Framework:	Freeband Communications
Project Title:	Adaptive Ad-hoc free Band Wireless Communications (AAF)

Document Number:	
Document Deadline:	
Date of Delivery:	
Document Title:	**Research on Cognitive Radio within the Freeband-AAF project**
Source:	
Workpackage:	WP3 and WP4
Task:	
Document Type:	W (White Paper)
Access Rights:	Project
Version:	P1
Author(s):	A.B.J. Kokkeler
	P. Pawelczak
	I. Budiarjo
	M. Heskamp
	Q. Zhang
Editor(s):	H. Nikookar
	A.B.J. Kokkeler

Abstract:	*The main objective of this documents is to present the results of the research on Cognitive Radio within the AAF projet*
Keyword List:	

Version history:

version	date	author(s)	description
01	2008 12 09	AK	P1 version released

Contents

Chapter 1

Introduction

As economy and technology grows, the need for wireless communication will grow along with it. New radio communication standards need spectrum. Since the usage of the radio spectrum has grown considerably and the amount of usable spectrum is finite, it is reasonable to predict shortage of spectrum in the future. Today, however, this physical limit is not reached by far. The perception that radio spectrum is scarce is caused by the bureaucratic way spectrum is managed, rather than actual physical shortage.

Historically the spectrum is managed by governmental agencies like the FCC in the united states, and Agentschap Telecom in The Netherlands. Such agencies give out licenses and solve interference issues by a process which is known as 'command and control' [1]. Until recently, this was the only sensible approach for a number of reasons. First, traditional radio equipment only works on a limited number of frequency bands which are hard wired into it during manufacturing. Second, radios had bad selectivity and were very susceptible to interference.

Today almost all new radios have a powerful computer onboard. Initially, the only task of this computer was to control the device, but nowadays this task only takes a fraction of the total computational power. Also the analog frontend capabilities have grown beyond what is necessary for merely demodulating the signal. This over capacity can be used to sense the radio environment and to apply artificial intelligence on this measurement data. This idea was first proposed by Mitola [2] to make devices more user friendly. Later the idea was adopted by the FCC which considers the possibility of dynamic spectrum access, also known as Cognitive Radio (CR). The FCC issued a request for comments about this subject. Hundreds of researchers from companies and universities replied, and their comments showed disagreement concerning the feasibility of such an approach. But neither the advocates or the opponents of cognitive radio could make a compelling point about whether it would work or not, which opened the road for many research projects.

1.1 Project description

The importance of wireless communication in emergency situations does not require much explanation. The question how such communication should be organized is much more difficult. Important lessons are learned from large disasters from the past like the hurricane Katrina in New Orleans or the fireworks factory explosion in Enschede, The Netherlands. For example, directly after the firework incident, it happened that outgoing phone calls where impossible because the base station was overloaded, but incoming calls where no problem. Apparently there was enough free spectrum, but the base station was unable to efficiently use it.

The research which is described in this document is part of the 'Adaptive Adhoc free Band Wireless Communications (AAF)' project. The AAF project aims to apply the ideas of cognitive radio to the field of public safety and emergency control communications.

Within the project there are five closely related research topics.

- Scenarios and products
- Media access en routing
- Implementation in reconfigurable hardware
- Adaptive modulation
- Spectrum sensing

The research on Cognitive Radio within the AAF project, described in this document, focusses on Opprtunistic Spectrum Acces (OSA). OSA is a promising new spectrum management approach that will allow co-existence of both licensed and opportunistic users in each spectrum band, potentially decreasing the spectrum licensing costs for both classes of users and increasing spectrum efficiency.

1.2 Opportunistic Spectrum Access

Finding new unassigned frequency slots pushes system designers to explore higher and higher frequencies, e.g., 60 GHz. However, as already mentioned, most of the already allocated frequencies are not used, or used sporadically. Therefore, it is logical to allow non-licensed users to use these frequencies when they are free at a specific place and time. Theoretically, such an approach will increase overall frequency reuse without any licensing costs and will boost the throughput for applications that opportunistically use the empty frequencies. This communication technique is called *Opportunistic Spectrum Access* (OSA).

To make OSA feasible, new dynamic spectrum management techniques have been developed [3, 4]. Promising dynamic spectrum management solutions are *Exclusive Spectrum Management* (ESM), the *Spectrum Commons* (SC) sharing model, and *Hierarchical Spectrum Management* (HSM). In Fig. 1.1, important spectrum management techniques and their hierarchy are introduced. The ESM model still gives exclusive channel use to each user or provider, but differs from a static assignment in the sense that the channels are allocated dynamically among possible licensees. In the SC model, different users compete for the assigned frequencies on equal terms. The HSM model gives *Primary (Licensed) Users* (PUs) more rights to use the spectrum then to other *Secondary (Non-Licensed) Users* (SUs). We can distinguish two HSM approaches. In *Overlay* HSM, only one user/system can use a frequency band at particular space

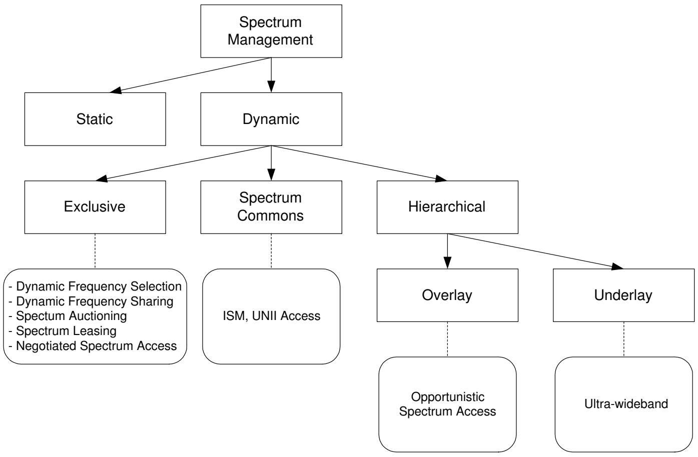


Figure 1.1: Modern spectrum management: Classification with the application examples (see also [4, Fig. 1]).

and time, and the SU has to back off when a PU is present. However, when no PU is present the SU can opportunistically use the frequency band, hence this technique is also referred to as OSA. In *Underlay* HSM, a SU can transmit in an already occupied band if this transmission does not increase the interference to the PU above a given threshold. A further classification of Overlay HSM (not shown in Fig 1.1) involves *Symmetric Coexistence* (when both SU and PU networks adapt) and *Asymmetric Coexistence* (when only the SU network adapts, obeying the PU requirements). For our research, we considered the case where the PU does not adapt to the operation of SU.

As a side note, we need to emphasize that different modern approaches of spectrum management outlined above are commonly mistaken with *Cognitive Radio* (CR) [2]. Fig. 1.2 explains the basic functional blocks of a Cognitive Functionality Wireless Communications Node (CFWCN). The Sensing Block and Policies Block (if available) are extensively used in deciding the availability of spectrum. These blocks also help to drive the Learning and Reasoning functions. The Learning and Decision Blocks may be implemented with fuzzy logic or neural networks. The decision database along with the input from the Sensing Block and Policies Block drives learning. The end result is that the radio is configured based on input from different layers of the communication stack as well as the environment. Concluding, OSA is a natural component of CFWNC, but not its synonym. Please refer to the IEEE P1900.1 standard for further discussion.

1.2.1 What to Expect from PU Spectrum Use

Although many researchers claim that the spectrum is used sparsely, it is in general very difficult to obtain good information about realistic spectrum use. To obtain an example of PU spectrum use, we measured the spectrum use in the frequency range $F =$ [446,04; 467,82] MHz on 13 March 2007 at different times between 11 AM and 8 PM in the Electrical Engineering department at the University of Twente in the Netherlands. Following the Dutch radio spectrum map, these bands are assigned to public mobile communication channels, with exception of those channels that are assigned to

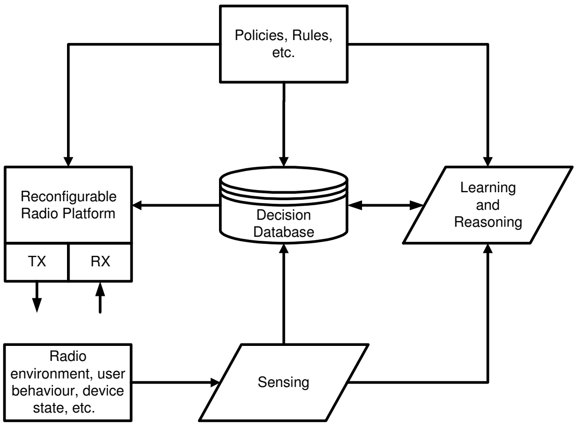

Policies, Rules, etc.
Reconfigurable Radio Platform
TX
RX
Decision Database
Learning and Reasoning
Radio environment, user behaviour, device state, etc.
Sensing

Figure 1.2: Cognitive Functionality Wireless Communications Node.

Feature	Value
Mean channel utilization	6.8%
Total number of busy bins	1.7%
Total number of free bins	15.7%
Total number of bins with PU duty cycles	82.6%
Slot to slot difference in available bins	47%
Number of free bins within frequency pool (Max)	85%
Number of free bins within frequency pool (Min)	16.6%
Average ON time	4.3 s
Average OFF time	58.9 s

Table 1.1: Results of the PU Channels Observations

the Dutch Ministry of Defence for aviation communication. We have extracted periods of PU signal activity (ON period) and PU non-activity (OFF period) for each frequency bin of 100 kHz, which made it possible to compute the PU activity metrics as listed in Table 1.1.

Only 1.7% of the frequency bins were busy the whole time, and could hence not be used by SUs at all. Also, 15.7% of all the observed frequency bins were free during the whole observation time. Therefore, the remaining 82.6% of all frequency bins showed ON and OFF patterns (with mean ON and OFF times of 4.3 s and 58.9 s, respectively). As a result, when a SU can not time-share a frequency bin with a PU, it can only achieve a spectrum utilization of 15.7%. A striking fact is that the total channel utilization of the measured frequency range F was only 6.8%. So, when time-sharing is possible, the SU can achieve a utilization of 93.2%, which is a significant improvement compared to 15.7%.

Next we studied the channel availability variations. For the chosen frequency range F, the average difference in available free frequencies between two consecutive time slots of 140 ms was 47%, which shows that the spectrum available to the SU can vary significantly. The minimum difference was 16.6%, which means there was always a variation. The maximum difference was 85%.

Our conclusion therefore is that dynamic spectrum access has the potential to dramatically improve spectrum efficienc but that any solution will require a high degree of spectral awareness and adaptivity. Early in the porject it was decided to concentrate on OFDM based radio systems, operating within the range from 200 to 800 MHz.

Based on the observations and the early decisions within the AAF project, the four PhD. students focussed their scientific research on Networking Aspects of Opportunistic Spectrum Access (2), Adaptive Baseband Processing for Adaptive Ad-hoc Freeband Cognitive Radio System (chapter 3), Spectrum Sensing for Dynamic Spectrum Access Radios (chapter 4) and Mapping Cognitive Radio onto a Reconfigurable Platform (chapter 5) and .

Chapter 2

Networking Aspects of Opportunistic Spectrum Access

by Przemek Pawelczak

2.1 Abstract

Opportunistic Spectrum Access (OSA) is a promising new spectrum management approach that will allow co-existence of both licensed and opportunistic users in each spectrum band, potentially decreasing the spectrum licensing costs for both classes of users. This chapter will provide information on the research results on the networking layer for OSA achieved so far within AAF project. Particularly, the focus of this chapter is on the advances in medium access control and transport layer design. First we show what are the challenges with implementing Transport Control Protocol (TCP) over OSA links, where we conclude that current TCP implementations can indeed achieve good performance on OSA links, only when selective acknowledgments are implemented in TCP design. Later we will show guidelines for the design of multichannel medium access control (MAC) protocols for OSA. There we conclude that only MACs that maximally spread control and data exchange among opportunistic channels can achieve very high throughput and low interference levels induced to licensed users.

2.2 Transport Control Protocol Performance over OSA Links

TCP has constantly evolved since it's original conception. A good overview in the context of wireless networks is given in [5]. Many versions ('flavors') of TCP are currently in use, but probably the most commonly used TCP in the Internet today is New Reno [6], which improves the Fast Recovery Algorithm of its ancestor Reno [7]. In the congestion avoidance phase, New Reno (and Reno) probe the network by additively increasing the sending rate by a segment per round-trip time, until a packet loss occurs. Thus, they use packet loss as an indicator of congestion, causing a periodic oscillation of the congestion window, which reduces throughput.

A promising new TCP variant is Vegas [8]. In the congestion avoidance phase, Vegas constantly measures the round-trip time of the connection, calculates from this the actual and expected segment flow rate, and from this the number of segments that (it believes) are queued in the network. Two parameters, called α and β control the size of the congestion window. Per round-trip time, when the calculated number of queued segments is less than α, the congestion window is increased by one segment, if greater than β, the window is decreased by one segment, else the window is not changed. The default values of α and β are 1 and 3, so Vegas in essence attempts to keep between 1 and 3 segments queued in the network. Because Vegas avoids congestion, it does not suffer from Reno's congestion window oscillations, and achieves better throughput in certain scenarios.

Most modern TCP stacks employ selective acknowledgments [9] (SACKs), which allow a TCP receiver to indicate up to 3 blocks of segments that have been correctly received. Old-style cumulative acknowledgments only allow the receiver to indicate the highest in-order segment received. The more precise SACK information enables the sender to re-transmit only those segments actually missing, and can result in much improved performance, especially in more dynamic network environments where multiple losses may occur more frequently (e.g., OSA links). In this work, we mainly consider SACK enabled TCP stacks, as these are the common case today.

Because of their different characteristics, especially in the congestion avoidance phase, these TCP flavors can be expected to perform differently over OSA links. Reno more aggressively probes the network and as a result, many packets are typically buffered in the network, perhaps allowing it to instantly grab capacity of a OSA link with packets already in the network. On the other hand, Vegas attempts to keep between only 1 and 3 segments queued in the network, which avoids oscillations in the congestion window and rate, but this may limit is ability to grab additional bandwidth. Also, Vegas' view of the network capacity may be disrupted by greatly varying RTT [10] due to abrupt capacity changes of OSA links.

The following section discusses in detail the performance modern TCPs achieve in networks using OSA links.

2.2.1 Simulation Setup

To investigate the performance of different TCP flavors in a OSA environment, we have constructed a basic simulation scenario shown in Fig. 2.1. A sender is connected to a Base Station (BS) by means of a wired connection, representing the Internet (IPv4). The receiver is connected to the BS via a OSA link of varying capacity. The BS buffers and forwards packets. A TCP connection is established between the sender and receiver, and an infinite flow of TCP segments travel from sender to receiver, while TCP acknowledgments flow in the opposite direction. We simulate the TCP connection, of which we discard the first 100 seconds, to remove the effect of TCP's startup phase. We record the number of segments TCP managed to transfer in the subsequent 10000 seconds. All simulations, as noted earlier, were performed using NS version 2.29, with TCP-Linux enhancement.

The wired connection has a fixed capacity of 10 Mbit/s and a constant delay representing the (simplified) delay a packet incurs while traveling the Internet. On the OSA link, a packet incurs no propagation delay. In addition to these delays, packets incur a transmission delay according to the current bit rate of a link, and queuing delays

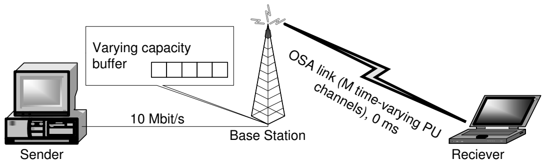


Figure 2.1: Basic OSA network used for TCP performance evaluation.

depending on occupancy and maximum size of the buffer in the BS. The bit rate of the wired link is chosen such that the OSA link is the bottleneck link.

The OSA link is constructed as follows. From the BS to the receiver, the BS has access to M channels, where each individual channel has equal capacity. The sum of all channel capacities is 2.4 Mbit/s. In addition, a small non-time-varying channel of 0.1 Mbit/s is always available to the BS, making the maximum and minimum available capacity 2.5 and 0.1 Mbit/s, respectively. Moreover, individual channels are occupied randomly and independently of each other by the PU, according to an exponential distribution, where parameters for arrivals and departures (μ and λ, respectively), are the same for every OSA channel. Thus, $1/\mu$ and $1/\lambda$ are the average 'on' and 'off' period of a channel. In the other direction, from the receiver to the BS, TCP acknowledgments can be transmitted by the receiver at a constant 2.5 Mbit/s rate. Furthermore, the BS' PU detection is perfect, and no errors occur on the wireless link.

In the simulations, the delay of the fixed link is varied between 5 and 100 ms, and the size of the BS's buffer between 5 and 100 packets, giving a wide range of network configurations one might encounter in the real world. For the OSA link, $1/\lambda, 1/\mu \in \{1.5, 5.5\}$ s (compare these values with ON and OFF values from Table 1.1), and it consists of $M \in \{3, 12\}$ channels. We have chosen the values of λ (OFF time) smaller than those extracted from measurements, resulting in a more dynamic OSA link, but representing possible combinations of arrivals and departures of the PU on a OSA link.

Given the fixed *total* capacity of 2.4 Mbit/s of these channels, individual channels are 200 kbit/s in 12 channel models, and 800 kbit/s in the 3 channel models. As discussed earlier, we mainly consider the Linux implementation of Reno and Vegas, but also simulate NS' implementation of New Reno, and Reno with selective acknowledgments (referred to as 'Sack' in the following). For Linux' New Reno and Vegas, and NS' Sack, the receiver uses selective acknowledgments, whereas for NS' New Reno it does not. The receiver sends one acknowledgment per received packet (i.e. no delayed acknowledgments), as this was shown to produce behavior closer to that of the actual Linux OS [11], that dynamically adapts its acknowledging strategy. The maximum segment size of TCP is set to 960 bytes, resulting in packets of 1000 bytes, after the IP header is added. We set the maximum congestion window to 1000 packets, well beyond the (maximum) bandwidth delay product (BDP) of the path, as setting it close to the BDP is not possible for a link of varying bandwidth. Finally, we set the minimum retransmission timeout to 0.2 s for all TCPs, as this is current practice. The BS buffer is a simple first in first out queue that drops arriving packets when it is full. We simulated all combinations of the above parameters.

Finally, for all simulations, we calculate in bytes the total OSA link capacity C_{tot} that was available to TCP over the entire measured period (from t_1 to t_2 s), and the number of bytes TCP actually managed to transfer in this period, referred as C_{act}. From these, we calculate the efficiency ϵ of TCP,

$$\epsilon = \frac{C_{act}}{C_{tot}} = \frac{A(t_2) - A(t_1)}{\int_{t_1}^{t_2} R_{lnk}(t)dt} \in [0, 1],$$

where $R_{lnk}(t)$ is the available OSA link rate at time t (bytes/s), and $A(t)$ is the number of bytes acknowledged at the sender at time t.

In summary, we simulate a single long-lived bulk TCP transfer over a network path where the OSA link is the bottleneck link, and measure the achieved efficiency. We compare the achieved efficiency of number of TCP flavors, and see which performs best and why for our simulated OSA link environments. We do not look at fairness among multiple TCP connections, nor do we consider short lived TCP connections (e.g. web-traffic). We simulate a OSA link with optimal and instantaneous PU occupancy measurements, without any wireless loss.

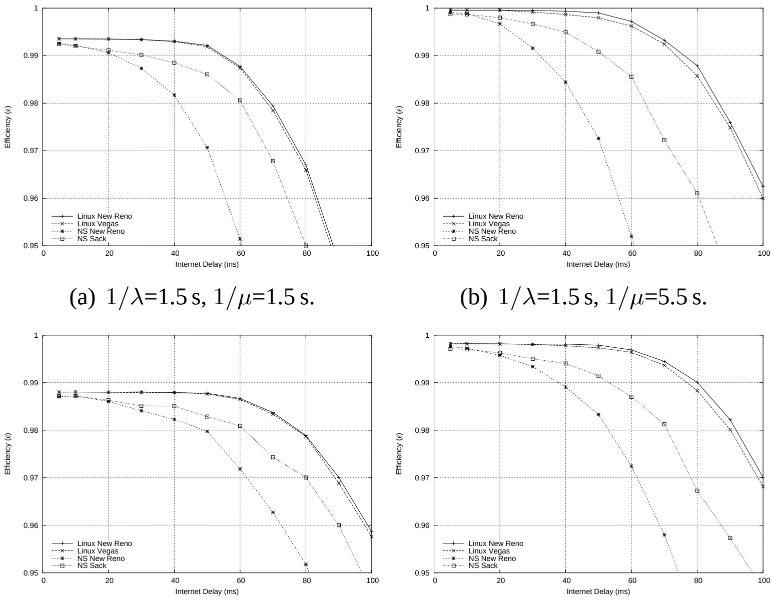
We now present the results of our simulations.

2.2.2 Simulation Results: Discussion of Models

Fig. 2.2 and 2.3 show TCP efficiency achieved by all TCP flavors in all 3 and 12 channel models, respectively. The efficiency is plotted as a function of wired link delay, for a (reasonable) buffer size of 50 packets. We can see that all TCPs achieve higher efficiency in 12 channel models, compared to their performance in 3 channel models, under otherwise equal conditions. The reason for this is the smaller link capacity change in 12 channel models when a channel becomes available or unavailable (recall individual channels are 200 kbit/s in 12 channel models, versus 800 kbit/s in 3 channel models, and they become (un)available independently of each other). Therefore, in the 12 channel models there is a relatively larger buffer to potentially i) grab capacity by transmitting packets queued in the buffer when the OSA link capacity is increased, and ii) absorb packets when link capacity is decreased until the sender can lower sending rate. Additionally, there is a low probability, due to the features of the exponential distribution, that more than one PU channel will change state simultaneously (or at almost the same time). Thus, the 12 channel model capacity will usually change by 200 kbit/s at a time, whereas in the 3 channel models, the granularity of change is 800 kbit/s (see Section 2.2.1).

Looking at the rate at which OSA link capacity changes occur, TCPs achieve better performance on links with long ‘on’ and ‘off’ periods, than on links with short ‘on’ and ‘off’ periods (compare, e.g., Fig. 2.3(a) and 2.3(d)). This is not surprising, as TCP needs to adapt less often because the OSA link changes capacity less often (for a given interval). Also, once TCP has converged to the new link capacity, it can operate there for a longer time.

Comparing the average duration of ‘on’ and ‘off’ periods, we see that for short delay, all TCPs perform better in the 12 channel model when $1/\lambda$=1.5, $1/\mu$=5.5, than when $1/\lambda$=5.5, $1/\mu$=1.5, achieving almost 100% efficiency in the former, see Fig. 2.3(b) and 2.3(c)). In this case, ‘on’ periods are easier to adapt to than short ‘off’ periods. Interestingly, the opposite becomes true as end-to-end delay increases. Here, we see performance start to drop beyond delays of approximately 80 ms for the



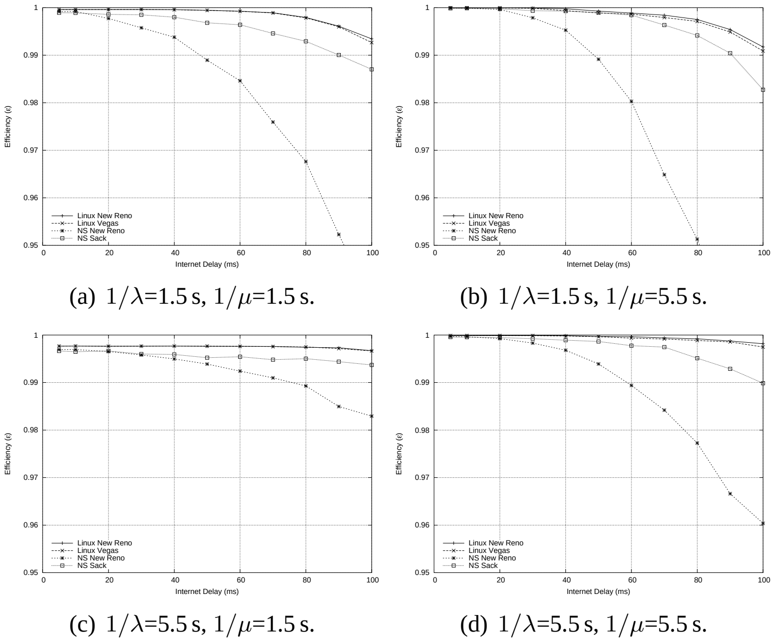
(a) $1/\lambda$=1.5 s, $1/\mu$=1.5 s.

(b) $1/\lambda$=1.5 s, $1/\mu$=5.5 s.

(c) $1/\lambda$=5.5 s, $1/\mu$=1.5 s.

(d) $1/\lambda$=5.5 s, $1/\mu$=5.5 s.

Figure 2.2: TCP efficiency of all analyzed TCP flavors as function of wired link delay; 3 channel model, BS buffer size of 50 packets.



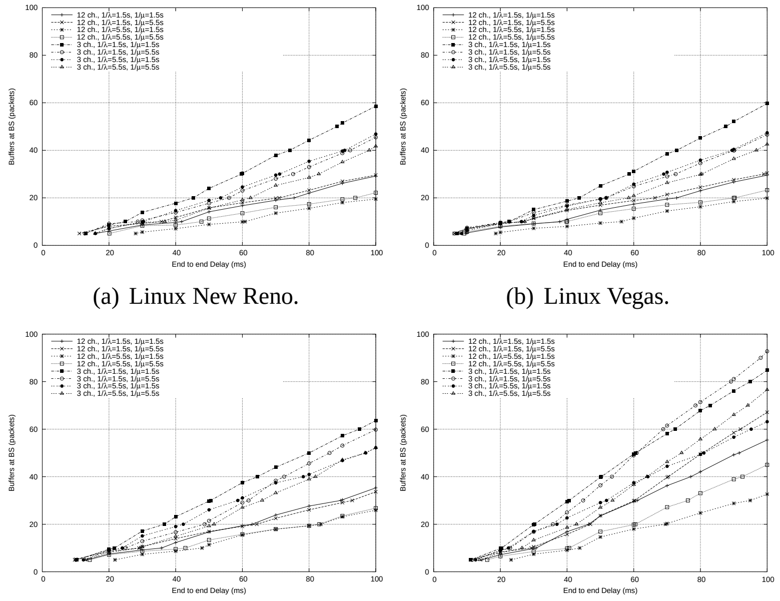
(a) $1/\lambda$=1.5 s, $1/\mu$=1.5 s.

(b) $1/\lambda$=1.5 s, $1/\mu$=5.5 s.

(c) $1/\lambda$=5.5 s, $1/\mu$=1.5 s.

(d) $1/\lambda$=5.5 s, $1/\mu$=5.5 s.

Figure 2.3: TCP efficiency of all analyzed TCP flavors as function of wired link delay; 12 channel model, BS buffer size of 50 packets.



(a) Linux New Reno.

(b) Linux Vegas.

(c) NS Sack.

(d) NS New Reno.

Figure 2.4: Buffers required to achieve 95% efficiency, for all models, grouped by TCP flavor. The data points are acquired via linear interpolation of the measured data.

link with $1/\lambda$=1.5, $1/\mu$=5.5, whereas for the link with $1/\lambda$=5.5,$1/\mu$=1.5, efficiency is unaffected by end-to-end delay (given a buffer size of 50 packets).

This is due to the following. For our 12 channel link models, *when end-to-end delay is large*, it is easier to utilize an ‘on’ period using packets from the buffer, than it is to adapt the sending rate to even a short ‘off’ period. A decrease in link capacity (‘off’ period) will likely lead to packets being lost as the BS buffer overflows. Loss leads to (multiplicative) reductions of the congestion window, and possibly even timeouts. We can conclude that, overall, grabbing extra bandwidth is easier for TCPs (as it is actually achieved by the BS buffer) than reducing the sending rate (while maintaining high efficiency). For the 12 channel models, λ has a greater effect than μ, and when $1/\lambda$ is small, TCP performance suffers most.

This effect can also be clearly seen in Fig. 2.4, where the number of buffers required at the BS to achieve 95% efficiency is plotted against the delay of the wired link (Internet delay). Focusing on the TCPs that employ selective acknowledgments in Fig. 2.4(a)-2.4(c) we can see the following. For the 12 channel models, for large delays, the number of required buffers is mostly determined by the duration of the ‘off’ period, as the curves are grouped according to the value of λ. The same cannot be said of the 3 channel models. Here, λ and μ both affect TCP performance. This is due to the relatively smaller buffer, compared to the change in link capacity, which is typically 800 kbit/s for 3 channel models. The buffer does not contain sufficient packets to keep the OSA link saturated after a capacity increase, until the sender can increase its rate, whereas it does for 12 channel models. As a result, the effect of an ‘on’ period is not hidden, as it was in the 12 channel case.

2.2.3 Summary of the Results

We have investigated the performance of a number of TCP flavors in a OSA environment. We can conclude that modern, real-world TCP stacks can achieve better than 95% efficiency on OSA links with widely varying characteristics, under a very wide range of network configurations, if i) a large (but not unrealistically so) buffer is available at the Base Station, and ii) the receiver employs Selective Acknowledgments. We have also seen that TCPs have trouble adapting to even brief reductions in capacity, if end-to-end delay is large. This implies that the probability of false alarm, a parameter of the OSA link's Primary User (PU) detection process, may have a larger effect on throughput than is apparent from theoretical analysis of TCP's *steady state* behavior.

2.3 Multichannel Medium Access Control

2.3.1 OSA QoS Tradeoffs

Our next goal is to quantify OSA dependability in terms of classical QoS parameters like throughput and delay, as a function of PU parameters such as load and tolerance to interference or collisions from the SUs. Intuitively, the OSA QoS will be improved when the PU is more tolerant to interference or when it has a lower load. Quantifying this however requires making assumptions about the OSA MAC protocol, since the optimal MAC design will result in the best joint SU-PU performance. Another question that hence needs to be addressed is "How should the SU exploit the available spectrum to achieve a reliable communication?". As we are focusing on the MAC design here, we answer this question by first listing all features that are important for OSA networks and showing how these have been addressed in the literature. Where possible, we quantitatively assess which solution is optimal and hence results in the best SU QoS for a given PU set of requirements.

2.4 Key Features of OSA MACs

Quantifying dependability for the scenario where SUs and PUs share a set of channels in time and frequency requires making assumptions about the OSA network operation. We have listed many important OSA MAC proposals found in literature and identified a set of key features required to enable OSA operation. Our focus is on decentralized MAC protocols only, i.e., each OSA node locally decides when and how to access the channel. In addition, many centralized solutions have been proposed where a coordinator organizes the channel access. For instance, the current proposal for IEEE 802.22 WRAN [12] is an example of such an OSA protocol. We are also aware of proprietary OSA MACs found in the OSA devices of Shared Spectrum Company, Philips and Microsoft, but since their specifications are not public we were not able to include them in the survey.

We briefly introduce the identified features, as listed in Table 2.1, for the protocols found in literature. Before the SU network can start operating, it should decide on the set of channels to use. This bootstrapping is hence a first SU MAC feature that deserves attention. Next, after the set of possible channels is identified, the network should decide on how to organize the SU communication over those channels. The

Protocol Name	Bootstr.	Type	Scan.	No. RFEs	Policies
BB-OSA [13]	No	DCC	No	1	—
ESCAPE [14]	No	DCC	Yes	1	P1,P2
C-MAC [15]	Yes	DCC	Yes	1	—
MMAC-CR [16]	No	DCC	Yes	1	P1
Choi et. al. [17]	No	DCC	No	2	—
Shu et. al. [18]	No	DCC	No	2	P1
AS-MAC [19]	Yes	DCC	Yes	1	—
DOSS [20]	Yes	DCC	Yes	3	—
HC-MAC [21]	No	DCC	Yes	1	—
Su *et. al.* [22]	No	DCC	Yes	2	—
SRAC [23]	No	SPCC	No	1	P1
HD-MAC [24]	Yes	SPCC	Yes	1	P1

Table 2.1: Survey of Representative OSA MACs Discussed in Section 2.4.5

more channels of a given bandwidth are used, the more throughput the SU network can achieve. Also, since each channel can potentially be claimed by a PU, the probability that a SU looses all its channels decreases when using more channels. We hence assume a multichannel OSA MAC, and selecting a MAC type is considered to be the next important feature. Next, OSA operation requires information about the presence of PUs, and how this is implemented is a third important design choice. Depending on the multichannel MAC type and the organization of the scanning, more or fewer front-ends are required to work in parallel, which is a fourth design choice. Finally, a policy is required to establish the coexistence rules with the PU. The stricter the policy, the more difficult it becomes for the SU. Below, we discuss each feature in more detail and quantify the effect on QoS where possible.

2.4.1 Bootstrapping

Bootstrapping is the process during which an SU node decides which PU channels are suited for opportunistic spectrum communication. In one scenario, third parties provide information about such channels so that the SU node only has to consult such third party when it wants to start or join a network. Other scenarios assume that each node finds those channels locally, which can involve a significant amount of spectrum scanning. Next to finding the channels, each node should distribute its set of channels to other users in the network. Interestingly, only a handful of proposed OSA MACs consider bootstrapping, i.e. C-MAC [15], AS-MAC [19], DOSS [20] (only for a Control Channel), and HD-MAC [24]. Usually MAC designers assume that each OSA node has a preprogrammed list of PU channels for use. In the rest of this chapter, we assume that each node has decided the set of opportune channels, and that this set of channels is available to each node in the SU network. Each SU node hence operates on the same set of channels. We assume that a channel is opportune when a PU is not using it constantly, i.e., channels with no PU present and channels with some PU activity. In most cases, this gives us a set with more than one channel. In the next section, we discuss how to organize the SU communication across those channels.

2.4.2 Control Channel Design

After the bootstrapping procedure, the SU network has decided on a set of possible channels. Now, for each data packet transmission, the SU transmitter and receiver have to coordinate which channel and time slot they will use for that transmission. This coordination is typically implemented with a *(common) Control Channel* (CC). From a reliability viewpoint, this CC is a very crucial element of the MAC design, since no SU data communication is possible when it is obstructed.

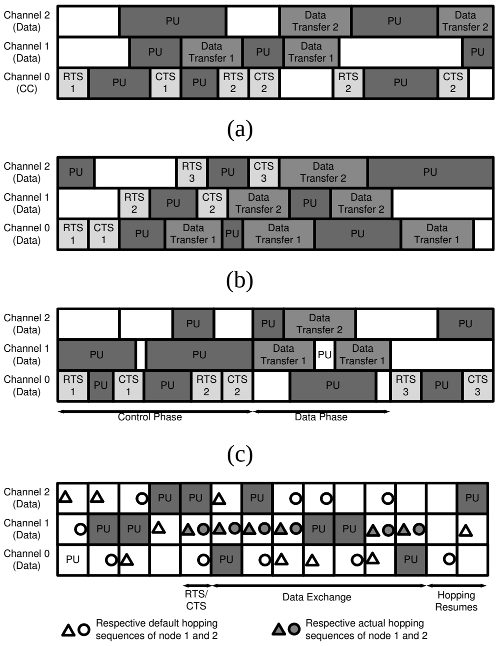
Using the approach defined in [25, Section II] for general multichannel MACs, we can identify four types of CC implementation, as listed in Fig. 2.5:

1. *Dedicated (Common) Control Channel* (DCC), where one SU channel is dedicated solely to the transport of control messages. All nodes should overhear the control data exchange, even during the data exchange. As a result, one Radio Front-End (RFE) needs to be dedicated to the exchange of control data. When only one RFE is used, transmission of control and data packets is time divided but then the operation of the protocol gets more complex. The drawback of the DCC approach in the context of OSA is that when a PU is active on the control channel, all communication is obstructed. It is hence often assumed that the CC should always be available or free from PU. We will discuss this issue in more detail below.

2. *Hopping Control Channel* (HCC), where all nodes hop between all channels following a predefined pattern. When both sender and receiver successfully exchange control messages on the current channel, they stop hopping and start transmiting data. After that, they come back to the original hopping pattern. HCC has the advantage that it uses all channels for transmission and control, whereas in DCC the CC can be used to transfer control packets only. Also HCC does not require a single channel to be free from PU activity.

3. *Split Phase Control Channel* (SPCC), where time is divided into control and data phases. During the control phases, all nodes switch their RFEs to the dedicated CC and decide on the channels to use for the upcoming data transfers. After each control phase, a data phase allows for data transmissions on the agreed channels. The advantage is that the control channel can be used during the data phases. Also, compared to DCC, no extra RFE for the control channel is needed. On the other hand, SPCC needs stronger synchronization to identify control and data phases.

4. *Multiple Rendezvous Control Channel* (MRCC), where multiple nodes can exchange control information at the same time, using all available channels. Each node knows the hopping pattern of the others (such hopping pattern is based on the seed of a pseudo-random generator), which makes control exchanges possible by following the intended receiver on its hopping sequence. MRCC maximally spreads both control and data exchanges across the channels, in a very random way. As a result, MRCC seems to be the most robust to PU activities on any of the channels. We will illustrate this quantitatively below. MRCC however also requires a more stringent synchronization between the hopping users since users have to keep track of meeting times.



(a)

(b)

(c)

(d)

Figure 2.5: Illustration of the operation of different multichannel MAC types, with PU activity on each channel: (a) DCC, (b) HCC, (c) SPCC, (d) MRCC.

We further study the different MAC types. First we assess the impact of PU activity on the control messages exchange. Next, delay and throughput of the SU network are determined for each of the four MAC types.

PU Activity on Control Channel

For HCC and MRCC the exchange of control messages on PU channels is inevitable since the exchange of control data is spread among all channels in the SU network. This certainly affects network availability and communication reliability. However for DCC and SPCC, the Control Channel does not necessarily need to be implemented on a channel with PU activity. More specifically, a single dedicated CC that does not suffer from PU activity can be built using a proprietary non-PU channel, e.g., ISM or UNII channels, or using a wideband transmission technique such as Code-Division Multiple Access (CDMA) or Ultra-wideband (UWB). The first approach is the most used in the literature, i.e., [15, 24]. Spreading the control channel among a wide bandwidth is very robust against PU activity, but limits the operating range of the network since UWB throughput decreases strongly with distance. When it is not possible to use a proprietary or wideband channel, the concept of a *Backup CC* has been proposed [15]. Indeed, the probability that both CCs are occupied by a PU simultaneously is smaller. However this solution is resource inefficient.

Since DCC is a very popular choice for OSA MAC protocols, see Table 2.1, we quantify the impact of PU activity in the CC on the throughput that can be achieved in a SU network using DCC. This will allow us to assess how important it is for a SU network to get a proprietary channel for its operation. For this, we have extended the analytical model for multichannel MACs proposed in [25] with a more detailed physical layer model to capture the impact of PU and SU interference. Also we have implemented the PU presence and the PU scanning process (for details see Section 2.4.3). In Fig. 2.6(a) we plot the impact of PU presence in the 2 Mb/s dedicated control channel on the SU throughput as function of SU data packet size. In the simulation, PU presence was modeled as a Bernouilli process with average presence rate q_{cc}. This presence is detected with a probability of 0.99 and with a probability of 0.03 the SU falsely assumes the PU to be present on the channel and pauses control message exchanges. A total of three SU channels are considered, and one of those is the control channel. The PU was assumed to be present in the control channel only. The interesting conclusion is that the SU can control how dependent it is on the control channel by tuning its data size. Indeed, for larger data packets, less control messages need to be exchanged, so the impact of the control channel is smaller. When the data size needs to be smaller, the impact of PU activity is larger, and in this case the concept of a backup control channel could be helpful.

PU activity versus SU Throughput and Delay

Because of their opportunistic nature, it is generally assumed that SU networks should be highly tolerant to delays. Indeed, it can happen that all channels are used by the PU, causing the communication to be suspended. Let us now quantify how large the SU delay becomes as a function of PU activity on the channels. This will give us important information on the type of applications that can be supported on OSA networks, or alternatively how the PU activity should be limited to be able to support a targeted SU application.

From Table 2.1 it is clear that the majority of OSA MAC proposals use DCC and only two use SPCC. Surprisingly, we were not able to identify HCC and MRCC proposals in the OSA literature. To be complete, we have however considered these four MAC classes. We have implemented all the protocols in a coarse time-slotted simulator. It allows to capture all the intrinsic features of the considered MACs, especially the way the control data exchange is organized. For a more detailed description of the simulator readers are referred to [25, Section V]. We have extended the simulator with PU activity patterns.

In Fig. 2.6(b) we plot the simulated delay of the SU applications as function of PU activity for each of the MACs. In the simulation, the OSA network consists of 20 devices, every user was generating traffic following a Poisson distribution with average transmission rate of 150 kb/s (the total SU load is 50% of the total bandwidth). There are 3 channels. Every channel has a fixed bandwidth of 2 Mb/s. PU activity was modeled through a geometrically distributed on-off process. The average PU packet was 0.8 ms (solid line in Fig. 2.6(b)) or 8 ms (dashed line in Fig. 2.6(b)), while the average off time was varied from 0.8 ms to 160 ms resulting in a range of average PU activity levels. We have assumed a perfect detection of PU activity on each individual channel. The striking fact is that MRCC is the best MAC among all, whatever the PU activity level is. Its immunity to temporal non-availabilities of the channel and efficient use of the whole channel capacity presents this type of MACs as a candidate for real-life implementation. This is because MRCC randomizes both control and data exchange significantly. Another observation is that the delay of all MAC classes becomes higher with increasing PU packet size. Because of its randomizing properties, MRCC suffers less. PU traffic can have ON times or packet durations in the order of seconds. For the given scenario, even when the PU activity reaches 30%, the delay experienced by the SU is still lower than 100 ms. This delay could even fit within the bounds for packet voice communication, where the round-trip delay for a voice conversation should not exceed 400 ms according to the ITU-T G.114 recommendation.

