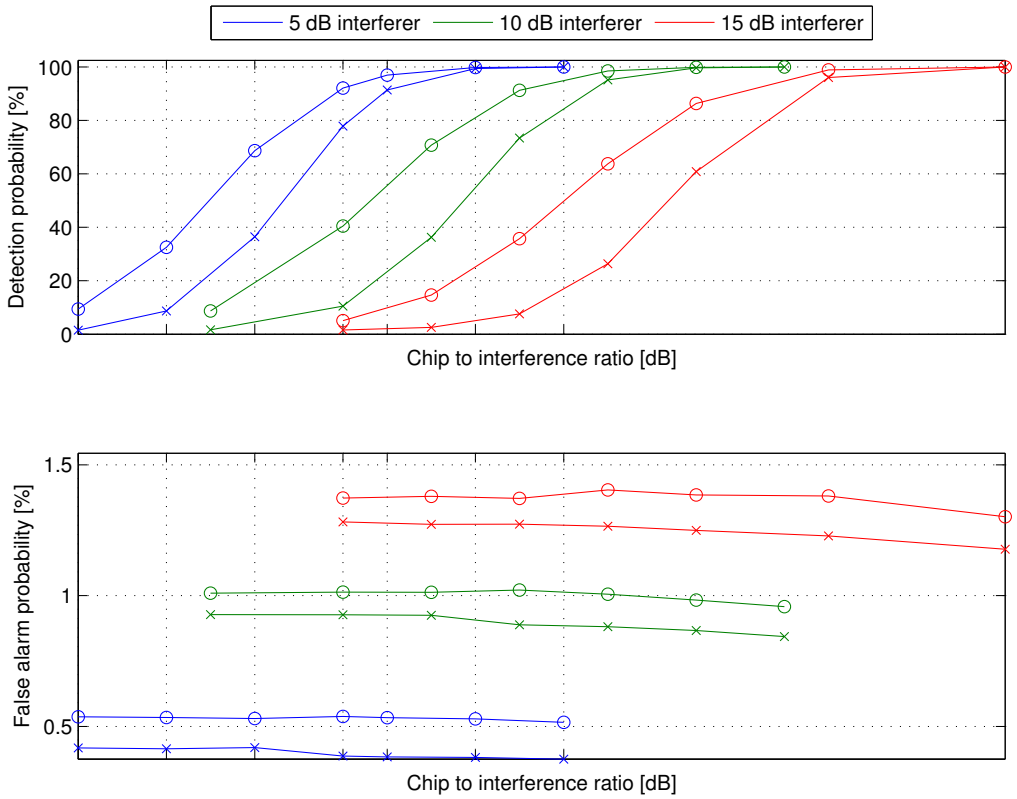


Figure 7.22: Case 6: FFT-based algorithm with spatial whitening (lines with circles) versus the baseline algorithm (lines with crosses)

Interferer[dB]	Baseline[%]	Spatial whitening[%]	Difference[%]	Ratio
5	36.48	68.67	32.19	1.88
10	36.3	70.75	34.45	1.95
15	26.37	63.78	37.41	2.42

Table 7.18: Case 6: Maximum difference in detection probability between the algorithms for different interference power levels.

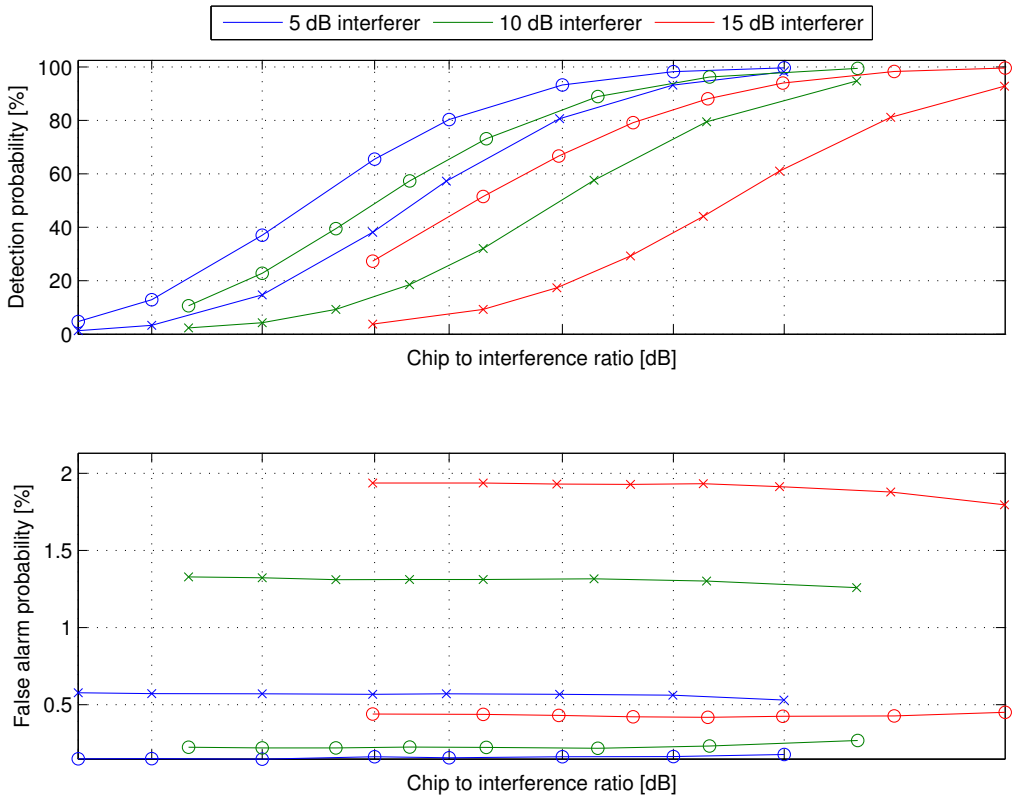


Figure 7.23: Case 7: FFT-based algorithm with spatial whitening (lines with circles) versus the baseline algorithm (lines with crosses)

Interferer[dB]	Baseline[%]	Spatial whitening[%]	Difference[%]	Ratio
5	38.18	65.56	27.38	1.72
10	32.04	73.19	41.15	2.28
15	29.31	79.19	49.88	2.7

Table 7.19: Case 7: Maximum difference in detection probability between the algorithms for different interference power levels.

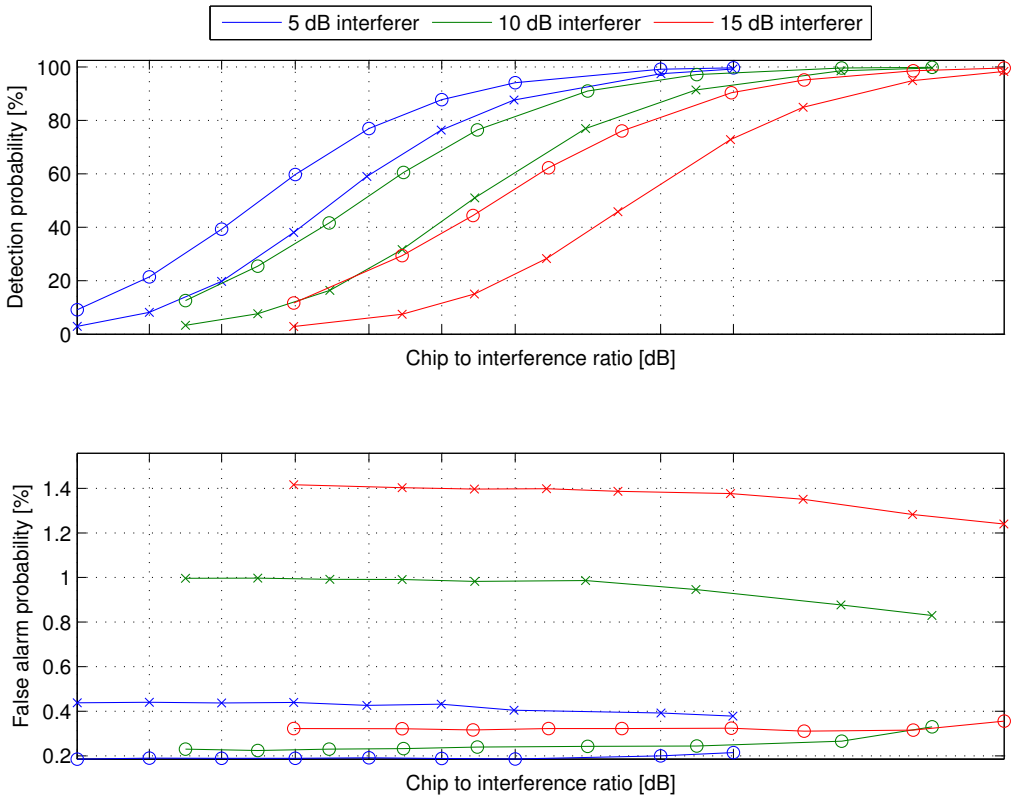


Figure 7.24: Case 8: FFT-based algorithm with spatial whitening (lines with circles) versus the baseline algorithm (represented as lines with crosses)

Interferer[dB]	Baseline[%]	Spatial whitening[%]	Difference[%]	Ratio
5	38.07	59.77	21.7	1.57
10	31.65	60.57	28.92	1.91
15	28.33	62.26	33.93	2.2

Table 7.20: Case 8: Maximum difference in detection probability between the algorithms for different interference power levels.

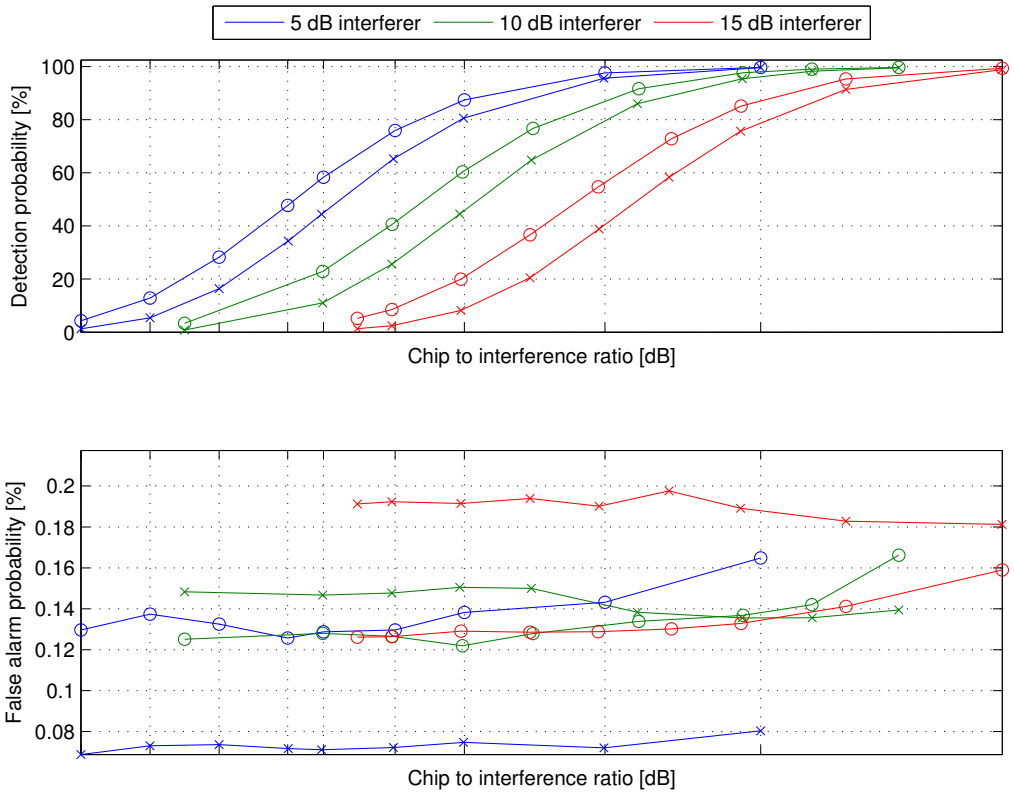


Figure 7.25: Case 9: FFT-based algorithm with spatial whitening (lines with circles) versus the baseline algorithm (lines with crosses)

Interferer[dB]	Baseline[%]	Spatial whitening[%]	Difference[%]	Ratio
5	44.44	58.33	13.89	1.31
10	44.52	60.35	15.82	1.36
15	20.49	36.64	16.15	1.79

Table 7.21: Case 9: Maximum difference in detection probability between the algorithms for different interference power levels.

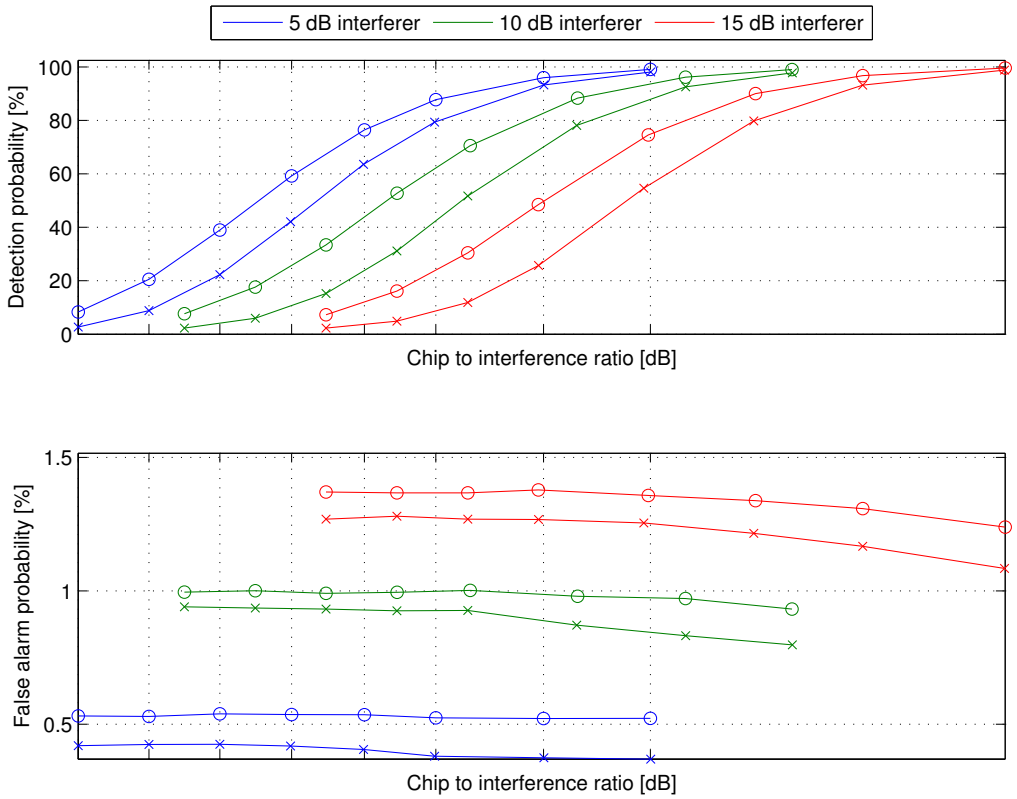


Figure 7.26: Case 10: FFT-based algorithm with spatial whitening (lines with circles) versus the baseline algorithm (lines with crosses)

Interferer[dB]	Baseline[%]	Spatial whitening[%]	Difference[%]	Ratio
5	42.1	59.23	17.13	1.41
10	31.15	52.75	21.61	1.69
15	25.74	48.53	22.79	1.89

Table 7.22: Case 10: Maximum difference in detection probability between the algorithms for different interference power levels.