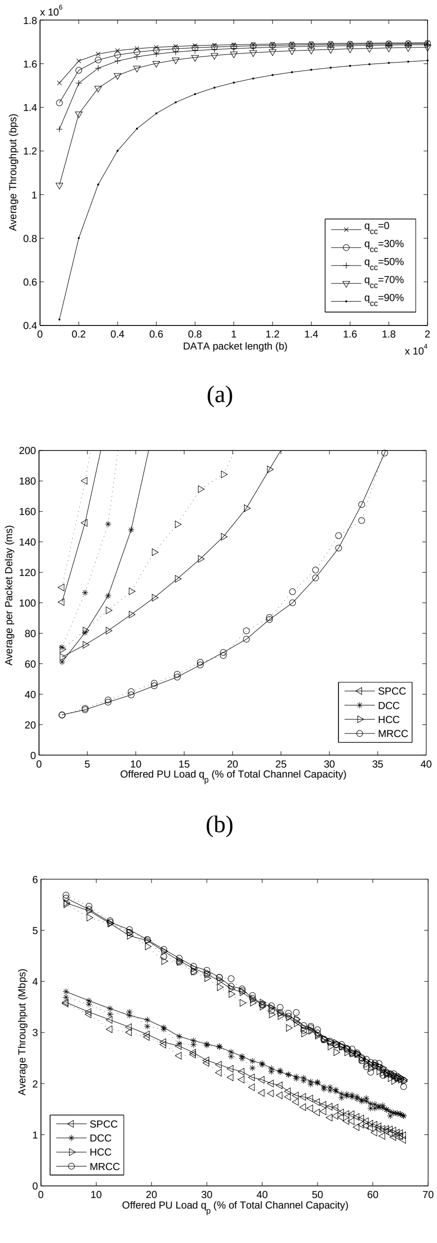
Next, in Fig. 2.6(c), we assess OSA network throughput as a function of PU activity for a similar scenario (the solid line is now 8 ms and dashed is now 80 ms PU packet size). As expected, the average SU throughput decreases linearly with PU activity. SPCC performs the worst in this case, since it wastes a lot of bandwidth on the data channels during the control phase. MRCC is still the best MAC design. The average throughput does not vary a lot with PU packet size.

2.4.3 Scanning Process

Since a SU cannot use the channel when a PU is present, it should obtain information about PU activities on each channel. Typically, this is implemented using PU detectors [26]. Alternatively, PU activity information can be assumed to be broadcast by a central device. We can thus classify OSA MAC protocols into a) sensing and b) non-sensing OSAs. From Table 2.1 we can conclude that the majority of the considered protocols assume having the scanning under their control.

Unfortunately, scanning increases the overhead since nodes cannot transmit when they are scanning. Since it is often difficult to distinguish SU and PU signals, the whole SU network has to be quiet during sensing which requires *Quiet Period Management* [15]. Scanning, or hence quieting the network, can be done periodically or before each transmission attempt. The distance between two consecutive sensing intervals varies, and is often a function of the policy. The more tolerant the PU to interference, the less often the sensing should be done. Noise, fading, multi-path shadowing and low PU signal levels make a reliable detection process difficult. Suboptimal detectors not only affect the PU QoS levels, but the SU QoS as well.

Scanning performance is measured in terms of the probability to detect a PU when present, and the probability to falsely detect a PU. In the former case, both SU and PU QoS is degraded, since a SU will transmit and collide with the PU resulting in packet loss for both PU and SU. In the latter case, SU QoS is degraded, since a SU will not transmit when the channel was actually free. It is well known that scanning performance improves with increasing scanning length [27], and in Fig. 2.7(a) we investigate the optimal sensing time in terms of SU QoS, for a scenario with DCC MAC. Scanning is performed using energy detection before each transmission attempt and Rayleigh fading is assumed. SU throughput indeed improves with increasing detection reliability. When detection performance is acceptable, the throughput starts to decrease since the scanning overhead dominates. This effect is less visible with the high PU activity, since the OSA network will hot have enough opportunities to communicate, therefore it will not loose much from the already small PU channel capacity. The impact on PU QoS will be discussed in Section 2.4.5 since PU QoS can be considered to be a policy constraint.

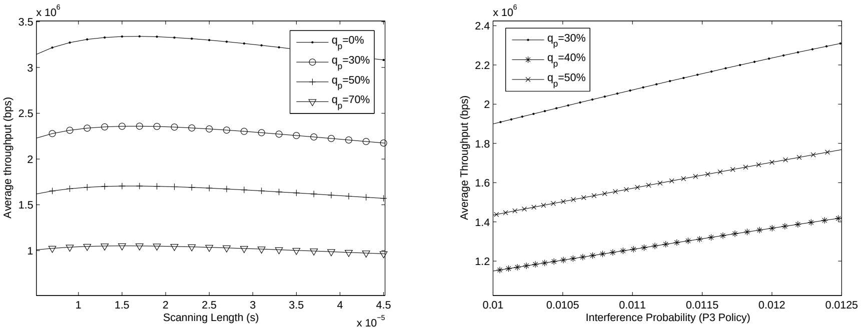


(a)

(b)

(c)

Figure 2.6: QoS assessment for 3 PU channels and 20 SU users: (a) Analytical throughput of OSA network as a function of DATA packet length for different levels of PU activity q_{cc} on CC for DCC MAC; (b) Simulated impact of PU channel occupancy rates for four different classes of OSA MACs in terms of delay and (c) throughput; dashed and solid lines represent different PU packet sizes (see text for more explanation).



(a) (b)

Figure 2.7: QoS assessment for 3 PU channels and 20 SU users: (a) OSA network analytical throughput as a function of scanning length for DCC MAC; (b) Analytical relation between level of interference to PU and a SU network throughput for DCC MAC. Throughput and interference have been computed as a function of scanning length varying from 1 to 45 μs (resulting in decrease of probability of false alarm from 0.23 to 0.024 and increase in probability of detection from 0.82 to 0.92) and three different levels of PU activities q_p on all channels.

2.4.4 Radio Frequency Front-Ends

The exact multichannel MAC operation and scanning implementation degrees of freedom depend on the number of front-ends that are available in each SU node. Indeed, when multiple RFEs are available, it is possible to use multiple channels simultaneously for transmission. Or, alternatively, spare RFEs can be used for scanning only, decreasing the impact of scanning on the network throughput. When only one RFE is available, sensing and communication should be split in time. Of course, increasing the number of RFEs increases the reliability of the system and decreases delay, but simultaneously increases the total cost. Typically this number varies from 1 to 3, see Table 2.1. In this chapter, we assume 2 RFEs for the DCC and a single RFE for the other MAC types.

2.4.5 Interference Management Policies

Since it is impossible to detect PU presence with certainty, harmful interference to the PU cannot be avoided. The maximum level of interference is typically specified through Interference Policies (IPs), i.e., policies that define how SUs can behave in certain PU bands while maintaining the QoS requirements of the PU. The more relaxed the IPs are, the better the SU can take advantage of spectrum opportunities. In other words, policies are rules that determine the trade-off between SU and PU QoS. Defining such IPs is however a very difficult task. In this chapter we want to see how much a SU could benefit from more relaxed PU policies.

From our literature search (Table 2.1), we can enlist three major policy classes for OSA networks:

P1 *Time based*: these policies define time metrics that regulate SU transmissions. An example metric is the Evacuation Time that defines how fast a SU should vacate a channel after a PU is detected.

P2 *Power based*: these policies define power limits that each SU needs to take into account when using PU channels. Example metrics are maximum (peak) power, power mask, and average transmit power.

P3 *Collision based*: these policies are defined at MAC layer, usually assuming packet based transmissions. They define collision probability limits, bounding the probabilities that a SU packet will harm a PU packet.

Depending on the PU system, one of these policies is most appropriate, e.g., the policy P3 can only be applied to packet-based networks. Also, a given policy can often be described, or implemented, differently. The exact description often significantly impacts the usability and cost of the SU network. For example, policy P2 can also be defined as a maximum distance between PU and SU, which requires the OSA network to embed expensive localization capabilities. We note that the definition of policies for OSA networks is a very hard problem and is an ongoing topic of research.

Next to the policy format, its level of PU protection can be too restraining. For a given policy (we use the P3 policy since we assume both SU and PU networks are packet based), we investigate what QoS the SU can achieve (Fig. 2.7(b)). The probability of collision with a PU packet and the SU throughput have both been computed as function of the scanning duration (and hence scanning quality). A stricter collision constraint is only achieved with an improved detection performance, requiring the SU to scan very long. When the PU constraint is relaxed, the SU can scan shorter, resulting in a throughput improvement of the SU.

The P3 policy, avoiding collisions with PU, can be implemented using a listen-before-send scanning. Since the PU does not scan for the SU presence, it is however possible that the PU will start a new packet during the SU transmission. This can only be avoided by assuming small SU packets, since it is not realistic to assume any synchronization between the PU and SU network. Assuming such synchronization is however very convenient for analysis [24], and we have also assumed such synchronization in our models. The only OSA MAC that can actually assume such synchronization is AS-MAC [19], which was specifically designed for operation on GSM channels, where slot boundaries can be captured easily. In general, since it is very hard to preserve SU/PU synchronization, certain policies, like P1 and P3 have to be defined very carefully.

2.4.6 Summary of the Results

In the previous section, many features that are important for OSA MAC design have been listed and discussed first by means of reviewing the proposals found in literature and also by quantitatively assessing the impact of some features on PU or SU QoS. A first conclusion is that most of the proposed solutions do not cover many of the crucial elements of a proper OSA MAC protocol design. Indeed, since the operating conditions of OSA networks are typically unknown during the design time phase, the bootstrapping procedure to setup the network before communication is very important. However, it is omitted in many of the protocol designs. In the case of OSA networking, this bootstrapping cannot be considered to be a one-time effort at the start of the communication network, so it is crucial to make it as efficient as possible and embed it in the MAC protocol design. Also, the required scanning for the presence of the PU is sometimes omitted in the protocol design or performance analysis. More importantly, the specification of policies to regulate the coexistence with PUs is often described very vaguely or even fully omitted. It can be concluded that although many individual contributions can be found, it is important to assess how these subtasks can be integrated together into a complete solution to be able to fully assess the expected QoS of OSA networks.

Often, the solutions proposed for the subtasks are suboptimal. In this chapter focus has been on the organization of the Control Channel since this is a very important aspect of multichannel OSA networking. Although the solutions proposed in literature always assume the availability of a fixed channel for control information exchange (for DCC this channel is only for control, in SFCC the channel is also used for data), we show that this is not necessarily optimal. Indeed, especially in the case that there are a lot of possible channels to use, a fixed control channel easily becomes the bottleneck. Also, when no channel can be assumed to be free from PU activity, it is best to spread the control exchanges over different channels as much as possible. As a result, we show that the MRCC actually outperforms DCC, HCC and SFCC over a broad range of PU traffic conditions.

Finally, we want to emphasize that no solutions found so far in literature assess the QoS given to the secondary network in detail. This is however a very crucial study since the introduction of OSA networks only makes sense if a sufficient level of QoS can be expected. In this chapter, we attempt to study the delay and throughput performance of a broad range of OSA designs as function of PU activity. Also, we assess the fundamental trade-off between PU QoS and SU QoS. The more freedom is given to the SU to access the channel, the more capacity it can use and the better its performance. However, more freedom to the SU means less guarantees for the PU and the success of OSA networking will depend on how well we can optimize this trade-off with a given policy.

2.5 Conclusions

In this chapter we have given important insights into the networking of Opportunistic Spectrum Access. Particularly we have focused on the performance of various flavors of Transport Control Protocols on Opportunistic Spectrum Access links and the performance of multichannel Medium Access Control protocols for Opportunistic Spectrum Access. The major conclusions can be summarized as follows:

- Hopping MACs not only increase the throughput of secondary networks, but also minimize the interference level induced to the primary users;

- Dedicated control channel MACs are not always the right solution for OSA networks–their performance depends strictly on the network setup like number of channels and users or offered load;

- For any TCP protocol sensing time (and introduced by it delay) is the most important factor responsible for throughput decrease of the OSA network. Errors introduced by sensing can be easily mitigated by the link layer retransmission mechanisms;

- Selective acknowledgments of TCP are a very efficient way of throughput increase for OSA networks;
- Length of the ON time has much more impact on the performance of TCP than the length of the OFF time, i.e. the bigger the disproportion between ON and OFF time, the worst performance of TCP protocol.

2.6 List of Relevant Publications by the Author

2.6.1 Magazines

- P. Pawełczak, S. Pollin, H.-S. W. So, A. Motamedi, A. Bahai, R. V. Prasad, and R. Hekmat, "Quality of service of opportunistic spectrum access: A medium access control approach," *IEEE Wireless Commun. Mag.*, 2008, conditionally accepted.
- R. V. Prasad, P. Pawełczak, J. Hoffmeyer, and S. Berger, "Cognitive functionality in next generation wireless networks: Standardization efforts," *IEEE Commun. Mag.*, vol. 46, no. 5, pp. 72–78, Apr. 2007.

2.6.2 Peer-Reviewed Conference Proceedings

- P. Pawełczak, S. Pollin, H.-S. W. So, A. Motamedi, A. Bahai, R. V. Prasad, and R. Hekmat, "State of the art in opportunistic spectrum access medium access control design," in *Proc. ICST/IEEE CrownCom'08*, Singapore, May 15–17, 2008, invited Paper.
- F. E. Visser, G. J. Janssen, and P. Pawełczak, "Multinode spectrum sensing based on energy detection for dynamic spectrum access," in *Proc. IEEE VTC'08-Spring*, Singapore, May 11-14, 2008.
- P. Pawełczak, R. V. Prasad, and R. Hekmat, "Opportunistic spectrum multichannel OFDMA," in *Proc. IEEE ICC'07*, Glasgow, Scotland, June 24-28, 2007.
- A. M. R. Slingerland, P. Pawełczak, A. Lo, R. V. Prasad, and R. Hekmat, "Performance of transport control protocol over dynamic spectrum access links," in *Proc. IEEE DySPAN'07*, Dublin, Ireland, Apr. 17–20, 2007.
- P. Pawełczak, G. Janssen, and R. V. Prasad, "Performance measures of dynamic spectrum access networks," in *Proc. IEEE GLOBECOM'06*, San Francisco, CA, USA, 27 Nov. - 1 Dec. 2006.
- P. Pawełczak, R. V. Prasad, H. Nikookar, and I. Niemegeers, "Performance analysis of periodical spectrum sensing for dynamic spectrum access networks," in *Proc. AWiN (IEEE GLOBECOM'05 Workshop)*, St. Louis, MO, USA, Nov. 28, 2005.

- P. Pawełczak, R. V. Prasad, L. Xia, and I. Niemegeers, “Cognitive radio emergency networks–requirements and design,” in *Proc. IEEE DySPAN’05*, Baltimore, MA, USA, Nov. 8–11, 2005.

2.6.3 Submissions

- P. Pawełczak, S. Pollin, H.-S. W. So, A. Bahai, R. V. Prasad, and R. Hekmat, “Performance analysis of multichannel medium access control algorithms for opportunistic spectrum access,” Mar. 20, 2008, submitted, *IEEE Trans. Veh. Technol.*

- P. Pawełczak, S. Pollin, H.-S. W. So, A. Bahai, R. V. Prasad, and R. Hekmat, “Comparison of opportunistic spectrum multichannel medium access control protocols,” Mar. 15, 2008, submitted, IEEE GLOBECOM’08.

- F. Granelli, P. Pawełczak, K. S. R.V. Prasad, R. Chandramouli, J. A. Hoffmeyer, and S. Berger, “Standardization and research in cognitive and dynamic spectrum access networks: IEEE SCC 41 efforts and open issues,” Jan. 22, 2008, submitted, *IEEE Commun. Mag.*

Chapter 3

Adaptive Baseband Processing for Adaptive Ad-hoc Freeband (AAF) Cognitive Radio System

by Ibrahim Budiarjo

3.1 Abstract

The growing demand on wireless communication systems that provide high data rates has implied the necessity of flexible and efficient use of the spectrum resource. While the spectrum is a scarce commodity, a vast majority of the available spectral resources has already been licensed, thus there is little or no room to add any new services, unless some of the existing licenses are discontinued.

Studies and measurements have shown that vast portions of the licensed spectra are rarely used which has raised the idea of Cognitive Radio (CR) where intelligence and learning processes aid the radio system in accessing the spectrum efficiently. A Cognitive Radio (CR) system is capable of learning by understanding (intelligent) in order to reach its goals. The scope of our research is the dynamic spectrum access aspect of Cognitive Radio, where the challenging problem on this aspect is the coexistence of the CR based rental (i.e., unlicensed) users with the licensed system while the target quality of service can still be achieved. It is expected that the rental users are allowed to transmit and receive data over portions of spectra when primary (i.e., licensed) users are inactive. This is done in a way that the rental users (RUs) are invisible to the licensed users (LUs). In such a setting LUs are ordinary mobile terminals and their associated base stations. They thus do not possess CR capabilities. The RUs, on the other hand, should possess the intelligence of sensing the spectrum and use whatever resources are available when they need them. At the same time, the RUs should give up the spectrum when a LU begins transmission.

In this chapter we present our research related to the techniques in realizing the coexistence between the CR based rental system and the licensed system.

3.2 Introduction

Orthogonal frequency division multiplexing (OFDM) has recently been introduced as a strong multicarrier modulation scheme and candidate to be applied in CR, due to its capability in notching part of its carriers in order to have the flexibility of spectrum access. Single carrier approach with transform domain communications systems (TDCS) has also been initiated. The transformation of both approaches (multicarrier and single carrier) can be applied by the aid of Fourier or wavelet basis functions. In the following sections we will describe the research that has been conducted in IRCTR TU DELFT with regards to OFDM and TDCS in the framework of AAF Cognitive Radio.

This chapter begins with OFDM on section 3.3, single carrier modulation TDCS is described in section 3.4, the multicarrier wavelet packet modulation is discussed in section 3.5, the work on MIMO V-BLAST as an effort to improve the BER and to increase the bit rate of the CR which is overviewed in section 3.6, and the demonstrator for IRCTR-AAF is presented in section 3.7.

3.3 OFDM

Modulation is a very important component in any transmission system. A modulation scheme takes information in the form of bits and modulates the transmitter carrier in such a way that a receiver is able to extract the information (bits) from the modulated carrier. Ideally a modulation scheme that is chosen such that it can transmit an infinite number of bits in an infinitely small time using an infinitely narrow bandwidth. But this is impossible. As explained earlier the radio spectrum is scarce, that is why chosen modulation schemes have to have a high spectral efficiency. The spectral efficiency is a measure that expresses the modulation schemes ability to transmit at a rate R ($\frac{bits}{s}$) within a channel of bandwidth W (Hz). The spectral efficiency ($\frac{bits/s}{Hz}$) is determined by calculating $\frac{R}{W}$. The more spectrally efficient a modulation scheme is, the less spectrum will be required in order to communicate at certain data rate. The spectral efficiency that can be attained by a modulation scheme depends on the noise and propagation conditions. The noise and propagation conditions may vary due to the atmospheric circumstances. The modulation scheme must therefore be able to adapt to these varying noise and propagation conditions. Therefore, the second requirement is adaptivity. A third requirement comes from the fact that there are multiple nodes, or users in a network. The modulation scheme must support a multiple users model.

The fourth requirement is that the modulation scheme must be resistant to frequency offset and Doppler shifting of the carrier frequency. Frequency offsets arise due to the inaccuracy in the frequency determining components of the carrier oscillator in the transmitter. Technically these are offset problems. While there are many modulation schemes that will fit all the above requirements, a reasonable choice is Orthogonal Frequency Division Multiplexing, or OFDM.

Orthogonal Frequency Division Multiplexing belongs to the multi-carrier modulation family. The modulator divides the incoming data across multiple carriers that are modulated at a lower rate compared to a single carrier modulation system. An OFDM symbol therefore consists of multiple carriers that can be individually modulated in amplitude and phase. The way the OFDM carriers are modulated can vary from BPSK for robust low rate communication to more complex QAM modulation for higher throughput. Therefore, OFDM is able to adapt to different noise conditions. It is also spectrally efficient and it is able to support multiple users through Frequency Division Multiplexing (FDM) and Time Division Multiplexing (TDM) techniques. Also, OFDM is very resistant to multi-path interference and when the Doppler shift is smaller than half of carrier spacing, this shift can be easily compensated.

In an OFDM system, the symbol rate of each carrier is much less than the serial source symbol rate. Therefore the effects of delay spread (i.e. ISI) significantly decrease, reducing the complexity of the equalizer at the receiver. Adding a guard interval between symbols will reduce the ISI. Increasing the symbol duration by a factor of N will make the ISI becomes negligible and it also reduces the fraction of the guard interval compared to the symbol duration.

The carriers of OFDM are overlapping, but between different modulated carriers they are mutually orthogonal. The carrier spacing is carefully selected such that each sub carrier is orthogonal to all other carriers. This can be achieved by making the carrier spacing equal to the reciprocal of the symbol period. Because the carriers are orthogonal, the spectrum of a carrier has nulls at the center frequencies of all other carriers, as shown in Figure 3.1.

During one OFDM symbol, the channel needs to be time-invariant and the fading per carrier can be considered as flat. Hence, the OFDM symbol duration should be smaller than the coherence time $(\Delta t)_c$ of the channel, and the carrier spacing should be smaller than the coherence bandwidth $(\Delta f)_c$ of the channel. By fulfilling these conditions, the realization of a low-complexity receivers is possible.

The OFDM transmission system is depicted in Figure 3.2. A data source sequence denoted by S_n consisting of N_c complex-valued symbols ($n = 1, ..., N_c$), is transmitted in parallel on N_c carriers. These symbols are interpreted as values in the frequency domain. The sequence S_n of rate$\frac{1}{T}$ are mapped onto N_c parallel substreams.

A typical condition that occurs in OFDM is that bit errors are concentrated in a set of severely faded carriers, while in the rest of OFDM spectrum often no bit errors are observed. The idea arises to have an adaptive OFDM system where the carriers that will exhibit high bit error probabilities are excluded. It will improve the overall BER with an exchange of a slight loss of the system throughput. The potential loss of throughput due to the exclusion of faded carriers is counteracted by employing higher order modulation modes on carriers which have a very small BER. In the framework of the AAF project we have investigated the application of adaptive bit loading as an adaptive OFDM method in anticipating the condition priorly described. As adaptive bit loading requires information about the channel state condition, we have studied the application of 2x 1-D Wiener filter channel estimation for providing the adaptive bit loading with the estimated channel transfer functions (ctfs).

In the effort of having coexistence between cognitive radio system (rental user) and the licensed user, OFDM is a strong modulation scheme candidate to be applied in cognitive radio system. OFDM has the capability of notching parts of its carriers which coincidentally are within the region of the licensed user's band. The method has been introduced and is called as ”Spectrum Pooling”. We have evaluated the application of spectrum pooling with OFDM, by deactivating the carriers on and adjacent to the licensed user's band combined with windowing in the effort of further interference reduction to the licensed system. In the case too many carriers need to be deacti-

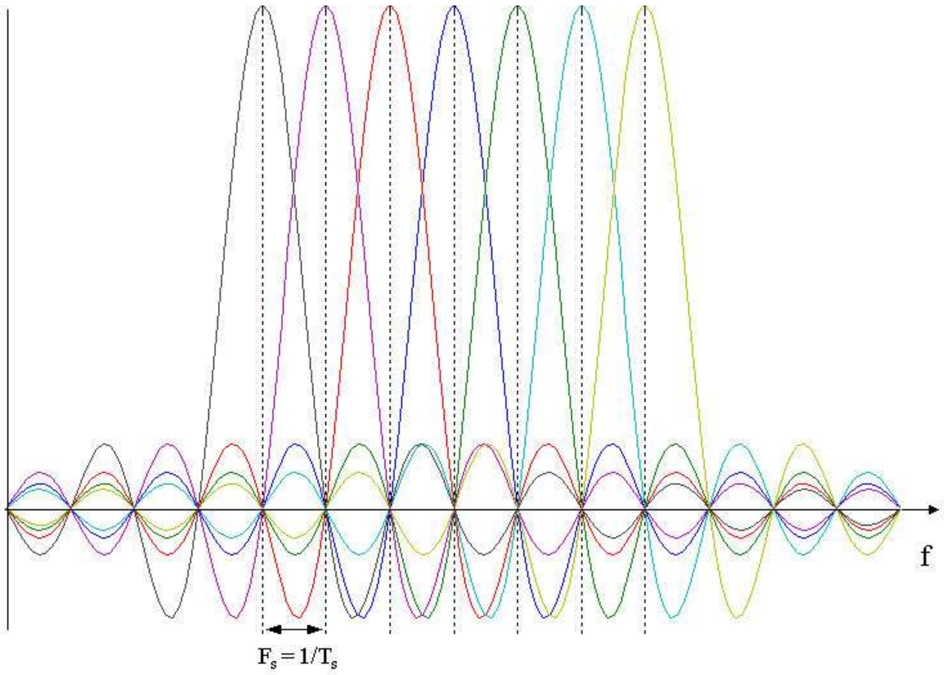


Figure 3.1: Spectrum of an OFDM signal.

vated, frequency hopping is applied. According to the spectrum sensing information, a new spectrum region that can fulfill the bit rate requirement is selected for transmission. The four topics : adaptive bit loading, Wiener filter channel estimation, spectrum pooling and frequency hopping will be described in the following paragraphs.

3.3.1 Adaptive Bit Loading

Adaptive bit loading is a method of allocating bits to carriers adaptively and where the carrier modulation modes are selected accordingly. The bits can be allocated carrier-wise or per subband (group of carriers). As a consequence of applying adaptive bit loading, the receiver needs information about the number of bits allocated on each carrier in order to decode the signal properly. This information can be derived blindly by examining the received signal or from dedicated signaling symbols. An example of the signaling symbol is depicted in Fig. 3.3, where the allowable constellation size is restricted to the set of 0,2,4,16 and 64. The inner rectangle which is the decision area for 0 modulation mode (no symbol is transmitted) can be made smaller up to a distance of about $3d/4$ from the other reference symbols, where d is the distance between outer points and the no symbol point (zero) shown in Fig. 3.3. This is due to the fact that a decision error (e.g no symbol transmitted but BPSK or other modulation mode is decided) is only caused by the noise (no contribution from channel fading). Next subsections describe several available bit loading algorithms.

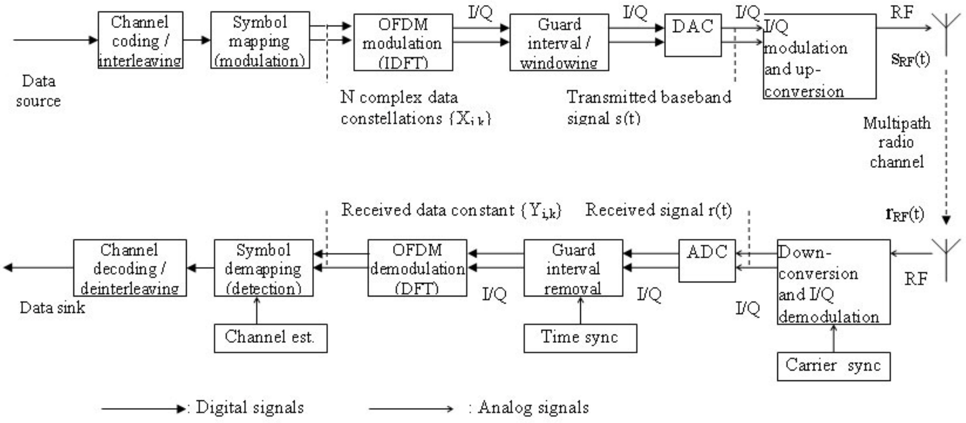

Channel coding / interleaving
Symbol mapping (modulation)
OFDM modulation (IDFT)
I/Q
Guard interval / windowing
I/Q
DAC
I/Q
I/Q modulation and up-conversion
RF
$s_{RF}(t)$
Data source
N complex data constellations $\{X_{i,k}\}$
Transmitted baseband signal s(t)
Multipath radio channel
$r_{RF}(t)$
Received data constant $\{Y_{i,k}\}$
Received signal r(t)
Channel decoding / deinterleaving
Symbol demapping (detection)
OFDM demodulation (DFT)
Guard interval removal
ADC
Down-conversion and I/Q demodulation
RF
I/Q
I/Q
I/Q
Data sink
Channel est.
Time sync
Carrier sync
: Digital signals
: Analog signals

Figure 3.2: OFDM transmission system [28].

Chow Algorithm

The algorithm is intended to maximize the allowable amount of noise while the minimum desired BER is still achieved. The equation for the optimum bit allocation of this algorithm is [29],

$$R_n = log_2\left[1 + \frac{S_n}{N_n\big(\Gamma + \lambda_{margin}\big)}\right] \tag{3.1}$$

where S_n is the signal power, N_n is noise variance and R_n is the transmission rate on carrier n. The noise margin γ_{margin} is the additional amount of noise (in dB) that the system can tolerate, and is iteratively calculated based on the gap between the total allocated bits from calculating R_n and the target total bits per OFDM symbol [29]. The SNR gap Γ is a constant which estimates the difference between the theoretical channel capacity and the actual capacity usage by the transmission scheme. Bits are only allocated to carriers with positive and nonzero value of R_n, and it is quantized into integer number.

The quantization error is the difference between R_n and its quantized value. If the total number of bits is equal to the target total bits per OFDM symbol then the calculation stops, otherwise substraction or addition of bits is conducted, by applying it first to the carriers which have the smallest quantization error, and so on until the target total number of bits is satisfied.

Fischer-Huber algorithm

This algorithm tries to make all carriers having the same SNR. Since the overall error rate is dominated by the carrier with highest error rate, the rate and power are distributed in such a way that the error probability on each subchannel is minimized [30]. The optimum bit allocation is based on the following equation [31]:

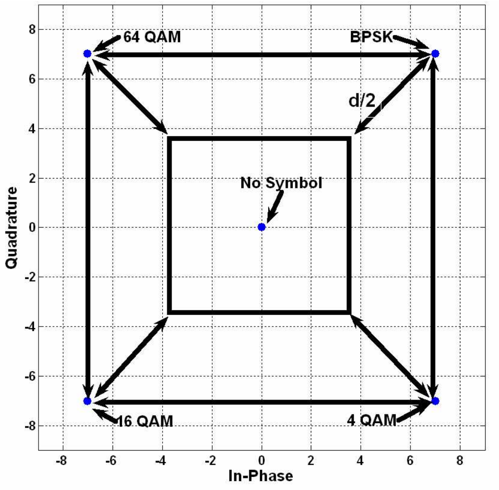


Figure 3.3: An example of signaling symbol points for adaptive bit loading with allowable M=0, 2, 4, 16, and 64

$$R_n = \frac{1}{N_c}\left[R_T + log_2\left(\prod_{k=0}^{K-1}\frac{N_k}{|H_k|^2}\right) - N_c log_2\left(\frac{N_n}{|H_n|^2}\right)\right] \tag{3.2}$$

where N_c is the number of carriers, R_T is the target bits per OFDM symbol, H_n is the complex channel gain and N_n is the noise variance on carrier n. R_n is quantized into an integer number. No bits are allocated to carrier n if the value of R_n is zero or negative. After several iterations and all values of quantized R_n are nonnegative and if the summation of bits is not equal to R_T, then addition or subtraction of bits (like in Chow algorithm) is applied.

Group wise loading with modified Fischer-Huber algorithm

In this method, the Fischer - Huber algorithm is modified by allocating bits per groups of carriers. The optimum equation becomes as follows [32],

$$R_i = \frac{R_T}{N'_B} log_2\left(\frac{\left(|H_i|^2\right)^{N'_B}}{\prod_{k\in\psi}|H_k|^2}\right) \tag{3.3}$$

where H_i denotes the channel gain of the i-th subband, which is the average value of the channel gains of the carriers affiliated to that subband. ψ contains the set of indices of the active subbands (carriers with nonnegative value of R_i), N'_B is the number of indices in the set. The grouping of carriers can be based on their indices (successive subband grouping) or based on the order of the channel gain (sorted subband grouping). Likewise in the Chow algorithm, quantization of R_i is applied, and if the target rate is not satisfied then subtraction or addition of bits is conducted.

Simple blockwise loading algorithm (SBLA)

Originally in [33] the bit loading is applied groupwise based on the indices of carriers. The subband's SNR is derived from the average SNR of the individual carriers in the group. The choice of modulation mode is determined with the help of the SNR grid in Fig. 3.4. The center of the grid (8-QAM region) is placed on the average SNR of the total carriers. The grid is given in [34].

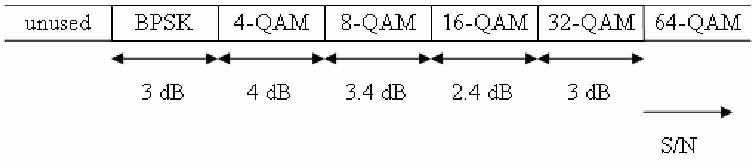


Figure 3.4: SNR gap for determining modulation modes [34]

The grouping based on the order of the channel gain will also be evaluated. The grid can be used for bit loading algorithm carrier-wise.

3.3.2 Wiener Filter Channel Estimation

The Wiener Filter optimizes channel estimation by minimizing the mean square error between the true channel gain and the estimated channel gain. Two cascaded 1-D Wiener Filtering perform similar to 2-D Filtering, but 2x 1-D Filter gives less computational complexity [35], therefore it is preferable to use 2x 1-D in our system.

In designing the filter for channel estimation, information about the correlation function of the channel is required. The correlation function of the channel in time is derived from the inverse Fourier transform of the Doppler power density, and the correlation function of the channel in frequency is derived from the Fourier transform of the power delay profile.

The pilots are arranged in a frame in such a way that it will fulfill the sampling theorem [35]:

$$\begin{aligned} d_t &\leq \frac{1}{2 f_{D_{max}} T_s'} \\ d_f &\leq \frac{1}{\tau_{max} \Delta F} \end{aligned} \tag{3.4}$$

where $f_{D_{max}}$ is the estimated maximum Doppler frequency, T_s' is OFDM symbol time(incl. guard interval), d_t is the distance between pilots in time direction, τ_{max} is the estimated maximum delay of the channel, Δ_F is the carrier spacing and d_f is the distance between pilots in frequency direction.

This balanced design will be optimal if the power delay spectrum and power Doppler spectrum have the same shape. The rectangular form of power delay spectrum and power Doppler spectrum produce the frequency ($\theta_{\triangle F}$) and time ($\theta_{\triangle_t}$) correlation function as [36]:

$$
\begin{aligned}
\theta_{\Delta F}(l-l") &= \frac{sin(\pi\tau_{max}\Delta F(l-l"))}{2\pi\tau_{max}\Delta F(l-l")} \\
\theta_{\Delta t}(k-k") &= \frac{sin(2\pi f_{D_{max}}T_s'(k-k"))}{2\pi f_{D_{max}}T_s'(k-k")}
\end{aligned}
\tag{3.5}
$$

The filter coefficients ($\boldsymbol{\omega}$) are generated from the multiplication of the channel correlation matrix ($\boldsymbol{\theta}$) and the inverse of the autocorrelation matrix of the channel in pilot positions ($\boldsymbol{\Phi}$), which is expressed as [35] :

$$
\boldsymbol{\omega}^T = \boldsymbol{\theta}\boldsymbol{\Phi}^{-1} \tag{3.6}
$$

the autocorrelation (in time or frequency) between the channel in pilot positions is defined as [35]:

$$
\phi_{n'-n"} = \theta_{n'-n"} + \frac{1}{\gamma_c} \tag{3.7}
$$

where γ_c is the average SNR on all carriers.

3.3.3 Spectrum Pooling

A strategy which is called *spectrum pooling* is proposed in [37], [38] where the public access is enabled to these bands without giving significant interference to the actual license owners. Spectrum pooling enables public access to spectral ranges of licensed frequency bands which are seldom used by overlaying a secondary rental user (RU) to an existing licensed user (LU). The LUs are radio systems authenticated to operate under licensed spectral bands. The RUs are intelligent CR systems that actively scan the landscape and opportunistically utilize available and unused resources. The RU relinquishes control over the resources (here spectrum) as and when the LU starts using them. To identify and utilize unused bands, the frequency bands of various radio systems (including licensed and rental users) are combined to obtain a common spectral pool. Cohabitation of LU and RU systems is actualized by shaping the transmission waveform of the RU in a way that it utilizes the unoccupied time-frequency gaps of the LU.

The pioneering work on the subject was conducted by Jondral et al [37] who devised a spectrum pooling scheme using multi-carrier modulation (MCM) where individual carriers on and adjacent to the occupied licensed user spectrum are deactivated. Fig. 3.5 illustrates a typical scenario. The combination of adaptive MCM and spectrum pooling can serve as a robust method to achieve a good quality of communication and efficient use of the spectrum.

An obvious MCM candidate for CR systems is OFDM. With advantages of flexibility, easiness of implementation and elegance in operation, OFDM is naturally suitable for CR system design.

The short-coming of the OFDM solution is the large sidelobes of the frequency response of filters that characterize the channel associated with each of carrier. The sidelobes amplitude of the Fourier based OFDM can be observed from its power spectral density (PSD),

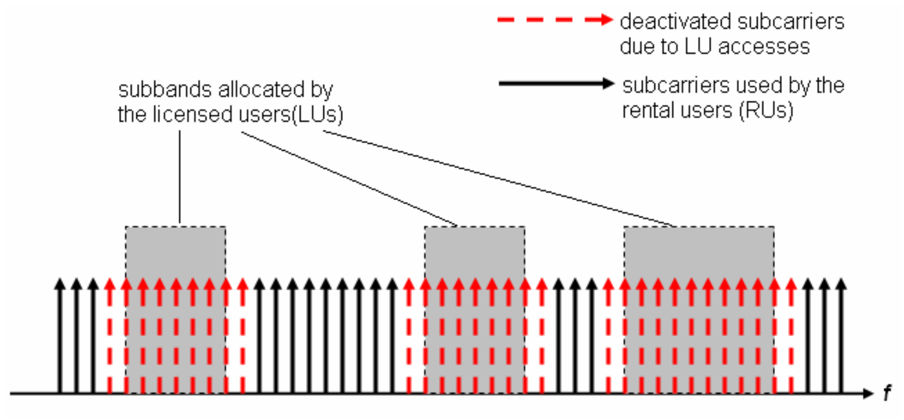

deactivated subcarriers due to LU accesses
subcarriers used by the rental users (RUs)
subbands allocated by the licensed users(LUs)
f

Figure 3.5: Illustration of Spectrum pooling block diagram.

$$PSD(f) = \frac{1}{N_{FFT}} \left| \sum_{m=0}^{N_{FFT}-1} \sqrt{p_m} A_m \int_{-(1+\alpha)\frac{T_u}{2}}^{(1+\alpha)\frac{T_u}{2}} g(t) e^{-j2\pi(f-f_m)t} dt \right|^2 \tag{3.8}$$

where N_{FFT} is the number of points of the FFT, T_u is the useful signal duration, $g(t)$ is the window function, α is the roll off factor of the window, f_m is the frequency on carrier m. p_m and A_m are the allocated power and the symbol from QAM or PSK mapping on carrier m, respectively. The large side lobes result in significant interference among the carriers that originate from different RUs and between LUs and RUs. Equation 3.8 shows that parameters like allocated power (p_m), symbol (A_m) amplitude and window ($g(t)$) can be set to resolve this problem. Other elements which can be varied to improve the performance of RUs are the channel coding parameters and carrier frequency. The utilization of the parameters to reduce the PSD in the LU's band and elements to improve the quality of service (QoS) of RUs will be further explored in the next following sub subsections.

Adaptive Power Allocation

The allocated power on the carriers adjacent to the LU's band can be set so the sidelobes becoming low, while no power is allocated to LU's band, hence the unallocated power can be distributed to other carriers.

The distribution can be applied equally or adaptively to the other carriers. Other options are to distribute the power only to data, or only pilots, or only to pilots/data which are not too close but also not too far from the LU's band with the intention not to largely increase the sidelobe to the LU's band but improve the signal to interference and noise ratio (SINR) of the pilots/data of RU on that position due to the possibility of the high sidelobes of LU to RU.

Adaptive power allocation is applied to the data or pilots based on the estimated channel gain. In [39] and [40] the power is allocated to each carrier with the objective of minimizing the overall BER. The allocated power to each carrier is derived from [39]:

$$
\begin{aligned}
P_m &= \frac{(CS-1)\sigma_m^2}{3\Phi_m} W(\Phi_m^2 \eta, \qquad m = 1, ..., N_{FFT} \\
& \quad subject \quad to \quad \sum_{m=1}^{N_{FFT}} P_m = P_T
\end{aligned} \tag{3.9}
$$

where CS is the constellation size of the symbol A_m in 3.8, σ_m^2 is the noise variance on carrier m, $W(.)$ is the Lambert function (i.e., inverse function of $y(w) = we^w$), $\Phi_m = |H_m|^2$ where H_m is the channel gain on carrier m, η is a constant and P_T is the total allocated power per OFDM symbol.