7.7.1 Discussion

The FFT-based algorithm with spatial whitening yielded a substantial gain in performance when compared to the baseline algorithm. The performance gain increased relative to the baseline algorithm, when the interferer increased the transmission power, as for temporal whitening algorithm. The algorithm showed the best performance in mildly dispersive channels, i.e., flat channels, such as the RA and PA channels, see Fig. 7.19 and 7.20. However, the algorithm demonstrated poor performance in TU and VA channels, see Fig. 7.21 and 7.22, which is mostly likely due to the fact that the spatial whitening does not equalize the frequency selectivity of the channel.

7.8 FFT-based algorithm with temporal whitening and spatial whitening

The following results show the performance of the FFT-based algorithm with temporal whitening and spatial whitening compared to the performance of the time-domain baseline algorithm. The algorithms are evaluated when the UE transmits over two different channels, with an interferer transmitting simultaneously over four different channels. The default preamble detection threshold for the FFT-based algorithm with temporal and spatial whitening is once again lowered to produce the same amount of false detections as the baseline algorithm.

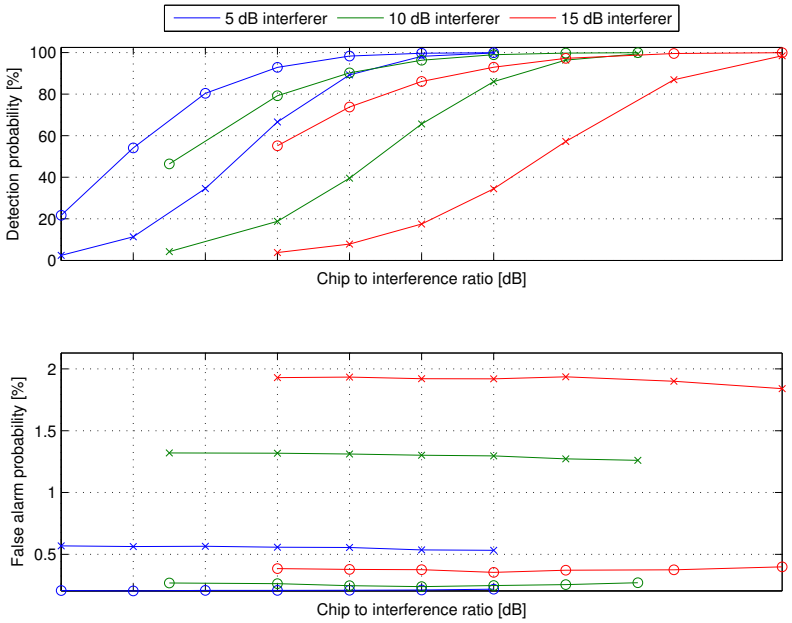


Figure 7.27: Case 3: FFT-based algorithm with temporal whitening and spatial whitening (lines with circles) versus the baseline algorithm (lines with crosses)

Interferer[dB]	Baseline[%]	Temporal and Spatial[%]	Difference[%]	Ratio
5	34.56	80.41	45.84	2.33
10	18.8	79.24	60.44	4.22
15	17.53	86.13	68.6	4.91

Table 7.23: Case 3: Maximum difference in detection probability between the algorithms for different interference power levels.

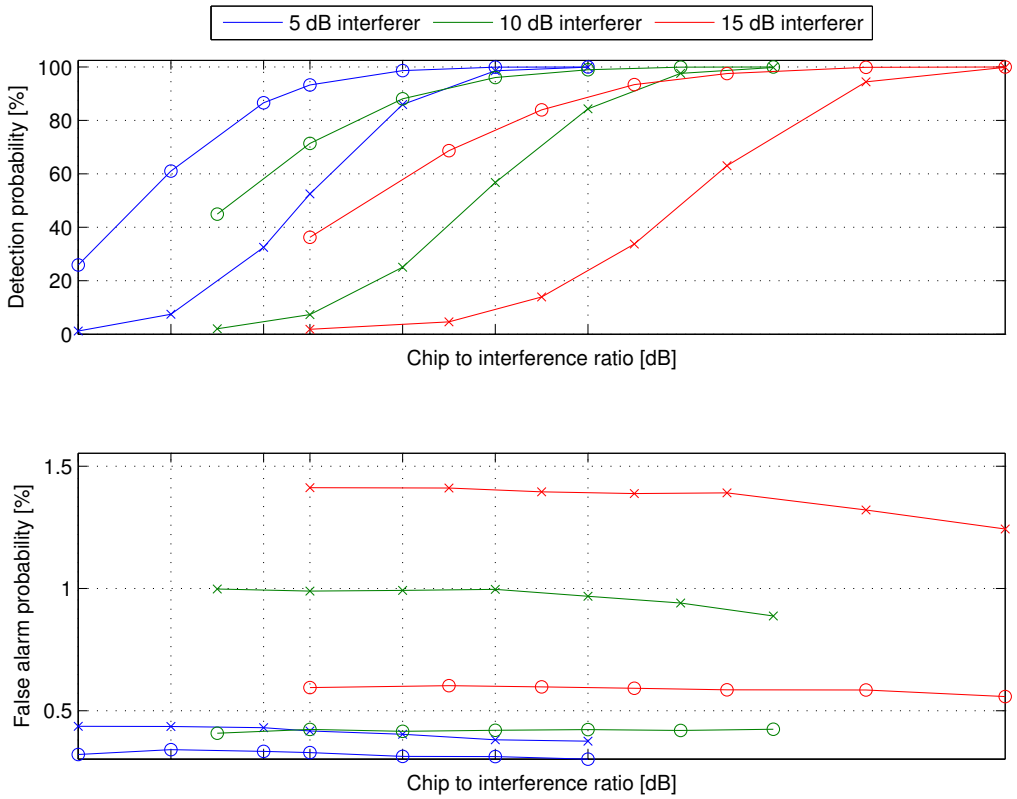


Figure 7.28: Case 4: FFT-based algorithm with temporal whitening and spatial whitening (lines with circles) versus the baseline algorithm (lines with crosses)

Interferer[dB]	Baseline[%]	Temporal and Spatial[%]	Difference[%]	Ratio
5	21.57	65.65	44.08	3.04
10	24.11	77.99	53.88	3.23
15	31.06	88.17	57.11	2.84

Table 7.24: Case 4: Maximum difference in detection probability between the algorithms for different interference power levels.

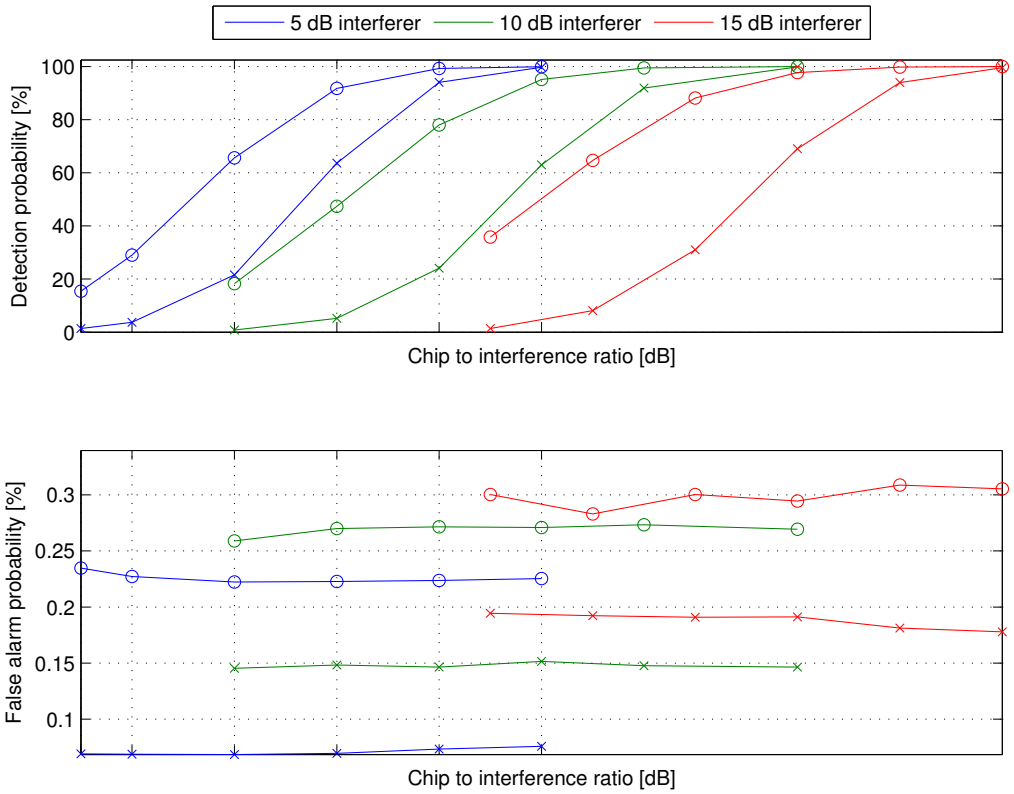


Figure 7.29: Case 5: FFT-based algorithm with temporal whitening and spatial whitening (lines with circles) versus the baseline algorithm (lines with crosses)

Interferer[dB]	Baseline[%]	Temporal and Spatial[%]	Difference[%]	Ratio
5	19.36	68.15	48.79	3.52
10	10.53	66.1	55.56	6.28
15	14.7	78.8	64.09	5.36

Table 7.25: Case 5: Maximum difference in detection probability between the algorithms for different interference power levels.

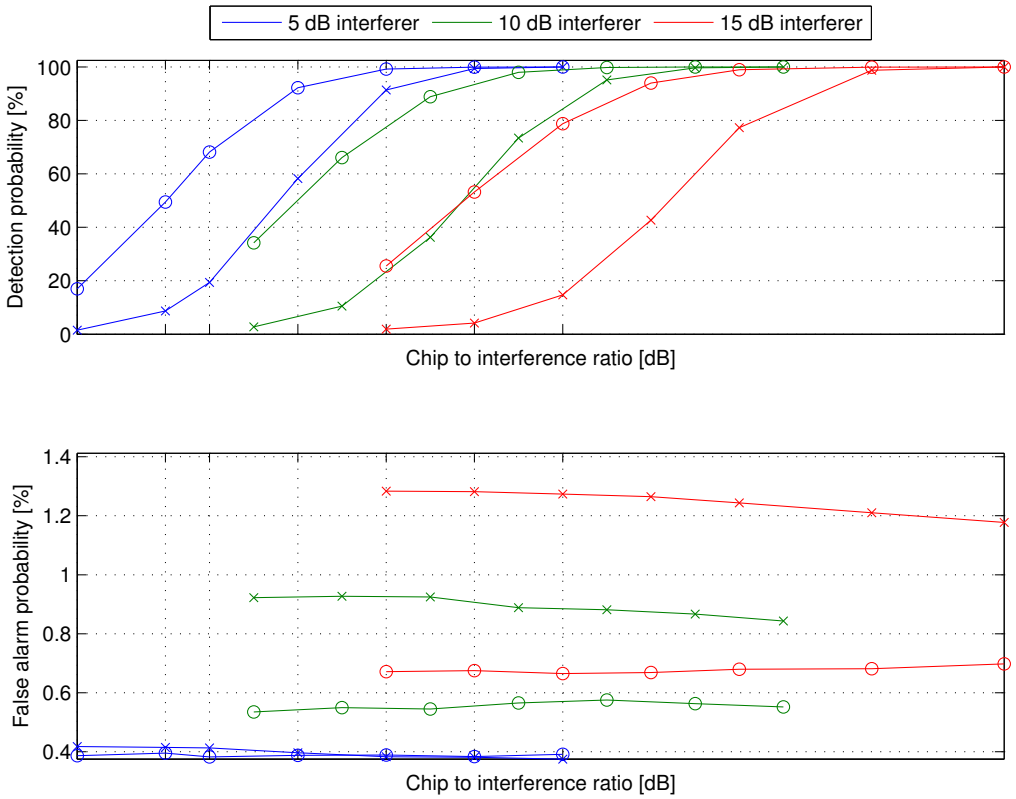


Figure 7.30: Case 6: FFT-based algorithm with temporal whitening and spatial whitening (lines with circles) versus the baseline algorithm (lines with crosses)

Interferer[dB]	Baseline[%]	Temporal and Spatial[%]	Difference[%]	Ratio
5	38.18	68.16	29.98	1.79
10	18.54	61.64	43.1	3.32
15	22.79	78.9	56.11	3.46

Table 7.26: Case 6: Maximum difference in detection probability between the algorithms for different interference power levels.

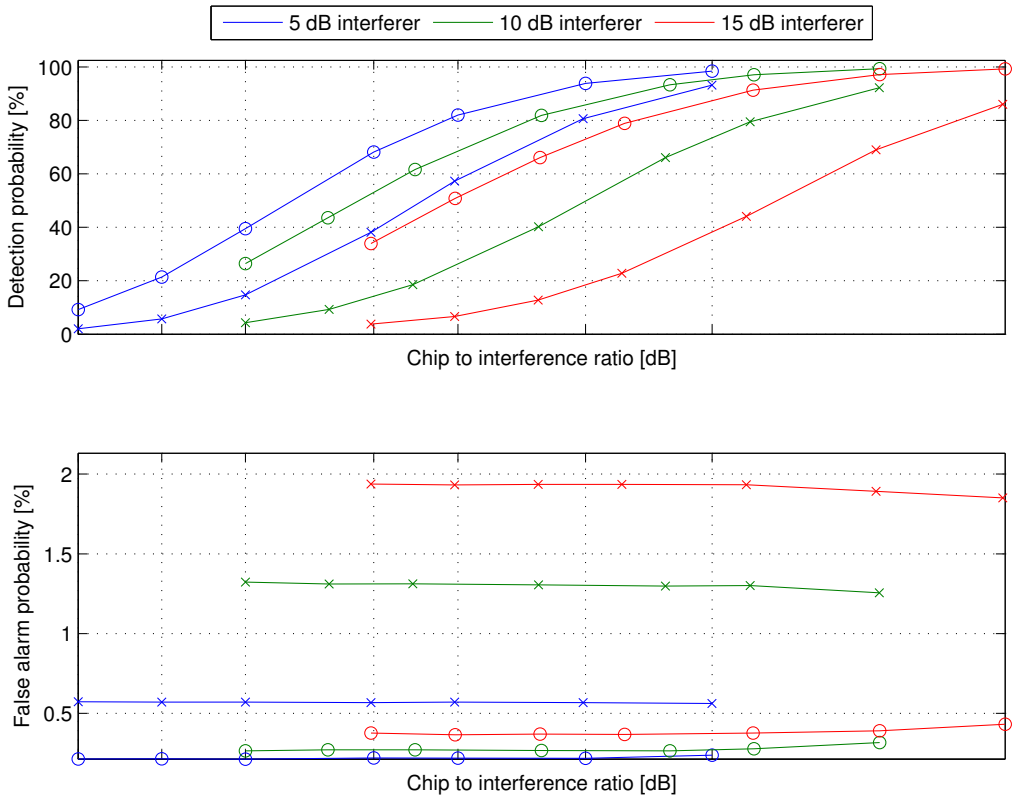


Figure 7.31: Case 7: FFT-based algorithm with temporal whitening and spatial whitening (lines with circles) versus the baseline algorithm (lines with crosses)

Interferer[dB]	Baseline[%]	Temporal and Spatial[%]	Difference[%]	Ratio
5	28.2	55.37	27.17	1.96
10	31.65	70.56	38.91	2.23
15	28.33	75.73	47.4	2.67

Table 7.27: Case 7: Maximum difference in detection probability between the algorithms for different interference power levels.