In [41] optimal power allocation by means of the steepest descent algorithm is introduced. In the beginning initial power to each carrier ($p_m(0)$) and initial step size parameter ($\mu(0)$) are set. The power is updated accordingly as [41]:

$$
p_m(i+1) = p_m(i) - \mu(i)\left(\frac{1}{N_{FFT}} \frac{d}{dp_m(i)} v\left(\frac{\Phi_m p_m(i)}{\sigma_m^2}\right)\right) + \lambda(i) \tag{3.10}
$$

where i is the iteration number, $v(.)$ is the BER function determined by the modulation mode used, and $\lambda(i)$ is determined by :

$$
\lambda(i) = \frac{1}{N_{FFT}^2} \sum_{m=1}^{N_{FFT}} \frac{d}{dp_m(i)} v\left(\frac{\Phi_m p_m(i)}{\sigma_m^2}\right) \tag{3.11}
$$

If all of the p_m values are positive then the new stepsize $\mu(i+1)$ is assumed as $\mu(i)$ and the power is further updated until the total power P_T constraint in 3.9 is fulfilled. Otherwise the stepsize on a specific carrier m with negative p_m is updated. The new μ_m becomes [41]:

$$
\mu_m(i) = \frac{1}{\frac{1}{N_{FFT}} \frac{d}{dp_m(i)} v\left(\frac{\Phi_m p_m(i)}{\sigma_m^2}\right) + \lambda(i)} \tag{3.12}
$$

and the new stepsize $\mu(i)$ becomes the multiplication of a scaling factor $\rho(0 < \rho < 1)$ with the smallest μ_m [41]. The power to those carriers (pm) is updated again according to 3.8 until all p_ms are nonnegative and the P_T constraint is achieved. A suboptimal power allocation is also introduced in [41] by utilizing the upper bound approximation of the BER function ($v()$) as:

$$
v\left(\frac{\Phi_m}{\sigma_m^2} p_m\right) \leq \frac{(\sqrt{CS}-1)}{\sqrt{CS}} log_2(\sqrt{CS}) e^{-\frac{3\Phi_m}{2(CS-1)\sigma_m^2} p_m} \tag{3.13}
$$

By applying the BER upper bound in 3.13 combined with the total power (P_T) constraint to the Lagrange optimization function, the power allocation becomes :

$$
p_m = \begin{cases} \frac{\lambda_0 \sigma_m^2}{\Phi_m} - \frac{2(CS-1)\sigma_m^2}{3\Phi_m} ln\left(\frac{\sigma_m^2}{\Phi_m}\right), & \frac{\Phi_m}{\sigma_m^2} \geq e^{-\frac{3\lambda_0}{2(CS-1)}} \\ 0, & \frac{\Phi_m}{\sigma_m^2} < e^{-\frac{3\lambda_0}{2(CS-1)}}. \end{cases} \tag{3.14}
$$

where λ_0 is derived from :

$$\lambda_0 = \frac{N_{FFT}\bar{P} + \frac{2(CS-1)}{3}\sum_{m\in S}\left(\frac{\sigma_m^2}{\Phi_m}\right)ln\left(\frac{\sigma_m^2}{\Phi_m}\right)}{\sum_{m\in S}\left(\frac{\sigma_m^2}{\Phi_m}\right)} \tag{3.15}$$

$\bar{P}$ is the average power and S is the set of carriers where its power is nonzero. In [42] the optimal and suboptimal power allocations are combined with Coded Weak Sub Carrier Excision (CWSE) method, where the weak carriers are excluded from transmission. Power allocation with the objective to maximize the bit rate subject to a finite total allocated power and minimizing the total power allocation subject to a finite target rate optimization problem is decribed in [43], while in [44] a look up table, to find the rate-SNR operating point, and a Lagrange multiplier bisection search method are applied to calculate the allocated power on each carrier.

Spectrum Shaping by Time Domain Windowing

In [37] an extension of each OFDM block with a long cyclic prefix and suffix samples and application of some windowing to reduce the side-lobes of the carrier channels are suggested. Obviously, this solution is at the cost of bandwidth loss because excessive time should be allocated to cyclic extensions that otherwise could be used for data transmission. Further in [37], interference reduction is extended by deactivating carriers located adjacent to the licensed system (LS) which provides a kind of shield to the LUs. Due to the channel propagation influence, OFDM bit errors are typically concentrated in a set of severely faded carriers. This fact indicates that there is possibility that the quality of service can not be maintained. Meanwhile, as mentioned before, the carriers resource is limited due to deactivation of RU carriers on the occupied LU band. Having adaptive OFDM in CR is thus a solution. The heavily faded and noisy carriers are excluded from transmission to improve the overall bit error rate (BER) of the system, and the loss of throughput is counteracted by applying higher order modulation modes to the carriers which have better signal to noise ratios (SNR) or BER.

A commonly used window type is the raised cosine window, which is defined as [37]:

$$g(t) = \begin{cases} \frac{1}{T_u} & 0 \le |t| \le \frac{T_u(1-\alpha)}{2}, \\ \frac{1}{2T_u}\left\{1 + cos\left[\frac{pi}{\alpha T_u}\left(|t| - \frac{T_u(1-\alpha)}{2}\right)\right]\right\} & \frac{T_u(1-\alpha)}{2} \le |t| \le \frac{T_u(1+\alpha)}{2}, \\ 0 & otherwise \end{cases} \tag{3.16}$$

where α is the roll off factor, and T_u is the useful OFDM symbol interval (without guard interval). The shape of the window in the roll-off region determines how rapid the OFDM spectrum goes down to zero. The PSD will hit zero on the frequencies in the interval of $\frac{1}{T_u}$. The Bartlett window is defined by equation 3.17, [45] while the Better than Raised Cosine (BTRC) Window which gives the lowest sidelobes among the three windows, is defined by equation 3.18 [46].

$$g(t) = \begin{cases} \frac{1}{T_u} & 0 \leq |t| \leq \frac{T_u(1-\alpha)}{2}, \\ \frac{1}{2T_u} - \frac{1}{T_u}\left[\frac{|t|}{\alpha T_u} - \frac{1}{2\alpha}\right] & \frac{T_u(1-\alpha)}{2} \leq |t| \leq \frac{T_u(1+\alpha)}{2}, \\ 0 & otherwise \end{cases} \tag{3.17}$$

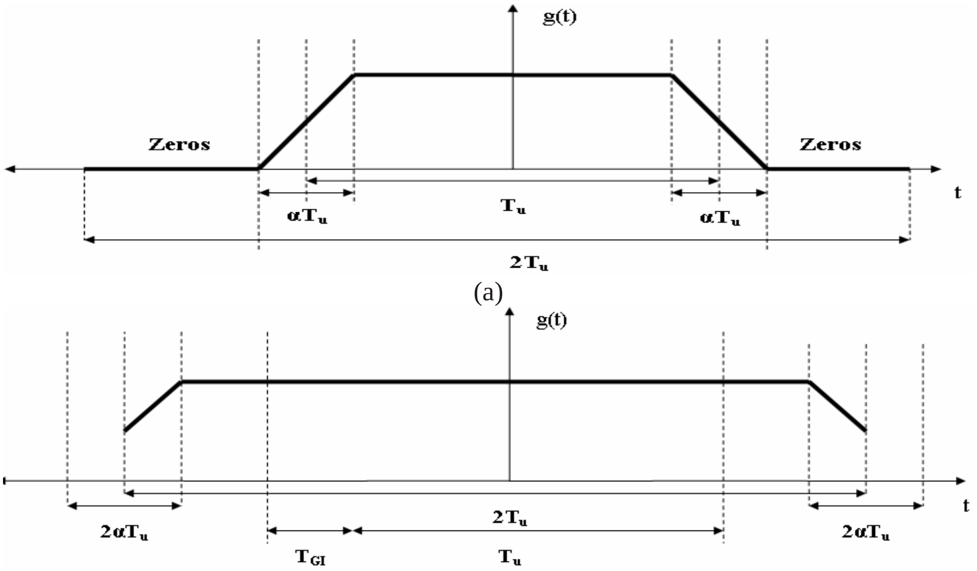
$$g(t) = \begin{cases} \frac{1}{T_u} & 0 \leq |t| \leq \frac{T_u(1-\alpha)}{2}, \\ \frac{1}{T_u} e^{-\frac{2ln(2)}{\alpha T_u}[|t| - \frac{T_u(1-\alpha)}{2}]} & \frac{T_u(1-\alpha)}{2} \leq |t| \leq \frac{T_u}{2}, \\ \frac{1}{T_u}\left\{1 - e^{-\frac{2ln(2)}{\alpha T_u}[\frac{T_u(1+\alpha)}{2} - |t|]}\right\} & \frac{T_u}{2} \leq |t| \leq \frac{T_u(1+\alpha)}{2}, \\ 0 & otherwise \end{cases} \tag{3.18}$$

A window with lower sidelobes than BTRC which is called the *Flipped- Inverse Hyperbolic Secant* (farcsech) window which is proposed in [47]. The window is designed according to 3.19,

$$g(t) = \begin{cases} \frac{1}{T_u} & 0 \leq |t| \leq \frac{T_u(1-\alpha)}{2}, \\ \frac{1}{T_u}\left[1 - \frac{1}{\alpha T_u \gamma} arcsech\left(\frac{1}{\alpha T_u}\left(\frac{T_u(1+\alpha)}{2} - |t|\right)\right)\right] & \frac{T_u(1-\alpha)}{2} \leq |t| \leq \frac{T_u}{2}, \\ \frac{1}{T_u}\left[\frac{1}{\alpha T_u \gamma} arcsech\left(\frac{1}{\alpha T_u}\left(|t| - \frac{T_u(1-\alpha)}{2}\right)\right)\right] & \frac{T_u}{2} \leq |t| \leq \frac{T_u(1+\alpha)}{2}, \\ 0 & otherwise \end{cases} \tag{3.19}$$

where, according to [48] the duration of an OFDM signal should be $2T_u$ in order to complete a total $2N_{FFT}$ samples to preserve the orthogonality, and zeros are added on the region outside the window $g(t)$. The window designs by equations 3.14- 3.17 affect the transmitted signal, as the consequence an error floor will be formed. Fig. 3.6(a) shows an example Bartlett window design according to 3.17 which affects the transmitted signal. It is emphasized in [49] that the applied window must not influence the signal during its effective period to avoid an error floor. In order to fulfill this requirement the window forms are expanded, i.e., the new T_u is becoming $2T_u$, and α is restricted in the range of $0 \leq \alpha \leq (1 - \frac{T_{GI}}{T_u})/2$, where T_{GI} is the guard interval duration. The orthogonality is preserved since the PSD hits zeros in the interval of $\frac{1}{2T_u}$ at the cost of longer duration of OFDM signal ($2T_u(\alpha + 1)$). As a solution the window is truncated to fit $2T_u$ OFDM duration, as depicted in Fig. 3.6(b). The orthogonality is preserved by applying the rectangular receiver filter with duration T_u , implemented by the DFT [49]. In this way there is more freedom in choosing a window in the region of $\frac{T_u(1-\alpha)}{2} \leq |t| \leq \frac{T_u(1+\alpha)}{2}$ as long as its PSD on the symbol boundaries is low, e.g Gaussian window or half sine window [50].

It should be emphasized that applying only windowing is not enough to reduce interference to the LU. A more powerful technique to suppress OFDM sidelobes is carrier deactivation. Deactivation of carriers adjacent to the licensed band provides flexible guard bands which will make the PSD sidelobes of RU's OFDM signal on the licensed band lower. Applying only one of the methods may not be sufficient, therefore combining both methods (windowing and adaptive deactivation of carriers adjacent



(a)

(b)

Figure 3.6: (a) Window design with influence to the transmitted signal (b) Window design without influence to the transmitted signal.

to licensed bands) is suggested to make the PSD sidelobes even lower in the region of LU's band. The loss of throughput due to carrier deactivation and long OFDM symbol duration can be compensated by applying adaptive bit allocation. Recently in [51] overlapped OFDM symbol transmission with long symbol duration is proposed to counteract the throughput loss. The scheme is described in Fig. 3.7.

The delay between one OFDM symbol to another should be designed in such a way that no intersymbol interference occurs, which means the next OFDM symbol should start after the end of the previous useful data part of an OFDM symbol. If zeros are inserted at the prefix of suffix (outside the T_{GI} and T_u area as depicted in Fig. 3.6(b)), the useful data of the next OFDM symbol should start after the last useful data of the previous OFDM symbol.

Optimum Pilot Pattern for OFDM Based Cognitive Radio

The main concern in estimating the channel by using pilot symbols in the context of the Cognitive Radio environment is where to place the pilot symbols if there is a frequency band that is occupied by the Licensed User. The main requirement of the pilot pattern design is that the designed pilot pattern will still be able to estimate the channel optimally with the presence of the Licensed User, meanwhile the performance of Licensed User in its spectrum band is not deteriorated by the presence of these pilot symbols. It is assumed that the transmitter and receiver of our system have the knowledge of the presence of Licensed User via the information obtained from spectrum sensing; our receiver and transmitter receive the information about which frequency bands are occupied by the Licensed User or in the context of OFDM modulation, which carriers

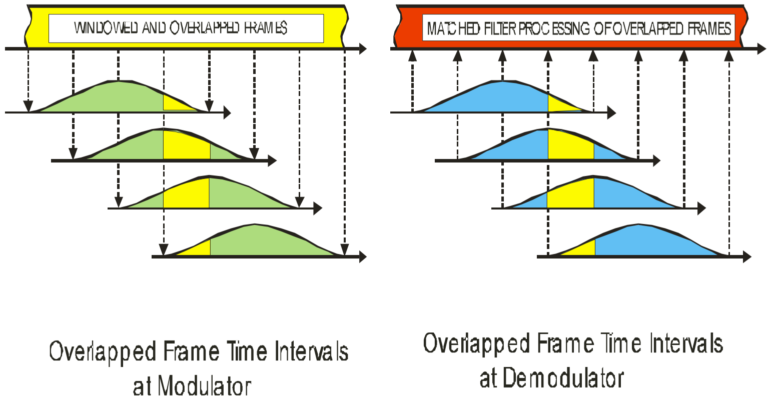

WINDOWED AND OVERLAPPED FRAMES
MATCHED FILTER PROCESSING OF OVERLAPPED FRAMES
Overlapped Frame Time Intervals at Modulator
Overlapped Frame Time Intervals at Demodulator

Figure 3.7: Overlapped OFDM symbol with zero ISI.

are to be deactivated in order not to interfere with the Licensed User. Most estimation degradation occurs as deactivated band edges are on the pilot positions, because the channel estimates on the position after these pilots will depend on the pilots before/after itself which are far in distance and less correlated. In order to counteract this problem we shift the pilot that supposed to be on the edge of the deactivated carriers one carrier position before/after [52]. By doing this we can still be able to sample the channel up to the border with the LU's band at the cost of losing some throughput. The proposed pilot pattern is depicted in Fig. 3.8. The hexagonal sampling pattern is considered to be the most efficient two-dimensional sampling scheme. This scheme requires 13.4% fewer samples than a rectangular sampling to represent the same circularly bandlimited continuous signal [53]. The complete design requirements for the hexagonal pilot pattern are described in [54]. Likewise in rectangular pilot pattern, in the presence of LU access, the shifting of the pilot that supposed to be on the edge of the deactivated carriers to one carrier position before/after (on the data position) is applied. The scheme is depicted in Fig. 3.9.

Further we have developed a concept of having virtual pilots to aid the channel estimation using a Wiener filter with a hexagonal pilot pattern. The virtual pilots simplify the implementation of Wiener filtering by reducing the set of filters required compared to hexagonal pilot pattern without virtual pilots. Fig 3.10 describes the Hexagonal pilot pattern with virtual pilots. At the receiver, the virtual pilots on the deactivated carriers position are estimated by extrapolating linearly the initial channel estimates on the position close to the virtual pilots. Further the virtual pilots between pairs of pilots are derived from linear interpolation of the pairs of the initial channel estimates on pilot positions combined with a decision directed method. As the results show in Fig. 3.10, the pattern is becoming the regular rectangular pattern, hence only one set of filter coefficients is required for channel estimation in the frequency direction. The noisy initial channel estimates can be further smoothed by applying the decision directed method. The process is shown in Fig. 3.11. X_v, Y_v, H_v, and H'_v are the symbol before demapping, the received symbol, the initial channel estimate from interpolation and the final initial channel estimate on the virtual pilot position after the decision directed

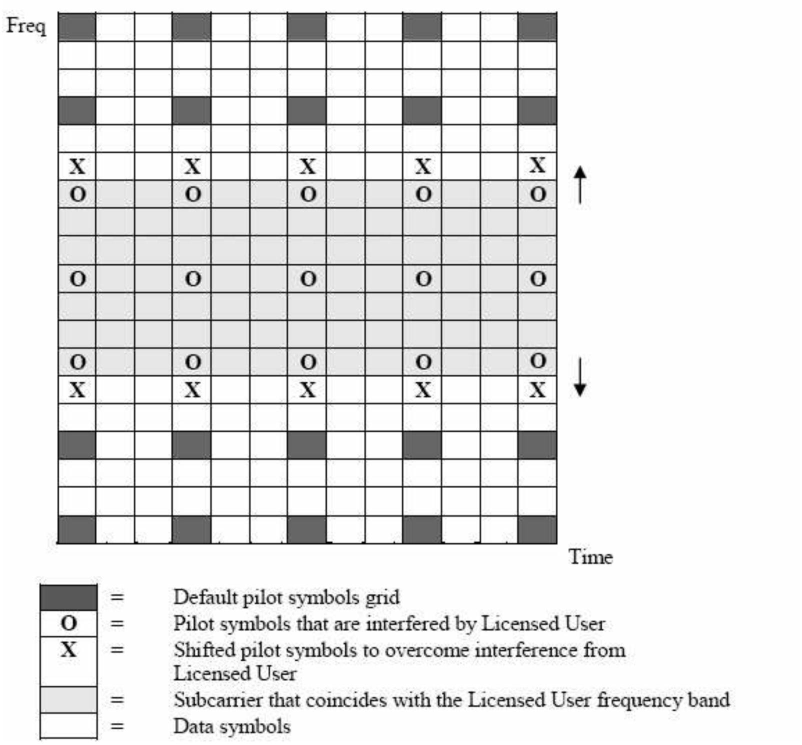

Freq
X
O
Time
= Default pilot symbols grid
O = Pilot symbols that are interfered by Licensed User
X = Shifted pilot symbols to overcome interference from Licensed User
= Subcarrier that coincides with the Licensed User frequency band
= Data symbols

Figure 3.8: Proposed pilot pattern for the cognitive radio system in the presence of the wideband LU.

method, respectively. In case the decision directed estimation module can perfectly estimate the transmitted symbol on the position of the virtual pilots, then the initial estimates on the virtual pilots position will be the same as if on that positions (virtual pilots) the transmitter sends the real pilots. The decision directed method requires the received signal, this method cannot be applied to the positions where the carriers are deactivated (LU's band) since the received signal on these positions are not available. The channel estimation is applied separately for the region before the deactivated band and after. As a consequence the more LUs exist the more carriers are deactivated, and the more edge effects in channel estimation occur that degrade the channel estimation performance. The edge effect can be avoided if the LU's band is less than the distance between pilots in frequency since the pilots on the deactivated carriers positions can still be nicely estimated by the extrapolation method.

3.3.4 Frequency Hopping GSM Channel Model At 900 MHz

In the application of spectrum pooling, problem arises if too many carriers need to be deactivated, hence there will not be enough resources to transmit data to fulfill the target bitrate. In [55] frequency hopping is proposed to deal with too many carriers deactivation in a selected band where the frame design is chosen with small bandwidth and the frame duration is based on the TDD Bluetooth frame. We have investigated the application of frequency hopping to an OFDM system at a GSM frequency 900

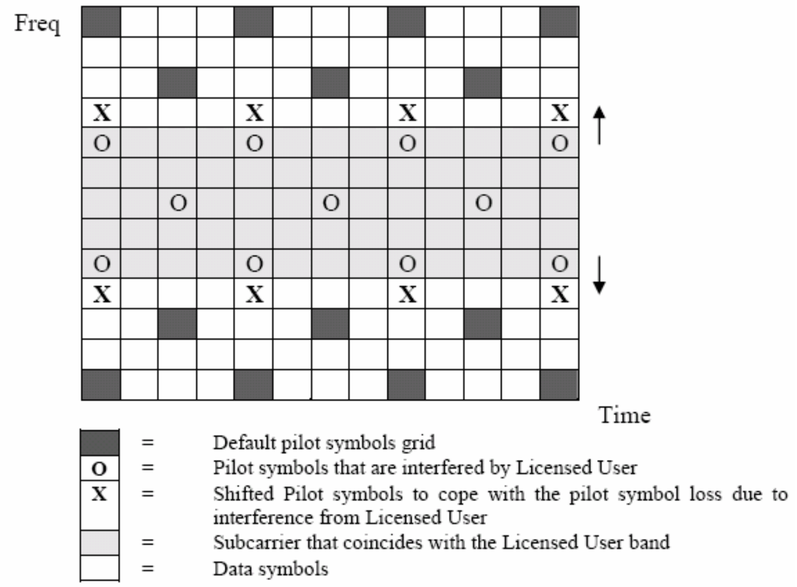

Freq
Time
X
O
= Default pilot symbols grid
O = Pilot symbols that are interfered by Licensed User
X = Shifted Pilot symbols to cope with the pilot symbol loss due to interference from Licensed User
= Subcarrier that coincides with the Licensed User band
= Data symbols

Figure 3.9: Transmitted modified hexagonal pilot pattern of cognitive radio system in the presence of wideband LU.

MHz using rural area and urban area propagation models [56]. The cognitivity of the system is the utilization of the spectrum between 800 to 900 MHz with bandwidth of 5 MHz in a short period of time, according to the Bluetooth TDD frame as if during that period the licensed users (LUs) do not acces the spectrum. A carrier frequency (f_c) is accessed after the scanning result of the CR device indicates that the frequency range is clean (unoccupied).

The propagation models are taken from the ETSI GSM model at 900 MHz [56]. We consider Rural area and Urban area power delay profile models. There are 2 alternative taps setting in each model. The coherence bandwidth for the rural area model of the first alternative taps setting is 1.6 MHz while the second alternative is 1.2 MHz. In case of the urban area model, the coherence bandwidth of the first alternative is 155 KHz while of the second alternative it is 159 KHz. The first alternative of the two area models suggests the sampling time of OFDM signal should be in the order of 0.1μs while for the second alternative a 0.2 μs sampling time is sufficient enough in order to have a sample spaced channel model and good channel estimation result. According to [55] in order to fulfill 2 pJ(pico joule)/ conversion step Figure of Merit of the analog to digital converter (ADC), the signal bandwidth must not exceed 5 MHz. Since the sampling frequency is twice the signal bandwidth, the sampling time must not be less than 0.1 μs.

The worst condition of the propagation model is found in the urban area propagation model where the maximum delay (τ_{max}) is 5 μs and non constant tap spacing. The OFDM frame is designed according to the TDD Bluetooth system as depicted in Fig. 3.12. Only about half duration of the slot can be used for transmission. After the system transmits the bits, it hops to a free (unoccupied) frequency band and it needs time to stabilize itself on that band. The frame duration must not exceed 366 μs. Table

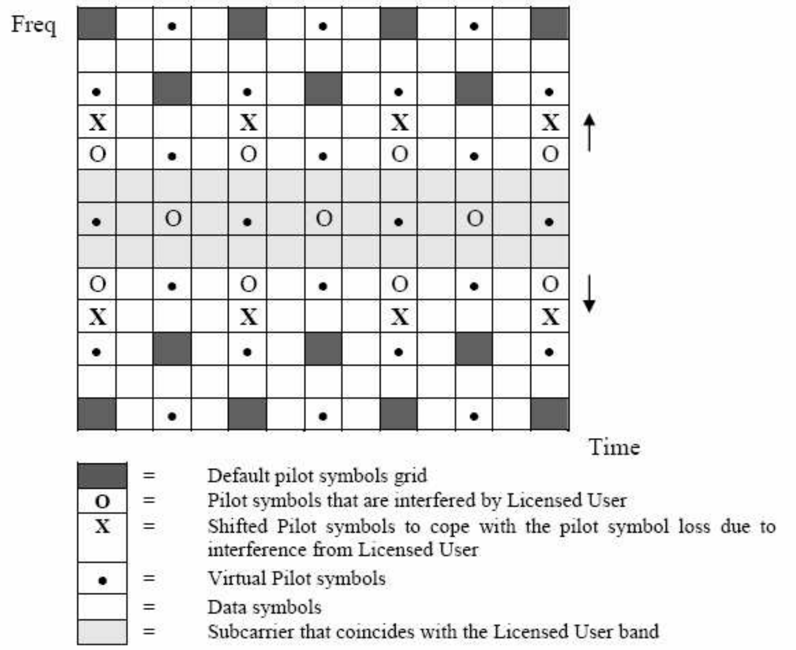

Freq
Time
= Default pilot symbols grid
O = Pilot symbols that are interfered by Licensed User
X = Shifted Pilot symbols to cope with the pilot symbol loss due to interference from Licensed User
● = Virtual Pilot symbols
= Data symbols
= Subcarrier that coincides with the Licensed User band

Figure 3.10: Receiver side hexagonal pilot pattern with virtual pilot symbols obtained from the extrapolation / interpolation combined with a decision directed method.

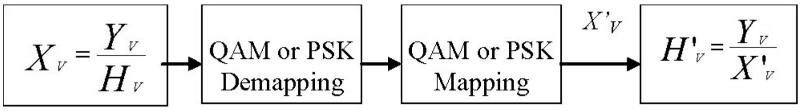


Figure 3.11: Decision directed process in estimating the virtual pilots.

3.3 shows the choices of OFDM parameters that fulfil the 366 μs frame duration constraint. The guard interval (GI) must exceed the worst case scenario in the urban area channel model with maximum delay of 5 μs. OFDM with a 2x1-D channel estimation technique requires 2 OFDM symbols for channel estimation purpose, while 1-D channel estimation requires only 1 OFDM symbol. Choosing 128 carriers with 11 OFDM symbols per frame is the best choice since it gives the highest throughput. The bit rate achieved with a 4 QAM constellation size on each carrier in case of 2x1-D channel estimation is around 3.7 Mbits/sec without channel coding. The frame is designed as a short packet. For the case of 1-D channel estimation only one training/pilot symbol is required, hence it gives higher throughput compared to 2x-1D, but there will be BER loss. In the 2x1-D channel estimation, training symbols are insterted in the beginning and at the end of the frame. Fig. 3.13 depicts the frame design.

Estimation is first applied by using Wiener filter sliding along the OFDM training symbols (frequency/vertical direction) only. In the next step, the filter slides along the time / horizontal direction using the channel estimates in the frequency domain as pilots. This process is repeated until the last carrier index / row. Then all the channel estimates of the frame are obtained. In 1-D channel estimation, the filter slides along

Tap Number	Relative Time (μs)		Average relative power (dB)	
	(1)	(2)	(1)	(2)
1	0.0	0.0	-0.0	0.0
2	0.1	0.2	-4.0	-2.0
3	0.2	0.4	-8.0	-10.0
4	0.3	0.6	-12.0	-20.0
5	0.4	-	-16.0	-
6	0.5	-	-20.0	-

Table 3.1: Propagation model for Rural Area GSM in 900 MHz [34]

Tap Number	Relative Time (μs)		Average relative power (dB)	
	(1)	(2)	(1)	(2)
1	0.0	0.0	-4.0	-4.0
2	0.1	0.2	-3.0	-3.0
3	0.3	0.4	0.0	0.0
4	0.5	0.6	-2.6	-2.0
5	0.8	0.8	-3.0	-3.0
6	1.1	1.2	-5.0	-5.0
7	1.3	1.4	-7.0	-7.0
8	1.7	1.8	-5.0	-5.0
9	2.3	2.4	-6.5	-6.0
10	3.1	3.0	-8.6	-9.0
11	3.2	3.2	-11.0	-11.0
12	5.0	5.0	-10.0	-10.0

Table 3.2: Propagation model for Urban Area GSM in 900 MHz [34]

N_{FFT}	$T_s(\mu s)$	Δf (kHz)	# of OFDM symbols	$\frac{T_G}{T_S}$
1024	204.8	4.9	1	$\frac{1}{32}$
512	102.4	9.7	3	$\frac{1}{16}$
256	51.2	19.5	6	$\frac{1}{8}$
128	25.6	39	11	$\frac{1}{4}$

Table 3.3: OFDM frame design according to TDD Bluetooth frame.

the only one pilot/training symbol aided to the frame. By assuming the channel doesn't change abruptly (slowly faded) during a frame duration then the channel estimates of the other OFDM symbols within a frame are equal to the channel estimates on the training/pilots position.

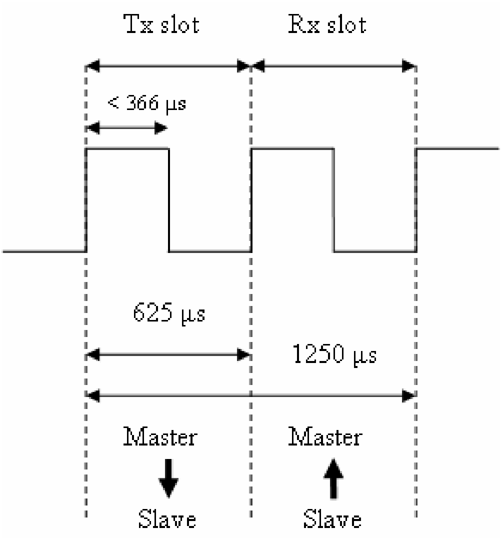


Figure 3.12: TX/RX cycle of the Bluetooth system [55].

As the spectrum scanning module indicates a 5 MHz bandwidth is available in a certain carrier frequency (f_c) (between 800 -900 MHz), then the front end module will hop to that frequency, stabilizes itself and then transmits data. With the assumption that the scanning process and transmitting data are independent to each other, the scanning is conducted in parallel with the front end's oscillator stabilization and data transmission process. After data transmission, information about the fc where 5 MHz bandwidth is free (provided by the spectrum scanning module) is available to each CR transceiver through a common control channel [57]. After receiving the f_c information, the front end will set the oscillator to the corresponding f_c. Time required for the transmitter's front end to transmit data and stabilizing itself to the new carrier frequency is about 625 μs, while during this period the spectrum scanner module also senses the frequencies from 800 to 900 MHz. The process flow of the frequency hopping including spectrum scanning and data transmission is described in Fig. 3.14.

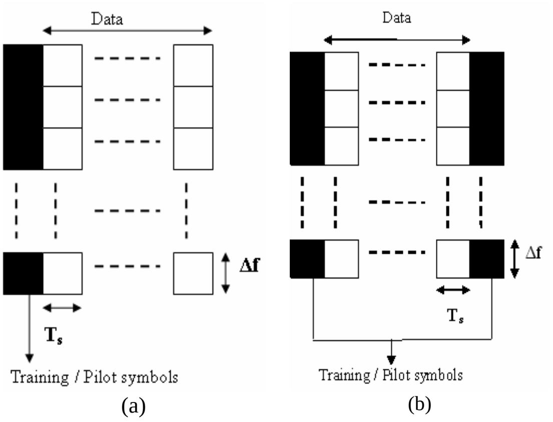


Figure 3.13: OFDM frame design for short packet transmission with (a) 1D channel estimation, and (b) 2x1D channel estimation.

OFDM Parameters		Channel Parameters	
Carr. Freq.	5.5 GHz	Chn. del. max.	5 μs
$\#$ of carriers	256	$\#$ of taps	11
Carr. spacing	31.25 KHz	Taps distance	0.5 μs
Guard time	5.25 μs	$\#$ of paths	12
Bit rate	7.5 Mbps	Veh. Speed	100 Km/hr
Chn. coding	OFF	Fading Model	Rayleigh

Table 3.4: Simulations Parameters I

3.3.5 Simulation Results and Discussions

In this subsection we present some major simulation results that we got from our experiments and investigations. In the beginning we have simulated the application of several adaptive bit loading algorithms combined with Wiener filter channel estimation. In this case the parameters used in the simulation are described in Table 3.4. Fig. 3.15 depicts the simulation of several bit loading algorithms taking the information about the channel from the Wiener filter . The allowable modulation modes are BPSK, 4-QAM,16-QAM, 64-QAM, and 256-QAM. The figure shows the superiority of the Fischer carrierwise algorithm. Fig. 3.16 depicts the comparison if the odd number of bits is allowed to be allocated on each carrier (allowable modulation mode : BPSK, 4-QAM, 8-QAM, 16-QAM, 32-QAM, 64-QAM, 128-QAM, 256-QAM). The results show that in case the Wiener filter is used, the performance difference is slight.

Further we have also examined the combination of windowing, carrier deactivation and adaptive bit loading in OFDM based Cognitive Radio systems. The simulations were running using parameters given in Table 3.5. Fig. 3.17(a) shows the comparison of the sidelobes for different kinds of windows. Among them the better than raised cosine (BTRC) window provides the lowest sidelobes. Deactivating carriers adjacent to the licensed system will lower the sidelobes even further, as can be seen in Fig. 3.17

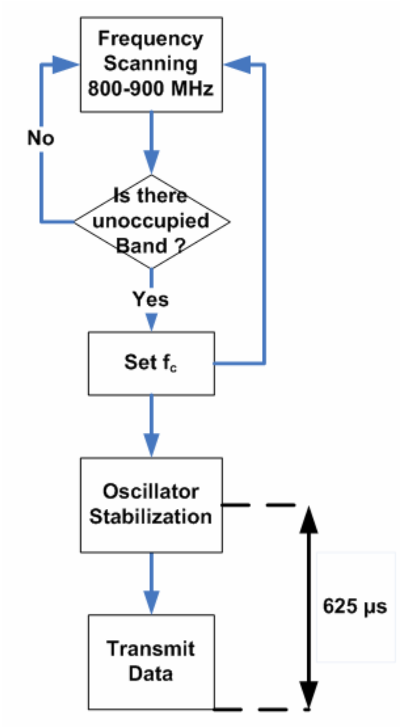


Figure 3.14: Process flow of hopping and data transmission for CR OFDM based with Bluetooth TDD frame design.

(b). The SNR vs BER results are shown in Fig. 3.18. Fig. 3.18 (a) shows the results of combined windowing with the design according to Fig. 3.6 (b) and adaptive bit loading with several bit loading algorithms. The Fischer carrierwise algorithm outperforms all other algorithms, with a gain of about 5 dB compared to conventional OFDM (without bit loading).

The proposed optimum pilot with a Hexagonal pattern for OFDM based Cognitive Radio combined with the concept of virtual pilots was simulated using the parameters in Table 3.6. Fig. 3.19 and 3.20 show that a tremendous performance improvement can be achieved at high SNR as the virtual pilots concept is utilized in the pilot pattern.

For the application of frequency hopping, the carrier frequency used is between

OFDM Parameters		Channel Parameters	
Carr. Freq.	5.5 GHz	Chn. del. max.	5 μs
# of carriers	256	# of taps	11
Carr. spacing	31.25 KHz	Taps distance	0.5 μs
Guard time	5.25 μs	# of paths	12
Bit rate	3 Mbps	Veh. Speed	100 Km/hr
Chn. coding	OFF	Fading Model	Rayleigh

Table 3.5: Simulations Parameters II

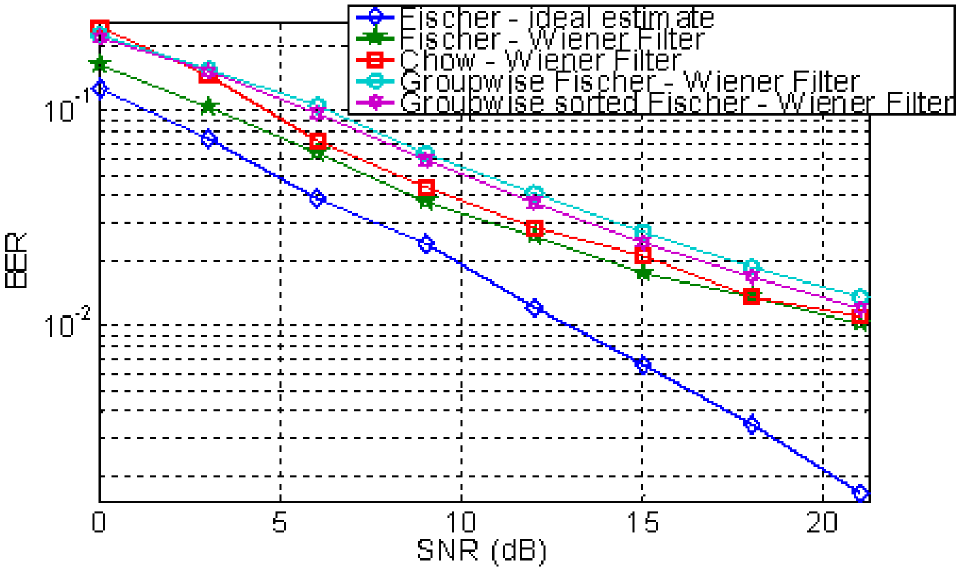


Figure 3.15: BER vs SNR (dB) for Adaptive OFDM with Rect Wiener Filter Channel Estimation with allowable modulation mode : BPSK, 4-QAM,16-QAM, 64-QAM, and 256-QAM.

800 -900 MHz. The hopping distance is in multiples of 10 MHz. The OFDM parameters are chosen as described in the previous section, where the number of carriers is 128, guard interval length is 6.4 μs, and 16 QAM constellation size. A frame consists of 11 OFDM symbols, and vehicular speed is 120 km/h. Since the frame duration is constrained to be 366 μs , the distance between pilot / training symbols even fulfils the 4 times oversampling assuming a vehicular speed of 300 km/h and fc 900 MHz. . The frame duration with those parameters is also much shorter than the coherence time of the channel. Rural and urban area propagation models are examined. Fig. 3.21 shows that the 1-D Wiener filter is the better choice considering its low complexity and slight BER degradation (around 0.5 dB) compared to the updated 2x1-D Wiener Filter. The BER performance is lower bounded by the filter matched to the power delay profile and power Doppler Spectrum of the GSM channel model. In case the CR system can access the spectrum longer than TDD frame duration, the distance between training symbols can be increased.

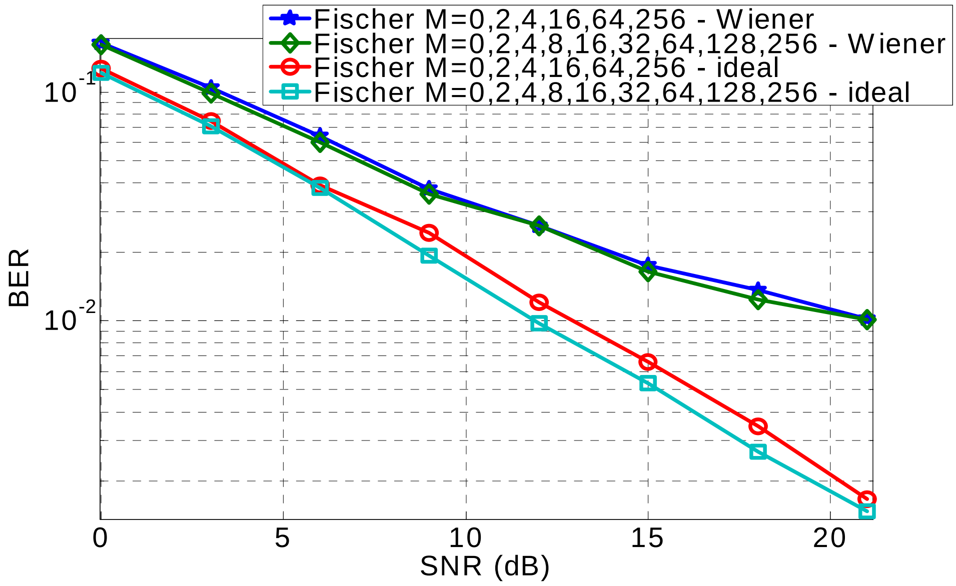
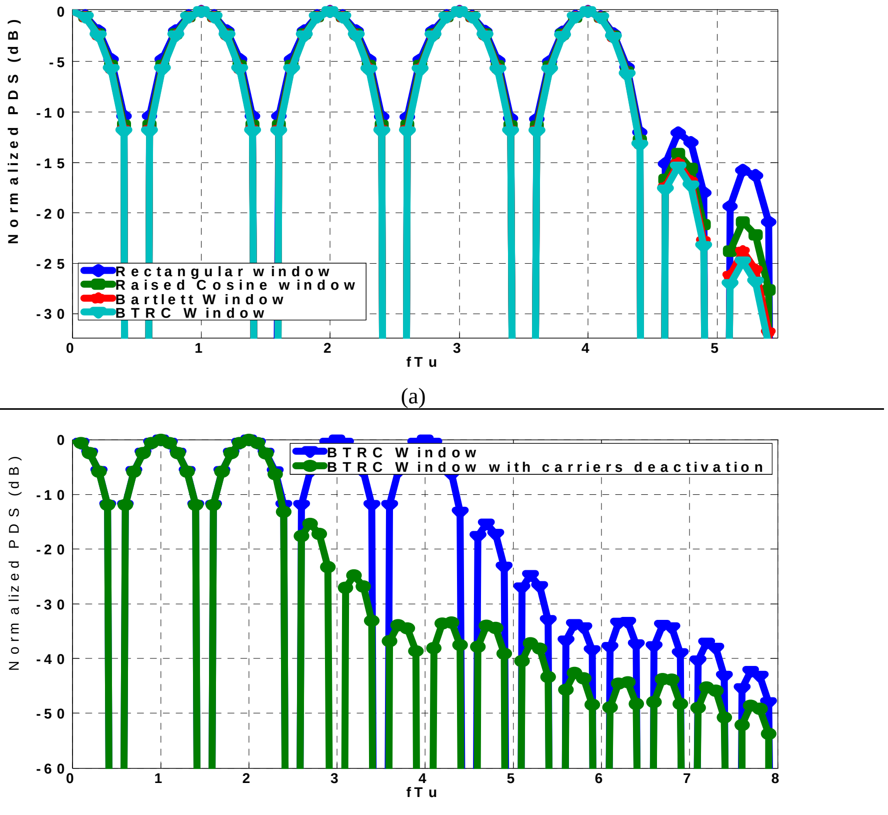


Figure 3.16: Comparison between allowable modulation modes (M) : BPSK, 4-QAM,16-QAM, 64-QAM, 256-QAM, and BPSK, 4-QAM, 8-QAM, 16-QAM, 32-QAM, 64-QAM, 128-QAM, 256-QAM.