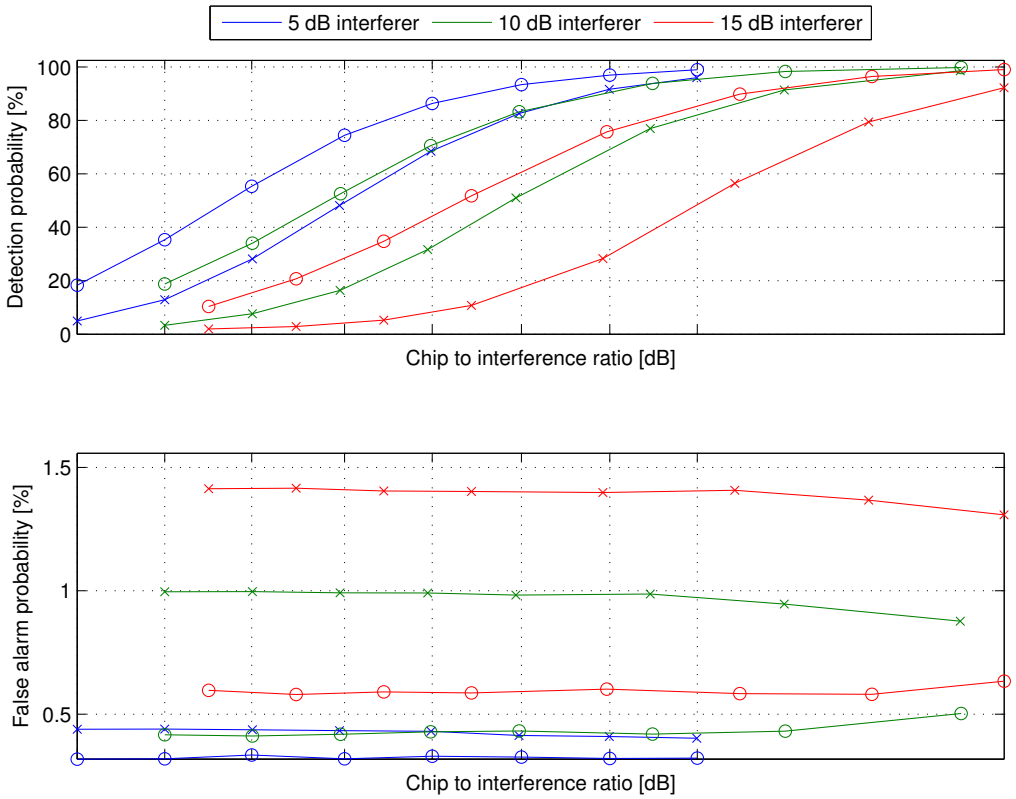


Figure 7.32: Case 8: FFT-based algorithm with temporal whitening and spatial whitening (lines with circles) versus the baseline algorithm (lines with crosses)

Interferer[dB]	Baseline[%]	Temporal and Spatial[%]	Difference[%]	Ratio
5	44.44	68.13	23.69	1.53
10	25.58	57.68	32.11	2.26
15	20.49	59.38	38.89	2.9

Table 7.28: Case 8: Maximum difference in detection probability between the algorithms for different interference power levels.

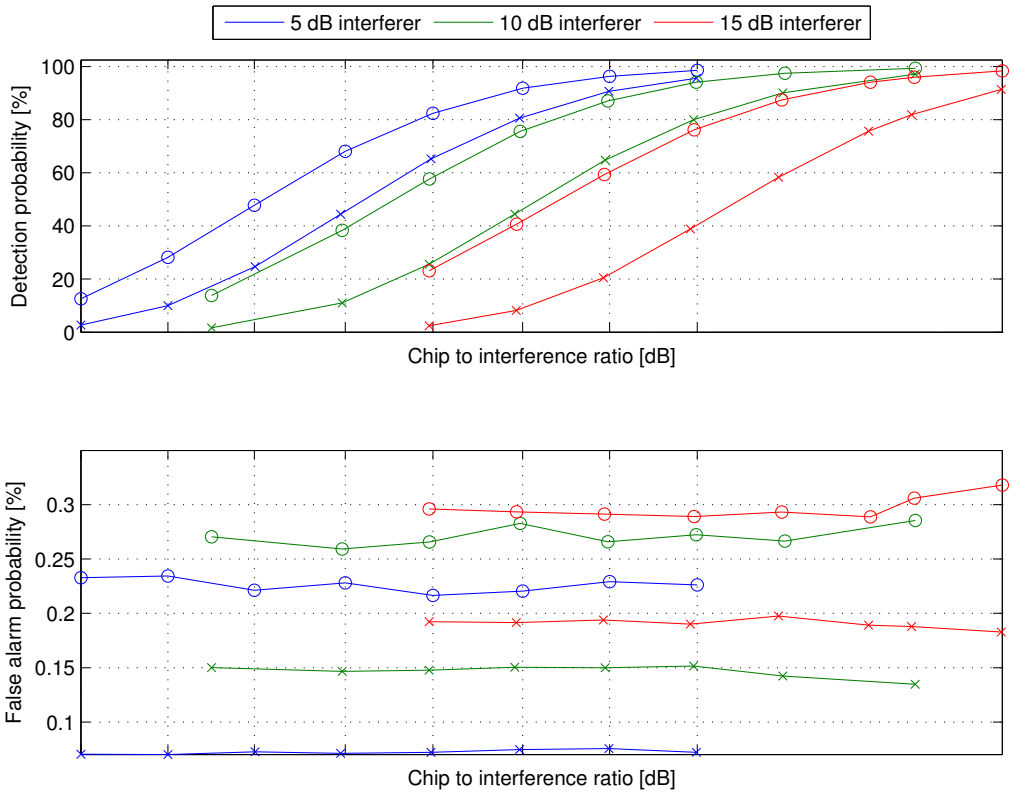


Figure 7.33: Case 9: FFT-based algorithm with temporal whitening and spatial whitening (lines with circles) versus the baseline algorithm (lines with crosses)

Interferer[dB]	Baseline[%]	Temporal and Spatial[%]	Difference[%]	Ratio
5	44.44	68.13	23.69	1.53
10	25.58	57.68	32.11	2.26
15	20.49	59.38	38.89	2.9

Table 7.29: Case 9: Maximum difference in detection probability between the algorithms for different interference power levels.

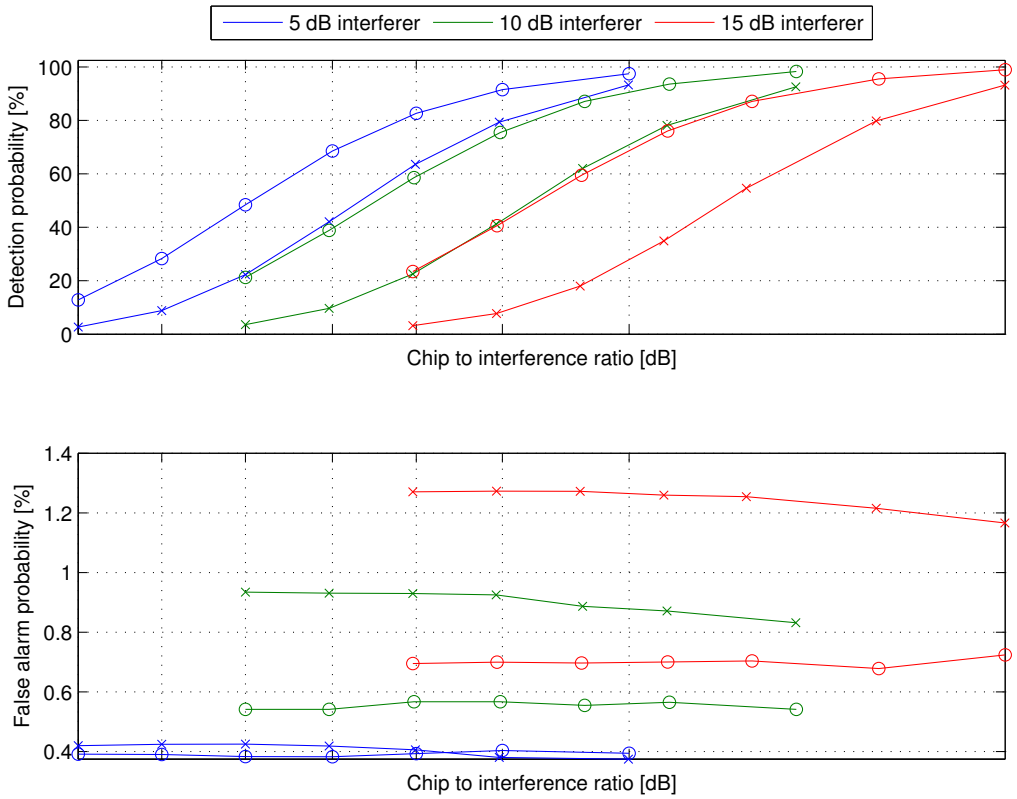


Figure 7.34: Case 10: FFT-based algorithm with temporal whitening and spatial whitening (lines with circles) versus the baseline algorithm (lines with crosses)

Interferer[dB]	Baseline[%]	Temporal and Spatial[%]	Difference[%]	Ratio
5	42.1	68.56	26.46	1.63
10	22.58	58.59	36.01	2.59
15	18.05	59.52	41.48	3.3

Table 7.30: Case 10: Maximum difference in detection probability between the algorithms for different interference power levels.

7.8.1 Discussion

The FFT-based algorithm with spatial and temporal whitening demonstrated the best performance out of all the previous algorithms. It presented better performance than the temporal whitening algorithm when the interferer was transmitting in TU and VA channels, which was the cases where it fared the best. This can be seen when comparing Fig. 7.29 and 7.30 with Fig. 7.9 and 7.10. However, it also presented better performance than the spatial whitening algorithm when the interferer was transmitting in RA and PA channels. This can be seen when comparing Fig. 7.27 and 7.28 with Fig. 7.19 and 7.20. Thus the spatial and temporal whitening algorithms can be used together, granting an overall improvement in their relative areas of performance gain.

7.9 All implemented algorithms versus the baseline algorithm

The following results show the performance of all the different algorithms, i.e., temporal whitening, spatial whitening and both temporal and spatial whitening, compared to the performance of the time-domain baseline algorithm. The algorithms are evaluated for a UE transmission over two different channels, without an interferer for the sake of evaluating performance without interference.

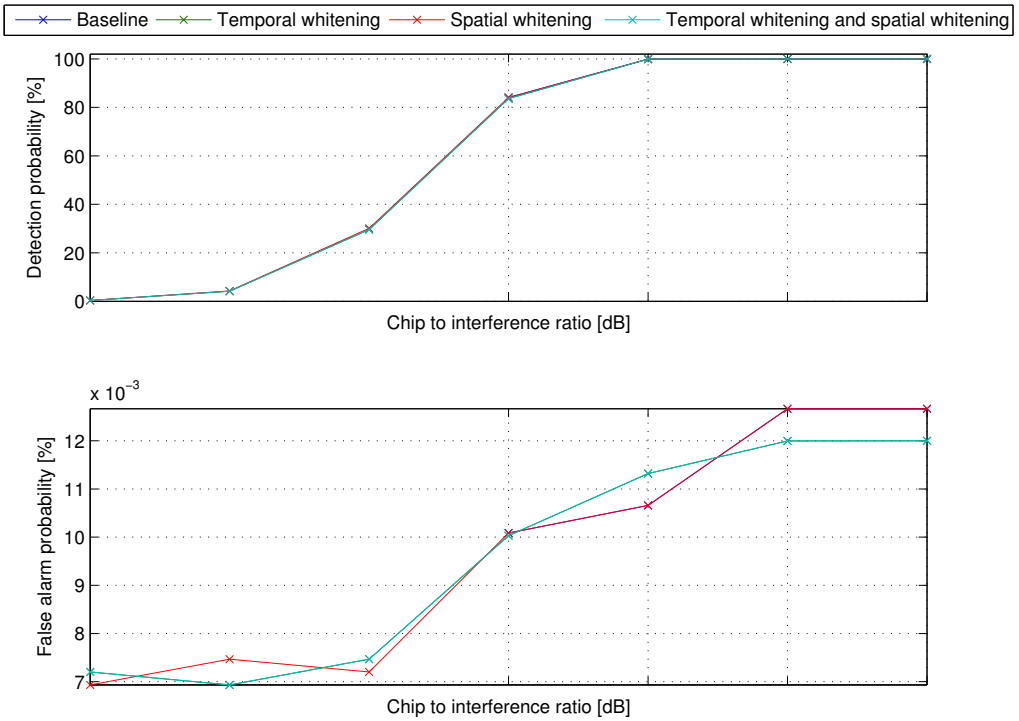


Figure 7.35: Case 1: All algorithms versus the baseline algorithm

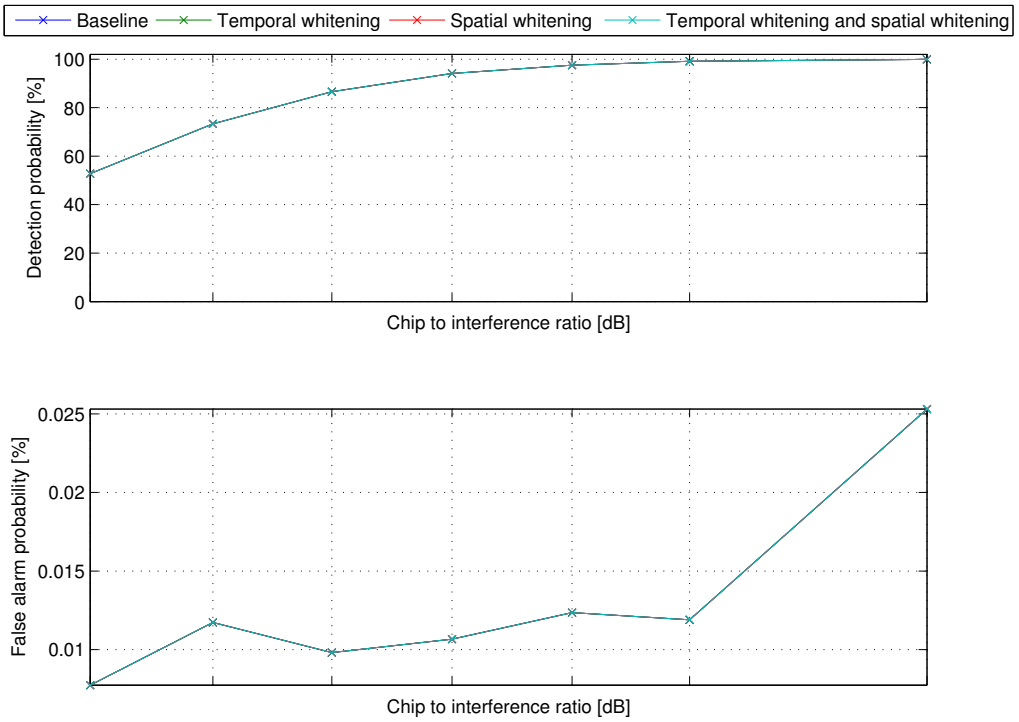


Figure 7.36: Case 11: All algorithms versus the baseline algorithm.

7.9.1 Discussion

The performance did neither degrade nor improve in the presence of no interferer, which further highlights the robustness of the operation of the whitening algorithms.

8

Conclusion

The purpose of this thesis was to evaluate and design an FFT-based RACH preamble detector in a WCDMA system, with the objective of increasing the detection probability in the presence of strong interference and reducing the complexity of the preamble detection algorithms.

Using receiver-side signal processing such as temporal and spatial whitening greatly improves the access procedure in a transmission over a channel with interference induced by an interfering high data rate user. The preamble detection algorithm with temporal whitening granted a maximum gain in detection probability of approximately 47 % compared to the baseline algorithm in the case of strong interference. Furthermore, the preamble detection algorithm with spatial whitening demonstrated a maximum gain in detection probability of approximately 62 %. Finally, using both temporal and spatial whitening collectively granted a maximum gain in detection probability of approximately 69 %.

The performance gain presented yields a higher probability of preamble detection for a specific transmission power. Thus, transmissions can be made with lower power and still yield the same detection probability as the baseline algorithm. Consequently, the coverage area can be increased for transmissions in the presence of strong interference. However, the increase in performance is, as mentioned, relative to the baseline algorithm, thus the performance is still degraded overall in the presence of strong interference.

Furthermore, the complexity of the presented FFT-based algorithm is lower than the time domain equivalent, which could lead to less computations for the preamble detector. This is due to a convolution in the frequency domain is equivalent to element wise multiplication, which is a less computationally complex [26]. Hence, the implementation of the FFT-based preamble detector could lead to lower latency for the entire system.

8.1 Future work

The preamble detector algorithms designed in this thesis may serve as a basis for evaluating the possibility of using an FFT-based RACH preamble detector with interference suppression in WCDMA. The complexity gain of using the FFT-based solution instead of the time domain may also be further investigated to determine exactly how many less clock cycles the FFT-based solution requires. Furthermore, the parameters governing the operation of temporal whitening algorithm may be examined in greater detail.

The spatial whitening algorithm is based upon a suboptimal solution and is therefore subject to improvements. Thus, removing the Cholesky decomposition and evaluating an alternate solution may improve the spatial interference suppression significantly. Furthermore, an FFT-based solution of the spatial whitening algorithm has not been investigated.

Moreover, combining the temporal and spatial whitening may be done in a different way where the covariance matrices can be combined, thus whitening spatio-temporally. This has already been tried in other types of implementations [22] [13], however, a FFT-based spatio-temporal whitening solution is yet to be made to our knowledge. Finally, the order of the whitening operations have not been fully investigated, although our initial testing presented better results when spatially whitening before the temporal whitening. Consequently, changing the order to temporally whiten first and then spatially whitening may yield some interesting results.

Bibliography

[1] Universal Mobile Telecommunications System (UMTS). "*Physical layer procedures (FDD)*", TS 25.214 version 12.2, Available at `http://www.etsi.org/deliver/etsi_ts/125200_125299/125214/12.02.00_60/ts_125214v120200p.pdf` , [May 2015].

[2] Worldwide Quarterly Mobile Phone Tracker, International Data Corporation. `http://www.idc.com/getdoc.jsp?containerId=prUS25407215`, [January 2015].

[3] Holma, H; Toskala, A. "*WCDMA for UMTS : HSPA Evolution and LTE*" 5. ed., Wiley 2010. pp.61-62

[4] Türke, U. "*Efficient Methods for WCDMA Radio Network Planning and Optimization*", Vieweg+Teubner 2007. pp 12-13.

[5] Sanchez, J; Thioune, M. "*Spread Spectrum and WCDMA*", ISTE, 2007. pp 3-5.

[6] Holma, H; Toskala, A. "*WCDMA for UMTS: Radio Access For Third Generation Mobile Communications*" 2. ed., Wiley 2002. pp.33-36.

[7] Holma, H; Toskala, A. "*WCDMA for UMTS: Radio Access For Third Generation Mobile Communications*" 2. ed., Wiley 2002. pp.89-99

[8] Agaian, S; Sarukhanyan, H; Egiazarian, K; Astola, J. "*Hadamard transforms*" Volume PM207, SPIE 2011. pp. 2-3.

[9] 3rd Generation Partnership Project (UMTS). "*Spreading and modulation (FDD)*", TS 25.213 version 12, Available at `http://www.etsi.org/deliver/etsi_ts/125200_125299/125213/12.00.00_60/` , [February 2015].

[10] Sanchez, J; Thioune, M. "*Spread Spectrum and WCDMA*", ISTE, 2007. pp 90-105.

[11] Sheikh, Asrar U. H. "*Wireless Communications: Theory and Techniques*", Springer Science + Business Media New York 2004, pp.670-675.

[12] Holma, H; Toskala, A. "*WCDMA for UMTS: Radio Access For Third Generation Mobile Communications*" 2. ed., Wiley 2002. pp.105-122

[13] Sahlin, Sihlbom, "*Spatial and temporal pre-equalization*", WO Patent 2011066851 A1, Jun 9, 2011.

[14] McDonough, R; Whalen, A. , "*Detection of signals in noise*" Academic Press, Inc 1995, pp. 340-341, 227-233.

[15] Goldsmith, A. "*Wireless Communications*" 1. ed, Cambridge University Press 2005, pp.46.

[16] Hlawatsch, F; Matz, G. "*Wireless Communications Over Rapidly Time-Varying Channels*", Elsevier Ltd 2011. pp.15-25.

[17] Goldsmith, A. "*Wireless Communications*" 1. ed, Cambridge University Press, 2005, pp. 86-92.

[18] Simon, M; Alouini, M. "*Digital Communication over Fading Channels*" 2. ed. Wiley-IEEE Press 2005. pp.20-23.

[19] Goldsmith, A. "*Wireless Communications*" 1. ed, Cambridge University Press 2005, pp.30-31.

[20] Kay, S.M. "*Fundamentals of Statistical Signal Processing, Volume II: Detection Theory*" Prentice-Hall PTR 1998, pp. 33-36.

[21] Vaidyanathan P.P. "*The Theory of Linear Prediction*", Morgan & Claypool Publishers, 2008, pp. 10-18.

[22] Humble, T.S; Mitra, P; Barhen, J; Schleck, B; Polcari, J; Traweek, M. "*Spatio-Temporal signal Twice-Whitening Algorithms on the hx3100 Ultra-Low Power Multicore Processor*" OCEANS 2010 IEEE - Sydney, pp.1-6.

[23] Sourceforge "*IT++ Documentation*", Available from: `http://itpp.sourceforge.net/4.3.1/`, [May 2015].

[24] International Telecommunication Union "*Guidelines for Evaluation of Radio Transmission Technologies for IMT-2000*", Rec. ITU-R M.1225. Available from `https://www.itu.int/dms_pubrec/itu-r/rec/m/R-REC-M.1225-0-199702-I!!PDF-E.pdf`, [May 2015].

[25] 3rd Generation Partnership Project (UMTS) "*Universal Mobile Telecommunications System (UMTS); Deployment aspects*", TR 25.943 version 9 Release 9, 2010-02, Available from `http://www.etsi.org/deliver/etsi_tr/125900_125999/125943/09.00.00_60/tr_125943v090000p.pdf`, [May 2015].

[26] Utha state university. "*Lecture 9: Convolution Using the DFT* ", Available at `http://ocw.usu.edu/Electrical_and_Computer_Engineering/Signals_and_Systems/9_5node5.html` , [May 2015].

[27] Zakharov, Y.V; Adlard, J and Tozer, T.C. "*The 11th IEEE International Symposium on Personal, Indoor and Mobile Radio Communication* Proceedings Volume 2, PIMRC 2000, pp. 82-86.

A

Appendix

A.1 Main

```
%{
/*-----------------------------------------------------------------
 *  Author:         Andreas Bring and Kim Rosberg
 *  Written:        2015-04-15
 *  Last updated:   2015-05-24
 *
 *
 *  FFT-based RACH preamble detection algorithm,
 *  including non-coherent MRC, spatial decorrelation and prewhitening
 *
 *-----------------------------------------------------------------
%}
clear all;clear all;clc;
%Load data from BCL simulation
addpath('/workspace/git/ekirosb/bcl/src/buildsupport/sinks/')
%Choose file format
list=dir('*.ascii');
geotag='preamble_detector_alg5';
tag='_ra_';
pathname='/workspace/git/ekirosb/bcl/src/buildsupport/sinks/';
fetch_data( geotag,tag,pathname)
load('preambledata')

%Number of frames
clc;close all
n_frame=1:22;
%Result vectors for false and true detections
true_detections_down=zeros(1,1:length(n_frame));
true_detections_prew=zeros(1,1:length(n_frame));
false_detections_down=zeros(1,1:length(n_frame));
false_detections_prew=zeros(1,1:length(n_frame));
counter=0;
%The correct sent signature
rightsignature=9;
%Number of samples to pick out from autocorrelation
n_acf_lag=16;
n_antenna=2;
%Choose coherent mode
COH=4;
```

```
%Number of signature
n_sign=16;
%Number of prewhitning parts
n_p_parts=4;
%Number of parts to split downsampled (received signal) per access sliot
%and antenna
l_rs=length(downsampled_0);
%Setup function
th_db=22.8-1;
l_psw=512;
extra_fft=0;

[l_fft,n_parts,H_scmf,RRCF,adj_factor,th_l,l_fft_pp,l_one_part] = ...
    setup(extra_fft,l_rs,n_p_parts,COH,scr_code,n_sign,th_db,l_psw);

for frame=n_frame
    sigma2=0;
    PPDP=zeros(n_sign,l_psw,n_antenna);
    PAPDP=zeros(n_sign,l_psw);
    for antenna=1:n_antenna
        if antenna==1 && n_antenna>1
            downsampled=[downsampled_0(frame,:);downsampled_1(frame,:)];
            downsampled = SIS(downsampled);
        elseif n_antenna==1
            downsampled=downsampled_0(frame,:);
        end
        [downsampled(antenna,:)] = prewhitening(downsampled(antenna,:), ...
            l_rs,n_p_parts,RRCF,n_acf_lag,l_fft_pp);
        %Signature and code matched filter with non-coherent MRC
        [PPDP] = scmf (l_psw,l_fft,downsampled(antenna,:),n_sign, ...
            n_parts,antenna,H_scmf,PPDP,l_one_part);
        [sigma2] = interference_estimate(downsampled(antenna,:), ...
            l_fft/2,adj_factor,sigma2);
    end
    threshold=sigma2*th_l;
    %Antenna combining
    [PAPDP] =antenna_combining(PPDP,n_sign,l_psw);
    if n_antenna==1
        ppdp=[ppdp_0{frame,:}];
    else
        ppdp=([ppdp_0{frame,:}]+[ppdp_1{frame,:}]);
    end
    %Results-------------------------------------------------------------
    detectionsdown=[];detectionspre=[];detectionstsis=[];
    ppdptmp=reshape(ppdp,[n_sign,l_psw]);
    counter=counter+1;
    for i=1:n_sign
        if i==rightsignature
            true_detections_down(counter)=sign(sum(ppdptmp(i,:)>= ...
                preamble_detector_alg3_ko_threshold(frame)));
            true_detections_prew(counter)=sign(sum(PAPDP(i,:)>=threshold));
        else
            detectionsdown=[detectionsdown,sign(sum(ppdptmp(i,:)>= ...
                preamble_detector_alg3_ko_threshold(frame)))];
            detectionspre=[detectionspre,sign(sum(PAPDP(i,:)>=threshold))];
        end
```

```
    end
    false_detections_down(counter)=sum(detectionsdown);
    false_detections_prew(counter)=sum(detectionspre);
    %Print results
    total_true_detection_BCL=sum(true_detections_down)
    total_true_detection_fftpreamble=sum(true_detections_prew)
    disp('-------------------------------------')
    total_false_detection_BCL=sum(false_detections_down)
    total_false_detection_fftpreamble=sum(false_detections_prew)
    disp('-------------------------------------------------------------------')
end
```

A.2 Setup

```
%{
/*-----------------------------------------------------------------
 *  Author:         Andreas Bring and Kim Rosberg
 *  Written:        2015-04-15
 *  Last updated:   2015-05-24
 *
 *
 *  Setup function to calculate static parameters for temporal whitening,
 *  interference estimator, signature and code matched filter.
 *
 *-----------------------------------------------------------------
%}
function [l_fft,n_parts,H_scmf,RRCF,adj_factor,th_l,...
    l_fft_pp,l_one_part] = setup(extra_fft,l_rs,n_p_parts,COH,scr_code, ...
    n_sign,th_db,l_psw)
%Root raised cosine with BETA 0.18, from it++ def
 RRCT=[-0.000232049 0.000278791 9.52054e-05 -0.0003361 9.22393e-05 ...
    0.000288235 -0.000275694 -0.00013596 0.000394084 -8.55033e-05 ...
    -0.000398454 0.000312746 0.000269414 -0.000470365 -2.78736e-05 ...
    0.000493469 -0.000265176 -0.000350475 0.00052141 5.91759e-05 ...
    -0.00064849 0.00030997 0.000578606 -0.000646533 -0.000294541 ...
    0.000827749 -0.00015487 -0.00075555 0.000656286 0.000393715 ...
    -0.001053 0.000205742 0.00118332 -0.000896834 -0.000928694 ...
    0.00146059 0.000260629 -0.00165297 0.00072624 0.00127324 ...
    -0.00180552 -0.000240546 0.00264104 -0.00133676 -0.00283904 ...
    0.00311389 0.00202266 -0.0044941 8.48386e-05 0.0046438 -0.00358568 ...
    -0.00252068 0.0083588 -0.00313274 -0.0140498 0.0139465 0.0201052 ...
    -0.0325711 -0.0258473 0.0651686 0.0305778 -0.134189 -0.0336914 ...
    0.444705 0.741884 0.444705 -0.0336914 -0.134189 0.0305778 ...
    0.0651686 -0.0258473 -0.0325711 0.0201052 0.0139465 -0.0140498 ...
    -0.00313274 0.0083588 -0.00252068 -0.00358568 0.0046438 8.48386e-05 ...
    -0.0044941 0.00202266 0.00311389 -0.00283904 -0.00133676 0.00264104 ...
    -0.000240546 -0.00180552 0.00127324 0.00072624 -0.00165297 ...
    0.000260629 0.00146059 -0.000928694 -0.000896834 0.00118332  ...
    0.000205742 -0.001053 0.000393715 0.000656286 -0.00075555 ...
    -0.00015487 0.000827749 -0.000294541 -0.000646533 0.000578606 ...
    0.00030997 -0.00064849 5.91759e-05 0.00052141 -0.000350475 ...
    -0.000265176 0.000493469 -2.78736e-05 -0.000470365 0.000269414 ...
    0.000312746 -0.000398454 -8.55033e-05 0.000394084 -0.00013596 ...
    -0.000275694 0.000288235 9.22393e-05 -0.0003361 9.52054e-05 ...
    0.000278791 -0.000232049];

%Root raised cosine fft, length dependet on number of parts
l_fft_pp=2^(extra_fft+nextpow2(l_rs/n_p_parts));
RRCF=fft(RRCT,l_fft_pp);
%Number of parts coherently accumulated
n_parts=2^(4-COH);
l_fft=2^(10+COH+extra_fft);

%Interference estimator
th_l=10^(th_db/20);
```

```
adj_factor=(2/l_fft)*((l_rs-l_psw)/l_rs)*(1/n_parts);
%Define matched filter coefficents
l_filter=length(scr_code(1,:));
had=hadamard(n_sign);
hrep=repmat(had,[1,(l_filter/n_sign)]);
h_scmf=zeros(n_sign,l_filter*2);
l_one_part=(2*l_filter)/n_parts;


% Define in frequency domain
H_scmf=zeros(n_sign,l_fft,n_parts);
% tmp=zeros(1,l_fft);
for l=1:n_sign
    h_scmf(l,1:2:end)=scr_code(1,:).*hrep(l,:);
    for k=1:n_parts
        tmp=fft(fliplr(h_scmf(l,1+(l_one_part*(k-1)):l_one_part*k)),l_fft);
        H_scmf(l,:,k)=tmp;
    end
end
end
```