Bandwidth	1.28 MHz
carriers	211
Carrier spacing	5 KHz
FFT length	256
Guard Interval (GI) length	43.75 μs
OFDM symbol duration (incl. GI)	$T'_s = 243.75\mu s$
OFDM symbols per frame	46
Modulation for pilot symbols	BPSK
Modulation for data symbols	4-QAM
Pilot spacing in frequency	d_f=4
Pilot spacing in time	d_t=3
Maximum Doppler frequency	$f_{D_{max}} = 82Hz$
Maximum delay	$\tau_{max} = 9.375\mu s$
RMS delay spread	$\tau_{rms} = 3.4\mu s$
Channel taps	4
Number of paths per tap	10
Power decrement per tap	1 dB
Tap spacing	3.125 μs

Table 3.6: Simulations Parameters III



(a)

(b)

Figure 3.17: Power density spectrum (PDS) of (a) the windows without carriers deactivation and (b) BTRC window with and without carriers deactivation. For both figures α=0.25 and window duration is $2T_u(\alpha+1)$.

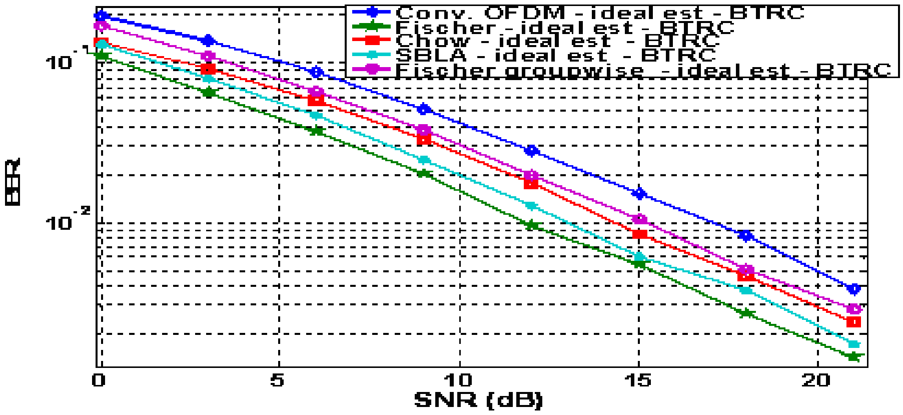


Figure 3.18: BER vs SNR (dB) Comparison of Adaptive Bit Loading algorithms using Better Than Raised Cosine Window form according to design in Fig. 1.6(b) with $\alpha = 21/128(TGI/Tu)$ and 48 deactivated carriers in the occupied spectrum.

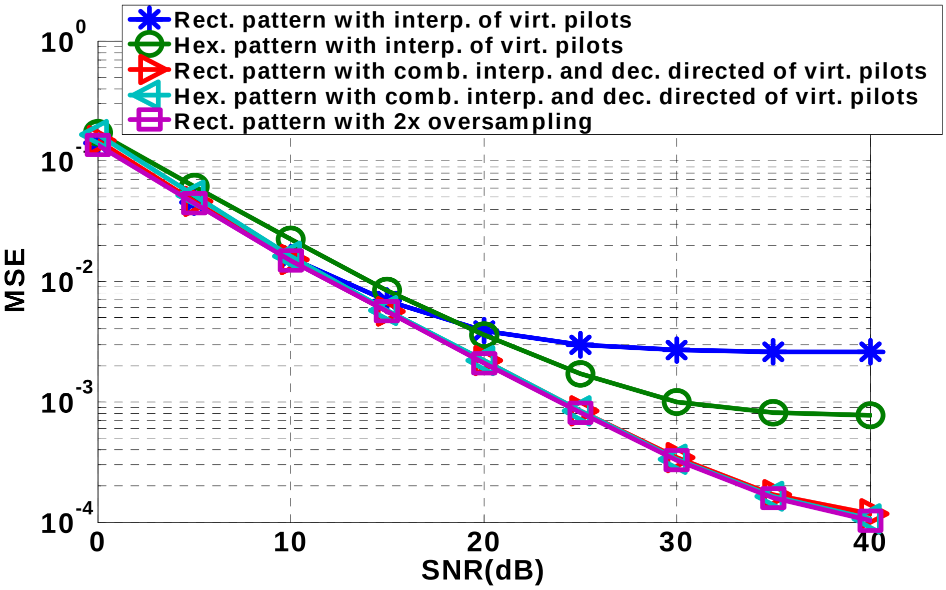


Figure 3.19: Performance comparison of rect., hex. pilot pattern with virtual pilots from interpolation./extrapolation, their combination with decision directed method and the 2x oversampling rect. pattern.

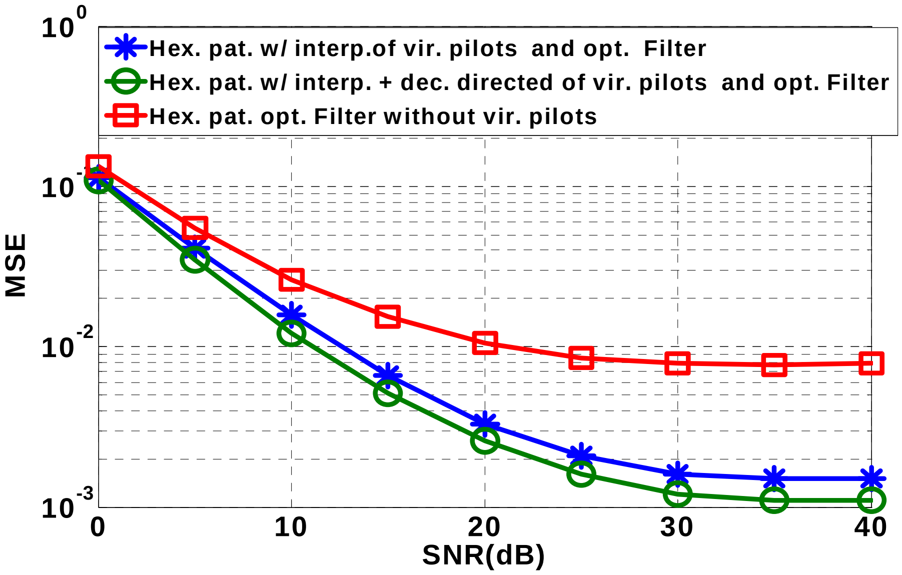


Figure 3.20: Performance comparison opt. Wiener Filter with hex. pilot pattern with virtual pilot symbols from interpolation/extrapolation and its combination with decision directed method, and the one without virtual pilots in the presence of wideband LU.

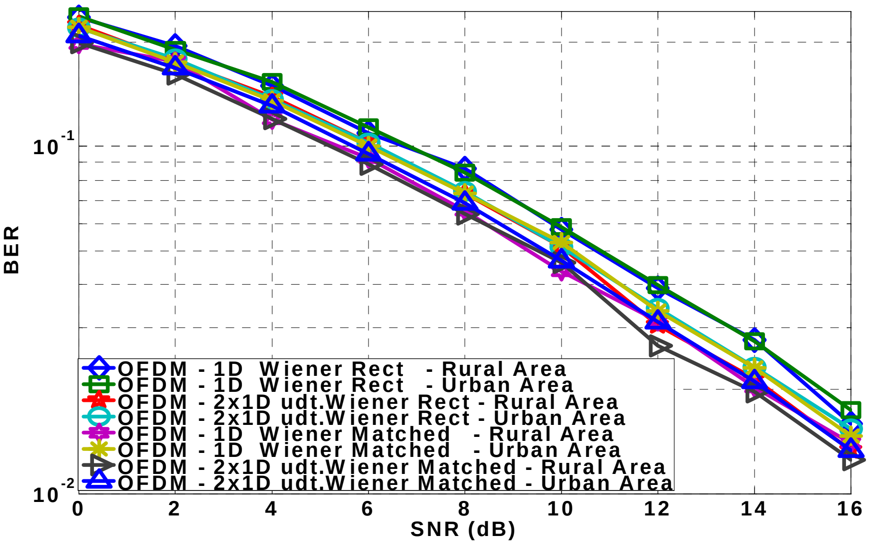


Figure 3.21: BER vs SNR(dB) performance of OFDM with 1-D and updated 2x1-D Wiener filter.

3.3.6 Publications with Respect to The Topic

- I. Budiarjo, M.K. Lakshmanan, and H. Nikookar, ”Cognitive Radio Dynamic Access Techniques”, in *Springer International Journal on Wireless Personal Communications special issue on Cognitive Radio Technologies*, 2008.

- I. Budiarjo, H.Nikookar, and L.P. Ligthart,”Overview of Adaptive OFDM in the Context of Cognitive Radio”, in *Proc. 12th Annual Symposium of IEEE on Communications and Vehicular Technology (SCVT) in the Benelux,* Nov. 3, 2005, Enschede, The Netherlands.

- I. Budiarjo, H. Nikookar, and L.P. Ligthart,”Performance Evaluation of Adaptive Bit Loading For OFDM With Channel Estimation Using 2x 1-D Wiener Filter”, in *Proc. 9th Annual International Symposium on Wireless Personal and Multimedia Communications (WPMC),* Sept. 17-20, 2006, San Diego, USA.

- I. Budiarjo, H. Nikookar, and L.P. Ligthart, ”Combined Spectrum Pooling and Adaptive Bit Loading for Cognitive Radio OFDM Based System”, in *Proc. 13th Annual Symposium of IEEE on Communications and Vehicular Technology (SCVT) in the Benelux*, Nov. 23, 2006, Liege, Belgium.

- I. Budiarjo, H. Nikookar, and L.P. Ligthart, ”Techniques for Cognitive Radio”, invited paper on *the 2nd Center for TeleInfrastruktur (CTIF) Workshop*, May 22-23, 2007, Aalborg, Denmark.

- I. Budiarjo, H. Nikookar, and L.P. Ligthart, ”Performance Evaluation of OFDM Based Cognitive Radio System With Wiener Filter Channel Estimation Using Frequency Hopping GSM Channel Model at 900 MHz”, in *Proc. 10th European Conference on Wireless Technology (ECWT),* Oct. 8-12, 2007, Munich, Germany.

- I. Rashad, I. Budiarjo and H. Nikookar, ”Efficient Pilot Pattern for OFDM based Cognitive Radio Part I”, in *Proc. 14th IEEE Symposium on Communications and Vehicular Technology (SCVT),* Nov. 15, 2007, Delft, The Netherlands.

- I. Budiarjo, I. Rashad and H. Nikookar, ”Efficient Pilot Pattern for OFDM based Cognitive Radio Part II”, in *Proc. 14th IEEE Symposium on Communications and Vehicular Technology*, Nov. 15, 2007, Delft, The Netherlands.

- I. Budiarjo, I. Rashad and H. Nikookar, ”On The Use of Virtual Pilots with Decision Directed Method in OFDM Based Cognitive Radio Channel Estimation Using 2x1-D Wiener Filter”, in *Proc. IEEE International Conference on Communications (ICC)*, May 19-23, 2008, Beijing, China.

3.4 Transform Domain Communications System (TDCS)

TDCS is a single carrier transmission system where the bandwidth can be divided into smaller subbands. In this way it is easier to locate the part of the band occupied by the LUs and to not put energy in that region. Information regarding the spectrum occupancy of the licensed users is distributed to each CR device. In our project the information is transmitted through a common control channel [57]. The transmitter architecture of TDCS system is depicted in Fig. 3.22. Upon receiving the information about the region of the LU's band, a vector map $\boldsymbol{A}'(\omega)$ is produced by the Spectrum Magnitude block where ω defines the index of subband. Zeros are placed on the subband indexes where the LU's band is located, and ones are placed on the other subband indexes [58], [59]. A guard interval (GI) is added to combat the multipath fading channel effect. Windowing can be added in order to reduce the sidelobes of the transmitted signal on the LU's band. TDCS gives more degrees of freedom in choosing a window to lower the sidelobes of its spectrum. Unlike in OFDM, the spectrum of the window of TDCS does not have to hit zero on the subband spacing interval as long as its sidelobes are very low, e.g through a half sine window [50]. Inter subband interference will not degrade the data detection at the receiver due to the robustness of autocorrelation method. Windowing can be replaced by deactivating more subbands adjacent to the LU's band (adding more zeros to the vector $\boldsymbol{A}'(\omega)$ at the subbands adjacent to LU's band) since it will not reduce the transmission bit rate.

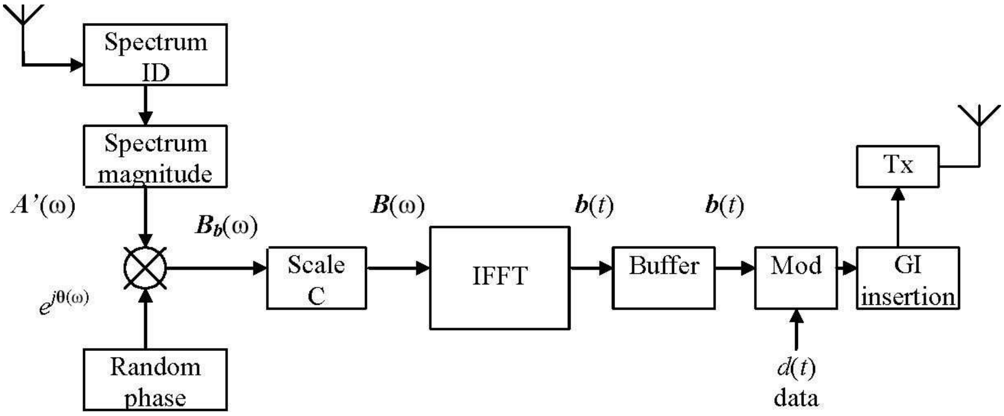


Figure 3.22: Transmitter architecture of TDCS.

The impact of zero insertion can be described from the power density spectrum (PDS) equation,

$$PSD(f) = \frac{1}{N_{FFT}} \left| \sum_{m=0}^{N_{FFT}-1} A(\omega_m) e^{j\theta(\omega_m)} \int_{-(1+\alpha)\frac{T_u}{2}}^{(1+\alpha)\frac{T_u}{2}} g(t) e^{-j2\pi(f-f(\omega_m))t} dt \right|^2 \tag{3.20}$$

where T is the usefull signal duration, $p(t)$ is the window function, α is the roll off factor of the window, and $\theta(\omega_m)$ is the phase on subband m produced by the random phase module. If $p(t)$ is a rectangular window and α is zero, the area within the

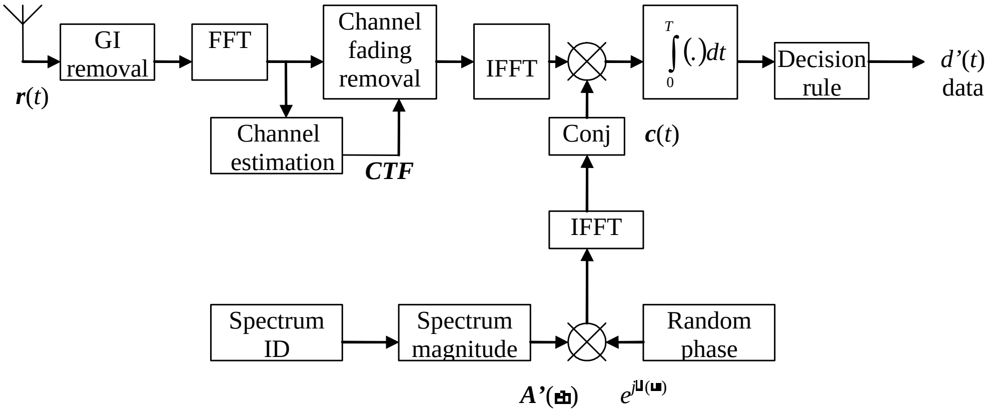


Figure 3.23: Receiver architecture of TDCS with Channel Estimation module.

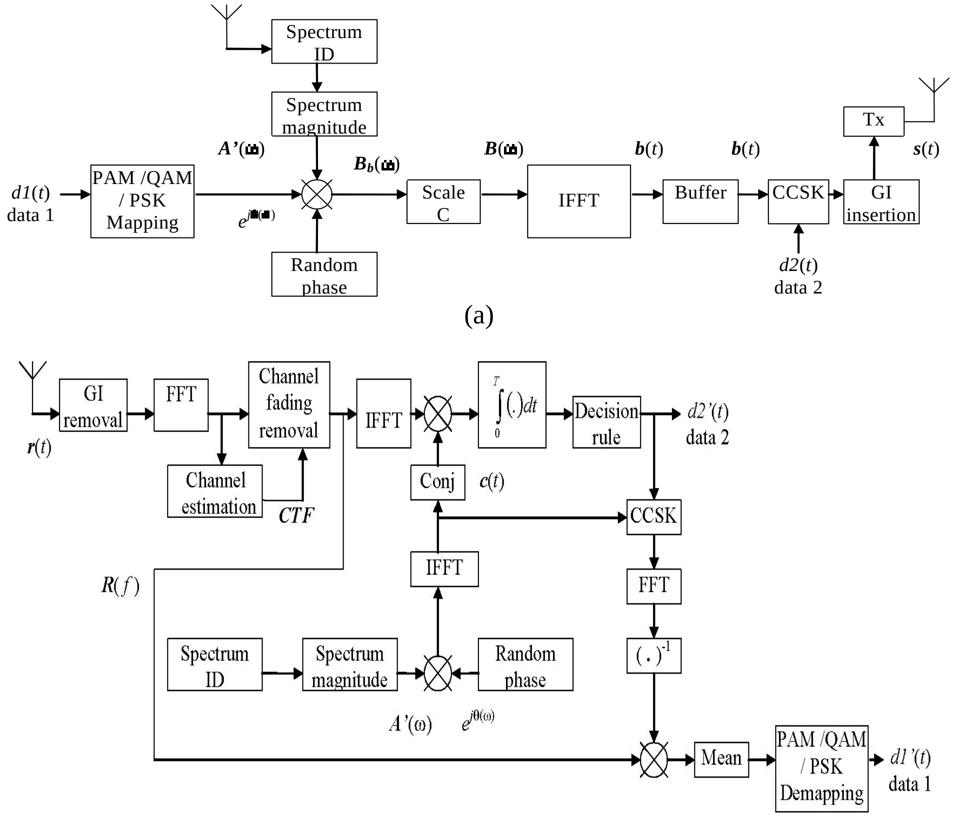
integral can be replaced by $Tsinc((f - f(\omega_m))T)$. Zero amplitude at the carrier ω_m will make the PDS on that carrier position becoming zero, and due to the orthogonality among carriers the power contributions from the other carriers are also zero. The time domain signal $b(t)$ is combined with the transmitted data $d(t)$ using Pulse Amplitude Modulation (PAM) or the Cyclic Code Shift Keying (CCSK) scheme. In binary PAM the bits are mapped as the binary signals are the negative of each other, while in binary CCSK the binary signals are the cyclic shift by half of the symbol period of each other. A symbol is represented by N_{FFT} samples, accordingly the bit rate of TDCS system can be calculated as given in 3.21,

$$R_{TDCS} = \frac{log_2 CS}{T_s'}, \tag{3.21}$$

where CS is the constellation size of modulation and $T_{S'}$ is the TDCS symbol duration including GI duration. The maximum CS that can be applied in CCSK is restricted by the number of subbands (N_{FFT}). Unlike PAM, CCSK has the tendency of lower probability of error as the CS increases [60], [61], hence the optimum constellation size for TDCS with CCSK modulation is equal to the number of subbands (N_{FFT}). Theoretically it will give a better bit error rate (BER) performance and also a higher bitrate. The outputs of the random phase modules at the transmitter and at the receiver are the same. At the receiver the random phase vector with the size of NFFT will be used for data detection. A modification to the receiver block diagram according to [59] has been made in our system. A channel estimation module is added as depicted in Fig. 3.23 to remove the effect of the fading. After the GI removal, the fading effect is cancelled in the frequency domain, after the FFT module. The signal is transformed again into the time domain by an IFFT, and the signal is decorrelated with the reference signal $c(t)$. Decorrelation for symbol demapping is simplified in [61] by choosing the maximum of the absolute value of the inverse Fourier transform of the product of the conjugated received signal and the reference signal in frequency domain.

3.4.1 Higher Rate TDCS with Extra Embedded Symbol

In an attempt to increase the bit rate of TDCS, in this subsection we describe the proposed embedding symbol from PAM or QAM or PSK modulation to TDCS. The transmitter and receiver architectures of the proposed system are depicted in Fig. 3.24. There are two bit sources available. The first one is for the embedded signal, and the second one is for the CCSK modulation. We decided to have N_{FFT} as the CS value of the CCSK modulation, while the constellation size of the embedded signal can be varied according to the desired target rate and expected overall BER. At the transmitter one embedded symbol will be multiplied with the vector resulting from the point to point multiplication between the random phase vector $\boldsymbol{\theta}(\omega)$ and the spectrum magnitude vector $\boldsymbol{A}'(\omega)$. Further, the normal procedure in producing a TDCS signal is applied. The receiver in the beginning detects the data from the CCSK modulation.



(a)

(b)

Figure 3.24: (a) Transmitter and (b) Receiver architecture of TDCS with extra embedded symbol.

After removing the fading effect in the frequency domain, the decorrelation in the time domain for detecting the CCSK data can be replaced by taking the maximum absolute value of the inverse Fourier transform of the product of the conjugated received signal and the reference signal in the frequency domain. The data from CCSK detection represents the estimated shift that was taking place at the transmitter for transmitting the data from the CCSK modulation. This estimated shift is then further used to shift the reference signal, followed by transforming it into the frequency domain signal with the aid of an FFT. The received signal in the frequency domain before CCSK data detection is then divided by the shifted reference signal (in the frequency domain). After taking the average value per N_{FFT} samples and detection by PAM or QAM or PSK demapping, the bits from the first symbol are reconstructed. In order to improve the BER performance further the channel coding and interleaving can be applied before the mapping process, meanwhile the deinterleaving and decoding are applied after demapping. These techniques will not be considered in this document. The new TDCS bit rate will become :

$$R_{NEW \quad TDCS} = \frac{log_2(CS.CS_{EM})}{T_s'}, \tag{3.22}$$

where CS_{EM} is the embedded symbol constellation size.

The new PSD becomes :

$$PSD(f) = \frac{1}{N_{FFT}} \left| Se^{j\varphi} \sum_{m=0}^{N_{FFT}-1} A(\omega_m) e^{j\theta(\omega_m)} \int_{-(1+\alpha)\frac{T_u}{2}}^{(1+\alpha)\frac{T_u}{2}} g(t) e^{-j2\pi(f-f(\omega_m))t} dt \right|^2 \tag{3.23}$$

where φ is the phase of the embedded symbol. For the purpose of having as low as possible sidelobes for dynamic spectrum access, according to 3.23, the PSK modulation mode is the most appropriate one due to its unity energy amplitude, while PAM and QAM will increase the sidelobes as the constellation size is bigger than 2. Moreover with regard to the sidelobes requirement, the PAM modulation is the worse option since for the same constellation size (bigger than 2), its maximum amplitude is bigger than QAM or PSK mapping.

3.4.2 Simulation Results and Discussions

The simulations for the higher rate TDCS we running using parameters described in Table 3.7. The results in Fig. 3.25 show that in a Rayleigh fading channel the CCSK detection is consistently improved as the constellation size of the embedded symbol increases. We see the maximum gain is achieved when the embedded symbol is using 256 PSK. More than 9 dB SNR gain is achieved, as the conventional TDCS reaches its BER at SNR 14 dB, the TDCS with embedded 256 PSK requires 9 dB less SNR to have the same BER.

TDCS Parameters	
Carr. Freq.	5.5 GHz
# of carriers	128
Carr. spacing	31.25 KHz
Guard time	5.25 μs
Chn. coding	OFF
Channel Parameters	
Chn. del. max.	5 μs
# of taps	6
Taps distance	1 μs
# of paths	12
Veh. Speed	100 Km/hr
Fading Model	Rayleigh

Table 3.7: Simulations Parameters

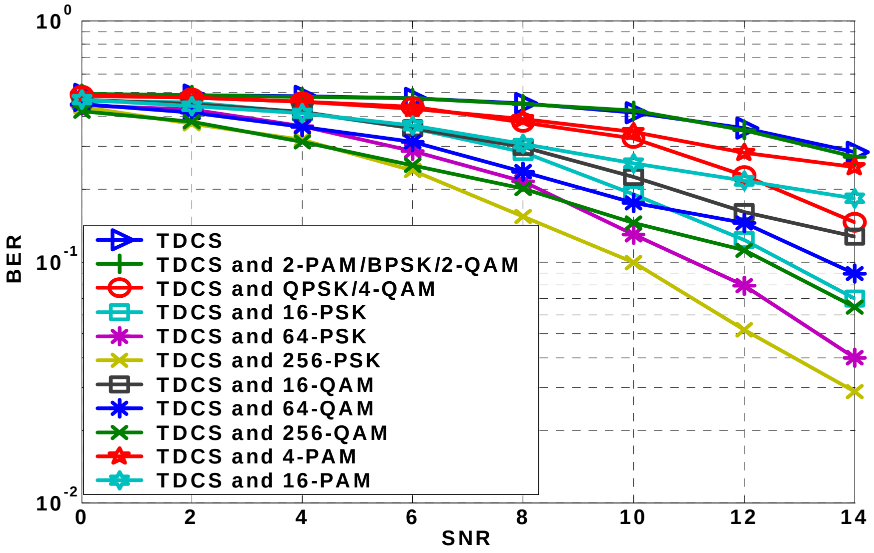


Figure 3.25: SNR (dB) vs BER performance of the TDCS-CCSK with PAM, PSK and QAM embedded symbol on data2 source in Rayleigh fading channel

3.5 Wavelet Packet Modulation for Cognitive Radio Systems

In traditional implementations of Multi carrier OFDM (MC-OFDM), the generation and modulation of the sub-channels are accomplished digitally using Fourier bases. In this subsection another modulation technique is introduced. The conventional Fourier-based complex exponential carriers of OFDM are replaced with orthonormal wavelet packet (WP) bases for use in cognitive radio environments. The WP bases are derived from perfect reconstruction of two-band FIR filter bank solutions [62], [63], [64], [65]. Cohabitation of the WP-MCM based CR systems with existing licensed users is actualized by shaping its transmission waveform and by adaptively activating or deactivating sub-carriers in a way that it utilizes the unoccupied time-frequency gaps of the LU. The idea is to dynamically sculpt the transmission signal so that it has no or very little time-frequency components competing with the LU. This way the CR can seamlessly blend with the LU operation. The steps to generate the WP-MCM transmission signal are elucidated in Fig. 3.26.

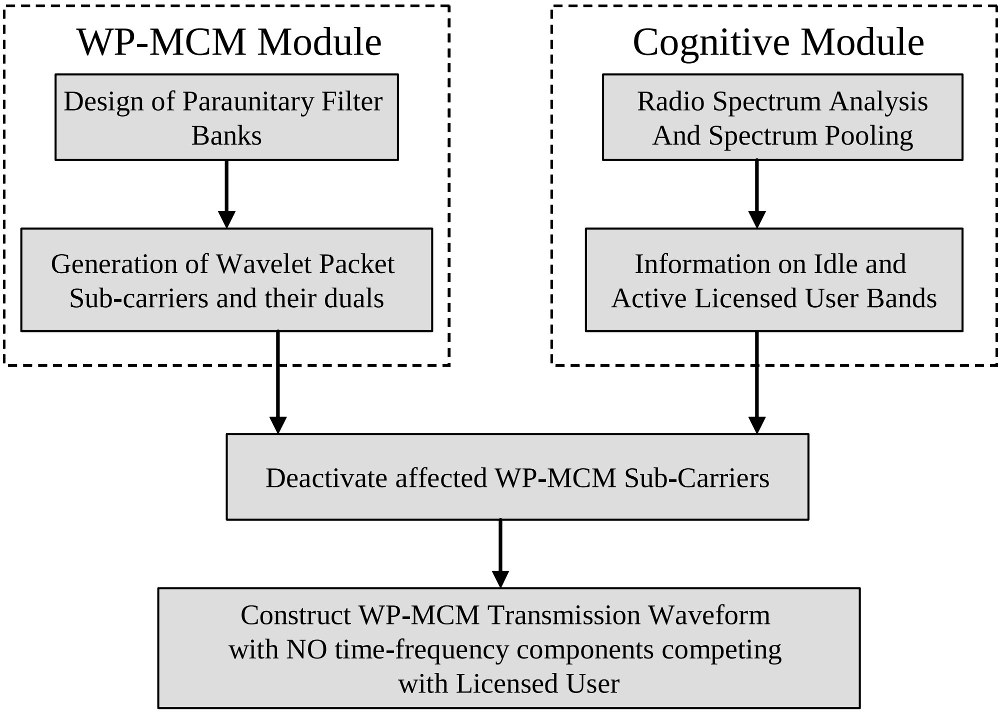

WP-MCM Module
Design of Paraunitary Filter Banks
Generation of Wavelet Packet Sub-carriers and their duals
Cognitive Module
Radio Spectrum Analysis And Spectrum Pooling
Information on Idle and Active Licensed User Bands
Deactivate affected WP-MCM Sub-Carriers
Construct WP-MCM Transmission Waveform with NO time-frequency components competing with Licensed User

Figure 3.26: Fundamental Blocks of WP-MCM based Cognitive Radio system.

The level and the number of bases generated is given as $M = 2J$. A level 3 tree structure generates up to 8 wavelet packet carriers and duals each. The WP-MCM transmission is realized by replacing orthonormal complex exponential basis functions, as used in OFDM systems, with orthonormal wavelet packet basis functions. The wavelet packet bases and their dual bases are derived from perfect reconstruction two-band FIR filter bank solutions from multistage tree-structured paraunitary filter banks derived by cascading 2-channel filter banks. The wavelet packet sub-carriers (to be used at the transmitter end) are generated through a multichannel filterbank consisting of cascaded two-channel filterbanks applying the synthesis filters (H' and G'). The process known as the synthesis procedure consists of binary interpolation (up-sampling) by 2, filtering and recombination at each level. To demonstrate the process of generation of wavelet packet bases, here we consider a cascaded level-3 tree structure as shown in Fig. 3.27(a). Such an arrangement can give rise to eight wavelet packet bases. This procedure is simplified further by applying the identities shown in Fig. 3.27(b). These expressions called the noble identities are popularly applied in the implementation of multi-rate systems. And in Figure 3.27(c) the equivalent eight-channel system after the application of the noble identities is shown. In general, to generate M bases or sub-carrier waveforms, a level-J tree has to be constructed. The wavelet packet duals (to be used at the receiver end) are obtained from a multichannel filter bank analysis too, though the processes are reversed. The duals are obtained from the analysis filters (H and G) through the analysis procedure which consists of filtering, decimation (downsampling) by 2 and decomposition at each stage. Fig. 3.28 illustrates the generation of 8 wavelet packet duals from a level-3 tree cascaded filter bank.

Choosing the right wavelet is a delicate and at times even an overwhelming issue. In theory any time and frequency limited function can be utilized. However in practice, the wavelet bases cannot be arbitrarily chosen and instead have to satisfy a number of requirements. With regard to the applicability to WP-MCM systems, the desirable properties may be listed as follows:

- The bases must be time and band limited.
- The bases and their duals must be orthogonal to one another to enable perfect reconstruction.
- The bases must be orthogonal to one another in order to have unique demodulation.

Considering these requirements, among several available wavelet such as: Coifflets, Daubechies, Haar, Symlets, the most suitable wavelet is the family of maximally frequency filter banks derived using a modified Remez exchange algorithm. It provides degree of freedom to shape its sidelobes by varying its regularity order, transition bandwidth and the filter length. The longer the filter length the more degree of freedom available in designing the frequency selectivity and also the regularity order of the signal. The smaller the transition bandwidth the more frequency selective the filter will be. Regularity corresponds to the sidelobes of the signal. The higher the regularity order, the smoother or the lower the sidelobes of the signal. More on this subject can be found in [66]. In [67] WP-MCM with Remez exchange algorithm as CR system is evaluated and compared with a Fourier based MC-OFDM CR system in the presence of an LU signal in an Additive White Gaussian Noise (AWGN) channel. The results show that for certain Remez parameters values, WP-MCM outperforms the Fourier based MC-OFDM.

In an ideal scenario the filter banks used to generate the wavelets have zero transition bands B (i.e., difference between pass and stop band frequencies). Under such an ideal scenario for a level-i decomposition, the wavelet packet bases derived have confined spectral footprints with bandwidth $(1/2i)$ times that of the Nyquist frequency.

However, available wavelet families are derived from filter banks that have a wide transition band and hence the resultant wavelet sub-carriers have a dispersed spectrum with footprints spilling into neighboring regions. The wider the transition bandwidth the greater the dispersion of the carrier's spectral footprint and therefore the greater the difficulty in isolating those sub-carriers that fall in the region of the licensed user. This greatly reduces the efficiency of the system. It is therefore important to design filter banks that have narrow transition bands. The fundamental theory on the design of frequency selective filter banks was developed by Rioul and Duhamel [66]. They devised the procedure to design maximally frequency selective filter banks under a given set of constraints using the modified Remez exchange algorithm. In addition to the frequency selectivity, two other properties are important in the design of wavelets. They are the paraunitary and regularity conditions. The paraunitary condition is an important property for many reasons. Firstly, it is a prerequisite for generating orthonormal wavelets [68]. Second, it automatically ensures perfect reconstruction of the decomposed signal i.e., the original signal can be reconstructed without amplitude or phase or aliasing distortion [69]. A rational transfer function $T(z)$ is said to be paraunitary when it obeys the relation $\tilde{T}(z)T(z) = 1$. Here $\tilde{T}$ is the paraconjugate of $T(z)$ and is given as $\tilde{T}(z) = T(z^{-1})$ where the subscript * denotes the conjugation of the coefficients. The regularity property is a measure of smoothness of the wavelet and is normally quantified by the number of times a wavelet is continuously differentiable. The simplest regularity condition is the "flatness" constraint which is normally stated on the low pass filter (LPF). A LPF is said to satisfy Kth order flatness if its transfer function $H(z)$ contains K zeroes located at the Nyquist frequency ($z = -1$ or $\omega = \pi$). The parameter is called the regularity index and for a filter of length L it satisfies the condition $0 \leq K \leq \frac{L}{2}$. The objective of the design is therefore to construct a maximally frequency selective filter bank with additional constraints that the filters:

- allow orthonormal expansion and perfect reconstruction of discrete-time signals
- have finite impulse response
- satisfy the paraunitary condition
- satisfy the flatness/regularity condition

For a filter of length L this is essentially solving L unknown filter variables from L linear equations. Of these L linear equations, $L/2$ equations come from the paraunitaryness constraint, K equations come from the regularity or flatness constraint and the remaining $L/2 - K$ conditions offer the possibility for establishing frequency selectivity. The larger the value of $L/2 - K$, the greater the degree of freedom for frequency selectivity and the greater the loss in regularity. There is therefore a trade-off between frequency selectivity and regularity. Wavelets such as the Daubechies family are maximally flat with regularity index $K = L/2$ and hence they are not frequency selective.

Such an optimization problem can be solved using the modified Remez exchange algorithm. The exact details of how these equations are solved to obtain the filter banks can be found in [66].

3.5.1 Simulation Results and Discussions

In this work we are considering an AWGN channel, while the Licensed User's band is considered to be comparable to 32 carriers (1/4th) of the CR system and located in the middle of the CR spectral band (Fig. 3.4). The Cognitive Radio system operates with 128 equally spaced carriers. The 128 wavelet packet carriers are derived from a level-7 cascaded tree. As shown in Fig. 3.29, the performance of both the OFDM and WP-MCM based CR systems worsens (WP-MCM is marginally more tolerant than OFDM) in the presence of an LU and perform better when their carriers in and around the LU spectral bands are removed. For OFDM the best results are obtained when 36 or more carriers are removed while for WP-MCM the number is about 40 carriers. The advantage of WP-MCM is that once its transmission signal is steered clear of the LU bands, it comfortably outperforms its OFDM counterpart. For e.g. for a given BER of 10^{-4} with 44 carrier removals, WP-MCM gives a gain of about 2 dB in comparison to OFDM.

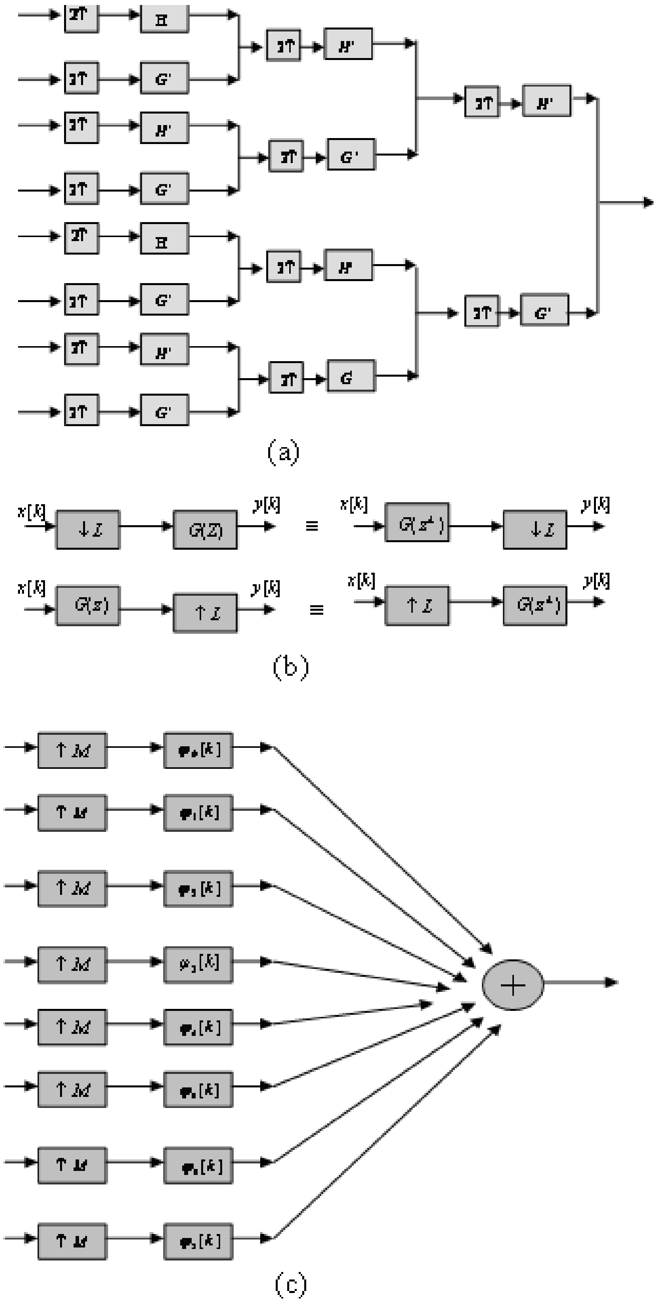


Figure 3.27: Generation of wavelets. (a) A level-3 tree gives 8 wavelet packet bases. The up arrows represent interpolation by 2. H' and G' denote the frequency responses of the low and high pass reconstruction filters, respectively; (b) Noble identities; (c) Equivalent 8-channel system after applying the noble identities, where φ' is equivalent reconstruction filter as results of the convolution combination H' and G' according to the tree lines. $M = 8$ for the example considered.

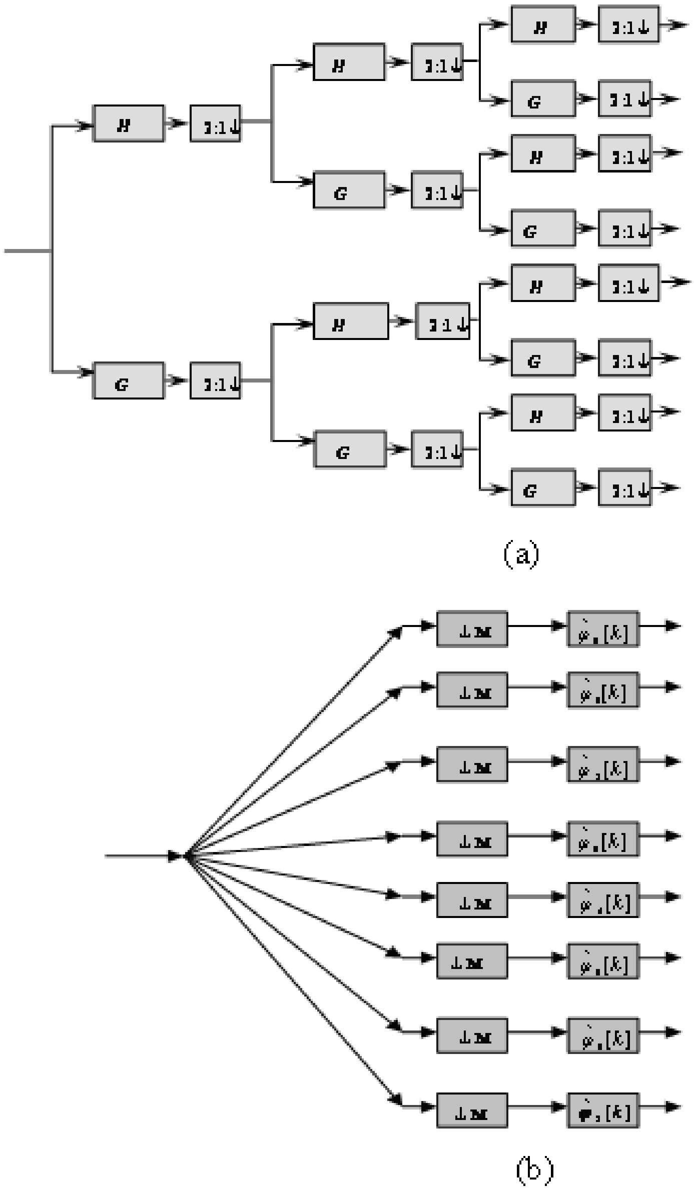


Figure 3.28: Generation of wavelet Duals. (a) A level-3 tree gives 8 wavelet packet dual bases. The down arrows represent decimation by 2. H and G denote the frequency responses of the low and high pass decomposition filters, respectively; (b) Equivalent 8-channel system after applying the noble identities, where φ is equivalent decomposition filter as results of the convolution combination H and G according to the tree lines $M = 8$ for the example considered.