A.3 Spatial interference suppression

```
%{
/*-------------------------------------------------------------------
 *  Author:         Andreas Bring and Kim Rosberg
 *  Written:        2015-04-01
 *  Last updated:   2015-05-24
 *
 *
 *  Spatially whitens incoming matrix with the help off Cholesky
 *  decomposition and outputs data as a matrix.
 *
 *-------------------------------------------------------------------
%}
function [sis_data] = SIS(inputmatrix)
L=chol(inputmatrix*inputmatrix','lower');
sis_data=L\inputmatrix;
end
```

A.4 Temporal interference suppression

```
%{
/*-----------------------------------------------------------------
 *  Author:         Andreas Bring and Kim Rosberg
 *  Written:        2015-04-15
 *  Last updated:   2015-05-24
 *
 *
 *  Temporally whitening the incoming data
 *
 *-----------------------------------------------------------------
%}
function [prewhitened_data] = prewhitening(sis_data,l_rs,n_p_parts,...
    RRCF,n_acf_lag,l_fft_pp)
part_l=l_rs/n_p_parts;
prewhitened_data=[];
sig=zeros(1,part_l);
for part=1:n_p_parts
    %Pick out a number of samples dependet on number of
    %prewhitening
    sig=sis_data(1+(part-1)*part_l:part*part_l);
    %Filter-----------------------------------------------------------
    %PSD
    xfft = fft(sig,l_fft_pp);
    psdx=(abs(xfft).^2);
    %Autocorrelation
    autocorrx=ifft(psdx);
    %Pick out samples dependent on nrofs
    autocorrx=[autocorrx(end-n_acf_lag+1:end),autocorrx(1:n_acf_lag+1)];
    psdh=fft(autocorrx,l_fft_pp);
    %Scaling factor for filter
    n_scale=max(abs(RRCF))*mean((psdh));
    psdh=n_scale./psdh;
    % Bandlimiting filter
    Hlimited =(sqrt(psdh)).*(RRCF);
    Hpre=abs(fft(ifft(Hlimited),l_fft_pp));

    %Filtering--------------------------------------------------------
    %Calc sigma
    sigma=sum(abs(Hpre).^2.*psdx);
    sigma=sigma/length(Hpre);
    %Normalizing filter to get variance 1 on output
    Hpre=Hpre/(sqrt(sigma));
    %Prewhitening
    prew=xfft.*(Hpre);
    %Received signal back to time domain
    tprew=ifft(prew);
    %Concatenate the different parts to one segment and removing
    %tail
    prewhitened_data=[prewhitened_data,tprew(1:length(sig))];
end
end
```

A.5 Signature and code matched filter

```
%{
/*-------------------------------------------------------------------
 *  Author:         Andreas Bring and Kim Rosberg
 *  Written:        2015-04-15
 *  Last updated:   2015-05-24
 *
 *
 *  Signature and code matched filter with non-coherent accumulation.
 *
 *-------------------------------------------------------------------
%}
function [PPDP] = scmf (l_psw,l_fft,prewhitened_data,n_sign,n_parts, ...
    antenna,H_scmf,PPDP,l_one_part)
for i=1:n_sign
    for l=1:n_parts
        SIG=fft(prewhitened_data(1*(l_one_part*(l-1))+1:(l_one_part*l) ...
            +l_psw),l_fft);
        tmp=ifft(H_scmf(i,:,l).*SIG);
        PPDP(i,:,antenna)=abs(tmp(l_one_part:l_one_part+l_psw-1)).^2+...
            PPDP(i,:,antenna);
    end
end
end
```

A.6 Antenna combining

```
%{
/*-------------------------------------------------------------------
 *  Author:         Andreas Bring and Kim Rosberg
 *  Written:        2015-04-15
 *  Last updated:   2015-05-24
 *
 *
 *  Combines output from signature and code matched filter from
 *  multiple antennas
 *
 *-------------------------------------------------------------------
%}
function [PAPDP] = antenna_combining(PPDP,n_sign,l_psw)
for i=1:n_sign
    for l=1:l_psw
        PAPDP(i,l)=sum(PPDP(i,l,:));
    end
end
end
```

A.7 Interference Estimator

```
%{
/*-----------------------------------------------------------------
 *  Author:         Andreas Bring and Kim Rosberg
 *  Written:        2015-04-15
 *  Last updated:   2015-05-24
 *
 *
 *  Interference estimator, calculated from the power spectral density.
 *
 *
 *-----------------------------------------------------------------
%}
function [sigma2] = interference_estimate(prewhitened_data,l_estimate, ...
    adj_factor,sigma2)
psdsig=(abs(fft(prewhitened_data(1:2:end),l_estimate)).^2);
sigma_antenna=sum(psdsig)*adj_factor;
sigma2=sigma_antenna+sigma2;
end
```

A.8 Fetch data

```
%{
/*-------------------------------------------------------------------
 *  Author:         Andreas Bring and Kim Rosberg
 *  Written:        2015-04-15
 *  Last updated:   2015-05-24
 *
 *
 *  Fetching .ascii files and saving the data in a .mat file
 *
 *
 *-------------------------------------------------------------------
%}
function [] = fetch_data( geotag,tag,pathname)
%Choose file format
addpath(pathname);
list=dir('*.ascii');
filen=[];
for i=1:length(list)
    filen{i}=list(i).name;
end
%Read files from algorithm two or three
filetosearhforextend=[geotag,tag]; algnum='';
all=1:length(list);
%Match with name
idx=regexp(filen,geotag);
idx=find(~cellfun(@isempty,idx));

for k=idx
    filename=list(k).name;
    fid = fopen(filename);
    filename=strrep(filename,filetosearhforextend,'');
    filename=strrep(filename,'.ascii',algnum)
    for i=1:4
        fgets(fid);
    end
    data=fscanf(fid,'%c');
    data=[filename,'=[',data,'];'];
    eval(data);
    fclose(fid);
end
save('preambledata.mat')
end
```

A.9 Parameter file

```
lstools_iop_file = 'exjobb.iop'
EbN0_dB = -10
Eb_definition = 'Ec Preamble'
nrof_antennas =[2]
carrier_frequency = 2000000000
dch_i_EcN0_dB = [20.0 20.0]
dch_i_multipath_channel_type = "Vehicular A"
dch_i_speed_km_h = 150
dch_interferer_on = 1
frequency_error = {[0]}
max_nrof_frames = 5
max_nrof_trans_preamble = 10
min_SF = 32
coh_mode = 2
multipath_channel_type = "Pedestrian A"
speed_km_h = 3
nrof_errors_exit_criteria = 10000
preamble_threshold = 25
ra_preamble_detector_method = 'alg5'
rtt_chip = 256
signature_list_bs = [8]
signature_list_ue = [8]
nrof_prewhitening_parts = 4;
prewhitening_on=1;
spatial_decorr_on=1;
extrafft=0;
nrof_acf_lag=16;
```

B

Appendix

In this appendix the results from the additional tests that were run, are shown.

B.1 FFT-based algorithm with temporal whitening and extended zero padding for FFT

The following results show the performance of the frequency-domain baseline algorithm with temporal prewhitening and extended zero padding for the FFT, compared to the performance of the temporal whitening algorithm with less zero padding.

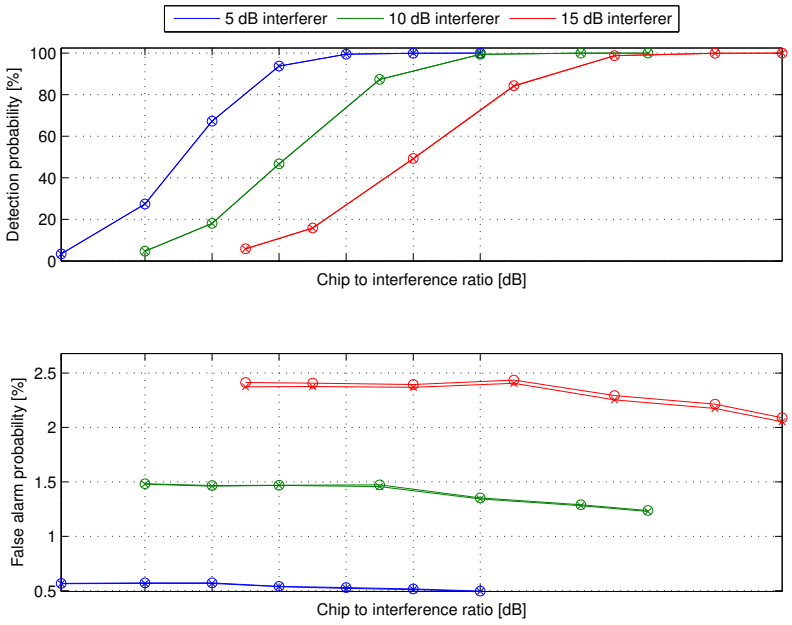


Figure B.1: Case 4: FFT-based algorithm with temporal whitening and extended zero padding (lines with circles) versus the temporal whitening algorithm with less zero padding (lines with crosses)

Interferer[dB]	Baseline[%]	Temporal whitening[%]	Difference[%]	Ratio
5	93.75	93.77	0.03	1
10	46.64	46.75	0.12	1
15	84.09	84.35	0.26	1

Table B.1: Case 4: Maximum difference in detection probability between the algorithms for different interference power levels.

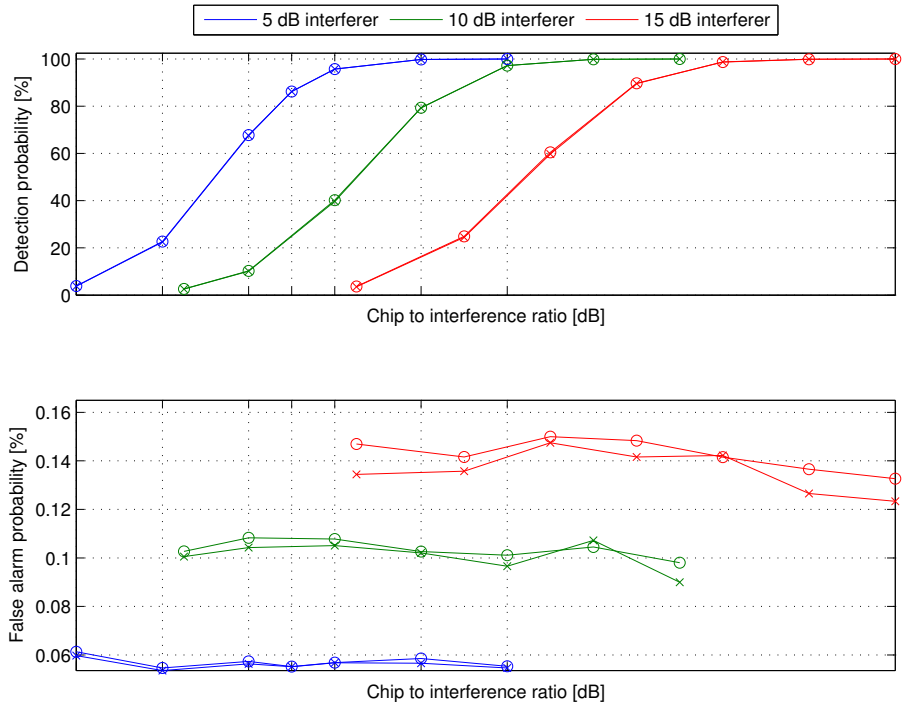


Figure B.2: Case 5: FFT-based algorithm with temporal whitening and extended zero padding (lines with circles) versus the temporal whitening algorithm with less zero padding (lines with crosses)

Interferer[dB]	Baseline[%]	Temporal whitening[%]	Difference[%]	Ratio
5	67.59	67.76	0.16	1
10	39.71	40.22	0.51	1.01
15	59.78	60.48	0.7	1.01

Table B.2: Case 5: Maximum difference in detection probability between the algorithms for different interference power levels.

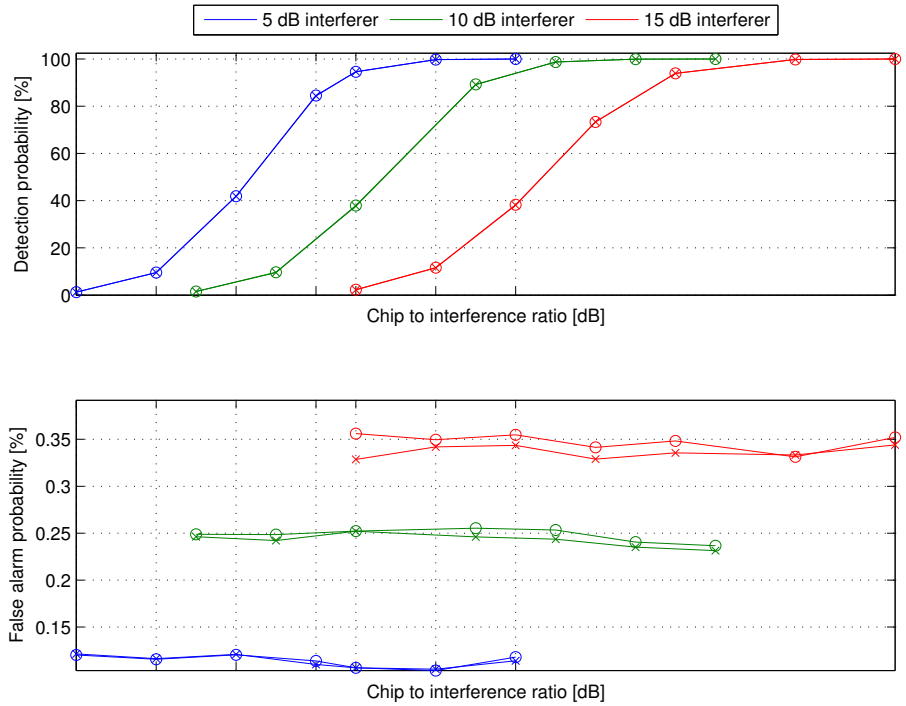


Figure B.3: Case 6: FFT-based algorithm with temporal whitening and extended zero padding (lines with circles) versus the temporal whitening algorithm with less zero padding (lines with crosses)

Interferer[dB]	Baseline[%]	Temporal whitening[%]	Difference[%]	Ratio
5	84.35	84.52	0.17	1
10	37.68	37.96	0.28	1.01
15	38.06	38.27	0.21	1.01

Table B.3: Case 6: Maximum difference in detection probability between the algorithms for different interference power levels.

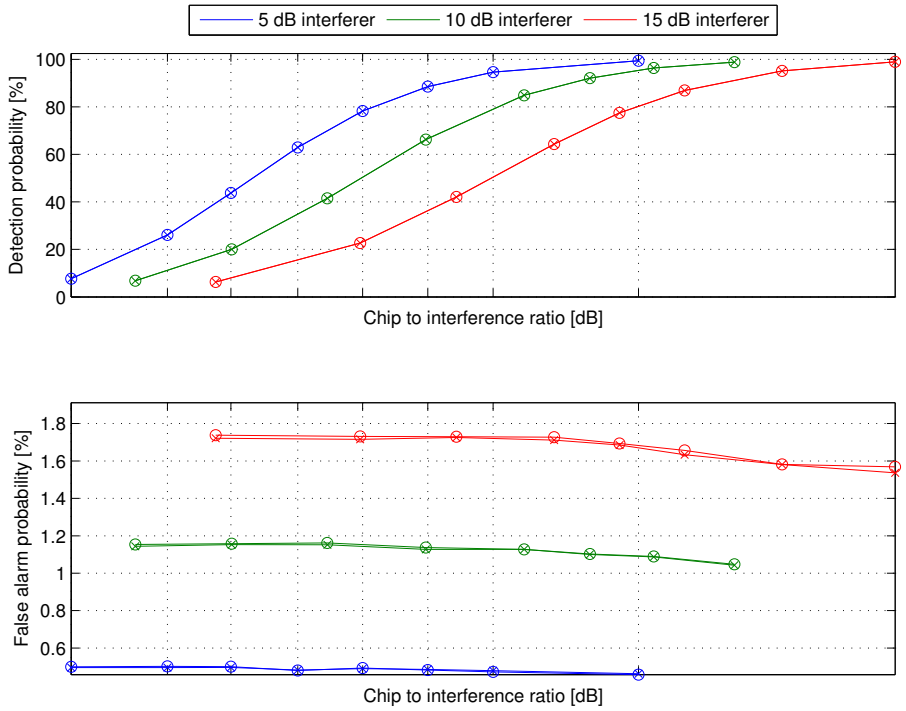


Figure B.4: Case 7: FFT-based algorithm with temporal whitening and extended zero padding (lines with circles) versus the temporal whitening algorithm with less zero padding (lines with crosses)

Interferer[dB]	Baseline[%]	Temporal whitening[%]	Difference[%]	Ratio
5	88.5	88.57	0.08	1
10	92.08	92.11	0.03	1
15	42.02	42.08	0.06	1

Table B.4: Case 7: Maximum difference in detection probability between the algorithms for different interference power levels.

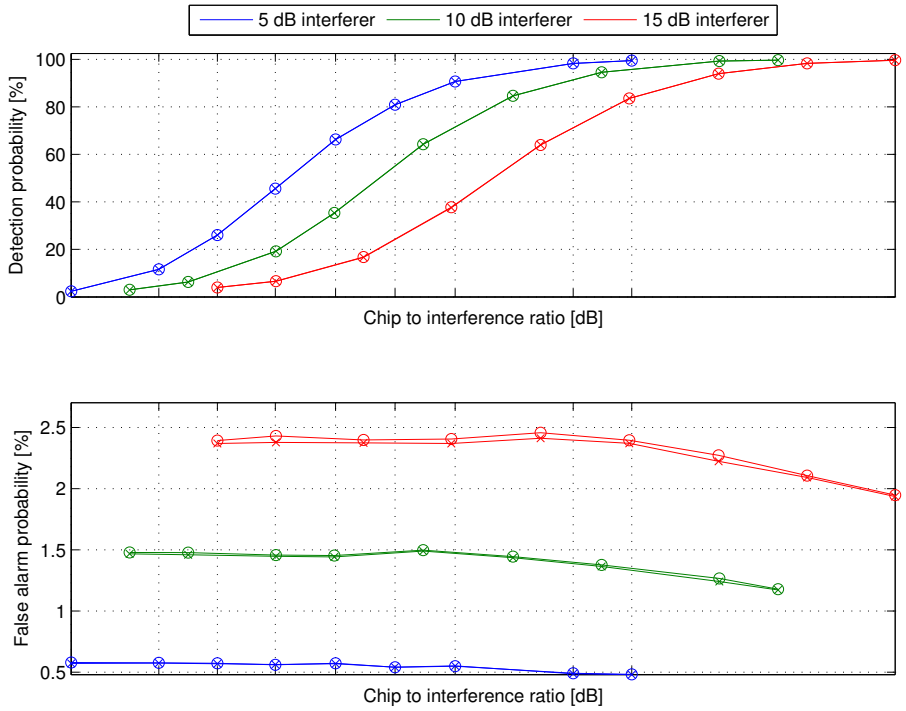


Figure B.5: Case 8: FFT-based algorithm with temporal whitening and extended zero padding (lines with circles) versus the temporal whitening algorithm with less zero padding (lines with crosses)

Interferer[dB]	Baseline[%]	Temporal whitening[%]	Difference[%]	Ratio
5	66.25	66.3	0.05	1
10	35.29	35.39	0.1	1
15	37.58	37.74	0.17	1

Table B.5: Case 8: Maximum difference in detection probability between the algorithms for different interference power levels.

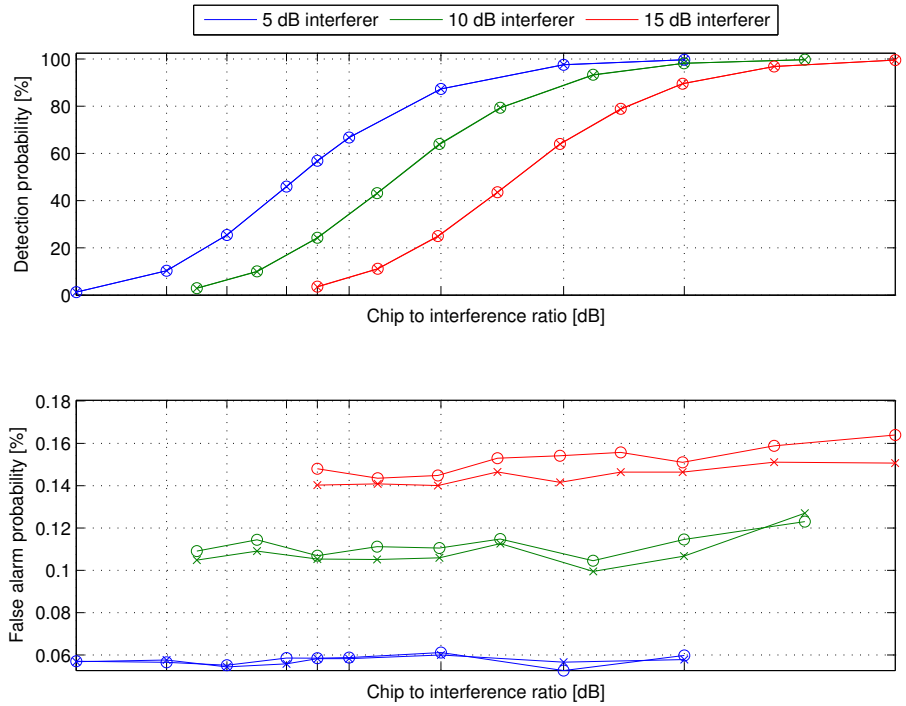


Figure B.6: Case 9: FFT-based algorithm with temporal whitening and extended zero padding (lines with circles) versus the temporal whitening algorithm with less zero padding (lines with crosses)

Interferer[dB]	Baseline[%]	Temporal whitening[%]	Difference[%]	Ratio
5	56.8	56.94	0.15	1
10	63.8	64.02	0.22	1
15	24.83	25.04	0.22	1.01

Table B.6: Case 9: Maximum difference in detection probability between the algorithms for different interference power levels.

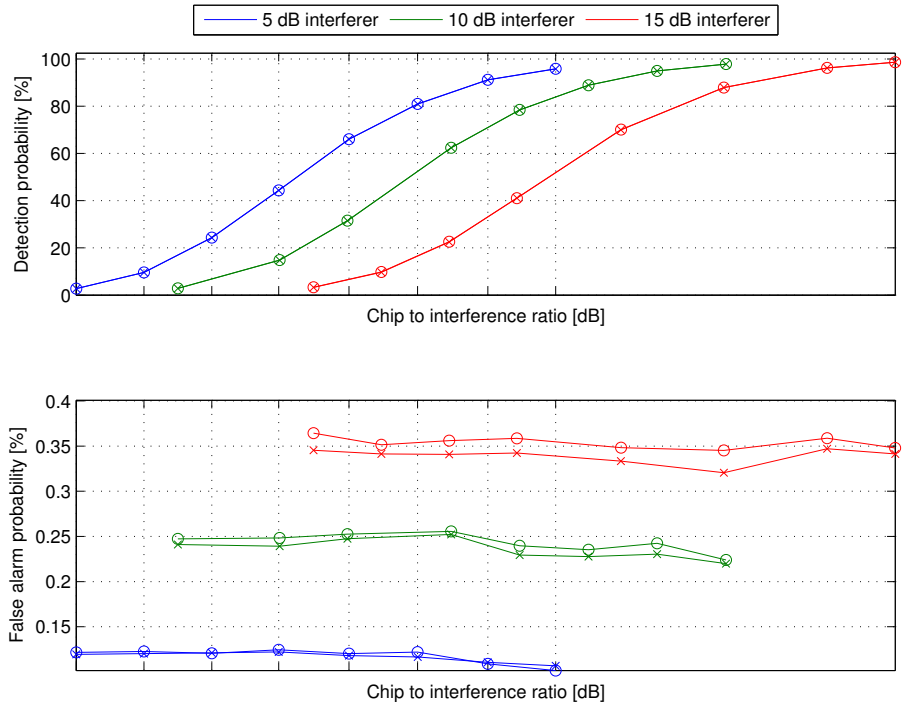


Figure B.7: Case 10: FFT-based algorithm with temporal whitening and extended zero padding (lines with circles) versus the temporal whitening algorithm with less zero padding (lines with crosses)

Interferer[dB]	Baseline[%]	Temporal whitening[%]	Difference[%]	Ratio
5	91.17	91.24	0.07	1
10	14.77	14.87	0.1	1.01
15	9.67	9.83	0.17	1.02

Table B.7: Case 10: Maximum difference in detection probability between the algorithms for different interference power levels.

B.2 FFT-based algorithm with temporal whitening with different numbers of autocorrelation function lags

The following results show the performance of the frequency-domain baseline algorithm with temporal whitening with different numbers of autocorrelation function lags.

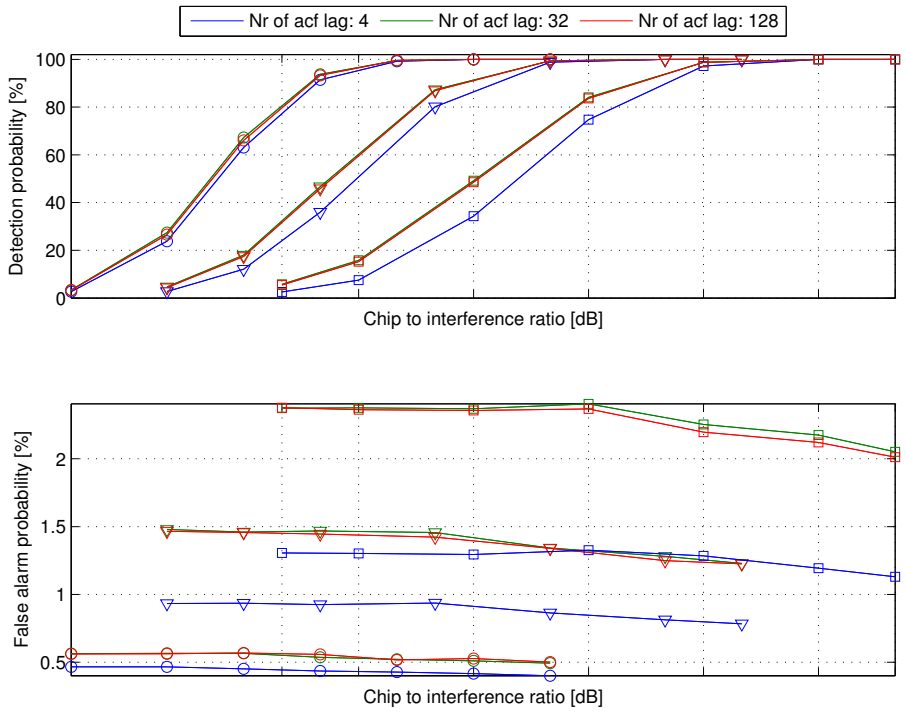


Figure B.8: Case 4: FFT-based algorithm with temporal whitening with different numbers of ACF lags (lines of different colors). Circles, triangles and squares represent 5 dB, 10 dB and 15 dB interference power respectively

Interferer[dB]	Nr of acf lag 32[%]	Nr of acf	Difference[%]	Ratio
5	99.52	128 lag at 99.56%	0.04	1
10	99.99	4 lag at 100 %	0.01	1
15	99.97	4 lag at 99.98 %	0.01	1

Table B.8: Case 4: Maximum difference in relation to the algorithm with 32 ACF lags in detection probability between the algorithms for different interference power levels.

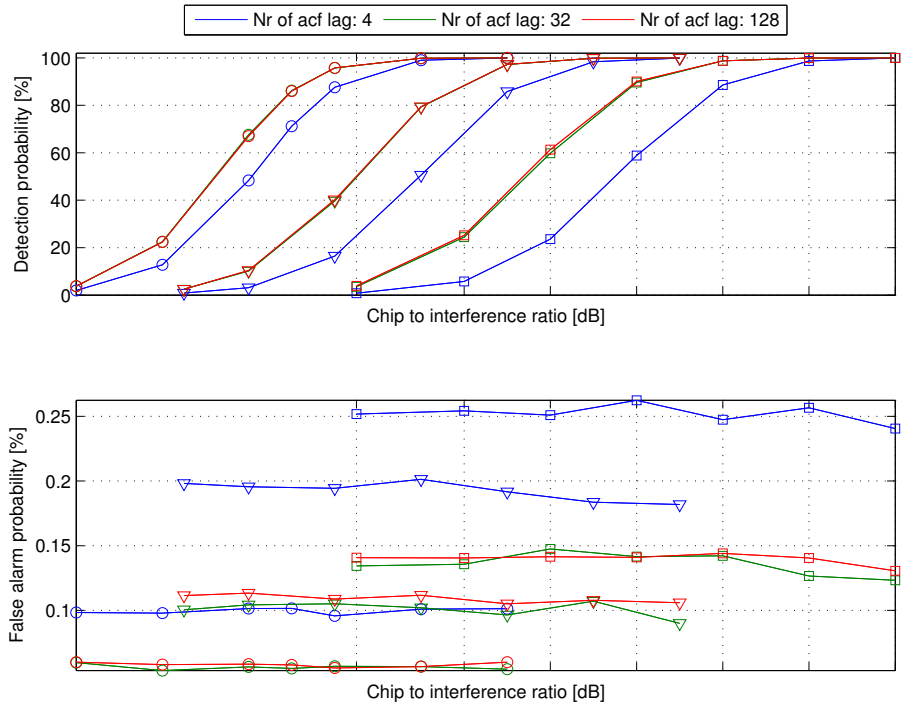


Figure B.9: Case 5: FFT-based algorithm with temporal whitening with different numbers of ACF lags (lines of different colors). Circles, triangles and squares represent 5 dB, 10 dB and 15 dB interference power respectively

Interferer[dB]	Nr of acf lag 32[%]	Nr of acf	Difference[%]	Ratio
5	95.72	128 lag at 95.77 %	0.05	1
10	39.71	128 lag at 40.18 %	0.47	1.01
15	59.78	128 lag at 61.31 %	1.54	1.03

Table B.9: Case 5: Maximum difference in relation to the algorithm with 32 ACF lags in detection probability between the algorithms for different interference power levels.