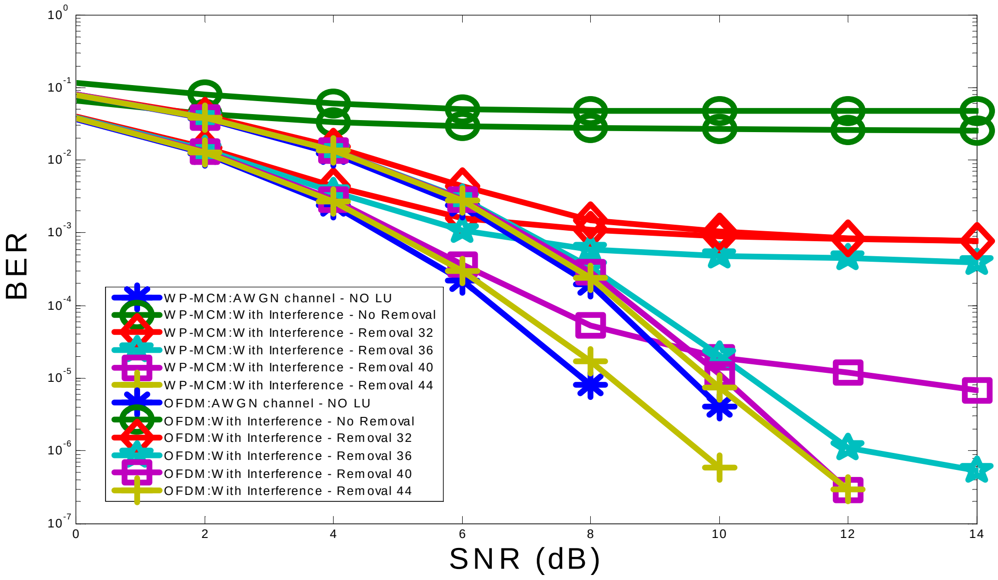


Figure 3.29: OFDM versus WP-MCM based CR. BER performance comparison of OFDM and WP-MCM based CR systems in the presence of LU.

3.5.2 Publication with Respect to The Topic

- M. K. Lakshmanan, I. Budiarjo and H. Nikookar, ”Maximally Frequency Selective Wavelet Packets Based Multicarrier Modulation Scheme for Cognitive Radio Systems”, in *Proc. 50th IEEE Global Communications (GLOBECOM) Conference*, Nov. 26-30, 2007, Washington, DC, USA.

3.6 MIMO V-BLAST Architecture

V-BLAST is one of the promising realizations of multiple input and multiple output (MIMO) systems [70]. At each symbol it detects the strongest layer of the transmitted signal, cancels the effects of this strongest layer from each of the received signals, then continues to detect the strongest on the remaining layers, and so on. The MIMO transmitter and receiver models for CR TDCS and OFDM based systems are depicted in Fig 3.30. At the receiver side of either TDCS or OFDM shown in Fig. 3.31, the received signals on each receiver antenna are transformed first into the frequency domain. V-BLAST algorithms will reconstruct the transmitted signal by removing the fading effect of the channel in the frequency domain. Wavelet packet OFDM and higher rate TDCS with embedded MIMO V-BLAST are depicted in Fig. 3.32 and 3.33. The received signals at each receiver antenna n from each of the transmit antenna m is :

$$Y_n = \sum_{m=1}^{M} H_{nm} X_m + \eta_n, \tag{3.24}$$

or in matrix form the received signal vector is :

$$\boldsymbol{Y} = \boldsymbol{H}\boldsymbol{X} + \boldsymbol{\eta} \tag{3.25}$$

where $Y = [y_1, y_2, , y_N]$ is the received signal vector, $X = [x_1, x_2, , x_M]$ is the transmitted signal vector, M is the number of transmit antennas, N is the number of receive antennas, $\boldsymbol{\eta} = [\eta_1, \eta_2, , \eta_N]$ is the noise vector and the channel matrix H is described as 3.26 where H_{mn} is the channel link between transmit antenna m and receive antenna n [71]. Fig. 3.34 depicts all the links between transmit and receive antennas.

$$\begin{bmatrix} H_{11} & H_{12} & \cdots & H_{1M} \\ H_{21} & H_{22} & \cdots & H_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ H_{N1} & H_{N2} & \cdots & H_{NM} \end{bmatrix} \tag{3.26}$$

By assuming that the receiver has the information of the channel matrix H, the V-BLAST algorithm is implemented as follows [71]:

- Build a Moore pseudo inverse matrix of $\boldsymbol{H}$,

  $\boldsymbol{G = H^{+} = (H^{*}H)^{-1}H^{*}}$

- Find the row of G where its Euclidean norm is the smallest one,

  $k =$ arg $min_j \; \|G_j\|$

  and j is the column of matrix $\boldsymbol{G}$. For the TDCS system the sum is applied for N_{FFT} samples since one transmitted symbol is represented by N_{FFT} samples.

- Take the row k of $\boldsymbol{G}$ as the nulling vector w,

  $w = (\boldsymbol{G})_k$

- Obtain the strongest transmit signal,

  $r_k = \boldsymbol{w} * \boldsymbol{y}$

  for the TDCS system r_k will be a vector of size N_{FFT}.

- The estimated value of the transmitted symbol $\hat{s}_k$ is derived by PSK or QAM demapping of r_k for OFDM while for TDCS the vector $\boldsymbol{r}_k$ will be decorrelated by the reference signal.

- After detection of the strongest transmitted signal, its effect must be cancelled from the received signal vector to reduce the detection complexity of the remaining signals, the symbol $\hat{s}_k$ is mapped again by PSK or QAM mapping for OFDM while in TDCS $\hat{s}_k$ is mapped into its frequency domain representation,

  $\boldsymbol{y} = \boldsymbol{y} - (\boldsymbol{H})_k * Mapping(\hat{s}_k)$

  where here k is the column index. The k-th column of channel matrix $\boldsymbol{H}$ is then zeroed for the purpose of detection of the strongest transmitted signal on the next layer. The algorithm returns to step 1 until the transmitted symbols on all layers are detected.

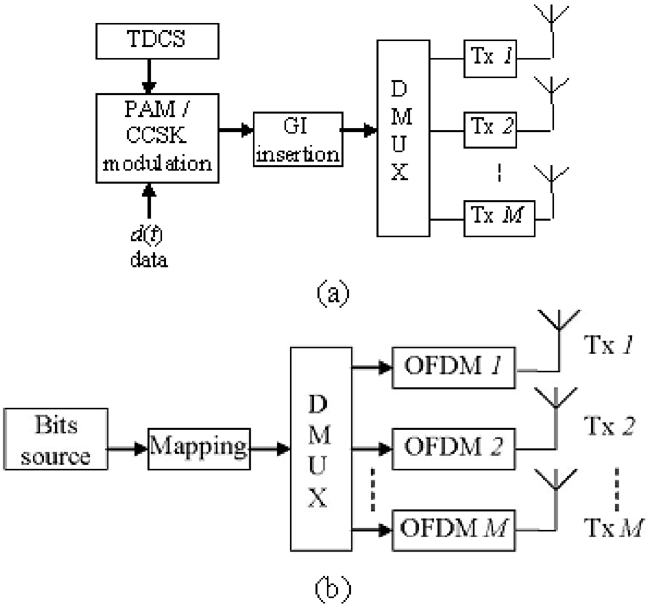


Figure 3.30: MIMO Transmitter architecture for CR (a) TDCS and (b) OFDM based system.

3.6.1 Simulation Results and Discussions

We have simulated MIMO OFDM and MIMO TDCS with parameters given in Table 3.7. The results in Fig. 3.35 show that the BER of the CR TDCS based system improves its performance and even outperforms the OFDM based system as the number of receive antennas and the SNR are increased. The achieved SNR gain of TDCS is more significant compared to OFDM system as the number of receive antennas increases while the number of transmit antennas remains the same.

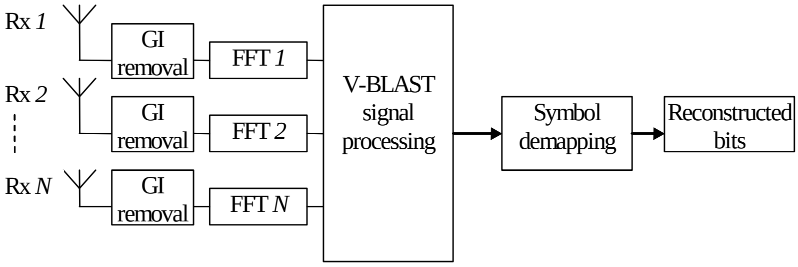


Figure 3.31: MIMO Receiver architecture for CR TDCS and OFDM based system.

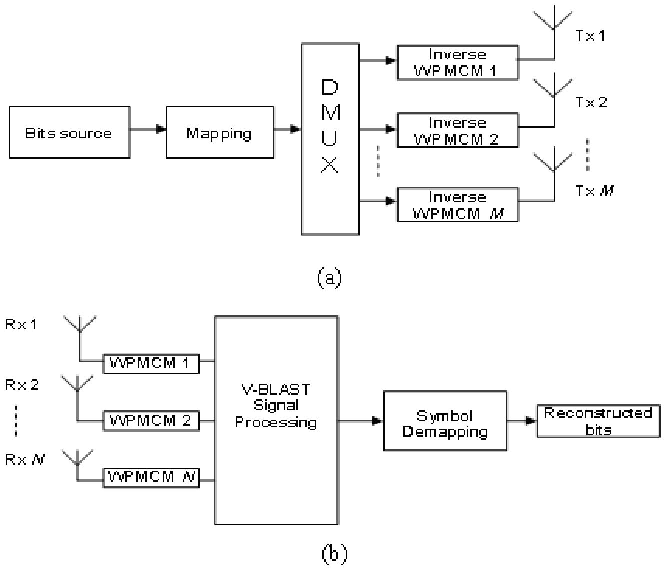
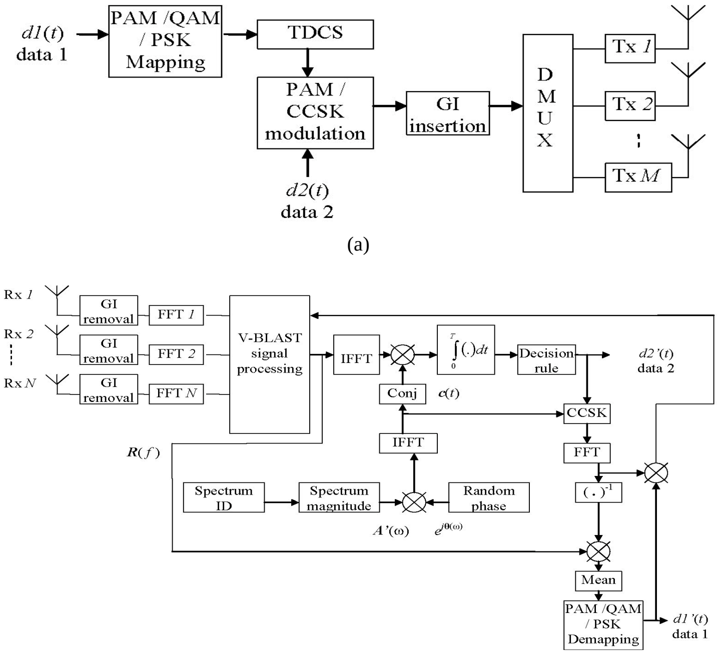


Figure 3.32: MIMO architecture for WPMCM based CR system (a) Transmitter (b) Receiver.

For the MIMO Wavelet packet multicarrier modulation (WP-MCM), an AWGN channel is used. Both the OFDM and WP-MCM scheme use the same set of transmission parameters - 128 carriers each with 40 carriers in the interference band removed. The wavelet used belongs to the family of maximally frequency selective filters. The curves (Fig. 3.36) are for the MIMO configurations 1x1, 2x2 and 3x3. For all the cases considered the WPMCM setup performs comparably well as its OFDM counterpart.

The application of MIMO to the TDCS with an extra embedded symbol (higher rate TDCS) was simulated using the parameters provided in Table 3.7. Fig. 3.37 shows the performance of the CCSK detection. In line with the finding that we have for the single input single output higher rate TDCS shown in subsection 2.2, we see that as the constellation size increases the CCSK correlation property is enchanced, particularly for the PSK embedded symbol. Adding MIMO with the composition of more receive antennas compared to transmit antennas improves the performance even further.



(a)

(b)

Figure 3.33: (a) MIMO Transmitter and (b) V-BLAST Receiver architecture of TDCS with extra embedded symbol.

N_{Tx}	R_{TDCS}		R_{OFDM}
	2 PAM	128 CCSK	
1	26.8 Kbps	187.6 Kbps	2.362 Mbps
2	53.6 Kbps	375.2 Kbps	4.724 Mbps
3	80.4 Kbps	562.8 Kbps	7.086 Mbps

Table 3.8: Bit rate comparison

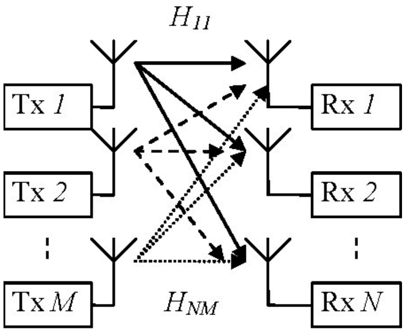


Figure 3.34: MIMO Channel model.

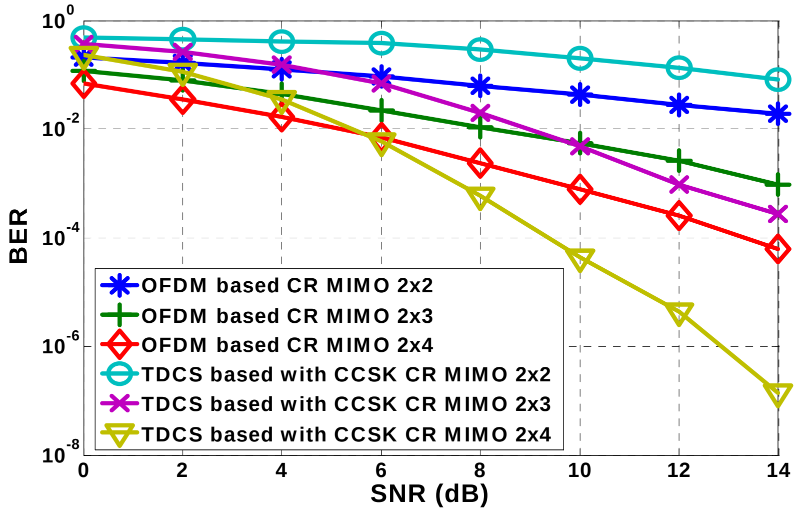


Figure 3.35: BER vs SNR (dB) performance of CR TDCS-CCSK, OFDM based with BPSK in MIMO V-BLAST architecture with 2 Tx antennas and different number of Rx antennas.

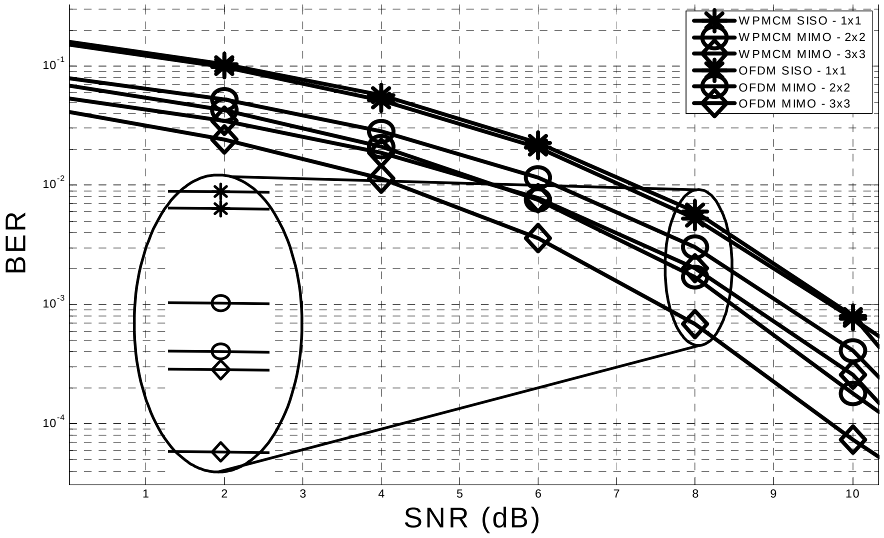


Figure 3.36: OFDM versus WP-MCM based CR. BER performance comparison of OFDM and WP-MCM based CR systems in the presence of LU. MIMO stands for Multiple input-multiple output, SISO for single input-single output.

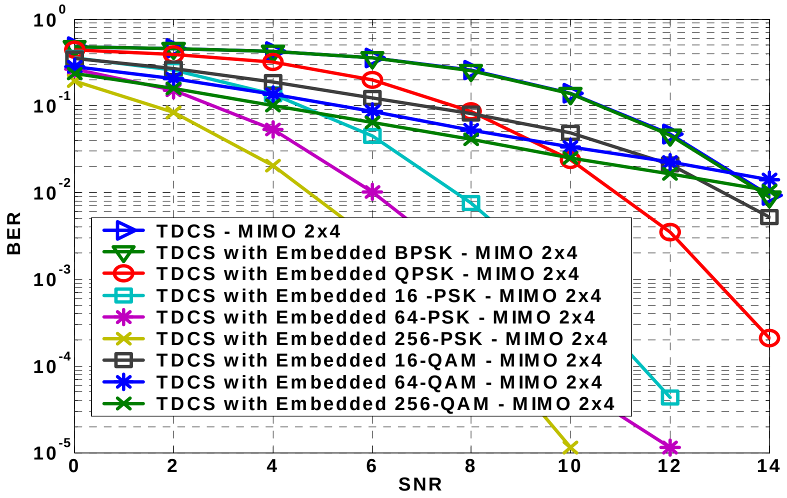


Figure 3.37: SNR (dB) vs BER performance on source data2 of the TDCS-CCSK with embedded QAM and PSK symbol in Rayleigh fading channel with 2x4 MIMO.

3.6.2 Publications with Respect to The Topic

- I. Budiarjo, H.Nikookar, L.P. Ligthart,"Cognitive Radio with Single Carrier TDCS and Multicarrier OFDM Approach with V-BLAST Receiver in Rayleigh Fading Channel", in *Proc. 2nd International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM)*, August 1-3, 2007, Orlando, Florida, USA.

- I. Budiarjo, H. Nikookar and L.P. Ligthart, "Cognitive Radio with Single Carrier TDCS and Multicarrier OFDM Approach with V-BLAST Receiver in Rayleigh Fading Channel", to appear in *ACM / Springer Mobile Networks and Application (MONET) Journal special issue on Cognitive Radio Oriented Wireless Networks and Communications*, 2008.

- M. K. Lakshmanan, I. Budiarjo and H. Nikookar," Wavelet Packet Multi-carrier Modulation MIMO Based Cognitive Radio Systems with VBLAST Receiver Architecture", to appear in *Proc. IEEE Wireless Communications and Networking Conference (WCNC)*, March 31 - April 3, 2008, Las Vegas, USA.

Parameter	Value
Topology	Mullti-hop
Modulation scheme	OFDM
Minimum frequency (RF) f_{min}	400 MHz
Maximum frequency (RF) f_{max}	900 MHz
Transmit power	30 dB(m)
Minimum channel SNR	15 dB
Minimum bandwidth B_{min}	1 MHz
Maximum bandwidth B_{max}	10 MHz
Maximum number of carriers	128
Maximum Doppler frequency f_D	15 Hz (at 900 MHz, 18km/h) 125 Hz(at 900 MHz, 150 km/h) 250 Hz(at 900 MHz, 300 km/h)

Table 3.9: Demonstrator Requirements

3.7 Demonstrator for AAF Cognitive Radio Systems

According to [55] the required parameters to be fulfilled for the demonstrator are given in Table 3.9. By setting the desired bandwidth(B) to 10 MHz, the sampling frequency (f_s) refering to the Nyquist rule is 20 MHz. The minimum desired number of bits b for the digital to analog and analog to digital converter (DAC and ADC) is formulated as :

$$b \cong \frac{20log(1 + \frac{f_{min}}{\Delta f}) + 4.77}{6.02} \tag{3.27}$$

where f_{min} is the minimum channel frequency and Δf is the carrier spacing. The log term is the dynamic range. Due to desired $B = 10$MHz, a number of carriers of 128 will make the $\Delta f = 78.125$KHz. If the $f_{max} = 900$MHz, then $f_{min} = 890$MHz. Filling in these parameters to (3.27) would give $b = 14.26$.

Among several available boards we have selected P25M as the suitable platform to implement the AAF demonstrator. Table 3.10 lists the ADC an DAC specifications of P25M board.

The block design of P25M is represented in Fig. 3.38, while the process scheme in P25M is represented by Fig. 3.39. In the beginning we will try to implement a simple OFDM based cognitive radio scheme with spectrum pooling. Spectrum pooling will be simply be the time domain windowing combined with adaptive deactivation of carriers located on the occupied licensed user's band. The scheme is comparable with the one depicted in Fig. 3.2. The OFDM frame itself will be designed according to Fig. 3.12(a). The channel estimation will be 1-D. By assuming the channel does not change abruptly during the frame period, the channel estimates are derived by a least square method, dividing the received signal on pilot positions by the pilot symbols. The channel estimates on this pilot OFDM symbol will be used to remove the fading effect on the rest of the OFDM symbol on that OFDM frame.

The schematic signal from of the OFDM based CR demonstrator is depicted in Fig. 3.40, while Fig. 3.41 shows the TDCS based CR demonstrator.

Parameters	ADC component	DAC component
Number of channels	4	4
Sample rate	25 MSPS	50 MSPS
Max Bandwidth	12 MHz	25 MHz
Bits width	16	16
Connection	To FPGA	To FPGA
Power Dissipation	unknown	unknown

Table 3.10: Innovative Integration P25M Specifications

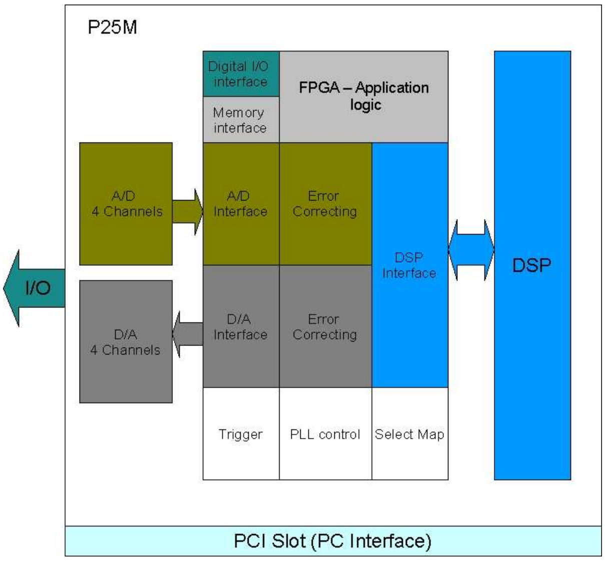


Figure 3.38: P25M Block design.

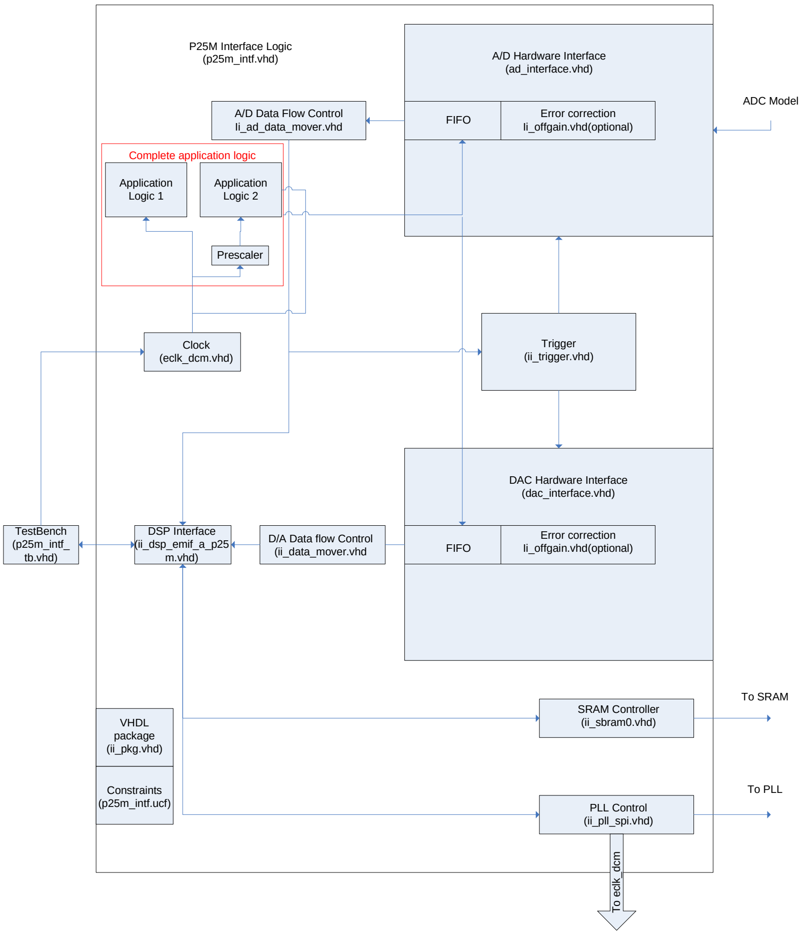

P25M Interface Logic (p25m_intf.vhd)
A/D Hardware Interface (ad_interface.vhd)
ADC Model
A/D Data Flow Control Ii_ad_data_mover.vhd
FIFO
Error correction Ii_offgain.vhd(optional)
Complete application logic
Application Logic 1
Application Logic 2
Prescaler
Clock (eclk_dcm.vhd)
Trigger (ii_trigger.vhd)
DAC Hardware Interface (dac_interface.vhd)
TestBench (p25m_intf_ tb.vhd)
DSP Interface (ii_dsp_emif_a_p25 m.vhd)
D/A Data flow Control (ii_data_mover.vhd
FIFO
Error correction Ii_offgain.vhd(optional)
VHDL package (ii_pkg.vhd)
Constraints (p25m_intf.ucf)
SRAM Controller (ii_sbram0.vhd)
To SRAM
PLL Control (ii_pll_spi.vhd)
To PLL
To eclk_dcm

Figure 3.39: P25M process schematics.

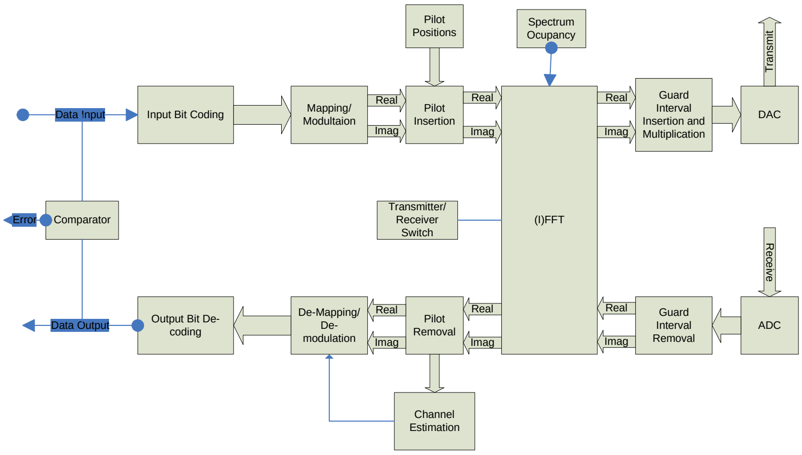


Figure 3.40: Signal flow of IRCTR-AAF OFDM based Cognitive Radio Demonstrator.

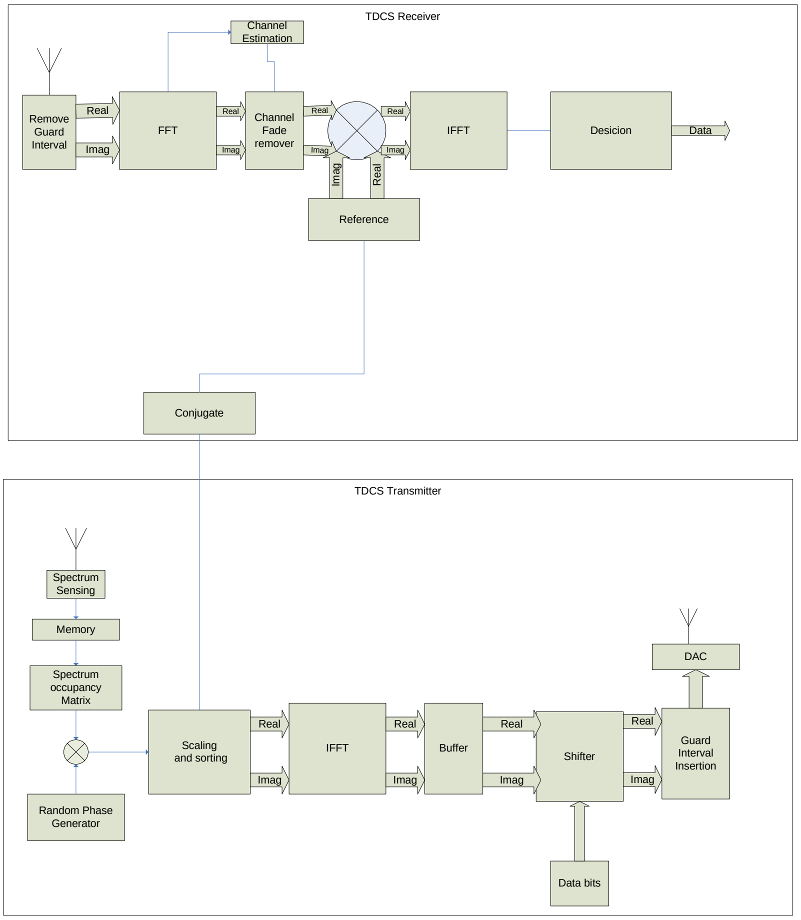


Figure 3.41: Signal flow of IRCTR-AAF TDCS based Cognitive Radio Demonstrator.

3.8 Conclusions

In this chapter we have reviewed the performance of several dynamic spectrum access techniques. OFDM is considered as the strong modulation mode to be applied in CR systems. The core optimization problem is the reduction of the interference quantity of the CR system due to its spectrum shape on the licensed system band. We have observed several available CR spectrum shaping techniques such as windowing, carriers deactivation, power allocation and the wavelet approach. The cost of spectrum shaping techniques in terms of bit rate reduction can be compensated by adding MIMO architecture to the system, which also requires extra effort.

Chapter 4

Spectrum sensing for opportunistic radio spectrum access

by Marnix Heskamp

4.1 Introduction

The research presented in this chapter is on spectrum sensing. Spectrum sensing is a key concept in cognitive radio, as it provides the input for the cognition cycle [72] that defines a cognitive radio.

At first glance, spectrum sensing may seem like a simple engineering problem. The main difficulty, however, is that a spectrum sensing system for cognitive radio must be orders of magnitude more sensitive than a normal receiver, in order to detect all relevant signals [73]. In fact, the main reason that prevents cognitive radios from wide scale deployment may be the hidden node problem [1].

There are two main research questions we address in this chapter. First we want to know what 'white space' actually looks like, and what should be the requirements are for a spectrum sensing system. Second, we want to know how much can be gained from smart algorithms that make use of the fact that radio signals are not completely random, but contain man-made artifacts like cyclostationarity.

This chapter is organized as follows: Section 4.1.1 gives a qualitative overview of current day spectrum usage, and establishes a frequency range in which the AAF system should operate. Next, in section 4.2, we introduce a simulation platform for testing detection algorithms for spectrum sensing. Radio propagation models that were implemented in the simulator are described in section 4.3. An overview of detection theory for spectrum sensing is given in section 4.4 along with first results from the simulation platform.

4.1.1 Spectrum usage

The choice of frequency for a cognitive radio system is important because it will have impact on all other system aspects. First, the transceivers must be dimensioned for the propagation conditions of a certain band. Second, different bands have different primary users which require different approaches of interference avoidance. Since it is unlikely to find enough bandwidth below 100 MHz, we do not consider those frequencies. Since an upper bound is much harder to give, and experimentation is easier done on lower frequencies, we first consider low frequencies.

Figure 4.1 shows a rough overview of the bandplan in the Netherlands between the FM and TV band. Between 88 and 108 MHz we find the FM broadcasting band. Signals in this band are generally very strong so that they will probably drown out the much weaker cognitive radio signal. Therefore we consider this band to be too hostile for the AAF system. Directly above the FM band we find the aviation band which appears much more quiet than the FM band, but because of the mission critical nature of this band it must be considered unsuitable for cognitive radio.

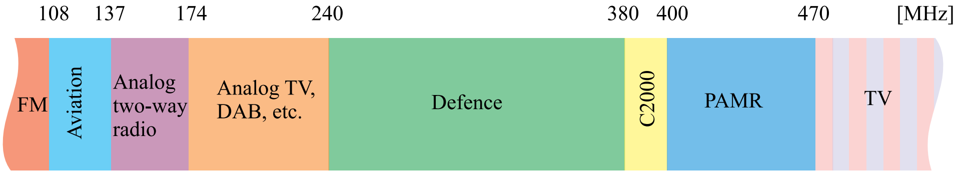


Figure 4.1: Bandplan

The band between 137 and 174 MHz is typically used for two-way radio communications for all sorts of purposes, like communication on construction sites, public transportation and taxies. Typical equipment used in this band are ‘push-to-talk’ radios that use analog FM in a 25 kHz raster. Because of this old-fashioned technology there is a lot of white space in this band, that potentially could be used by a cognitive radio.

The band between 174 and 240 MHz was traditionally used for television (VHF Band III) but is now used for various purposes and for Digital Audio Broadcasting (DAB). The physical layer aspects of DAB can be useful for the AAF system design, because like AAF it uses an OFDM based physical layer and interference issues of DAB are well studied.

Directly after VHF band III, a large military band follows which we will not consider for AAF. Between the 380 and 400 MHz we find the radio system for emergency service and public safety called C2000. It is based on the TETRA standard from ETSI. This system is important for the AAF project because it implements many AAF requirements, except that it can not search for additional free spectrum, has no broadband functionality, and is designed with a fixed infrastructure in mind.

Between 400 and 470 MHz we find a second band for various applications. Typical usage is for Public Access Mobile Radio (PAMR). A PAMR network is a trunked radio system that is operated by a telecom service provider that licenses capacity to its customers. Digital PAMR systems often use TETRA and older networks often use the analog MPT1327 standard. The service of PAMR is somewhat similar to that of a cellular phone but it is faster, more reliable and offers special facilities like user groups.

Between 470 and 846 MHz we find a very large band for television (TV) broadcasting (UHF Band IV and V). It contains 47 channels that are 8 MHz wide, which in the past only contained analog TV signals. Because TV signals carry relatively far because of high transmit powers and high antennas, and analog TV is very sensitive to interference, in the past most of the channels remained unused in order to prevent interference. Since the beginning of 2007 analog TV is switched off in the Netherlands in favor of Digital Video Broadcasting (DVB). Because DVB is much more robust against interference and can combine multiple TV streams in one channel, now a lot of the TV bands are empty. The white space between TV channels is relatively easy to use for cognitive radio, because of its static nature. In the United States the upper part of the TV spectrum between 700 and 800 MHz is reassigned from television to public safety, and on 17 February 2009 analog broadcast will be switched of. Other bands of the TV spectrum are considered to open up for the IEEE 802.22 standard.

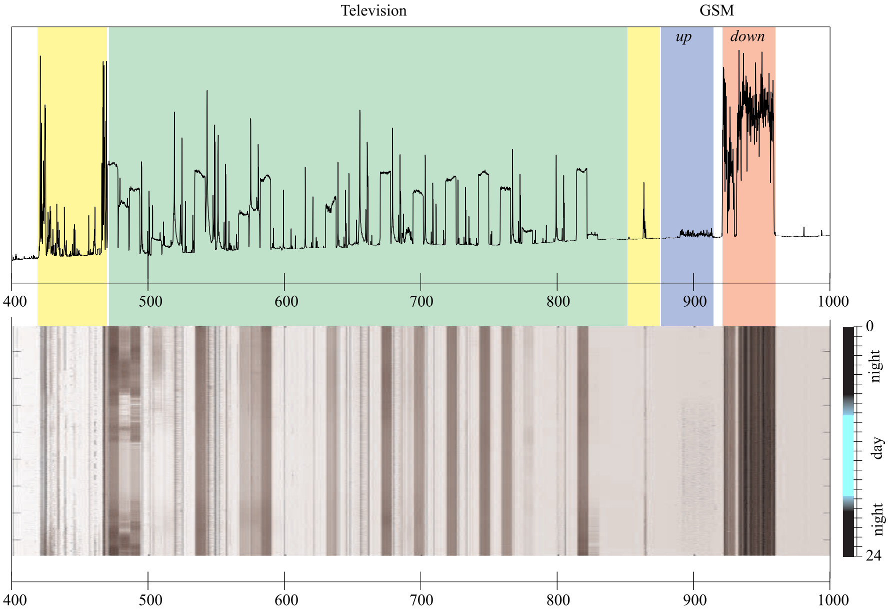


Figure 4.2: Spectrogram measured in Eindhoven in October 2005.

Figure 4.2 shows an overview of the radio spectrum between 400 and 1000 MHz based on spectrum measurements from Agentschap Telecom (AT). In the upper part of this figure the average field strength is plotted. In the lower part the field strength spectrogram is plotted as a gray scale bitmap. In the television band one can clearly identify the 8 MHz wide channels of analog and digital television. Around 900 MHz the GSM-900 band can be seen. The band between 876 and 915 MHz contains the 194 uplink channels with a bandwidth of 200 kHz. The band between 921 and 960 contain the downlink channels. It is interesting to note the large difference in magnitude between the up and downlink. This can partially be explained by the fact that the basestations are on a elevated position and thus more likely have a line of sight with the measurement antenna which was placed 50 m above the ground. Furthermore, mobile handsets are designed to minimize transmit power and basestation have more sensitive receivers.

Figure 4.3 shows a close-up of the GSM uplink band. Most activity is in the

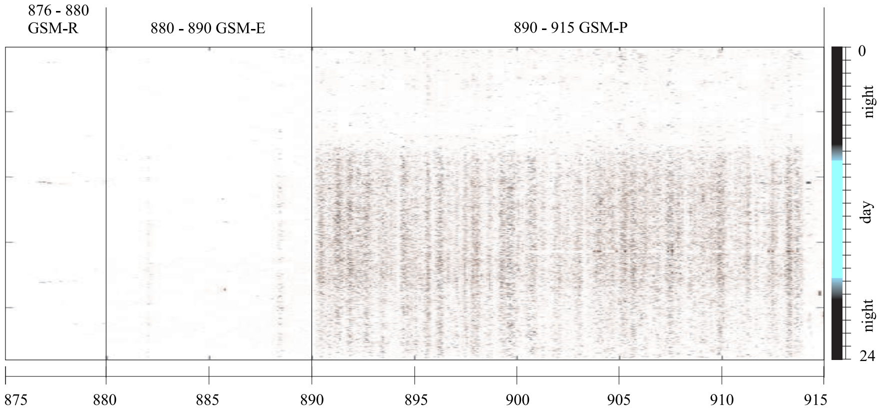


Figure 4.3: Close up of the GSM up-link band.

primary GSM band (GSM-P), although there is some sporadic usage visible in the extended (GSM-E) and the railway (GSM-R) bands. In this figure a remarkable correlation is visible between sunrise (on 8:05) and sunset (on 18:45) and the apparent usage of the spectrum. From Figure 4.2 it is clear that the GSM basestations have a more or less constant power level during a twenty-four hours period. This also explains why the uplink has a higher average power.

In Figure 4.4 the average field strength (blue line) and the maximum field strength (red line) over 24 hours are plotted. From this figure it becomes clear that the uplink band has a much higher peak to average ratio than the downlink band. If the maximum field strengths are compared, the uplink is only about 15 dB weaker than the downlink.

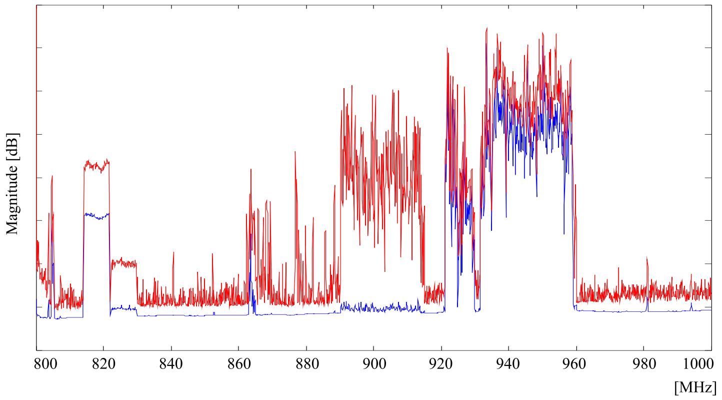


Figure 4.4: Average (blue line) and maximum (red line) field strength values

Figure 4.5 shows the field strength over 24 hours on the 900 MHz up link frequency and the 945 MHz down link frequency. Figure 4.6 shows a close up between 12:00 and 13:00.