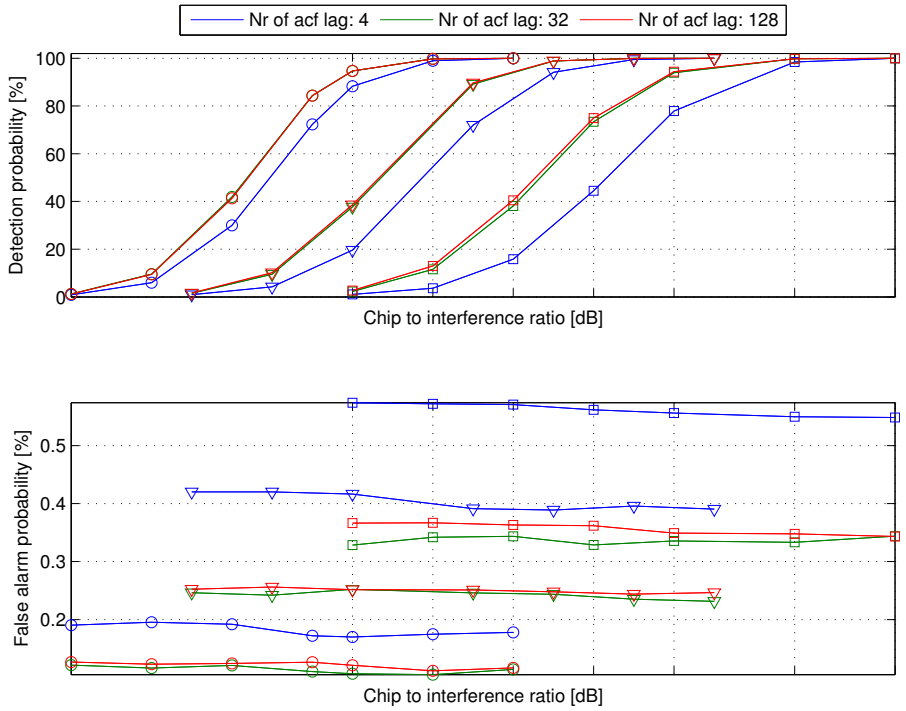


Figure B.10: Case 6: FFT-based algorithm with temporal whitening with different numbers of ACF lags (lines of different colors). Circles, triangles and squares represent 5 dB, 10 dB and 15 dB interference power respectively

Interferer[dB]	Nr of acf lag 32[%]	Nr of acf	Difference[%]	Ratio
5	94.63	128 lag at 94.68 %	0.05	1
10	37.68	128 lag at 38.6 %	0.92	1.02
15	38.06	128 lag at 40.46 %	2.4	1.06

Table B.10: Case 6: Maximum difference in relation to the algorithm with 32 ACF lags in detection probability between the algorithms for different interference power levels.

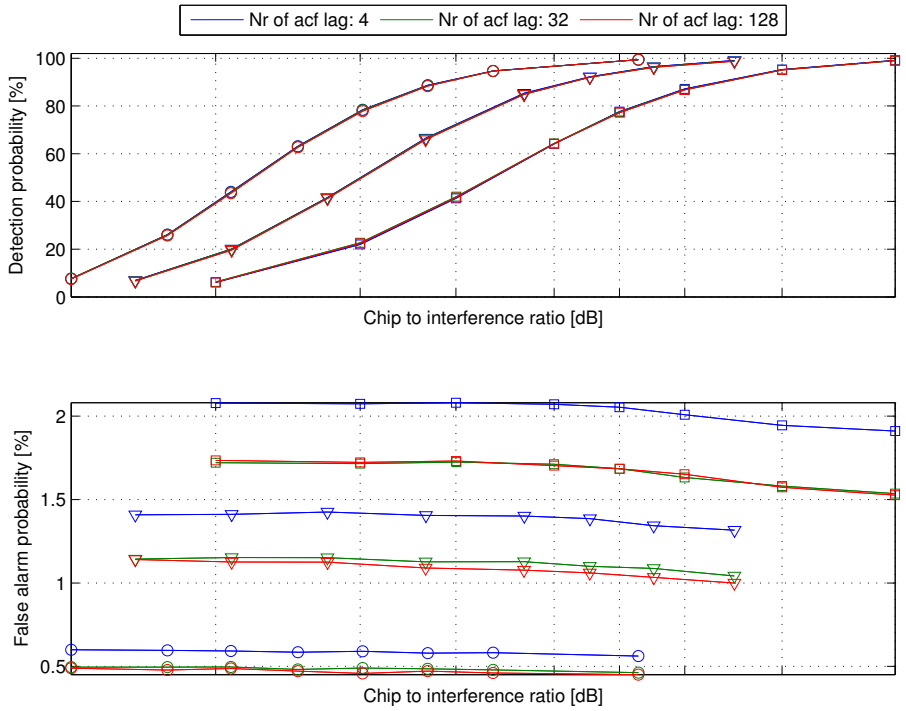


Figure B.11: Case 7: FFT-based algorithm with temporal whitening with different numbers of ACF lags (lines of different colors). Circles, triangles and squares represent 5 dB, 10 dB and 15 dB interference power respectively

Interferer[dB]	Nr of acf lag 32[%]	Nr of acf	Difference[%]	Ratio
5	43.74	4 lag at 43.98 %	0.24	1.01
10	84.94	4 lag at 85.32 %	0.38	1
15	86.87	4 lag at 87.16 %	0.3	1

Table B.11: Case 7: Maximum difference in relation to the algorithm with 32 ACF lags in detection probability between the algorithms for different interference power levels.

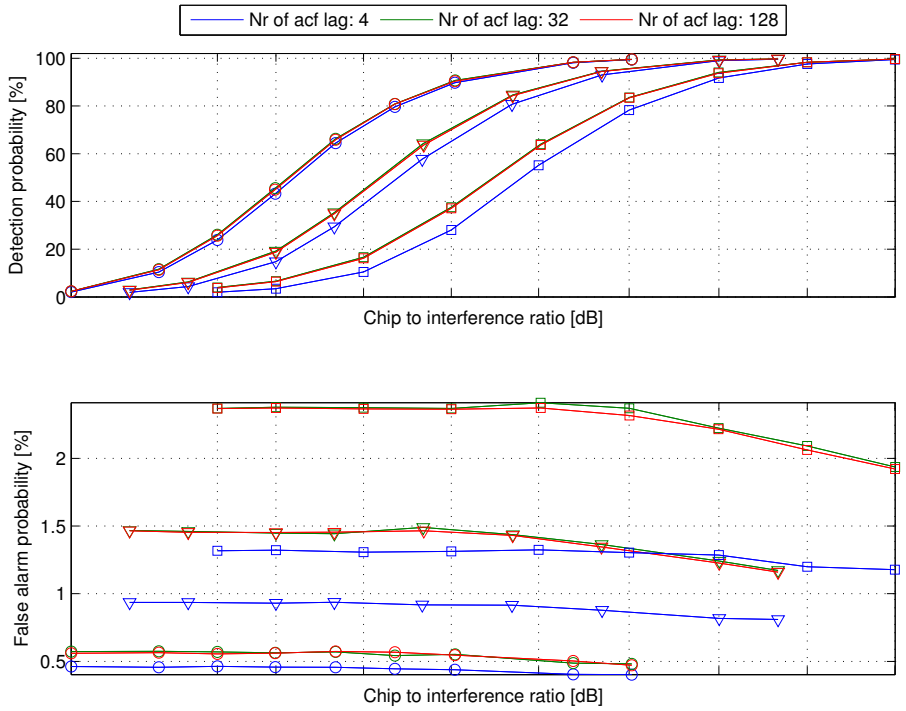


Figure B.12: Case 8: FFT-based algorithm with temporal whitening with different numbers of ACF lags (lines of different colors). Circles, triangles and squares represent 5 dB, 10 dB and 15 dB interference power respectively

Interferer[dB]	Nr of acf lag 32[%]	Nr of acf	Difference[%]	Ratio
5	2.35	128 lag at 2.38 %	0.02	1.01
10	99.69	128 lag at 99.69 %	0	1
15	98.33	128 lag at 98.3%	-0.03	1

Table B.12: Case 8: Maximum difference in relation to the algorithm with 32 ACF lags in detection probability between the algorithms for different interference power levels.

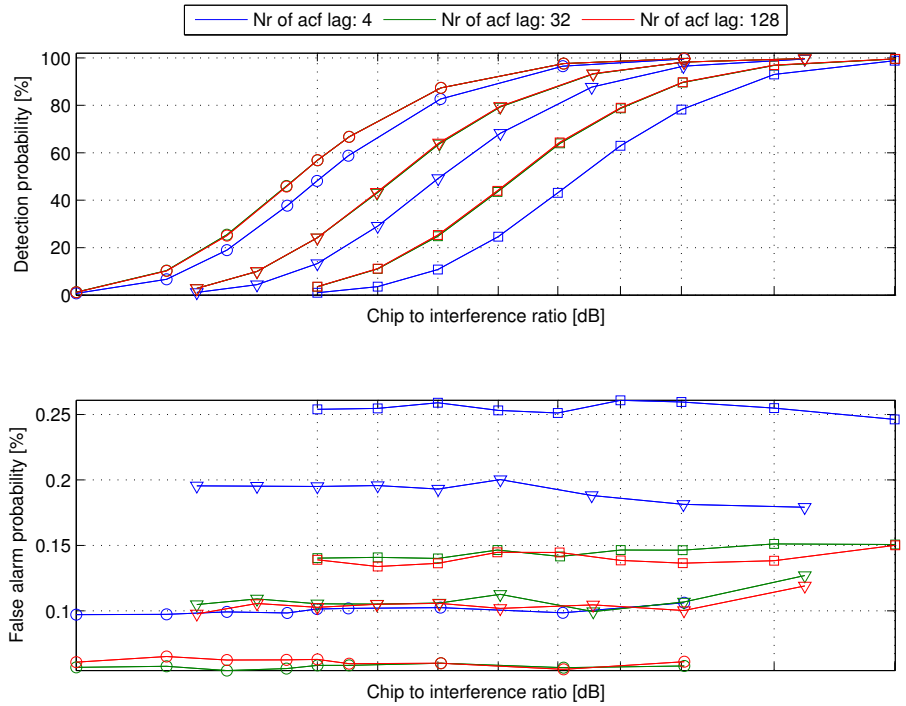


Figure B.13: Case 9: FFT-based algorithm with temporal whitening with different numbers of ACF lags (lines of different colors). Circles, triangles and squares represent 5 dB, 10 dB and 15 dB interference power respectively

Interferer[dB]	Nr of acf lag 32[%]	Nr of acf	Difference[%]	Ratio
5	56.8	128 lag at 56.86 %	0.06	1
10	63.8	128 lag at 64.17 %	0.37	1.01
15	24.83	128 lag at 25.34 %	0.51	1.02

Table B.13: Case 9: Maximum difference in relation to the algorithm with 32 ACF lags in detection probability between the algorithms for different interference power levels.

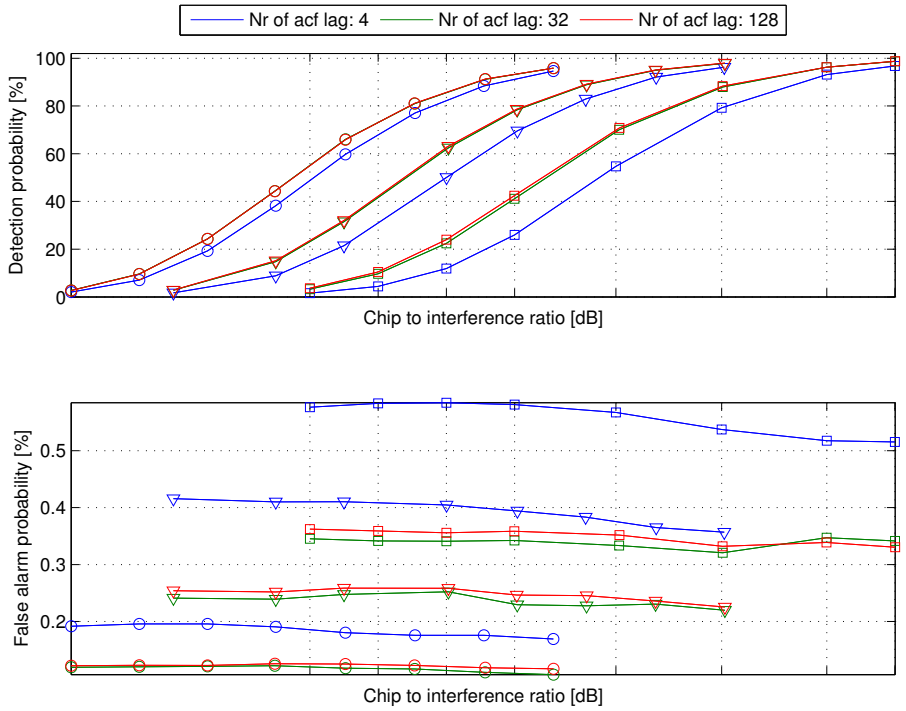


Figure B.14: Case 10: FFT-based algorithm with temporal whitening with different numbers of ACF lags (lines of different colors). Circles, triangles and squares represent 5 dB, 10 dB and 15 dB interference power respectively

Interferer[dB]	Nr of acf lag 32[%]	Nr of acf	Difference[%]	Ratio
5	80.96	128 lag at 81.09 %	0.13	1
10	62.41	128 lag at 63.12 %	0.72	1.01
15	22.46	128 lag at 24.02 %	1.56	1.07

Table B.14: Case 10: Maximum difference in relation to the algorithm with 32 ACF lags in detection probability between the algorithms for different interference power levels.

B.3 FFT-based algorithm with temporal whitening of the signal in parts

The following results show the performance of the frequency-domain baseline algorithm with temporal whitening of the signal in parts, compared to the performance of the time-domain baseline algorithm.