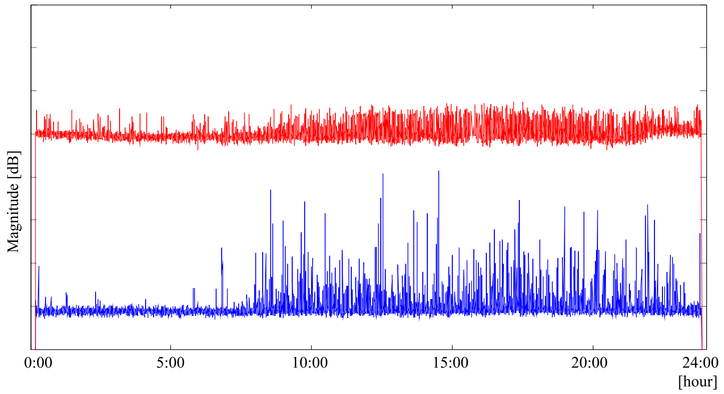


Figure 4.5: Twenty-four hours field strength plot in the GSM uplink band on 900 MHz (blue line) and the downlink band on 945 MHz (red line)

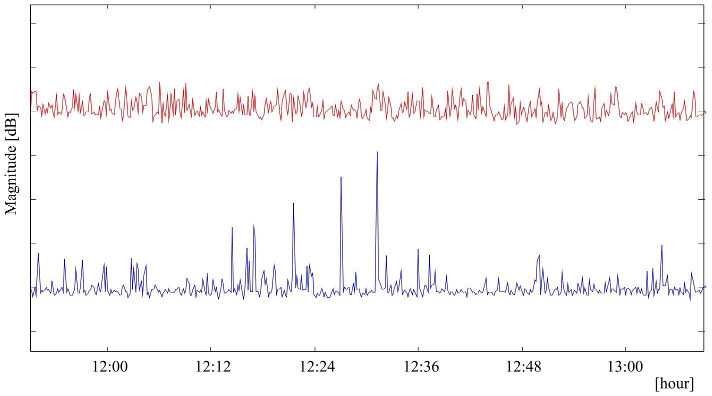


Figure 4.6: Close up of Figure 4.5

4.2 The Simulator

In this section a simulation platform is described for testing cognitive radio spectrum sensing algorithms. Such a simulator is needed because it is too difficult to perform cognitive radio experiments in the real world, since one has no control over propagation and primary users. The main purpose of the simulator is to test detection algorithms under reasonably realistic conditions. To simplify the simulation, only the first order effects of the analog world are taken into account. The include path loss, noise and quantization, but not phenomena that have a wide frequency span like intermodulation, because all simulation where done on digital baseband representations.

4.2.1 Programming environment

The programming language that was used to implement the simulator was C++. Its main advantage over for example C and MATLAB is that C++ is an object oriented language. Object oriented languages are better suited for implementation of simulators than procedural languages, because class instances have a natural correspondence with physical objects. For example, filters and channel models have an internal state that must be retained between successive function calls. In a procedural languages this internal state must be returned after each function call and stored somewhere. If multiple filters and channels are used in one simulation, the bookkeeping of these internal states becomes cumbersome. However, if a filter or channel is implemented as a C++ object, the internal state resides inside the object, along with other implementation details that are kept hidden from the user of the object.

All simulations are performed on time discrete complex baseband representations of the RF signals. C++ itself has no build in support for complex numbers, so the `complex` class from the standard library was used, which internally represents a complex number as two double precision floating point numbers.

To prevent function call overhead the processing of the samples is done on blocks of samples. In C++ there are various ways to form a block of samples, like static or dynamic arrays or standard library container classes. We have chosen to use the vector object provided by the IT++ library [74]. IT++ is a open source signal processing library targeted at simulations of communication systems. It defines vector and matrix objects along with a large collection of signal processing functions. Inside the vector object the samples are stored contiguously and a pointer to the raw data can be retrieved. This makes it possible to bypass any overhead for performance critical sections of the code.

4.2.2 The analog to digital converter

The analog to digital converter (ADC) is a significant bottleneck of every digital radio system, since it inherently distorts the signals. One can divide the ADC distortion in three components: sampling, clipping and quantization. A typical ADC has a voltage range from 2 Volt peak-peak and an impedance of 1 kΩ. If we assume that the amplitude distribution is Gaussian we can clip the signal above 4 times the RMS voltage without causing much distortion. This leads to a maximum ADC input power level of

$$P_{max} \;=\; \frac{V^2}{R} \;=\; \frac{0.25^2}{1000} \;=\; -12\ \mathrm{dBm}. \tag{4.1}$$

Quantization will add quantization noise to the signal, which has a uniform frequency and amplitude distribution if the quantization is not too coarse, and an RMS voltage of

$$V_{noise} = \frac{\Delta V}{\sqrt{12}} \tag{4.2}$$

in which the quantization step size $\Delta V = V_{pp} \cdot 2^{-bits}$. If our ADC has 14 bits the average quantization noise power is -89 dBm, so the effective dynamic range is about 77 dB.

In order to prevent aliasing, the ADC in each quadrature channel must sample at a rate equal or higher than the signal bandwidth. Because the frequency response of the channel selection filter falls off with only 20 dB per decade per section, a factor of over sampling is needed to prevent interference.

The power consumption of an ADC is given by

$$P \;=\; \mathrm{FoM}\; f_s \cdot 2^{bits} \tag{4.3}$$

where f_s is the sampling frequency and $bits$ the number of resolution bits. FoM is a figure of merit that for current day ADC is about 0.6 pJ per conversion step, and approximately halves every three year.

With this figure of merit one can estimate a reasonable bandwidth for the sensing receiver. A typical battery of a handheld device can provide 2000 mAh at 4 V. If we want to be able to use the device for 8 hours without recharging and spend 10% of the power budget on the ADC, the power consumption of the ADC is 100 mW. A dynamic range of 14 bits will cost 16384 conversion steps per sample. With a FoM of 0.6 pJ we can have a sampling bandwidth of about 10 MHz. With an 8-th order anti-aliasing filter this gives a clean digital bandwidth of 2.5 MHz.

4.2.3 Signal sources

In the simulator multiple signal sources are used. All sources are derived from a `Source` base class. This base class defines the functions `SetGain()`, `SetFreq()` and `GetSamples()`. If the gain is set to one, a source is calibrated to produce unity average power. The frequency is normalized between -0.5 and 0.5. The corresponding real world frequency at RF can be found by multiplying the relative frequency with the sampling rate and adding the RF center frequency.

If, for example, 1024 OFDM samples at a relative frequency of 0.4 are needed, the following source lines could be used:

```
cvec u1;
u1.SetSize(1024);
Source* s = new OFDMSource();
s->SetFreq(0.4);
s->GetSamples(u1);
```

in which `cvec` is a complex IT++ vector. The `OFDMSource` object maintains its internal state, so that an unlimited sequence of samples can be obtained by calling `GetSamples` repeatedly.

Sinusoidal source

One of the most elementary signals are the sinusoids. They can represent an unmodulated carrier or a local oscillator signal in an other signal source. In a complex baseband representation sinusoidal signals are represented as complex exponentials of the form

$$u[n] \;=\; \exp(j2\pi fn) \;=\; \cos(2\pi fn) + j\sin(2\pi fn). \tag{4.4}$$

If explicitly a $\cos()$ or $\sin()$ signal is needed it can easily be obtained by taking the real or imaginary part respectively. When used as a signal source (4.4) is always evaluated for successive values of n. This makes it possible to generate the samples recursively by

$$u[n] \;=\; u[n-1]\cdot dz \qquad \text{with} \;\; dz \;=\; \exp(j2\pi f) \tag{4.5}$$

which is much faster than evaluating the $\sin()$ and $\cos()$ functions.

In the simulator, a sinusoidal is called `ComplexExpSource`. All other sources have internally a `ComplexExpSource` object to generate an oscillator signal that sets the output frequency.

White noise source

To model the noise floor of the receiver a white Gaussian noise source is needed. The easiest way to generate white noise in a computer simulation is with a pseudo random number generator. There are two reasons to use a pseudo random source instead of a true random source. First a truly random sequence can not be generated digitally. Secondly, a pseudo random signal source can be reset to a known initial condition so that it can reproduce a sequence. This provides the practical convenience that simulations can be exactly reproduced many times so that not all results have to be collected at one time. This makes it also possible to estimate ensemble averages. The uniformly distributed samples of the random number generator can be easily converted to Gaussian samples. Because the samples from a random number generator are uncorrelated, its power spectral density is white.

Bandpass noise source

Pure sinusoids and sums of sinusoids are not a good model of a radio signal since they can not convey information. Furthermore, pure sinusoids do not exist in practice, because the amplitude and phase will always have some random fluctuations caused by noise. So, even if a signal with a very narrow bandwidth is needed, it is not advisable to use a pure sinusoid. These drawbacks of sinusoids can be solved by multiplying them with a slowly varying random function $c[n]$

$$u[n] \;=\; c[n]\;\exp(j2\pi f n). \tag{4.6}$$

The function $c[n]$ is complex valued, so that it includes both amplitude and phase fluctuations. It is generated by passing pseudo random noise $w[n]$ with Gaussian distributed uncorrelated samples, through a low pass filter with impulse response $h[n]$

$$c[n] \;=\; w[n] * h[n]. \tag{4.7}$$

The advantage of the bandpass noise source is that it provides a good model for a signal that approaches the theoretical limit of spectral efficiency. Because it is generated by a LTI system, it is wide sense stationary, so that there are no correlation between frequency bins. This makes the bandpass noise source a good test signal for cyclostationary feature detectors, because it is supposed to have no features at all. So, if an ensemble of white noise and bandpass noise sources is applied to a cyclostationary feature detector it must show now response.

4.2.4 Interpolated sources

The complex bandpass noise source has two main disadvantages. First the filtering operation forms a large computational load because it operates on full bandwidth noise, even if the resulting signal is narrow banded. Secondly, it does not contain the modulation artifacts that are typical for most real world radio signals. For these reasons all other signal sources in the simulator are formed by interpolation:

$$c[n] \;=\; \sum_{m=0}^{M-1} d[n \,-\, mR]\, p[m] \tag{4.8}$$

in which $d[n]$ is the signal that has to be interpolated, $p[m]$ the interpolation kernel with duration M, and R the interpolation ratio. In the simulator the interpolation is done with an `Upsampler` object. Note that $d[n]$ can be regarded as a baseband representation with respect to the full bandwidth signal $c[n]$, which itself is a baseband representation of the RF signal. After the interpolation the resulting narrow band signal is mixed with a complex exponential, to shift it to the desired frequency. Interpolation ratios used in the simulator are 16 for wide band signals and 256 for narrow band signals.

Frequency modulation

One of the oldest known modulation types is frequency modulation (FM) which also today is still widely used. An FM signal $u(t)$ can be described by

$$u(t) \;=\; \exp(j2\pi f_c t)\; \exp(\, j2\pi \int_{-\infty}^{t} m(\tau)d\tau\,) \tag{4.9}$$

in which f_c is the RF center frequency and $m(t)$ the analog audio or message signal.

The digital baseband representation of this signal is

$$d[n] \;=\; \exp(\, j2\pi \sum_{k=-\infty}^{n} m[k]\,). \tag{4.10}$$

In the simulator we use an audio file with a sampling rate of 4 kHz for the message signal, which is first up-sampled with a factor 16. Next (4.10) is applied to modulate the signal. This modulated signal is then again up-sampled with a factor 16 and frequency shifted. So the (pseudo) source code for the `FM_Source` objects is

```
AudioSource->GetSamples(u1);
Upsampler1->Process(u1, u2);
FM_modulator->Process(u2, u3);
Upsampler2->Process(u3, u4);
osc->GetSamples(u5);
out = u4*u5;
```

in which `FM_modulator` implement (4.10).

QAM, PSK and OFDM

Phase shift keying (PSK) and quadrature amplitude modulation (QAM) are special types of pulse amplitude modulation (PAM), in which the amplitude is restricted to a complex constellation. For example, in the simulator PSK or QAM samples are generated by the following pseudo code:

```
b1 = rand();
u1 = modulator(b1);
Upsampler->Process(u1, u2);
osc->GetSamples(u3);
out = u3*u2;
```

in which `modulator()` maps bits to complex symbols. OFDM can be generated in a similar way, by applying an FFT on blocks of samples and adding a cyclic prefix.

4.2.5 Hardware signal sources

When designing a simulation, it is difficult to capture all relevant aspects of reality. For example, it is difficult to simulate a source and receiver that work on sampling frequencies with a true non-integer ratio, which in fact is always the case in real world situations. In the simulation this will lead to implicit synchronization between source and receiver. Especially in the study of sensitive detection algorithms in which a large number of samples are used, this implicit synchronization may lead to overly optimistic results.

One of the main advantages of software defined radio is that a simulation can be easily converted into an implementation. The only thing that is required is a general purpose analog front-end, analog to digital converters, and digital IO functionality. The simulator described in this section can be connected to a Universal Software Radio Peripheral (USRP), and to proprietary AD-converter hardware that is used at the Signals and Systems group. In the simulator, a hardware signal source is handled in the same way as a simulated signal source. The main distinction is that a call to `GetSamples()` will block until the samples are available, and that samples are missed if `GetSamples()` is not called frequently enough.

Figure 4.7 shows the spectrum of TETRA signals from the C2000 downlink band obtained from a hardware signal source. The signals where received trough a television tuner analog front-end. The 43.75 MHz intermediate frequency (IF) signal from the tuner was down converted to baseband and digitized with a sampling rate of 8 MHz. Figure 4.7a shows the power spectral density plot of of the full captured bandwidth. The slanting edges are caused by the SAW filter of the tuner and the anti-aliasing filters before the ADC. It shows a part of the C2000 downlink band, which extends from 390 to 395 MHz. Because the full 8 MHz signal was too wide for easy processing, we down sampled it by a factor 16, as can be seen in Figure 4.7b. In section 4.4.10 we use this signal to demonstrate the extraction of cyclostationary features.

4.3 Radio propagation

In radio communication power is transported as electromagnetic waves consisting of a magnetic field **H** and an electric field **E**. The product of those quantities is the power

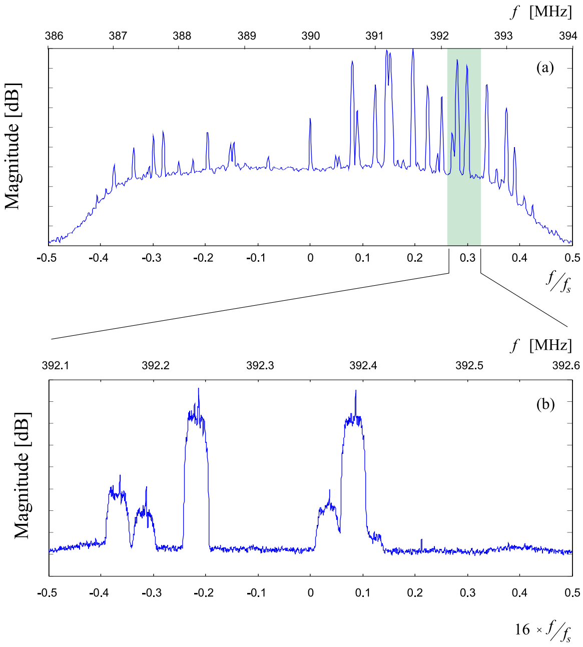


Figure 4.7: (a) TETRA signals in the C2000 downlink band. (b) Close up of a 500 kHz sub-band centered around 392.35 MHz, showing two strong signals and three weaker signals.

flux $\mathbf{S} = \mathbf{E} \times \mathbf{H}$, which is the amount of power that passes through an area. The magnitude of the $\mathbf{E}$ and $\mathbf{H}$ field will vary if the wave goes through differed media like vacuum or air, but the power remains constant. When received by an antenna the electromagnetic signal is converted into a voltage V and current I. The resulting power $P = V \cdot I$ will be equal to the electromagnetic power flux $\mathbf{S}$ multiplied by the effective area A_{eff} of the antenna. For parabolic dish antennas the effective area is similar to its physical dimensions, but for all other antennas the effective area is only a virtual area which represents the antennas ability to collect electromagnetic flux.

There are two important effects that lead to corruption of a radio signal. First, electromagnetic signals tend to propagate in three dimensions, so only a tiny fraction of the transmitted power arrives at the receiver. This would not be a problem if there was not a second fundamental problem, which is noise. Signals can carry hundreds of kilometers through space without picking up any significant noise, but as soon as the signal is inside the receiver front-end it is easily lost in thermal noise.

Every radio receiver has a noise floor caused by thermal noise equal to $P_{\text{noise}} = k\,T$, in which T is the absolute temperature, and $k = 1.3807 \times 10^{-23}$ $\text{Joules}/\text{Kelvin}$ is the Boltzmann constant. On a room temperature of $15.2°\text{C}$ the thermal noise is -174 dBm/Hz. Besides the thermal noise, also the amplifiers and other electronics in the receiver add noise. The effect of noise is largest when the signal is weak, so in practice the noise produced by the first amplifier is dominant. For this reason, the first amplifier in a receiver is a low noise amplifier (LNA). Because of linearity, the noise contributions of all components can be combined in a single number called the noise figure (NF), which can be added to the thermal noise floor. A typical value for the NF of an LNA is around 2 dB.

Figure 4.8 shows a simplified schematic representation of an antenna frontend receiving a plane electromagnetic wave. In the electromagnetical domain, the antenna is represented by a square area that collects electromagnetic power. At its electrical terminals the antenna can be represented by its Thevenin equivalent circuit, consisting of a voltage source V_a and the antenna impedance. Note that the antenna impedance is a radiation impedance rather than a ohmic impedance, so it does not produce thermal noise. In order to convert all available electromagnetic power into electrical power, the antenna terminals must be connected to a matched load. This matching load is formed by the impedance of the low noise amplifier (LNA). This load impedance is an ohmic resistor, which produces thermal noise, represented by noise source V_{th}.

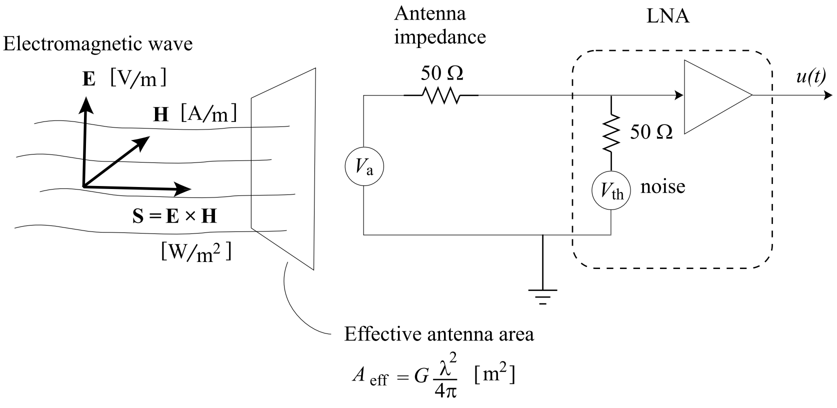


Figure 4.8: Simplified representation of the antenna input stage.

The antenna gain G determines the power conversion ratio between the electromagnetic and the electrical domain. Unfortunately G is not a simple constant. It is dependent on the angle of arrival of the signal, and also on objects near to the antenna. Therefore, in a handheld device, the received power may vary orders of magnitude depending on the position of the device.

4.3.1 Path loss

A good understanding of propagation is important for cognitive radio, because it determines over which distance interference can be caused. At UHF frequencies that are considered in this research, ionospheric reflections are not very common, so we assume that the range of a signal is in principle limited by the horizon. The distance to the horizon is approximately given by

$$d \approx \sqrt{2\,k\,r\,h} \tag{4.11}$$

in which k is a factor that corrects the bending around the earth which is about 4/3, r is the radius of the earth, which is about 6371 km, and h the height of the transmit antenna. So the signal of a 30 m high basestation can not be detected beyond about

23 km, and the signal of a handheld cognitive radio can not cause interference beyond about 5 km.

In free space RF energy is equally spread in three dimensions, which is described by the Friis transmission equation

$$P_{rx} = P_{tx}\, G_1\, G_2\, \left(\frac{c}{4\pi f d}\right) \tag{4.12}$$

in which c is the speed of light and G_1 and G_2 the antenna gain of the transmit and receive antenna respectively. This free space path loss can be expressed in decibels as

$$\text{Loss}(dB) \;=\; G_1(dB) + G_2(dB) + 147.56 - 20\log(f) - 20\log(d). \tag{4.13}$$

So, if the distance between transmit and receive antenna is doubled the signal level drops 6 dB and if the distance is increased by a factor of 10, the signal level drops 20 dB. Note that the dependency on frequency is caused by the fact that on higher frequencies the effective antenna area is smaller than for low frequencies. The fact that in practice low frequencies propagate further than high frequencies is mainly caused by the fact that low frequencies easier bend around obstacles like buildings.

The free space model is not valid on earth. Instead of a path loss exponent of 20 dB per decade, actual measurements are much closer to 40 dB per decade. The situations on earth can be described by the plane earth propagation model, which is given by

$$P_{rx} \;=\; 4\;\, P_{tx} \left(\frac{c}{4\pi f d}\right)^2 sin^2\left(\frac{2\pi\, f\, h_{tx}\, h_{rx}}{c\, d}\right). \tag{4.14}$$

Figure 4.9 compares the path loss predicted by the free space model with that of the plane earth model for a frequency of 400 MHz and antenna heights of 1.5 m. As can be seen from this figure, the plane earth model has two regions, that are separated by the Fresnel breakpoint distance d_0 which is given by

$$d_0 = \frac{4\,\pi\, f\, h_{tx}\, h_{rx}}{c} \tag{4.15}$$

which in this example is about 37 m. Before this distance the path loss varies wildly around the 20 dB/decade line, and after this distance it falls off smoothly with 40 dB/decade.

4.3.2 Channel fading

Fading caused by environment clutter can be considered as the multiplication of many uncorrelated attenuations of the signal. On a logarithmic scale these multiplication factors can be summed. From the central limit theorem it follows that the sum of uncorrelated terms has a distribution that is approximately Gaussian. Therefore, the attenuation in dB caused by shadow fading is a Gaussian random variable. This type of fading is commonly referred to as Lognormal or Shadow Fading.

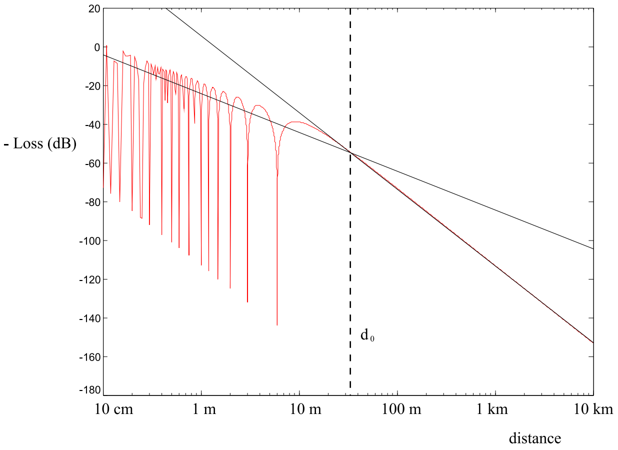


Figure 4.9: Negative of the Path loss (dB) as function of the distance.

4.3.3 The hidden node problem

The single most important issue in cognitive radio is the *hidden node* problem [1]. Spectrum sensing only gives information about the presence of transmitters, but interference is not caused to transmitters but to receivers. If a primary user does not transmit, it remains undetected by the cognitive radio. The signal strength of radio signals can vary tens of decibels over a short geographical distance because of fading. This means that it is easy to think of scenarios in which a single cognitive radio can not detect all meaningful signals. Such a scenario is sketched in figure 4.10. Cognitive radio network node 1 wants to send to node 2 and uses a channel that appears empty in its own sensing results. Because of shadow fading it can not detect the primary user transmitter on the other side of the wall. Therefore it is going to cause interference to the primary user receiver.

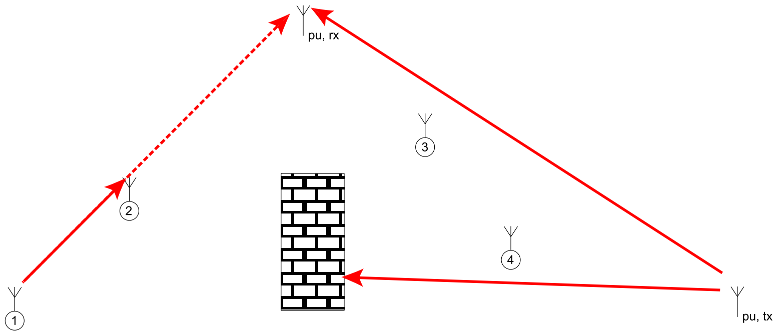


Figure 4.10: The hidden node problem

To mitigate the hidden node problem cooperative sensing is required. For example, in Figure 4.10 nodes 3 and 4, which can detect the primary user signal, should communicate their sensing results via node 2 to node 1. Multiple node sensing will only prevent interference if there is a sensing node behind each obstruction.

Cooperation of cognitive radios will not help much in bands where the signals are very weak, as for example in radio astronomy or satellite bands. For example, the signals used for the global position system (GPS) are already weak because the long signal path, however, the power spectral density is even further reduced by the spread spectrum modulation that is used. So, unless the correct de-spreading code is used a GPS signal is undetectable by a normal receiver. Based on bandplan assignments, a cognitive radio should avoid these frequencies. Cognitive radio should also avoid bands where the absence of a signal also has significance for example on channels that are used for alarms, or tactical operations by military and police.

On the other hand, bands that are particulary suitable for cognitive radio are those in which the receiver sends back acknowledgement signals to the transmitter, like for example IEEE 802.11.

4.4 Spectrum Sensing

At the heart of a cognitive radio resides the spectrum sensing function. It is responsible for detecting all relevant primary user systems. The main problem with all forms of spectrum sensing is that they detect the presence of transmitters, whereas interference is caused to receivers. Because of fading and hidden nodes this means that important primary users may be missed. On the other hand, the sensing receiver does not have to decode the signal in a way a normal receiver does. Therefore signals can be sensed that are too weak for detection by a normal receiver. Furthermore, the signal may be integrated over a considerable time interval. To make this document self contained and establish notation, in the next section the theoretical basis for the concept of *spectrum* is described. Next some spectrum sensing algorithms are discussed.

4.4.1 Filters

The idea of dividing the radio spectrum in frequency bands is based on the fact that signals can be selected and suppressed by *filters*. A system $\Theta()$ is a filter if it is both linear and time invariant (LTI), so

$$\begin{aligned} \Theta(\ a(t)\ +\ b(t)\ ) \quad &= \quad \Theta(a(t))\ +\ \Theta(b(t)) \\ \Theta(a(t)) = b(t) \quad \Longleftrightarrow \quad &\Theta(a(t+\tau))\ =\ b(t+\tau). \end{aligned} \tag{4.16}$$

If a Dirac delta function $\delta(t)$ is applied to the input of a filter it will produce the impulse response $h(t)$ on its output. From (4.16) and the definition of $\delta(t)$ it directly follows that the response $v(t)$ on an arbitrary input signal $u(t)$ is given by the convolution

$$v(t) = u(t) * h(t) = \int_{-\infty}^{\infty} u(\tau)h(t-\tau)\,d\tau \tag{4.17}$$

where $*$ denotes the convolution operator.

4.4.2 The frequency domain

The Fourier transform $\mathscr{F}$ is a linear coordinate transform to a basis of eigenfunctions of the convolution operator, which are sinusoidal functions, and is defined as

$$\begin{aligned} U(f) &= \mathscr{F}(\, u(t)\,) = \int\limits_{-\infty}^{\infty} u(t)\exp(-j2\pi f t)\, dt \\ u(t) &= \mathscr{F}^{-1}(\, U(f)\,) = \int\limits_{-\infty}^{\infty} U(f)\exp(j2\pi f t)\, df. \end{aligned} \tag{4.18}$$

The most important property of the Fourier transform is that it converts a convolution in one domain into a multiplication in the other domain. Equation (4.17) can therefore be written in the frequency domain as

$$V(f) = \mathscr{F}(v(t)) \;=\; \mathscr{F}(u(t)) \cdot \mathscr{F}(h(t)) \;=\; U(f) \cdot H(f). \tag{4.19}$$

In this equation $U(f)$ and $V(f)$ are the amplitude spectral densities of the signals $u(t)$ and $v(t)$ respectively, and $H(f)$ the frequency response of the filter.

The Fourier transform does not exist for all signals. There are two main groups of signals that have a Fourier transform. The first group are the transient functions or finite energy signals for which

$$\int\limits_{-\infty}^{\infty} |u(t)|^2 dt < \infty. \tag{4.20}$$

The second group are the periodical signals. A signal is periodical if there exist a T so that $u(t+T) = u(t)$ for all t. These signals have a Fourier transform that consist of delta functions. Also the summation of a finite number of periodical signals has a Fourier transform consisting of delta functions, even if the sum of these signals itself is not periodical.

4.4.3 Random processes

According to the principles of information theory, signals that carry information must have an unpredictable and random character, otherwise the information content would be zero. It is not necessary that the signal is completely random, although this will give the maximum information rate. In an informal way, a random process is defined by an infinite dimensional probability density function that specifies the joint probability of all possible combinations of amplitudes on all possible moments in time. Note that this model contains deterministic signals as a special case in which the probability density is a multidimensional delta function.

Instead of the infinite dimensional probability density function itself, usually only its first and second order moment are of interest. The first order moment, called the expectation, is defined as

$$\mathrm{E}\{u(t)\} = \int_{-\infty}^{\infty} \nu f_u(\nu; t)\, d\nu \tag{4.21}$$

in which $f_u(\nu; t)$ is the probability density function of the amplitude of u on time t. The expectation can also be defined as an ensemble average

$$\mathrm{E}\{u(t)\} = \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N} u_n(t) \tag{4.22}$$

where $u_n(t)$ are independent realizations of the process $u(t)$. In practice the exact distribution is often unknown and one has only one single realization of the process. Therefore (4.21) and (4.22) are only of theoretical value. If two conditions are met, the ensemble average can be replaced by a time average over one realization

$$\langle u(t) \rangle \ = \ \lim_{T\to\infty} \frac{1}{T} \int_{T} u(t)\, dt. \tag{4.23}$$

The first condition is that the moment of interest is stationary over time. The second condition is ergodicity. Informally, a process is ergodic if the variations that occur between various realizations, also occur during one single realization. This is of great practical importance, because one often receives only one realization. If a process has a stationary and ergodic first order moment we have

$$\mathrm{E}\{u(t)\} \ = \ \langle u(t) \rangle\,. \tag{4.24}$$

4.4.4 The power spectral density

Radio signals are often persistent and non periodical, so that they have infinite energy and (4.20) does not hold. Therefore such signals are not Fourier transformable, and thus do not have an amplitude spectral density. However, if a process is ergodic in its first and second order moments, it can be characterized in the frequency domain by its power spectral density. The power spectral density is defined by the Wiener-Khintchine theorem

$$R_{uu}(\tau) \quad \overset{\mathscr{F}}{\rightleftharpoons} \quad S_{uu}(f) \tag{4.25}$$

in which $R_{uu}(\tau)$ is the stationary and ergodic autocorrelation function

$$R_{uu}(\tau) \ = \ \mathrm{E}\{u(t)u^*(t-\tau)\} \ = \ \langle u(t)u^*(t-\tau) \rangle\,. \tag{4.26}$$

If the signal $u(t)$ is applied to a filter with impulse response $h(\tau)$, the autocorrelation of the output signal $v(t)$ will be

$$R_{vv}(\tau) \ = \ R_{uu}(\tau) * h(\tau) * h^*(-\tau). \tag{4.27}$$

If we take the Fourier transform of this double convolution, we get

$$S_{vv}(f) \ = \ S_{uu}(f) \cdot |H(f)|^2 \tag{4.28}$$

in which $S_{uu}(f)$ and $S_{vv}(f)$ are the power spectral densities of $u(t)$ and $v(t)$ respectively. The average power of a signal can be found by

$$P_{\text{avg}} = \left\langle |u(t)|^2 \right\rangle = R_{uu}(0) = \int_{-\infty}^{\infty} S_{uu}(f)\, df. \tag{4.29}$$

A more intuitive but equivalent way to define the power spectral density is by the schematic in Figure 4.11.

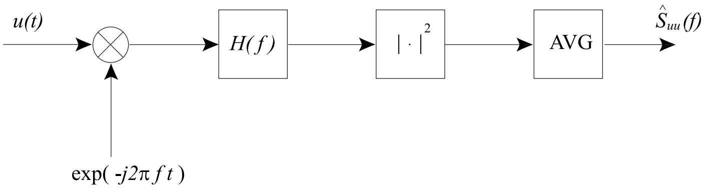


Figure 4.11: Block scheme of a spectrum analyzer.

In this schematic $H(f)$ is the resolution bandwidth (RBW) filter, which is a narrow band low-pass filter with impulse response $h(t)$, and AVG an averaging operation. The output of the RBW filter is

$$[u(t) \cdot \exp(-j2\pi f t)] * h(t). \tag{4.30}$$

The output of the power detector is given by

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} u(t-\tau_1)u^*(t-\tau_1-\tau_2)h(\tau_1)\, h^*(\tau_1+\tau_2)\, \exp(-j2\pi f\tau_2)\, d\tau_1 d\tau_2. \tag{4.31}$$

The output of the averager is

$$\hat{S}(f) = \int_{-\infty}^{\infty} R_{uu}(\tau) \cdot [h(\tau) * h^*(-\tau)] \; \exp(-j2\pi f\tau)\, d\tau \tag{4.32}$$

which is just a Fourier transform, so we have

$$\hat{S}(f) = S(f) * |H(f)|^2 \tag{4.33}$$

which is the true power spectral density convolved with the resolution bandwidth filter.

4.4.5 The periodogram

Because the problems with the convergence of the Fourier transform of persistent signals obviously comes from the infinite integration, a first step to solve it is to limit the signal to a finite interval of length T

$$U_T(f) \; = \; \int\limits_T u(t)\exp(-j2\pi ft)\,dt. \tag{4.34}$$

If one tries to let T go to infinity, the amplitudes become infinitely high and the phase infinitely erratic. The erratic behavior of the phase can be fixed by just removing it by taking the squared magnitude. The amplitude values can be kept finite by dividing by T. These two modification lead to the well known periodogram

$$S_T(f) = \frac{1}{T}|U_T(f)|^2. \tag{4.35}$$

The expectation of the periodogram is given by

$$\mathrm{E}\{S_T(f)\} \; = \; T\,\mathrm{sinc}(Tf)^2 \; * \; S(f) \tag{4.36}$$

which is the convolution between the square of a $\mathrm{sinc}()$ kernel and the actual power spectral density. Because the squared kernel, the estimate is non negative for each frequency, which is desirable since power can not be negative. Furthermore, the square of a $\mathrm{sinc}()$ will approach a delta function if T is made large, so the expectation of the periodogram converges to the true spectrum, even though this convergence is very slow.

The variance of the periodogram [75] is given by

$$\mathrm{VAR}\{S_T(f)\} = \begin{cases} 2S(0)^2 & f = 0 \\ S(f)^2 & |f| \gg 1/T. \end{cases} \tag{4.37}$$

So, although the expectation of the periodogram converges to the true spectrum, the periodogram itself does not, because its variance does not decrease for large T.

4.4.6 Sampling in time and frequency domain

In a digital implementation continuous time and frequency parameters are sampled. If t_s is the sampling time, we form the sampled signal

$$u[n] = u(n\;t_s) \tag{4.38}$$

in which the square brackets denote that n is an integer and $u[n]$ a sampled function. The autocorrelation function can be sampled in the same way

$$R_{uu}[n] = R_{uu}(n\;t_s). \tag{4.39}$$

The amplitude spectrum of a sampled signal is defined by the Discrete Time Fourier Transform (DTFT)

$$U(f') = \sum_{n=-\infty}^{\infty} u[n] \exp(-j2\pi n f') \tag{4.40}$$

in which $f' = f\, t_s$ denotes the relative frequency that lies between -0.5 and 0.5. For notational convenience we drop the accent if it is clear that relative frequencies are used.

For signals with finite duration L the frequency parameter can also be sampled. This leads to the Discrete Fourier Transform (DFT)

$$U[k] = \sum_{l=0}^{L-1} u[l] \exp(-j2\pi lk/L) \tag{4.41}$$

in which k is the discrete frequency parameter, such that

$$U[k] = U(k\,\Delta f) = U(\frac{k}{L\, t_s}) \tag{4.42}$$

in which Δf is the frequency bin size. The periodogram is calculated as

$$S_L[k] = \frac{1}{L} \left| \sum_{l=0}^{L-1} u[l] \exp(-j2\pi lk/L) \right|^2 . \tag{4.43}$$

Figure 4.12 show the periodogram of different lengths. The shown spectrum band contains two strong and three weaker TETRA signal from the C2000 downlink as described in section 4.2.5.

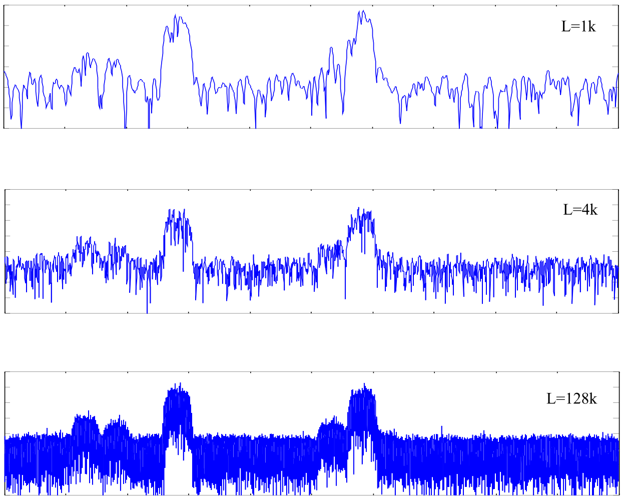


Figure 4.12: Periodogram of 1024 (1k), 4k and 128k points

4.4.7 Welch method

The main shortcoming of the periodogram is its noisiness, which does not improve if L is made larger, as can be seen from Figure 4.12. An first improvement to this situation was made by Barlett, who suggested to split up a signal of length NL in N blocks of length L and average the periodograms of those N blocks. Later this method was improved by Welch [76], who applied a data window to the samples, and allows the blocks of samples to overlap.

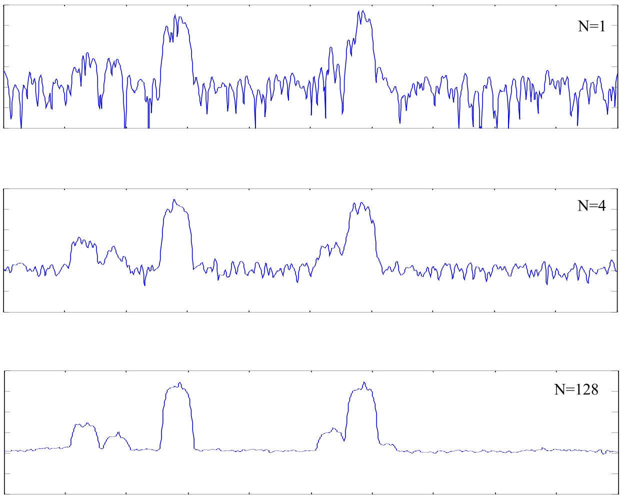


Figure 4.13: Averaged periodograms

Figure 4.13 shows the effect of periodogram averaging. It shows the same spectrum as Figure4.12 using the same amount of samples for each plot, but now L is kept constant on 1024 points, and periodogram averaging is applied. In the first plot no averaging is applied. In the second plot four periodograms are averaged, which reduces the noise considerably. The third plot show the result for $N = 128$.

4.4.8 Threshold detection

After the power spectral density is estimated, a discission has to be made about whether a channel is occupied or empty. If the channel bandwidth and center frequency of a primary user signal is known, one can integrate the power spectral density over this bandwidth to obtain the signal power. If such channel information is not available, the power spectral density itself is the decision metric, which is sub-optimal with respect to channelized detection. Detection methods based on the power estimations are known as radiometric or energy detection. Their advantage is that they do not require detailed knowledge of the signals internal structure.

The amplitude of a DFT bin can be considered complex Gaussian distributed, if the bin size is much smaller that the bandwidth of the narrowest signal, or equivalently, if the DFT size L is much longer than the extend of the autocorrelation function. This can be explained by the fact that a single DFT bin is formed by the (weighted) summation of many independent signal components.

Let $\hat{p}$ be the estimation of the true power p in a frequency bin, formed by the averaging of N squared magnitudes. The probability density function (pdf) of $\hat{p}$ will be a Chi-squared distribution with $2N$ degrees of freedom

$$\mathrm{pdf}(\hat{p}) = \frac{2N}{p}\chi^2_{2N}(\frac{2N\hat{p}}{p}) \tag{4.44}$$

with mean and variance given by

$$\mathrm{E}\{\hat{p}\} = p, \quad \mathrm{VAR}\{\hat{p}\} = \frac{p^2}{N}. \tag{4.45}$$

The received power is the sum of noise and signal power

$$p = p_{noise} + \varepsilon\, p_{signal} \tag{4.46}$$

in which $\varepsilon \in \{0, 1\}$ is a binary parameter that represents the presence of the target signal. Figure 4.14 shows the distributions of $\hat{p}$ for $N = 10$ and $N = 100$ and a signal to noise ratio $\gamma = 1$. The blue curves are for $\varepsilon = 0$ and the red curves for $\varepsilon = 1$.

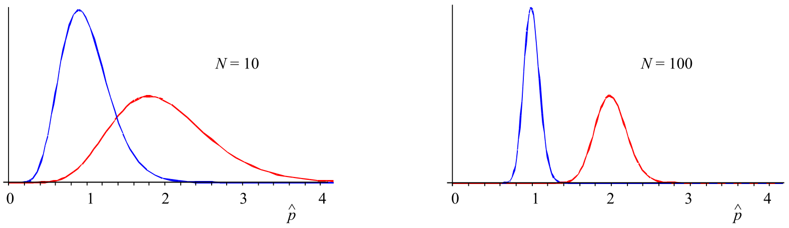


Figure 4.14: Chi Squared distributions

In order to find the actual detection probabilities it is easier to work with the cumulative Chi-squared distribution

$$\Phi_{2N}(\lambda) = \int_{-\infty}^{\lambda} \chi^2_{2N}(x)\, dx. \tag{4.47}$$

For the probabiliy that $\hat{p}$ is smaller than a threshold λ we then have

$$\Pr\{\hat{p} < \lambda\} = \Phi_{2N}(\frac{2N}{p}\lambda). \tag{4.48}$$

To evaluate the performance of threshold detection, two error probabilities are of interest: the False Alarm Rate (FAR) and the False Rejection Rate (FRR). The FAR is the probability that a channel is considered occupied, while in fact it is free. The FRR is the probability that a channel is considered free, while in fact a signal is present. The FAR and FRR are given by the following equations:

$$\begin{aligned} \text{FAR} = \Pr\{\hat{p} > \lambda \mid \varepsilon = 0\} &= 1 - \Phi_{2N}(2N\lambda) \\ \text{FRR} = \Pr\{\hat{p} < \lambda \mid \varepsilon = 1\} &= \Phi_{2N}(\frac{2N\lambda}{1+\gamma}). \end{aligned} \tag{4.49}$$

By adjusting the threshold λ a trade-off between both errors can be made. If the cost of both errors is equal, the threshold can be placed so that both errors are equal, which gives the Equal Error Rate (EER).

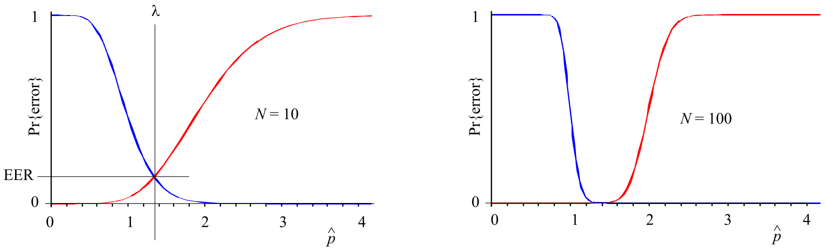


Figure 4.15: FAR (blue line) and FRR (red line) curves for a SNR of 0 dB

Figure 4.15 shows the FAR and FRR curves for 10 and 100 times averaging and a signal to noise ratio of one. For $N = 10$ the EER threshold lies on $\lambda \approx 1.34$ and the EER is about 14%. For $N = 100$ the threshold lies on $\lambda \approx 1.38$ and the EER is about 0.028%.

For a given number of signal samples, the DFT size must be made smaller if more averaging has to be done, which reduces the spectral resolution. So a trade-off must be made between accuracy and resolution. Figure 4.16 shows the relation between the SNR and the required averaging for various equal error rates. The plots are generated by evaluating (4.49) numerically. The left plot shows N on a logarithmic scale since it becomes very large for negative SNR. The right plot shows N only for positive SNR on a linear scale.

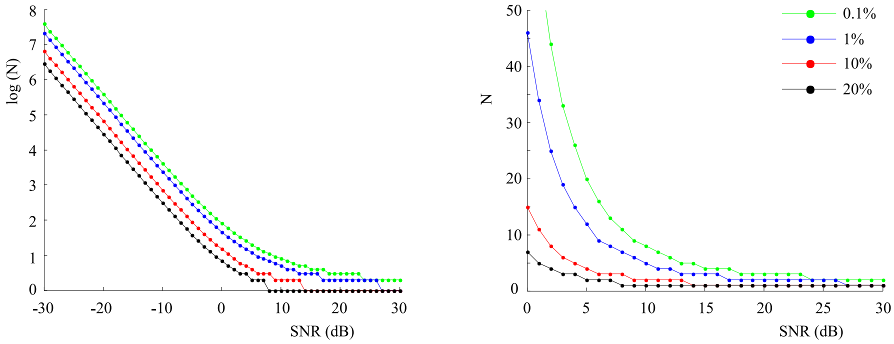


Figure 4.16: Required averaging as function of SNR for various EER

For a negative SNR and a high number of averages the relation between $log(N)$ and SNR(dB) becomes a straight line with a slope of $10^{-\mathrm{SNR(dB)}/5}$.

At first glace Figure 4.16 may seem to suggest that sub noise detection is possible with threshold detection of power measurements. However, in practice this will not work. First we see that the required number of averages becomes very high for low SNR values. For example, a SNR of -20 dB requires about 256 k averages to obtain a error of 1%. If we have a bandwidth of, for example, 2.56 MHz and a bin size of 2.5 kHz, a DFT size of 1 k points is required. Then the number of signal samples for the estimation will be 256 M which at the given bandwidth will take 100 s of acquisition time. In practice power levels are often not constant over such a long time interval. A second problem is that the sub-noise performance predicted by Figure 4.16 requires that the threshold is placed exactly on the right value, which requires exact knowledge of the noise floor and the received primary user signal power. Even if the transmit power and location of a primary user is exactly known, channel fading will give a signal level uncertainty of some dB's. Also the noise level is difficult to estimate with hight accuracy. For these reasons radiometric detection only works for positive SNR's.

4.4.9 Feature detection

In the last section we assumed a DFT bin size much smaller than the smallest signal bandwidth, so that the amplitude in each bin becomes Gaussian distributed. This is of coarse not possible if there are delta function present in the power spectral density, or equivalently, pure sinusoids in the time domain. With a few exceptions, like in amplitude modulation, pure sinusoids are not transmitted, because they cost power without conveying information. However, if they are present in a signal, this is favorable for detection, because delta functions have an infinite amplitude in the power spectral density, so they stick out above the noise for every SNR.

In [77] we describe a method to retrieve a sinusoidal component from PSK and low order QAM signals by applying a non linear operation to the signal. If we raise an M-ary PSK signal to the M-th power we get on the sampling moments

$$\begin{aligned} c[n] &= \exp(j2\pi\varphi[n]/M) \ \text{ with } \ \varphi[n] \in \{1 \cdots M\} \\ |c[n]|^2 &= 1, \quad c[n]^M = \exp(j2\pi\varphi[n]) = 1 \end{aligned} \tag{4.50}$$

which is a constant value. In between the sampling moments the amplitude is not confined to the constellation, so there the signal remains random after the power law operation. Figure 4.17 illustrates the principle for a QPSK signal.

Figure 4.18 shows the resulting spectrum if the method is applied to a TETRA signal. TETRA uses $\pi/4$-QPSK modulation, which means that the symbols are drawn from a QPSK constellation, and then the odd numbered symbols are given an extra $\pi/4$ phase shift, so

$$c[n]^4 = (-1)^n. \tag{4.51}$$

The baseband representation of a TETRA signal is given by

$$u(t) = \sum_{n=-\infty}^{\infty} c[n] \cdot p(t - nT). \tag{4.52}$$

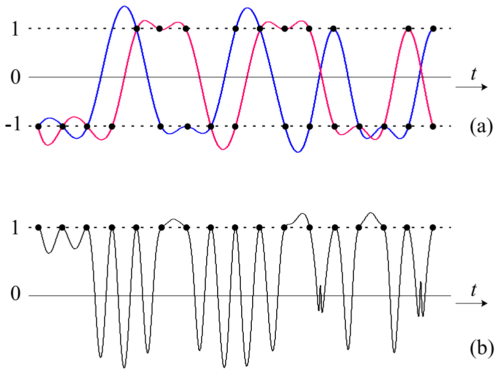


Figure 4.17: Periodical level crossings obtained by squaring a QPSK signal

If we raise $u(t)$ to the fourth power and take the expectation we get

$$\mathrm{E}\{u[n]^4\} \;=\; \mathrm{E}\{\left(\sum_{n=-\infty}^{\infty} c[n]\cdot p(t-nT)\right)^4\} \;=$$

$$\sum_{n=-\infty}^{\infty} (-1)^n\, p(t-nT)^4 = \tag{4.53}$$

$$p(t)^4 \;*\; [\frac{1}{T}\,\mathrm{III}(t/T)\cos(\pi t/T)]$$

which Fourier transforms into

$$P(f) * P(f) * P(f) * P(f) \cdot \mathrm{III}\left(\frac{f}{T} - \frac{1}{2T}\right). \tag{4.54}$$

We see that the channel filter $P(f)$ is convolved four times with itself, which gives a function that is about four times wider. This function is multiplied with the pulse train function which is shifted over halve the pulse bandwidth. From (4.54) we see that we can expect delta functions in the spectrum, which is very useful because delta functions in theory have an infinite height in the power spectral density. In practice the peak height is limited by the finite observation time, but still the peaks will be much narrower than normal signals.

The spectral peaks that are obtained by this method may be used in several ways. First they can be used to recognize that PSK of a certain order is used. Secondly, they can be used to estimate the exact center frequency, bandwidth and phase of a signal. If they can be used to enhance detection for low SNR is a subject for further research.

4.4.10 Cyclostationarity

It is well recognized that communication signals are often cyclostationary. The cyclostationarity of signals may provide a signature from which the modulation type and

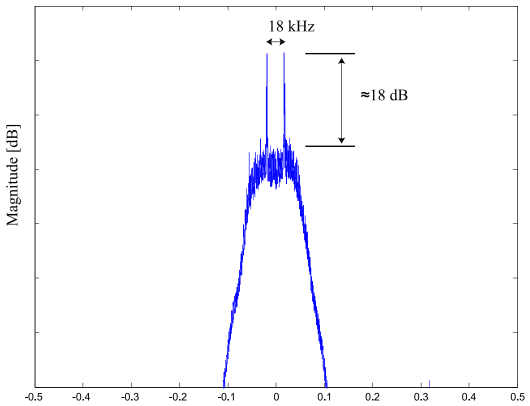


Figure 4.18: Spectral peaks in the TETRA signal from Figure 4.7 on 392.24 MHz after selecting it with a bandpass filter and raising it to the 4-th power.

other parameters can be derived. Furthermore, it is often suggested that cyclostationarity of a signal can be used to detect it far under the noise floor of the receiver [78]. The concept of cyclostationarity was introduced into the field of telecommunications mainly by Gardner. To keep this document self contained, we repeat here some definition that are mainly adapted from [79].

The instantaneous autocorrelation function (IAF) is defined as

$$R_{uu}(t,\tau) \;=\; \mathrm{E}\{u(t+\tfrac{1}{2}\tau)u^*(t-\tfrac{1}{2}\tau)\} \tag{4.55}$$

in which $\mathrm{E}\{\}$ is the ensemble average or Expectation operator, $u(t)$ a random process, t the absolute time, and τ the time-lag parameter. It is a common property of a random process that its autocorrelation goes to zero or becomes periodical for large τ, so it is Fourier transformable over τ. The Fourier transform over τ is the Wigner-Ville distribution

$$R_{uu}(t,\tau) \quad \overset{\mathscr{F}}{\rightleftharpoons} \quad W_u(t,f). \tag{4.56}$$

The fourier transform over t gives the radar ambiguity function

$$R_{uu}(t,\tau) \quad \overset{\mathscr{F}}{\rightleftharpoons} \quad A_u(\alpha,\tau). \tag{4.57}$$

The two dimensional Fourier transform over both parameters gives the spectral correlation function

$$R(t,\tau) \quad \overset{\mathscr{F}^2}{\rightleftharpoons} \quad S(\alpha,f). \tag{4.58}$$

If the Fourier transform $U(f)$ of $u(t)$ exist, the name spectral correlation function becomes immediately clear, as we then have

$$S(\alpha,f) = \mathrm{E}\{U(f+\tfrac{1}{2}\alpha)U^*(f-\tfrac{1}{2}\alpha)\}. \tag{4.59}$$

Telecommunication signals are usually persistent and non periodical, so that their Fourier transform does not exist and (4.59) becomes meaningless. Also the autocorrelation of such signals is persistent over time, so that (4.57) and (4.58) do not exist as ordinary functions. If the autocorrelation is constant over time, the Fourier transform over t will be a Dirac delta function, which appears in $S_{uu}(\alpha, f)$ as a line mass on $\alpha = 0$. If the autocorrelation is periodical over t, as sketched in Figure 4.19, $S_{uu}(\alpha, f)$ will have multiple line masses parallel to the f axis, as is sketched in Figure 4.20. Such a process is called cyclostationary if a single period time exist, or polycyclostationary or almost cyclostationary if it contains multiple periodicities that are not harmonically related [79].

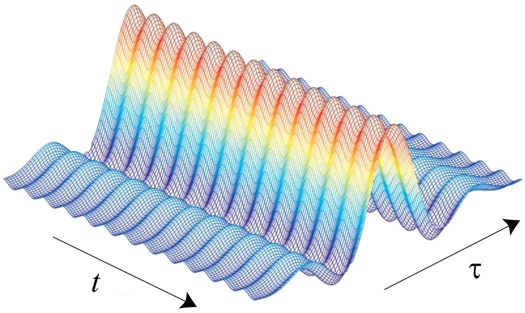


Figure 4.19: Example of a time dependent autocorrelation function $R_{uu}(t, \tau)$ of a cyclostationary process $u(t)$.

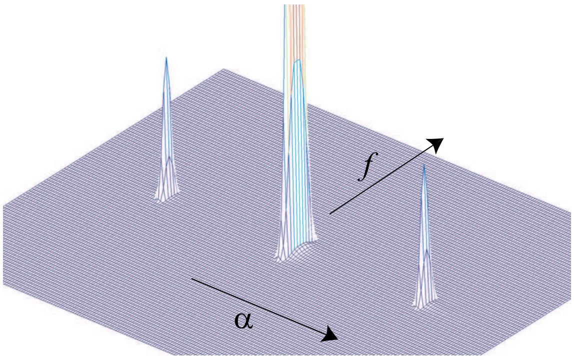


Figure 4.20: Two dimensional Fourier transform $S_{uu}(\alpha, f)$ of the autocorrelation of Figure 4.19. The large peak in the center is the power spectral density, and the two smaller peaks are cyclostationary features.

From these definitions one can readily derive some useful properties. If a random process $u(t)$ is multiplied by a function $a(t)$ we see that we have

$$\begin{aligned} v(t) &= a(t) \cdot u(t) \\ R_{vv}(t,\tau) &= a(t+\tfrac{1}{2}\tau)a^*(t-\tfrac{1}{2}\tau) \cdot R_{uu}(t,\tau) \\ S_{vv}(\alpha,f) &= A(f+\tfrac{1}{2}\alpha)A^*(f-\tfrac{1}{2}\alpha) ** S_{uu}(\alpha,f) \end{aligned} \tag{4.60}$$

in which the operator $*$ is a one dimensional convolution, $**$ a two dimensional convolution and $A(f)$ the Fourier transform of $a(t)$.

If the process $u(t)$ is filtered with a filter with impulse response $h(t)$ and frequency response $H(f)$ we get

$$\begin{aligned} v(t) &= h(t) * u(t) \\ R_{vv}(t,\tau) &= h(t+\tfrac{1}{2}\tau)h^*(t-\tfrac{1}{2}\tau) ** R_{uu}(t,\tau) \\ S_{vv}(\alpha,f) &= H(f+\tfrac{1}{2}\alpha)H^*(f-\tfrac{1}{2}\alpha)\, S_{uu}(\alpha,f). \end{aligned} \tag{4.61}$$

If the process $u(t)$ is time shifted over a distance σ the SCF is multiplied with a complex exponential

$$\begin{aligned} v(t) &= u(t+\sigma) \\ R_{vv}(t,\tau) &= R_{uu}(t+\sigma,\tau) \\ S_{vv}(\alpha,f) &= \exp(j2\pi\sigma\alpha)\, S_{uu}(\alpha,f). \end{aligned} \tag{4.62}$$

Note that for the power spectral density $S_{vv}(0,f)$ a time shift has no consequence.

Many digital modulation schemes are a form a pulse amplitude modulation (PAM). A PAM signal has distinct cyclostationarity features which are derived in [80]. At baseband a PAM signal can be written as the convolution

$$u(t) = \sum_{n=-\infty}^{\infty} c[n] \cdot p(t-nT) \tag{4.63}$$

in which $c[n]$ are the complex data symbols, $p(t)$ is the pulse shape and T the symbol duration time. The instantaneous autocorrelation function of this signal is

$$R_{uu}(t,\tau) = p(t+\tfrac{1}{2}\tau)p^*(t-\tfrac{1}{2}\tau) * \frac{1}{T}\mathrm{III}(t/T) \tag{4.64}$$

where $\mathrm{III}(t)$ denotes an infinite train of Dirac delta functions with unit mass and spacing. The two dimensional Fourier transform gives the spectral correlation density

$$S_{uu}(\alpha,f) = P(f+\tfrac{1}{2}\alpha)P^*(f-\tfrac{1}{2}\alpha) \cdot \mathrm{III}(\alpha T) \tag{4.65}$$

which is graphically depicted in Figure 4.21d.

To test the theory of cyclostationary feature detection, it was tried on the signals described in section 4.2.5. After capturing the signals, we have tried to reproduce the plots similar to Figure 4.20 and Figure 4.21d. There are two possible approaches. First,

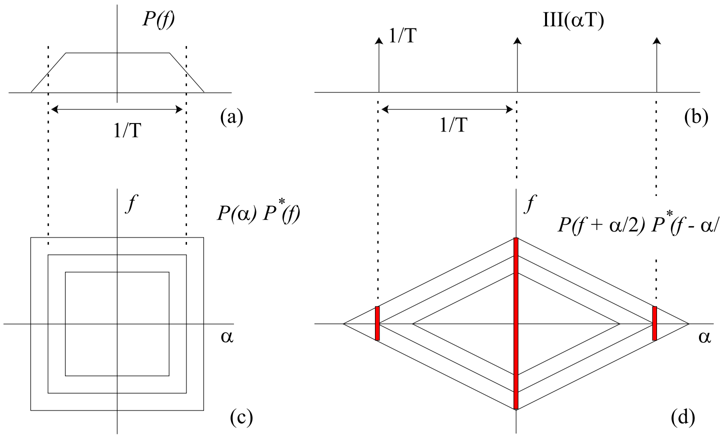


Figure 4.21: Graphical representation of (4.65). (a) Frequency response of a Nyquist Pulse Shape, (b) Pulse train function, (c) Pulse Shape in the (α, f) plane, (d) The SCF of a PAM signal. The red lines are the cyclostationary features.

one can try to calculate (4.59) in which $U(f)$ is approximated by the discrete Fourier transform (DFT) from a windowed section of the signal. Because the cyclostationary features are delta functions, in practice the values of α on which these features appear are never exactly aligned with the DFT bins. So, in order to get a good resolution over the α axis a very large DFT length N must be used. The consequence of this is that the $N \times N$ matrix that results from correlating each bin with every other bin becomes soon prohibitively large. A second problem is that there is no way to implement the expectation operator that occurs in (4.59). As is explained in [81] averaging over multiple estimates of $U(f)$ will lead to non-coherent addition of cyclic features, that can even lead to complete cancellation of features.

A second way to calculate the SCF is by implementing the two dimensional Fourier transform in (4.58) with a two dimensional DFT. First, we down sample the 8 MHz wide signal with a factor 16 to get the 500 kHz wide signal shown in Figure 4.7b. From this down sampled signal we selected a block of 128k samples and calculated the autocorrelation over 512 time lags. Because with time discrete signals it is not possible to use the symmetrical shift used in (4.55), we implemented it as

$$\hat{R}[n,m] \;=\; u[n]u^*[n-m] \tag{4.66}$$

in which we let m run from -256 to +255. The correlation is implemented cyclically, so that samples that are shifted out at the right side, are reinserted at the left side. This results in a $M \times N$ matrix with $M = 512$ rows and $N = 131072$ columns of double precision complex floating values. Again, there is no good way to implement the expectation operator. An ensemble average is impossible because we only have one signal realization. Time averaging will destroy the cyclostationarity, because the block size is in practice never an exact multiple of the cyclic period time. Aligning the DFT blocks with the cyclic period is in practice difficult because for an unknown signal yet to be detected, there is inherently no synchronization between transmitter and receiver. Moreover, if more than one signal is present in the captured band, the total signal is polycyclostationary, so there is no single period time over which one can average.

Next, we compute the two dimensional DFT of the correlation matrix. First, we apply an FFT over the rows, yielding the cyclic autocorrelation function. To correct for the asymmetrical time lag we multiply the transformed row with $\exp(j\pi\alpha m)$, which corresponds with a halve sample shift in the time domain. The resulting matrix is an approximation of the cyclic autocorrelation function which is defined in [79] as

$$\begin{aligned} R^{\alpha}[m] = \\ \exp(j\pi\alpha m) \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N} u[n]u^*[n-m] \cdot \\ \exp(-j2\pi\alpha n) \end{aligned} \tag{4.67}$$

Finally, we apply an FFT over each column to get the SCF estimation

$$\hat{S}[\alpha, f] = \underset{m\to f}{\mathrm{DFT}} \left[\exp(j\pi\alpha m) \underset{n\to\alpha}{\mathrm{DFT}} \left[u[n]u^*[n-m]\right]\right]. \tag{4.68}$$

Figure 4.22 shows a bitmap image of this estimation. To make the very narrow features visible in the gray-scale bitmap image, the α axis was decimated by a factor 256. This decimation consists of dividing the row vectors in blocks of 256 samples, of which only the highest value was kept. Also the logarithm of the absolute value of the samples was taken because of the large dynamic range of the values.

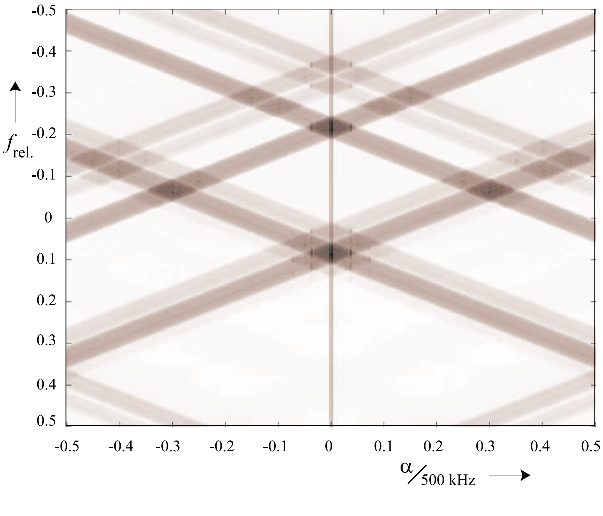


Figure 4.22: Measured spectral correlation function from signals shown in Figure 4.7b

Most prominently visible in the figures are the diagonal stripes, which unfortunately, are just artifacts. They are caused by the fact that the variance of the estimation is high on regions where the power spectral density is high. The actual cyclostationary features are the tiny vertical lines at $\alpha = 18$ kHz. Figure 4.23 shows a close-up in which the cyclostationary features are better visible. Figure 4.24 shows the same close-up as a three dimensional plot.

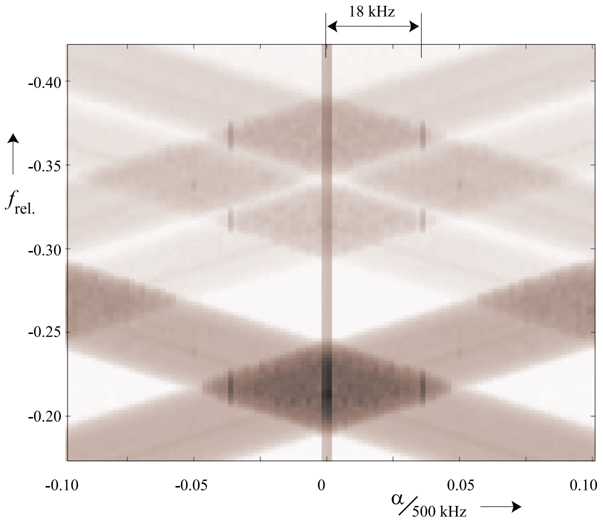


Figure 4.23: Close up from Figure 4.22

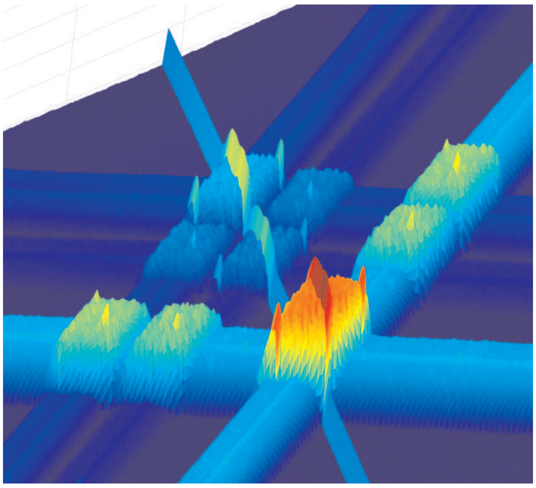


Figure 4.24: 3D view of the SCF from Figure 4.23

Although the number of samples used in the estimation is not too high, the computational load to calculate the SCF is considerable, because a two dimensional $N \times M$ DFT has to be done. However, the size of the DFT only needs to be large in the cyclic frequency dimension N. In this dimension the transform contains delta functions, which have a height in the spectrum proportional to the DFT length. In the M dimension the SCF behaves like an ordinary power spectral density. In this dimension the DFT only needs to be large enough to resolve the signals in the captured bandwidth. At the stage in the receiver where feature detection is performed, the signal is usually down sampled to a bandwidth small enough to only contain a few signals, so M does not need to be very large to resolve them. An other advantage of making M much smaller than N is that it automatically provides smoothing in the frequency domain, as the truncation of τ in the time domain results in a convolution over f in the frequency domain. This convolution provide the averaging that is needed to reduce the noise in the estimation. If M would be made as long as N this form of smoothing would not occur, which gives the SCF estimation the same erratic noise behavior as the periodogram.

The features obtained by cyclostationary feature detection provide accurate information about the center frequency and symbol duration. This information can possibly be used to synchronize a matched filter on a signal, which would then be able to perform optimal detection on preambles and other known signal parts. Further research is needed to see how well the used methods work for low or even negative SNR levels. Based on the current results we expect that an FFT size much larger than 128k is needed for sub-noise detection.

Chapter 5

Mapping Cognitive Radio onto a Reconfigurable Platform

by Qiwei Zhang

Abstract

Cognitive Radio is considered as a promising technology to address the paradox of spectrum scarcity and spectrum under-utilization. It has to operate in different bands under various data rates and combat adversary channel conditions. Therefore, Cognitive Radio needs an adaptive physical layer which has to be supported by a reconfigurable baseband processing platform. We proposed an MPSoC platform to fulfill the requirements of reconfigurability, speed and energy efficiency. The key element on this platform is the Montium processor. Supporting adaptive DSP algorithms for Cognitive Radio on our platform are our main interest. The mapping of the algorithms has been considered at system level. By using a task transaction level interface, we build system level model for MPSoC implementation and extract some profile information. The mapping of algorithms onto the Montium has also been considered.

5.1 Introduction

The research within the AAF project concerning Cognitive Radion using Reconfigurable platforms focuses on the study of Cognitive Radio physical layer algorithms and how to efficiently map these algorithms onto a reconfigurable platform. This chapter of the white paper presents all our work done so far concerning mapping Cognitive Radio onto a reconfigurable platform. The chapter is organized as follows: In section 5.2, we introduce the targeted reconfigurable platform: a heterogeneous reconfigurable multiprocessor System-on-Chip platform. Section 5.3 introduces the adaptive algorithms we have considered for Cognitive Radio. A system level design methodology for the proposed platform is introduced in section 5.4. Section refsection5 presents our work on mapping algorithms onto the Montium processor, a key element in our proposed platform.

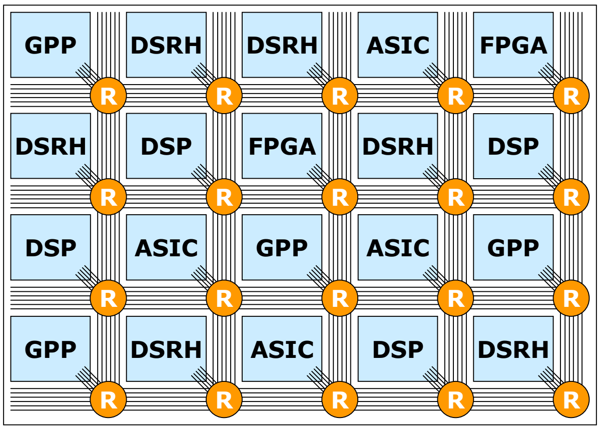


Figure 5.1: An example of a heterogeneous System on Chip (SoC). DSRH = Domain Specific Reconfigurable Hardware

5.2 A heterogeneous reconfigurable System-on-Chip architecture

5.2.1 The Montium based MPSoC platform

With the evolution of semiconductor technology, more and more transistors can be integrated on a single chip which makes it possible to build large systems, on a chip level rather than on a board level. This approach is called System-on-Chip (SoC). The reconfigurable platform we propose for Cognitive Radio is a heterogeneous reconfigurable SoC architecture shown in Figure 5.1.

This SoC is a heterogeneous tiled architecture, where tiles can be various processing elements including General Purpose Processors (GPPs), Field Programmable Gate Arrays (FPGAs), Application Specific Integrated Circuits (ASICs) and Domain Specific Reconfigurable Hardware (DSRH) modules. The tiles in the SoC are interconnected by a Network-on- Chip (NoC). Both the SoC and NoC can be dynamically reconfigurable, which means that the programs (running on the reconfigurable processing elements) as well as the communication links between the processing elements are configured at run-time. Different processing elements are used for different purposes. The general purpose processors are fully programmable to perform different computational tasks, but they are not energy-efficient. The dedicated ASICs are optimized for power and cost. However, they can not be reconfigured to adapt to new applications. FPGAs which are reconfigurable by nature, are good at performing bit-level operations but not that efficient for word level DSP operations. The Domain Specific Reconfigurable Hardware (DSRH) is a relatively new type of processing element, where the configurable hardware is tailored towards a specific application domain. The Montium [82] tile processor (see Figure 5.2) developed at the University of Twente, and recently commercialised by Recore Systems, is an example of DSRH. It targets the digital signal processing (DSP) algorithm domain, which is the heart of the wireless baseband processing. In our previous work [83] [84] [85], several DSP algorithms used in wireless communication have been mapped onto the Montium architecture. The implementation results show that the Montium architecture is flexible enough to

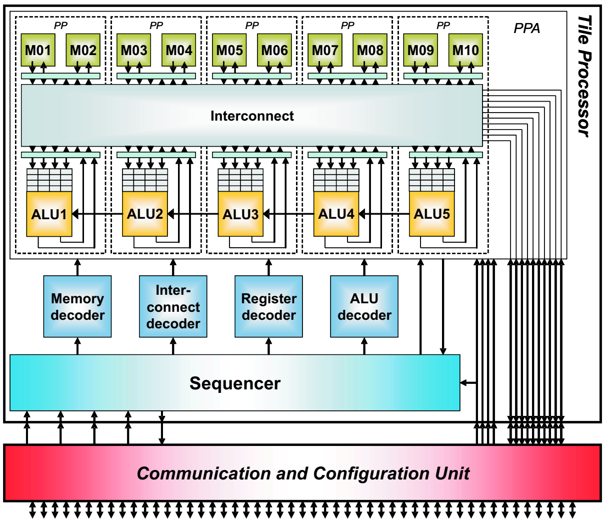


Figure 5.2: An example of DSRH: a Montium processor

adapt to different algorithms with good energy-efficiency. In a broader sense, working intelligently in an energy-efficient way is an important feature of Cognitive Radio. For Cognitive Radio devices working in the emergency network, energy-efficiency is really a crucial issue because the battery life of radio devices can be a limitation for successful operations. Therefore the reconfigurable platform we propose not only targets flexibility but also energy-efficiency.

5.2.2 Publications with respect to the topic

- Zhang, Q. and Hoeksema, F.W. and Kokkeler, A.B.J. and Smit, G.J.M., *Towards Cognitive Radio for emergency networks*, In: Mobile Multimedia: Communication Engineering Perspective. Nova Publishers, U.S.A. ISBN 1-60021-207-7

- Zhang, Q. and Kokkeler, A.B.J. and Smit, G.J.M., "A reconfigurable radio architecture for Cognitive Radio in emergency networks," In: *European Conference on Wireless Technology*, 10-15 September 2006, Manchester, UK. pp. 35-38. IEEE Communication Society. ISBN 2-9600551-5-2

- Zhang, Q. and Smit, G.J.M. and Smit, L.T., "A Reconfigurable Platform For Cognitive Radio," In: *Proceedings of Second International Conference on Mobile Technology, Applications and Systems*, 15-17 Nov 2005, Guangzhou, China. pp. 39-47. IEEE Computer Society. ISBN 981-05-4571-1.

5.3 Adaptive baseband processing for Cognitive Radio

In the AAF project, we mainly focus on mapping the digital baseband processing in Cognitive Radio onto the reconfigurable platform. The baseband processing of Cognitive Radio mainly consists of two parts: baseband transmission and spectrum sensing. These baseband processing algorithms have to be adaptive to enable Cognitive Radio to discover the free spectrum and adapt its transmission accordingly.

5.3.1 Adaptive multicarrier transmission for Cognitive Radio

The basic idea of the multicarrier based Cognitive Radio is to deactivate the subcarriers causing interference to licensed users and optimally use the remaining part of the targeted spectrum. This technique is also known as *spectrum pooling* [38]. To optimally using the remaining part of the spectrum, the adaptive resource allocation on each subcarrier, which includes adaptive bit loading and adaptive power loading, can be applied. In [86], we proposed the adaptive bit loading and power loading rule for OFDM based Cognitive Radio, but can be applied to other multicarrier systems. We could maximize the data rate of the system under a certain power constraint. It is formulated as follows:

$$\begin{aligned}
&\textit{Max}\ \ R = \sum_{k=1}^{K} \frac{F_k}{K} \log_2(1 + \frac{h_k^2 p_k}{N_0 \frac{B}{K}}) \\
&\textit{Subject to:} \sum_{k=1}^{K} p_k \leq P_{total} \\
&F_k \in \{0, 1\} \text{ for all } k \\
&p_k = 0 \text{ for all } k \text{ which satisfies } F_k = 0
\end{aligned} \tag{5.1}$$

where R is the data rate; K is the number of the subcarriers; N_0 is the noise power density, B is the band of interest for Cognitive Radio, h_k is the subcarrier gain and p_k is the power allocated to the corresponding subcarrier. F_k is the factor indicating the availability of subcarrier k to Cognitive Radio, where $F_k = 1$ means the kth carrier can be used by Cognitive Radio. The system power minimization can also be applied under the constraint of a constant data rate. We formulate it as follows:

$$\begin{aligned}
&\textit{Min} \sum_{k=1}^{K} p_k = P_{total} \\
&\textit{Subject to: } R = \sum_{k=1}^{K} \frac{F_k}{K} \log_2(1 + \frac{h_k^2 p_k}{N_0 \frac{B}{K}}) \\
&F_k \in \{0, 1\} \text{ for all } k \\
&p_k = 0 \text{ for all } k \text{ which satisfies } F_k = 0
\end{aligned} \tag{5.2}$$

OFDM based Cognitive Radio

OFDM is one of the candidates for multicarrier based Cognitive Radio. It can be easily implemented by FFT/IFFT and relatively easy to be recovered from channel distortion and delay. A functional diagram of the system is presented in Figure 5.3. A bit allocation vector indicates how many bits are loaded on each subcarrier. The number of bits corresponds to the different modulation types used for each subcarrier. The bit allocation vector is determined by the spectrum occupancy information from spectrum sensing and the SNR of subchannels. The bit allocation vector is disseminated via a signaling channel, so that both transmitter and receiver have the same information. We assume the bit allocation vector does not change frequently for instance during several frames.

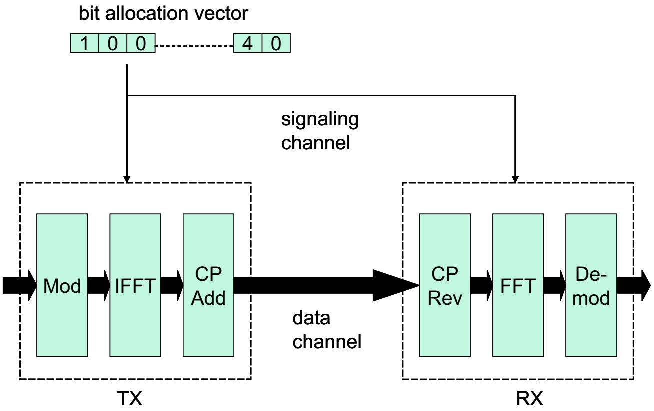


Figure 5.3: OFDM for Cognitive Radio

In the context of the OFDM based Cognitive Radio, we proposed a novel *sparse FFT* [87] for Cognitive Radio based on the observation that there could be a large number of zero inputs/outputs for the IFFT/FFT when a large part of the spectrum is not available to Cognitive Radio or there are many bad channels. In this case, the normal radix-2 IFFT/FFT is inefficient due to the wasted operations on zeros. The sparse FFT takes the sparse structure in the OFDM based Cognitive Radio algorithmically to reduce the operations of the standard algorithms.

The proposed sparse FFT is modified from transform decomposition in [88]. The basic idea is to divide a large FFT into two smaller size FFTs, one out of which results in a reduced computation. We explain the algorithm in details here. The DFT is defined as:

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{nk} \; k = 0, 1, ..., N-1 \tag{5.3}$$

, where $W_N^{nk} = e^{\frac{-j2\pi nk}{N}}$. We consider the case where L outputs are nonzero. Let N be factorized as two integers N_1 and N_2, so $N = N_1 N_2$. To facilitate the implementation, we make the size of FFT N a power-of-two integer. We choose N_1 as the nearest power-of-two integer larger than L and as a factor of N, denoted as $N_1 = \lceil L \rceil_{pow2}$. Therefore, N, N_1 and N_2 are all power-of-two integers. The index n can be written as:

$$\begin{aligned} n &= N_2 n_1 + n_2 \\ n_1 = 0, 1, ..., N_1 - 1 &\quad n_2 = 0, 1, ..., N_2 - 1 \end{aligned} \tag{5.4}$$

Substitute n in (5.3) with (5.4) and then the DFT can be rewritten as:

$$\begin{aligned} X(k) &= \sum_{n_2=0}^{N_2-1} \sum_{n_1=0}^{N_1-1} x(N_2 n_1 + n_2) W_N^{(N_2 n_1 + n_2)k} \\ &= \sum_{n_2=0}^{N_2-1} [\sum_{n_1=0}^{N_1-1} x(N_2 n_1 + n_2) W_N^{N_2 n_1 k}] W_N^{n_2 k} \end{aligned} \tag{5.5}$$

We define:

$$\begin{aligned} X_{n_2}(\langle k\rangle_{N_1}) &= \sum_{n_1=0}^{N_1-1} x(N_2 n_1 + n_2) W_{N_1}^{n_1 k} \\ &= \sum_{n_1=0}^{N_1-1} x_{n_2}(n_1) W_{N_1}^{n_1 k} \qquad (5.6) \end{aligned}$$

where $\langle\rangle_{N_1}$ denotes modulo N_1. So (5.5) can be written as:

$$X(k) = \sum_{n_2=0}^{N_2-1} X_{n_2}(\langle k\rangle_{N_1}) W_N^{n_2 k} \qquad (5.7)$$

The original N-point DFT with L nonzero outputs is decomposed into two major parts: the N_2 N_1-point DFTs in (5.6) which can be implemented as N_2 N_1-point FFTs and the multiplications with twiddle factors and recombinations of the multiplications in (5.7). Because the index k only consists of L nonzero values, only L twiddle factors are multiplied with each $X_{n_2}(\langle k\rangle_{N_1})$ for $n_2 = 1, 2, ..., N_2$. This multiplication part results in a computation reduction.

The number of complex multiplications in a N point sparse FFT with L nonzero outputs equals:

$$Mul_{sparse} = (N_2 - 1) * L + \frac{N}{2} \log_2 N_1 \qquad (5.8)$$

, where $N_1 = \lceil L \rceil_{pow2}$ and $N = N_1 N_2$. From eq. 5.8, we can see the algorithm is more efficient for small L. The complexity increases with the number of non zeros L and reaches the break even point when $L = \frac{N}{2}$. Since we set constraints $N_1 = \lceil L \rceil_{pow2}$ and $N = N_1 N_2$, the sparse FFT will be calculated as normal N point FFT when $L > \frac{N}{2}$. These constraints are important modifications to the algorithm in [88] in the sense that they facilitate hardware implementations by exploiting power-of-two integers. Unlike [88], they restrict their discussions on the case when only the first L consecutive values are non-zeros, our proposed algorithm can be applied to any non-zero distributions. In figure 5.4 we show an example where only 32 out of 1024 subcarriers are available for Cognitive Radio, the sparse FFT offers 30% saving of computations. Because FFT and IFFT are the most computational intensive parts in an OFDM transceiver, the savings can significantly reduce the computational complexity of the overall system. Therefore, transform decomposition can be an efficient option for an OFDM based Cognitive Radio system when only a small number of subcarriers are available for Cognitive Radio.