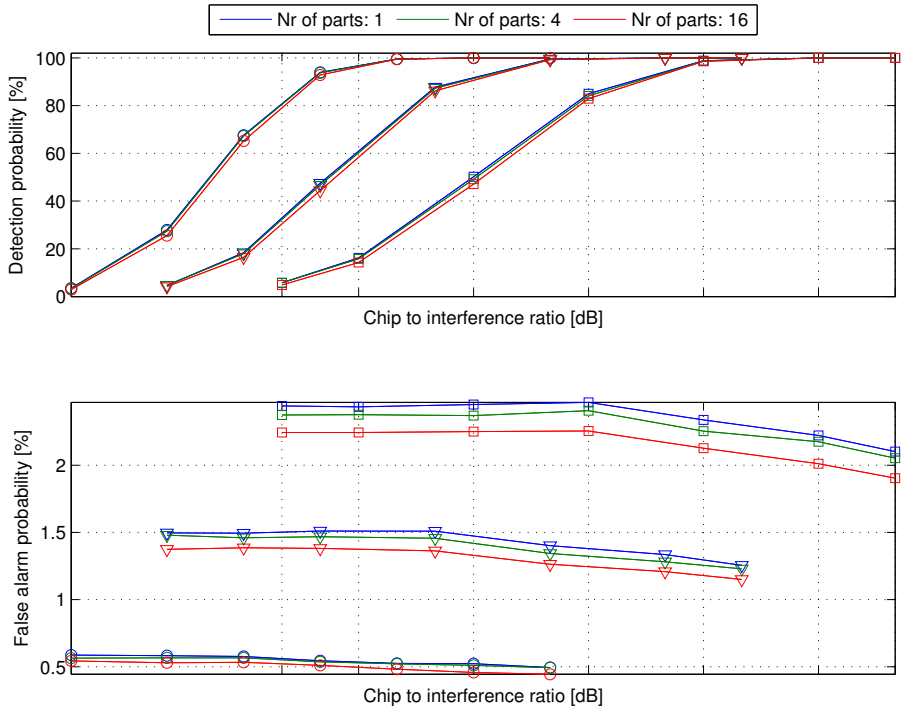


Figure B.15: Case 4: FFT-based algorithm with temporal whitening of the signal in parts (lines of different colors). Circles, triangles and squares represent 5 dB, 10 dB and 15 dB interference power respectively

Interferer[dB]	Nr of parts 4[%]	Nr of parts	Difference[%]	Ratio
5	27.4	1 part at 27.89 %	0.49	1.02
10	46.64	1 part at 47.48 %	0.84	1.02
15	49.15	1 part at 50.17 %	1.03	1.02

Table B.15: Case 4: Maximum difference in detection probability for the algorithm with four parts in relation to other amount of parts for different interference power levels.

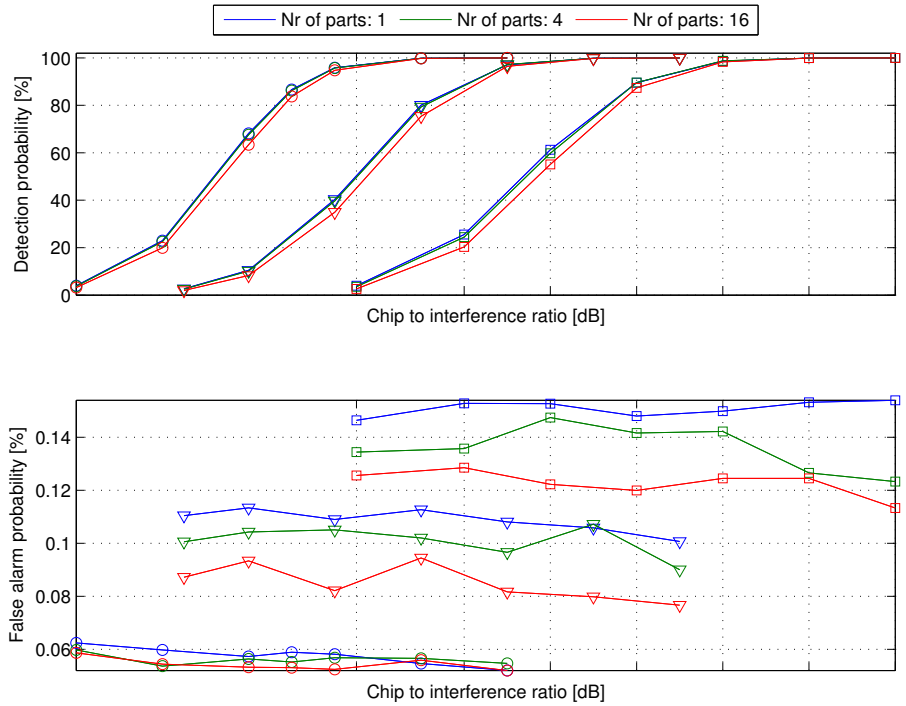


Figure B.16: Case 5: FFT-based algorithm with temporal whitening of the signal in parts (lines of different colors). Circles, triangles and squares represent 5 dB, 10 dB and 15 dB interference power respectively

Interferer[dB]	Nr of parts 4[%]	Nr of parts	Difference[%]	Ratio
5	67.59	1 part at 68.17%	0.58	1.01
10	79.3	1 part at 80.1 %	0.79	1.01
15	59.78	1 part at 61.24 %	1.46	1.02

Table B.16: Case 5: Maximum difference in detection probability for the algorithm with four parts in relation to other amount of parts for different interference power levels.

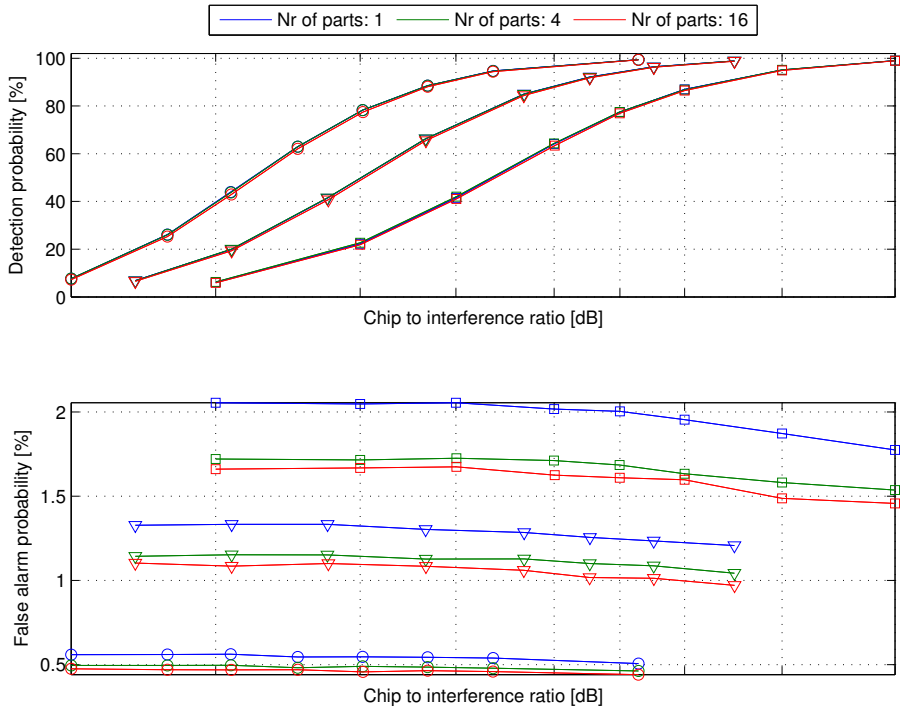


Figure B.17: Case 7: FFT-based algorithm with temporal whitening of the signal in parts (lines of different colors). Circles, triangles and squares represent 5 dB, 10 dB and 15 dB interference power respectively

Interferer[dB]	Nr of parts 4[%]	Nr of parts	Difference[%]	Ratio
5	43.74	1 part at 43.94 %	0.19	1
10	92.08	1 part at 92.17 %	0.09	1
15	86.87	1 part at 86.95 %	0.08	1

Table B.17: Case 7: Maximum difference in detection probability for the algorithm with four parts in relation to other amount of parts for different interference power levels.

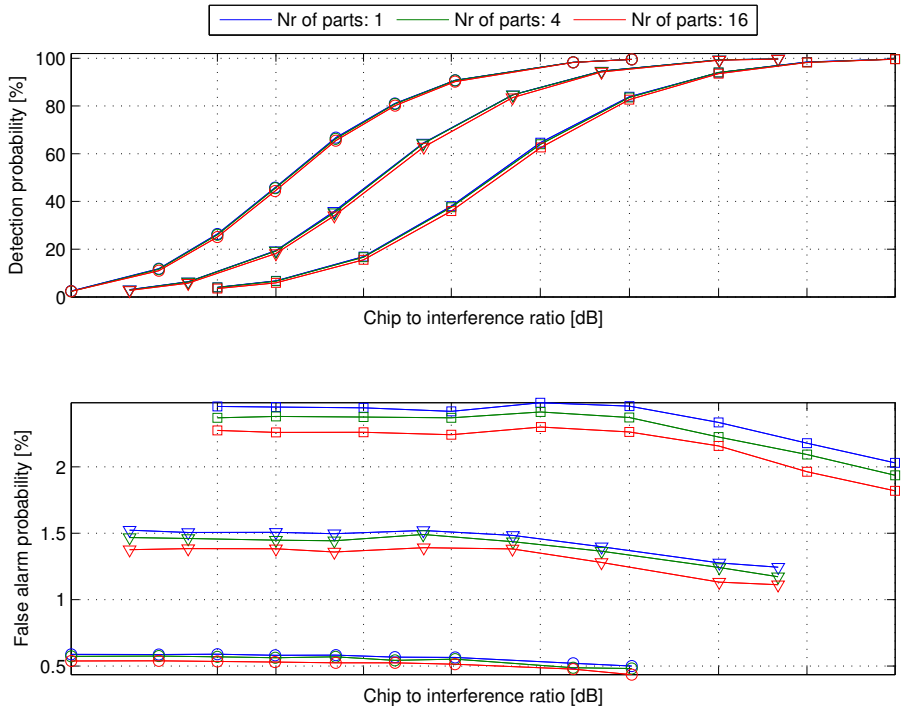


Figure B.18: Case 8: FFT-based algorithm with temporal whitening of the signal in parts (lines of different colors). Circles, triangles and squares represent 5 dB, 10 dB and 15 dB interference power respectively

Interferer[dB]	Nr of parts 4[%]	Nr of parts	Difference[%]	Ratio
5	66.25	1 part at 66.72 %	0.47	1.01
10	35.29	1 part at 35.9 %	0.61	1.02
15	64	1 part at 64.65 %	0.65	1.01

Table B.18: Case 8: Maximum difference in detection probability for the algorithm with four parts in relation to other amount of parts for different interference power levels.

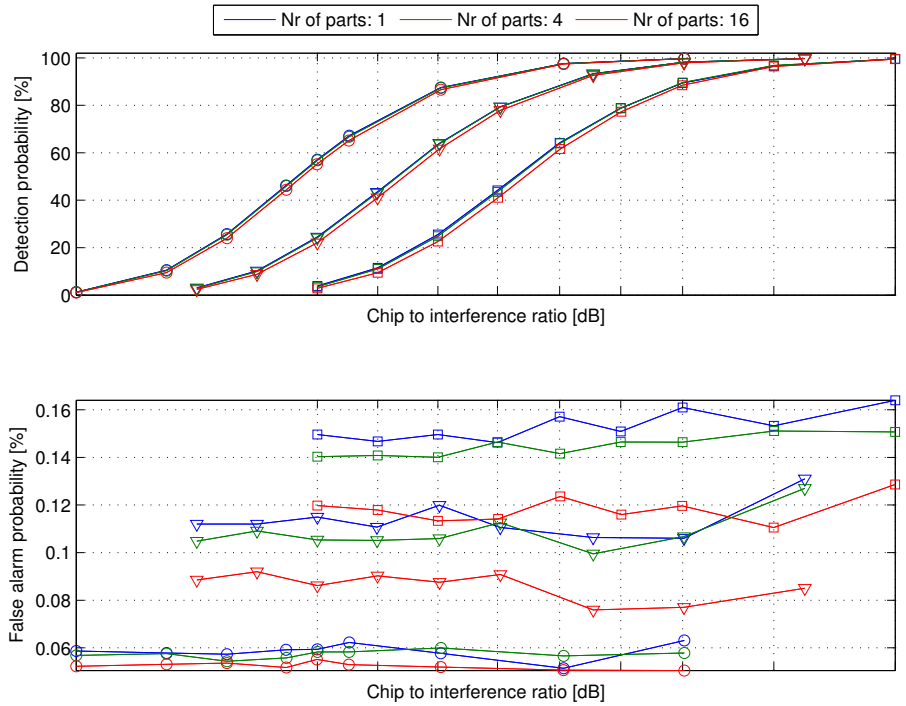


Figure B.19: Case 9: FFT-based algorithm with temporal whitening of the signal in parts (lines of different colors). Circles, triangles and squares represent 5 dB, 10 dB and 15 dB interference power respectively

Interferer[dB]	Nr of parts 4[%]	Nr of parts	Difference[%]	Ratio
5	66.75	1 part at 67.19 %	0.44	1.01
10	9.93	1 part at 10.23 %	0.3	1.03
15	24.83	1 part at 25.54 %	0.72	1.03

Table B.19: Case 9: Maximum difference in detection probability for the algorithm with four parts in relation to other amount of parts for different interference power levels.

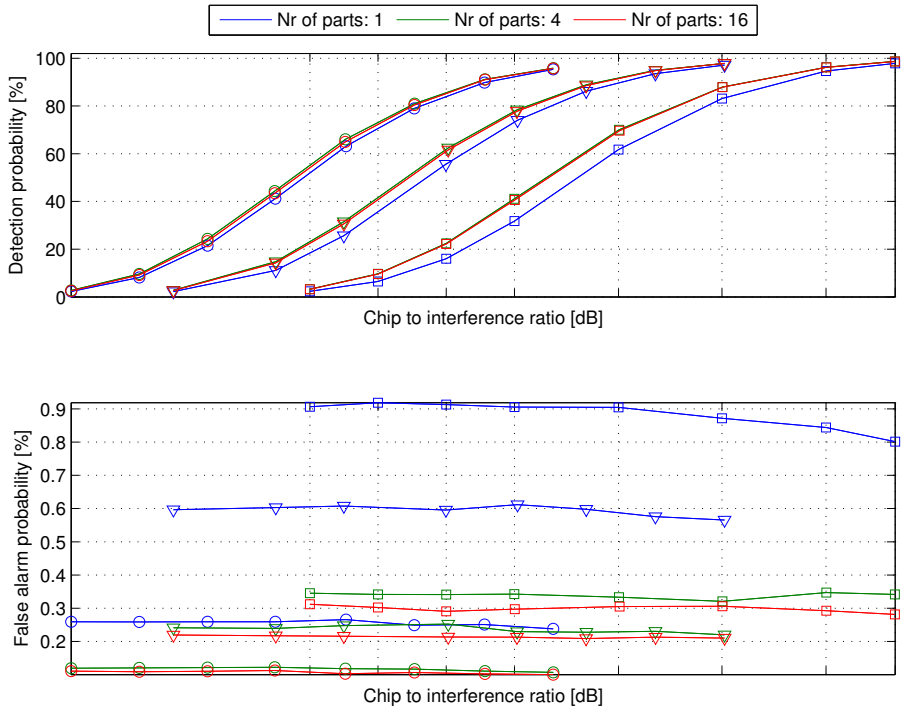


Figure B.20: Case 10: FFT-based algorithm with temporal whitening of the signal in parts (lines of different colors). Circles, triangles and squares represent 5 dB, 10 dB and 15 dB interference power respectively

Interferer[dB]	Nr of parts 4[%]	Nr of parts	Difference[%]	Ratio
5	95.84	16 part at 95.75 %	-0.09	1
10	97.85	16 part at 97.83 %	-0.02	1
15	96.22	16 part at 96.17 %	-0.05	1

Table B.20: Case 10: Maximum difference in detection probability for the algorithm with four parts in relation to other amount of parts for different interference power levels.