The computational structure of the proposed sparse FFT is shown in figure 5.5. Basically the computation can be divided into two stages: FFTs and multiplications with recombination. Before the FFT computation, the input samples are mapped to N_2 memory blocks. Then the N_1-point radix-2 FFT is performed on each of the N_2 memory blocks. The results in the $k\ mod\ N_1$ memory positions in each memory block are multiplied by twiddle factors and recombined to produce the output X_k. We can find the regularity in the computational structure: the memory addressing is constantly hopping from the same position in one block to another. Because N, N_1 and N_2 are all power-of-two integers, we can use the most significant bits to indicate the block

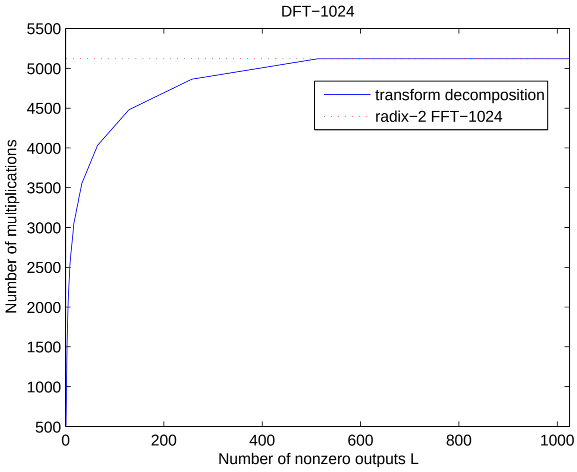


Figure 5.4: Comparison of computation for radix-2 FFT and transform decomposition N=1024

address. The block based addressing is done by changing the most significant bits. This regularity facilitates efficient hardware implementations. Reconfiguration from radix-2 FFT to the sparse FFT is also easy because we can the reuse of radix-2 FFT in the first part.

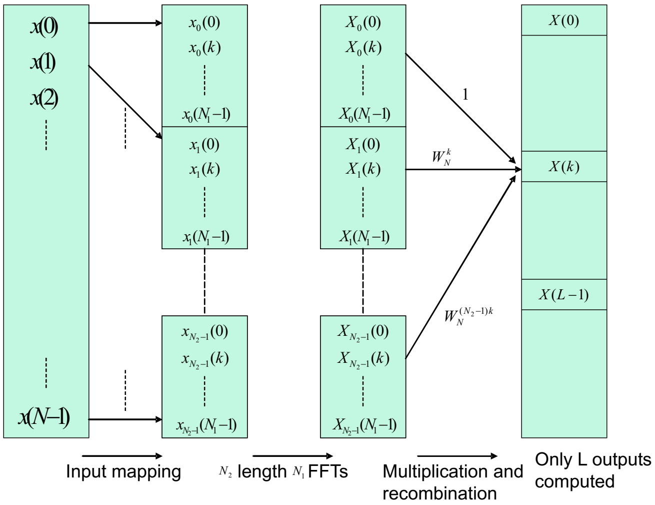


Figure 5.5: Computational structure of the sparse FFT

Filter bank based Cognitive Radio

Due to the rectangular window in the time domain, the OFDM subcarriers have large sidelobes which cause significant power leakage to adjacent subcarriers. Therefore, significant power from adjacent subcarriers will leak to the deactivated subcarriers causing potential interference to licensed users. This fact has also been recognized in [38]. They proposed two methods to mitigate the interference to the licensed user:

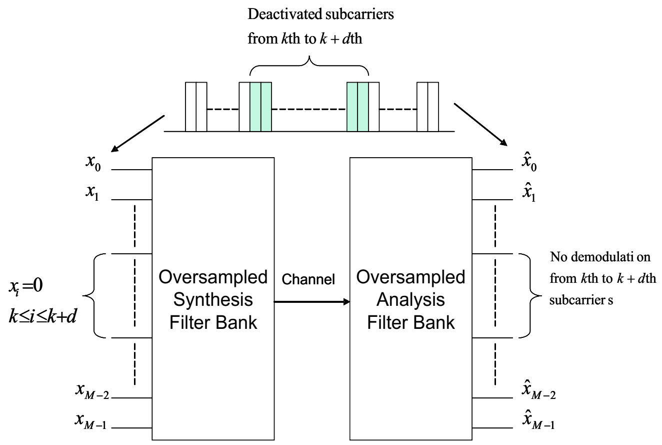


Figure 5.6: An OSFB multicarrier system for Cognitive Radio

deactivating more subcarriers adjacent to the licensed system or applying non-rectangular windows to reduce the spectrum leakage. Both methods mitigate the interference at the cost of bandwidth efficiency. Moreover, two methods didn't consider the system implementation issues. We propose an oversampled filter bank multicarrier system for Cognitive Radio based on the idea of filtered multitone (FMT) in [89]. The oversampling will increase intercarrier spacing thus reduce the overlapping of subcarriers. The scheme is shown in figure 5.6. The deactivation can be realized by loading zeros on the intended subcarriers while others are loaded with modulated complex symbols at the transmitter which is an M band oversampled synthesis filter bank. An M band oversampled analysis filter bank is on the receiver side and send the symbols from those active subcarriers for demodulation.

5.3.2 Adaptive spectrum sensing for Cognitive Radio

Since spectrum sensing plays a vital role in Cognitive Radio to establish communications on unused spectrum, it has been an active research area in Cognitive Radio. On the signal processing level, two methods are often mentioned in literature: energy detection and feature detection [90]. Energy detection measures the signal power at a certain time interval and for a certain frequency band. The detection decision is based on a noise threshold. Energy detection can be implemented by averaging frequency bins of an FFT. Feature detection detects modulated signals by exploiting their hidden periodicity. Feature detection is more robust than energy detection in bad SNR scenarios, but it requires more computations.

A multi-resolution sensing for Cognitive Radio

In the context of energy detection, we propose a novel multi-resolution spectrum sensing based on an efficient dynamically reconfigurable FFT. The basic idea of multi-resolution sensing is that the total bandwidth is first sensed using coarse resolution followed by fine resolution sensing performed on a portion of the interested bands for Cognitive Radio. In such a way, Cognitive Radio avoids sensing the whole band at the maximum frequency resolution. Therefore, the sensing time is reduced and the power

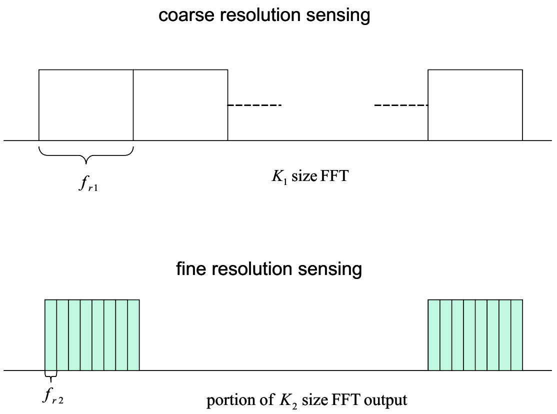


Figure 5.7: An multi-resolution sensing example

has been saved from unnecessary computations. Here, we assume a single antenna Cognitive Radio receiver which can digitize the total targeted bandwidth of B_{tot}. The frequency resolution $f_r = \frac{B_{tot}}{K}$, where K is the size of the FFT which produces the spectrum. Figure 5.7 shows an example of multi-resolution sensing. A coarse resolution sensing is done by using a smaller K_1 size FFT with a resolution $f_{r1} = \frac{B_{tot}}{K_1}$. The energy on each FFT bin E_i for $i = 0, 1, ...K_1$ is compared with a threshold th_1. We define Per as the percentage of the total number of bins where the energy are larger than th_1. If this percentage is larger than a limit p, $Per > p$, we assume the total band is too crowded to accommodate Cognitive Radio. If no bins have been found with significant energy (no i where $E_i > th_1$), namely $Per = 0$, we think the band is empty. In these two conditions, fine resolution sensing is not needed and Cognitive Radio will either start communication or wait for a licensed user to free the spectrum. Otherwise, Cognitive Radio will continue with fine resolution sensing with a resolution of f_{r2} to focus on those high energy bands (like the colored potion in figure 5.7) where licensed users are potentially active. However, the specific method to select the interested bands is not considered in our discussion. The interested portion of the spectrum is actually a part of the output bins of a larger K_2 size FFT, where $K_2 = \frac{B_{tot}}{f_{r2}}$. Based on the result of fine resolution sensing, Cognitive Radio will determine the transmission scheme and wait for the next sensing cycle. A flowchart describing the multi-resolution sensing scheme is shown in figure 5.8. The total cost of multi-resolution sensing can be expressed as:

$$C_{tot} = \{ \begin{array}{c} C_{coarse} \text{ others} \\ C_{coarse} + C_{fine} \text{ if } 0 < Per < p \end{array} \tag{5.9}$$

, where C denotes costs. Here, we make an important observation: only a portion of the larger FFT outputs is needed for fine resolution sensing. In this case, the naive implementation of a larger size FFT is inefficient. Therefore, an efficient algorithm which produces only a part of FFT outputs is desirable. Furthermore, there are several requirements for the efficient FFT implementation: 1) it can produce FFT outputs in any position depending on the need of Cognitive Radio; 2) it has reconfigurability from a normal FFT and vice versa; 3) it is easy to implement on hardware for real time processing.

We think the sparse FFT can be also applied to the multi-resolution sensing system

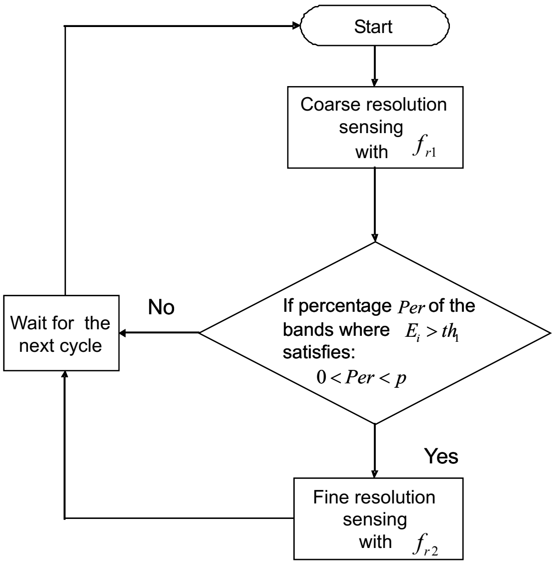

Start
Coarse resolution sensing with f_{r1}
If percentage Per of the bands where $E_i > th_1$ satisfies: $0 < Per < p$
No
Wait for the next cycle
Yes
Fine resolution sensing with f_{r2}

Figure 5.8: Flowchart of multi-resolution sensing

because it fulfills all aforementioned requirements. Let us consider a concrete multi-resolution sensing example. First, a coarse resolution sensing is done by FFT-128 with a resolution of $\frac{B_{tot}}{128}$. If Cognitive Radio finds 5% of the total bandwidth needs fine resolution sensing, the system is reconfigured to the sparse FFT-2048 with the resolution of $\frac{B_{tot}}{2048}$ (16 times finer) to only focus on those interested bands. In this case, the total cost of multi-resolution sensing is the complexity of the sparse FFT-2048 with 0.05 nonzero ratio plus the complexity of the radix-2 FFT-128, totally about 9000 complex multiplications. Comparing this with the fixed resolution sensing by the radix-2 FFT-2048, it gives about 20% saving. However, this saving will diminish with the increasing percentage of the bandwidth which requires fine resolution sensing. The break even point in our example is about 25% of the total bandwidth which requires fine resolution sensing. Beyond this point, the complexity of the radix-2 FFT-128 and sparse FFT-2048 will exceed the complexity of the radix-2 FFT-2048. Clearly, the condition to apply the proposed multi-resolution sensing method is that only a small fraction of total bandwidth requires fine resolution sensing.

Cyclostationary feature detection

Cyclostationary Feature Detection (CFD) consists of a combination of an energy detector and a single correlator block. Because we aim at the implementation of CFD in the digital domain, we will give the time discrete expressions for CFD (DCFD). We first define the sampled signal.

$$x_k = x(k \cdot \frac{1}{f_s}) \tag{5.10}$$

where f_s indicates the sampling frequency. The discrete Fourier Transform is applied to K samples.

$$X_{n,v} = \sum_{k=0}^{K-1} x_{n+k} \cdot e^{j2\pi \frac{n+k}{f_s} v} \tag{5.11}$$

Finally, the Discrete Spectral Correlation Function (DSCF) is determined.

$$S_f^a = \frac{1}{N} \sum_{n=0}^{N-1} X_{n,f+a} \cdot X_{n,f-a}^* \tag{5.12}$$

where * indicates the complex conjugate. In case $N = 2^n$, where $n = 1, 2...$, the Discrete Fourier Transform becomes a Fast Fourier Transform (FFT) and the number of complex multiplications that are involved becomes $\frac{1}{2}N\ ({}^2\log N)$. Determining the DSCF involves $\frac{1}{4}N^2$ complex multiplications. As an example, calculating the DSCF for a 256 point spectrum involves 16 times as many complex multiplications than the determination of the spectrum itself. For the analysis of the platform requirements, we will therefore concentrate on calculating the DSCF.

5.3.3 Publications with respect to the topic

- Zhang, Q. and Kokkeler, A.B.J. and Smit, G.J.M., “An Efficient FFT For OFDM Based Cognitive Radio On A Reconfigurable Architecture,” In: *IEEE International Conference on Communications 2007 Cognet Workshop*, 24-28 Jun 2007, Glasgow, UK. pp. 6522-6526. IEEE Communication Society. ISBN 1-4244-0353-7

- Zhang, Q. and Kokkeler, A.B.J. and Smit, G.J.M., “An Efficient Multi-resolution Spectrum Sensing Method for Cognitive Radio,” In: *IEEE CHINACOM 2008 COGCOM Workshop*, Aug, 2008, China, to appear

- Zhang, Q. and Kokkeler, A.B.J. and Smit, G.J.M., “An oversampled filter bank multicarrier system for Cognitive Radio,” In: *IEEE PRIMRC 2008 CRNET Workshop*, Sep, 2008, France, to appear

- Zhang, Q. and Kokkeler, A.B.J. and Smit, G.J.M., “A reconfigurable radio architecture for Cognitive Radio in emergency networks,” In: *European Conference on Wireless Technology*, 10-15 September 2006, Manchester, UK. pp. 35-38. IEEE Communication Society. ISBN 2-9600551-5-2

- Kokkeler, A.B.J. and Smit, G.J.M. and Krol, T. and Kuper, J., “Cyclostationary Feature Detection on a tiled-SoC,” In: *DATE2007 Proceedings*, 16-20 April 2007, Nice, France. pp. 171-176. European Design and Automation Association. ISBN 978-3-9810801-2-4

5.4 MPSoC design method

To implement Cognitive Radio on a heterogenous reconfigurable platform, we propose a design methodology which has two new features: 1) Transaction level modelling of an application into a parallel task graph, 2) Run-time spatial mapping of tasks onto heterogenous processing tiles.

5.4.1 High level system design using TTL (Task Transaction Level) interface

The platform we proposed for the AAF project is a heterogeneous reconfigurable platform which consists of various hardware and software modules. However, designing such a platform integrating different processing elements is a challenging work. To integrate all these different processing elements, hardware interfaces are used. For example, the usual way of working is to use bus interfaces for integration of hardware devices, with ad-hoc mechanisms based on memory mapped registers and interrupts to synchronize hardware and software modules. Designers have to deal with low-level issues so that the design efforts are increased. Low-level interfaces also hamper the reusability of hardware and software modules.

A task transaction level (TTL) interface based approach [91] is proposed to raise the level of the abstraction. Such an abstract interface should help to close the gap between the application models used for specification and the optimized implementation of the application on a heterogeneous platform. In the AAF project, TTL interfaces will be used to model parallel baseband applications. Assuming the application has been partitioned into a parallel task graph, TTL interface functions are invoked by those tasks to communicate with each other.

Applications are developed under the TTL framework. The system can be simulated and verified at the early stage which facilitates the following steps in our design flow and provides necessary information for the run-time mapping tool. TTL interfaces hide underlying implementation details which make tasks more reusable and easier for application developers. Further, TTL interfaces provide freedom for implementing the platform infrastructure. A broad range of platform implementations are allowed, for example both shared memory and message-passing architectures. Optimizing the implementation parts of the platform is also possible. Platform evolution is supported, for example the smooth transition from bus-based connections to Network-on-Chip technology.

The logic model provides the basis for the definition of the TTL inter-task communication interface and identifies the relevant entities and their relationships. A *task* is an entity that performs computations and that may communicate with other tasks. Multiple tasks can execute concurrently to achieve parallelism. The medium through which the data communication takes place is called a *channel*. The task that writes to the channel is called *producer*; The task that reads from the channel is called *consumer*. A task is connected to a channel via a *port*. A channel is used to transfer values from one task to another. A *variable* is a logical storage location that can hold a value. A *private variable* is a container for a value that is accessible by one task only. A token is a variable that is used to hold a value that is communicated from one task to another. A token can be either *full* or *empty*. Full tokens are tokens containing a value; empty tokens do not contain valid values but merely space for a task to put a value in. We also refer to full and empty tokens as *data* and *room*, respectively. Tasks communicate with other tasks by calling TTL interface functions on their ports. A set of communicating tasks is organized as a task graph.

The TTL interface offers a set of interface types to support different communication styles which satisfy needs for various applications and platform implementations. Each interface type is easy to use and implement. All interface types are based on the same logical model, which enables interoperability across interface types. A task designer has to select the interface type for each port. Different interface types can be used in a single model and even in a single application. Seven interface types are defined. The classification is based on whether the synchronization and the data transfer are combined (C) or separated, whether the task blocks (B) or does not block (N) when the synchronization condition is not met, whether the intra-task synchronization is in-order (I) or out-of-order (O), whether the data transfer is direct or indirect and whether the token reference is absolute or relative.

- **Interface Type CB:** This interface type is named as CB since it combines (C) synchronization and data transfer in a single function with blocking (B) semantics. CB is the most abstract TTL interface type since the programmer does not have to consider how to do synchronization in the program. By calling ttlCbInRead and ttlCbOutWrite functions, the data transfer and the associated communication are performed. This interface type can be implemented efficiently on message-passing architectures or on shared memory architectures where the processors have local buffers that can hold the values that are read or written.

- **Interface Type RB and RN:** The other TTL interface types offer separate functions for synchronization and data transfer. The programmer has to explicitly use functions like acquire and release to check the availability of room and release room when data transfer has completed. RB and RN can selectively load a part of the data from the channel by using a relative offset, thereby allowing cost reduction of the data transfer. However, they are less abstract compared with CB and more difficult to use by the programmer.

- **Interface Type DBI and DNI:** The RB and RN interface types hide the memory addresses of the tokens from the tasks. This may incur inefficiencies upon data transfers. To avoid such inefficiencies, TTL offers interface types that support direct data access like DBI and DNI. In-order means that tokens are released in the same order as they are acquired. These interface types can be implemented efficiently on shared memory architectures and are suited for software tasks that process coarse-grain tokens.

- **Interface Type DBO and DNO:** In some cases tasks do not finish the processing of data in the same order as the data was acquired. For this purpose, TTL interface offer DBO and DNO types. They support efficient use of memory at the cost of a more complex implementation of the channel.

TTL also offers different ways for tasks to interact with the scheduler in order to support different forms of multi-tasking. Three task types are supported by TTL:

- The task type process is for tasks that have their own (virtual) thread of execution and that do not explicitly interact with the scheduler.

- The task type co-routine is for cooperative tasks that interact explicitly with the scheduler at points in their execution where task switching is acceptable.

- The task type actor is for fire-exit tasks that perform a finite amount of computations and then return to the scheduler, similar to a function call.

In [91], only static task graphs were considered. However, embedded systems are becoming more and more complex, supporting multiple use cases. Therefore, TTL has been extended to model dynamic applications [92]. Dynamic applications result in changes in the task graph. Three types of task graph changes are considered: 1) *Topology*: removing or adding tasks and channels, 2) *Binding*: tasks may be processed on different processors, 3) *Parameters*: certain parameters for tasks (e.g. radio transmission modes). The reconfiguration is realized by introducing an extra entity, a *Configuration Manager* (CM), to the TTL logical model. The CM is responsible for initiating the task graph and configure the tasks at run-time. This feature makes TTL a suitable design method to model dynamic applications such as Cognitive Radio and as an interface for the MPSoC platform implementation.

5.4.2 Parameterizable OFDM modelling in TTL

We use the TTL to model an parameterizable OFDM system for Cognitive Radio. The OFDM baseband for Cognitive Radio is a parameterizable OFDM processing chain which is configured by the configuration manager, see figure 5.9. By applying different parameter settings in each task, the OFDM system can be adaptive to various channel conditions and provides various data rates. The parameters considered in our system are shown in Table 5.1, but they are not limited by this table and can be extended to add more flexibility to the system. The number of OFDM symbols per frame is limited by the channel coherence time, during which the channel characteristics are constant. The number of guard samples is chosen to deal with different channel delay spreads. Generally not all data is used to carry useful information. A part of data (e.g. pilots) is used to guarantee reliable transmissions. Different pilots are used for different purposes such as channel estimation or phase offset estimation. They can be placed in the preamble section prior to each frame or embedded in the OFDM symbol. All the information concerning the pilots is contained in a table. Modulation modes indicate the modulation type for the OFDM samples which carry useful information. The modulation mode can be the same in one OFDM symbol but it can also differs on subcarrier basis. In the Cognitive Radio case, the modulation mode can be set to zero to nullify carriers. A format table contains the information on the organization of the data frame, the preamble and the pilots. A generic task graph for the processing of OFDM data symbols on the receiver side is shown in figure 5.10. Based on the OFDM parameters of our Cognitive Radio system in [93], we modelled the OFDM processing chain on the receiver side (see figure 5.10) in the TTL framework. The major system parameters [93] are shown in Table 5.5. The system can operate in two different sampling rates f_s (bandwidth B) and the number of samples N in one

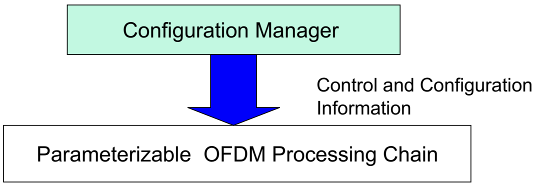

Configuration Manager
Control and Configuration Information
Parameterizable OFDM Processing Chain

Figure 5.9: OFDM for Cognitive Radio

B	bandwidth of OFDM system
N_{sym}	number of OFDM symbols per frame
$N_{preamble}$	preamble length per fame
N	number of OFDM samples per symbol
N_g	number of guard samples per symbol
N_{data}	number of useful data per symbol
mod	modulation modes
tab_{pilot}	table for pilot information
tab_{format}	table for format information

Table 5.1: The parameter set for the parameterizable OFDM

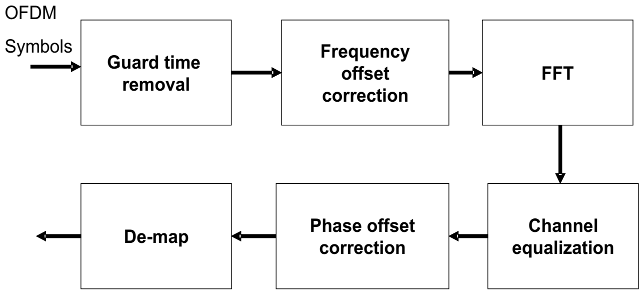


Figure 5.10: The task graph of an OFDM receiver

OFDM symbol is 128 or 512. The subcarrier spacing is 10kHz and the useful symbol

$B = f_s$ $[MHz]$	N	Δf $[kHz]$	T_u $[\mu s]$
5.12	512	10	100
1.28	128	10	100

Table 5.2: OFDM parameters: Sample frequency and symbol duration.

duration T_u is 100μs for both bandwidths. To cope with different delay spreads, the ratio between the number of guard samples N_g and the number of OFDM samples N can be set to 1:4, 1:8, 1:16, 1:32. As a working assumption, one OFDM data frame contains 25 OFDM data symbols thus one frame duration is less than the coherence time (4-5 ms in the considered frequency 400-800MHz) of the channel. The frequency offset and the channel equalization coefficients are determined in the preamble prior to each OFDM frame and updated on a frame basis. The pilots in the OFDM symbols are used for the phase offset correction. We define the ratio of the useful data carriers to the number of total carriers N_{data}:N as 3:4. Modulation modes are chosen from BPSK, QPSK, 16QAM, 64QAM and zero. The zero modes are used to nullify the subcarrier. The tasks are implemented in C/C++ and inter-task communications are function calls from the TTL library. The CM is added as a process on top of the task graph. The parameters for the OFDM tasks are set by the CM and sent via the configuration channel to the tasks. Each task can read the parameters from the configuration port. The TTL implementation in C/C++ can run on a Linux PC and we can verify the functional correctness of parameterizable OFDM at system level.

$N : N_g$	T_g	Freq.	FFT	Channel	Phase
	rem.	cor.		eq.	offset cor.
128 : 32	800	3200	11200	3200	2425
128 : 16	400	3200	11200	3200	2425
128 : 8	200	3200	11200	3200	2425
128 : 4	100	3200	11200	3200	2425
512 : 128	3200	12800	57600	12800	9625
512 : 64	1600	12800	57600	12800	9625
512 : 32	800	12800	57600	12800	9625
512 : 16	400	12800	57600	12800	9625

Table 5.3: Computation workload for each OFDM frame (25 OFDM symbols) in terms of the number of complex multiplications

The TTL run-time environment can generate high level profile information in terms of computation workload and communication workload. The computation workload is measured by counting the number of annotated instructions while the communication workload is measured by counting the number of tokens (data units) that are travelling through the TTL channels. We made a computation workload analysis based on the TTL model. The instructions for the complex multiplication are annotated for analysis because they are the major contributors to the computational complexity. The computation workload for processing one OFDM data frame is shown in Table 5.3 based on the given system parameters. The de-map task is not included in the table because no complex multiplications are involved and only decision and look-up table operations are performed. The computation workload increases significantly if the system parameter N (the number of OFDM samples) changes from 128 to 512. The change of guard samples N_g results in a workload change for the guard time removal task due to the changing length of the correlation window. From the table we can see that the FFT task is the most computationally intensive task. Considering that the worst case execution time (WCET) of the system should be less than the symbol duration 100μs plus the guard time, the system has to be able to compute a 512-point FFT which needs 2304 complex multiplications within 100μs plus the guard time. Therefore the minimum processing capacity required by the parameterizable OFDM system is 23×10^6 complex multiplications per second. The profile information provided by the TTL run-time environment is platform independent. However, it can help to generate the platform dependent profile for specific implementations. By associating execution times with instructions, and by multiplying these execution times with the instruction counts, one can get a rough estimate of the total execution time of a task on a certain processor. Suppose that the OFDM system is mapped onto one or more Montium processors and the CM is running on a GPP. A Montium processor can execute one complex multiplication instruction in one clock cycle. Therefore we can easily derive the profile information for processing one OFDM frame on the Montium. Taken the parameter set $N = 512$ and $N_g = 128$ as an example, the profile information for one OFDM frame on the Montium running at 100MHz is derived from the TTL profile in Table 5.4. The total execution time equals 960 μs. The total available time equals 3.125 ms, so one Montium can be used for OFDM baseband processing for Cognitive

f_{clk} 100MHz	T_g rem.	Freq. cor.	FFT	Channel eq.	Phase offset cor.	Total
Exec. (μs)	32	128	576	128	96	960

Table 5.4: Execution time on the Montium for one OFDM frame ($25 \times 125\mu s = 3.125ms$) where $N = 512$ and $N_g = 128$

Radio. The power consumption of the Montium in 0.13 μm technology is estimated at 0.577 mW/MHz [82]. We can further estimate the energy consumption of each task for processing one OFDM frame, see figure 5.11. The most energy hungry part is the FFT task which costs more than 33μJ for processing one OFDM frame. Therefore to have an efficient FFT implementation is crucial to reduce the energy consumption of the whole OFDM baseband. To reduce the computational complexity of the FFT task for Cognitive Radio, we propose the FFT task can be reconfigured to a sparse FFT in [87].

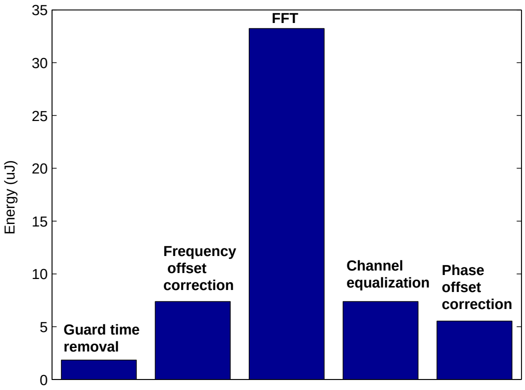


Figure 5.11: The energy consumption on the Montium for one OFDM frame

5.4.3 Reconfigurable sparse FFT in TTL

As the core algorithm of OFDM and multi-resolution spectrum sensing, reconfigurable sparse FFT has been implemented for the following goals: 1) to verify the algorithm at system level; 2)to obtain high level profile information in terms of computation workload and communication workload which help to make the implementation trade-offs on processors.

During the first step, we create the task graph (see figure 5.12) of the reconfigurable sparse FFT. The source task generates the input samples which are sent via the data channel to the FFT task. The destination task consumes the output samples from the FFT task. A configuration manager decides the type of FFT algorithm, depending on the number of non-zero values L in the bit allocation vector. If $L < N/2$, the configuration manager will generate the configuration data for a sparse FFT. Then it indicates the FFT task to perform sparse FFT and sends all the configuration data to the FFT task via the configuration channel. The configuration manager will go to standby

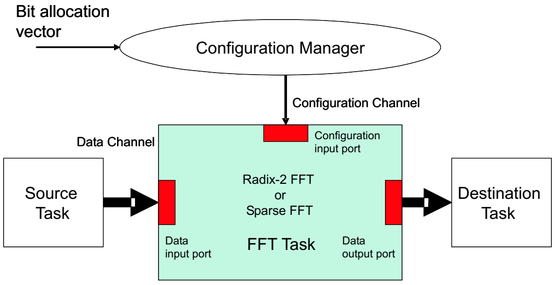


Figure 5.12: Task graph of reconfigurable sparse FFT

until a new bit allocation vector arrives. Depending on L, the FFT task either performs radix-2 FFT or the sparse FFT.

The TTL functions are called from the TTL C/C++ library to create tasks, define communication interfaces and generate the task graph. At the system level, the tasks are coded in C/C++. But in the platform implementation, the tasks can be implementations on a particular processor. Here we give a pseudo code example of the TTL implementation to show how the reconfiguration is done for the FFT task.

```
Task Task_FFT {initialization; while(true)
 {local variables;
 \\check the configuration updates
 tryAcquiredata(Task_FFT->config_inport)
 {\\update L
 ttl_read(Task_FFT->config_inport, L);
 \\read in configuration
 ttl_read(Task_FFT->config_inport, SFFT_CONFIG_DATA);
 }
 \\read in data
 for(i=0; i<num_samples; i++)
 ttl_read(Task_FFT->data_inport, proc_buffer[i]);
 \\sparse FFT or radix-2 FFT
 if (L<num_samples/2)
 {\\sparse_FFT processing
 call sparse_FFT;
 }
 else
 {\\radix-2 FFT
 call rad2_FFT;
 }
 \\write out results
 for(i=0; i<num_samples; i++)
 ttl_write(Task_FFT->data_outport, proc_buffer[i]);
 }
}
```

The FFT task checks the updates from the configuration channel. If a new configuration is generated by the configuration manager, the FFT task will read in the configuration data from the configuration channel via the configuration input port. Then the FFT task reads in samples from the source task. After reading the samples and the configuration data, the sparse FFT or radix-2 FFT procedure will be executed depending on L. After the FFT processing, the results will be written out to the data channel

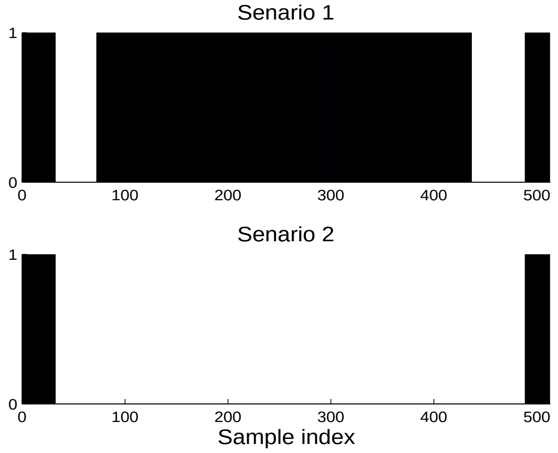


Figure 5.13: Example of reconfiguration

via the output data port. Both synchronization and data transfer are done by the TTL read and write functions. The procedures for sparse FFT and radix-2 FFT are software implementations in C/C++, but they can also be replaced by equivalent hardware implementations; for example, configurations for the Montium processor.

We applied the reconfigurable sparse FFT to an OFDM receiver based on the specification in [93] for the AAF system. The OFDM parameter set under consideration

$B = f_s$	N	Δf	T_u
$[MHz]$		$[kHz]$	$[\mu s]$
5.12	512	10	100

Table 5.5: OFDM parameters: Sample frequency and symbol duration.

is shown in Table 5.5. The bandwidth for the OFDM system is 5.12MHz and there are 512 subcarriers in one OFDM symbol. Thus, subcarrier spacing Δf is 10kHz and the useful symbol part duration T_u is $100 \mu s$. Therefore, we need a 512-point FFT to process an OFDM symbol, where some subcarriers might be zero according to the bit allocation vector. In figure 5.13, we show an example of the sparse FFT reconfiguration for the given OFDM system. We denote all the zero output indexes of the FFT with 0 and non-zero indexes with 1. In Scenario 1, 420 of 512 indexes are non-zeros which means that most of the subcarriers can be used by Cognitive Radio. To avoid causing interference to a potential licensed user, Cognitive Radio has to switch off a certain number subcarriers and re-assign the transmitted information to the available subchannels. This corresponds to Scenario 2 where only 56 out of 512 indexes are non-zeros. From Scenario 1 to Scenario 2, the FFT task is reconfigured from a radix-2 FFT to a sparse FFT. The high level TTL implementation has been run on a Linux PC. The computation result verifies the functional correctness of the sparse FFT. The TTL run-time environment can generate high level profile information in terms of computation workload and communication workload. The computation workload is measured by counting the number of annotated instructions while the communication workload is measured by counting the number tokens (data units) that are travelling through the TTL channels. Table 5.6 shows the computation workload of the FFT task in two scenarios generated by TTL. The reduction of computation in Scenario 2 is due to the sparse FFT which takes advantage of sparse data. Considering that the worst case execution time (WCET) for processing an OFDM symbol is $100\mu s$, we estimate the minimum required processing capacity for the S_1 and S_2 in Table 5.7. The profile information generated by the TTL run-time environment is independent of platforms. However, it can help to generate the platform dependent profile for specific implementations. By associating execution times with instructions, and by multiplying these execution times with the instruction counts, one can obtain a rough estimate of total execution time of a task on a certain processor. Considering the Montium for the FFT task, the Montium can execute one complex multiplication instruction in one clock cycle. From Table 5.7, we find that the Montium has to run at at least 23MHz for Scenario 1 and 19MHz for Scenario 2. In other words, the sparse FFT will save 16% processing capacity. Such a reconfiguration only takes place when the bit allocation vector has been updated. We expect the bit allocation vector not to change very often: at least it will be constant over several OFDM frames. Therefore the reconfiguration overhead is relatively small compared to the saving of computations. From the TTL profile information, we compare the computation workload of 512-point sparse FFT for various L with different zero distributions. Figure 5.14 shows that the computation workload increases with the number of non-zero L.

	complex multiplication (instruction count)
S_1	2304
S_2	1928

Table 5.6: Computation workload of the FFT task

	complex multiplications (per second)
S_1	23×10^6
S_2	19×10^6

Table 5.7: Minimum processing requirements

5.4.4 Run-time mapping

A System-on-Chip (SoC) architecture is a composition of parallel processing blocks, most frequently composed of commercially of-the-shelf intellectual property (IP) blocks. There is a trend to develop generic heterogeneous SoC architectures, which are flexible enough to run different applications. Mapping an application to such a heterogeneous SoC is more difficult than mapping to a homogeneous architecture. Todays common practice is to map the applications to the architecture at design-time. In this proposal we will consider how to perform the mapping at run-time. Run-time mapping offers a number of advantages over design-time mapping. In particular it can:

- adapt to the required resources. These resources may vary over time due to

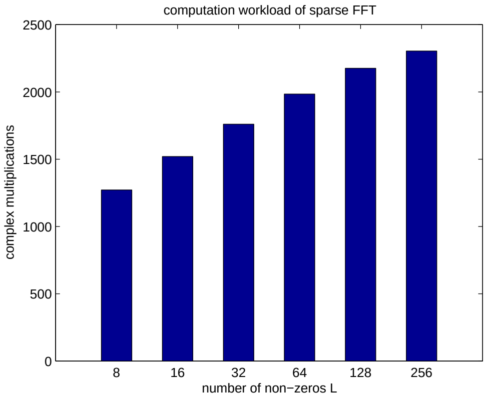


Figure 5.14: Computation workload of sparse FFT for 512 samples

applications running simultaneously or adaptation of algorithms to the environment.

- enable unforeseeable upgrades after first product release time, e.g. new applications and new or changing standards.

- avoid defective parts of a SoC. Larger chips mean lower yield. The yield can be improved when the mapper is able to avoid faulty parts of the chip. Also aging can lead to faulty parts, which are unforeseeable at design-time.

The proposed mapping algorithm maps applications to a heterogeneous SoC architecture at run-time. A number of inputs are required for this algorithm: a description of the applications, a library of process implementations and a description of the architecture.

- **Application Description:** An application is assumed to be described as a set of processes and interconnections between processes needed for communication. In this project, we assume that applications are described as TTL task graphs. In practice the generation of the task graph is done manually by an experienced designer, and we do no consider this process in this proposal. Besides a description of the structure of such an application, also Quality of Service (QoS) requirements have to be specified in the application. Examples of such QoS requirements are throughput, latency, maximum energy consumption etc.

- **Library Description:** For each process of an application, one or more process implementations have to be provided. A process implementation is the implementation of a process on a particular tile, e.g. object code for an ARM or a DSP or configuration data for an FPGA. A process implementation has several characteristics, e.g. the amount of energy it takes to execute the process on a particular tile of the architecture (see section below). Other examples are delay or tile utilization. This library (including the characteristics) is composed at designtime. In our approach, only the selection of process implementations is done at run-time.

- **Architecture Description:** The heterogeneous SoC architecture consists of multiple tiles of different types (e.g. ARM, FPGA, DSP) interconnected by a Networkon-Chip (NoC). For each tile, a number of characteristics has to be provided beforehand, such as the type of the tile, the amount of available memory, the clock frequency, etc. The NoC consists of routers and links. The links are used to interconnect routers or a tile with a router. It is possible to have different links in parallel between the same source and destination. Also the NoC characteristics have to be provided, such as the topology of the network, the frequency of the clock of the network, latency per router, etc.

These mapping problems are known to be NP-hard and in practice the corresponding optimization algorithms may run for hours on a workstation. When static design-time (thus off-line) mapping is used for thousands of devices, it is perfectly acceptable to have such a long execution time. However, when we perform the mapping at run-time for each individual device, the mapping process itself needs to be more flexible and efficient. This requires a quite different set of algorithms, because the exhaustive optimization methods used in most static allocation approaches are not practical anymore. To deal with the complexity of the problem, a hierarchical iterative approach is proposed in [94]. The idea is to solve the problem using multiple levels. At each level a particular decision is made that shrinks the search space. Decisions of previous levels are considered to be fixed at the lower levels. On higher levels not all details are taken into account to improve the speed of evaluation.

5.4.5 Publications with respect to the topic

- Zhang, Q. and Kokkeler, A.B.J. and Smit, G.J.M., “Cognitive Radio Design on an MPSoC Reconfigurable Platform,” In: *IEEE 2nd International Conference on Cognitive Radio Oriented Wireless Networks and Communications, CrownCom 2007*, 31 Jul - 03 Aug 2007, Orlando, USA. IEEE Communications Society. ISBN 1-4244-0815-6

- Zhang, Q. and Kokkeler, A.B.J. and Smit, G.J.M., “Cognitive Radio Design on an MPSoC Reconfigurable Platform,” In: *ACM/ Springer Mobile Networks and Application (MONET) Journal special issue on Cognitive Radio Oriented Wireless Networks and Communications*, 2008, to appear

- Zhang, Q. and Kokkeler, A.B.J. and Smit, G.J.M., “A System-level Design Method for Cognitive Radio on a Reconfigurable Multi-processor Architecture,” In: *IEEE International Symposium on System-on-Chip 2007*, 19-21 Nov 2007, Tampere, Finland. pp. 3-6. IEEE Circuits and Systems Society. ISBN 1-4244-1368-0

- Zhang, Q. and Kokkeler, A.B.J. and Smit, G.J.M., “Adaptive OFDM System Design For Cognitive Radio,” In: *11th International OFDM-Workshop*, 30th - 31st Aug. 2006, Hamburg, Germany. pp. 91-95. IEEE Communications Society, Germany Chapter.

- Smit, L.T. and Smit, G.J.M. and Hurink, J.L. and Broersma, H.J. and Paulusma, D. and Wolkotte, P.T., “Run-time mapping of applications to a heterogeneous

reconfigurable tiled system on chip architecture," In: *Proceedings of the International Conference on Field-Programmable Technology*, 2004.

5.5 Mapping algorithms onto the Montium

One of the major tasks of mapping Cognitive Radio baseband processing onto the proposed platform is implementing algorithms onto the Montium, the key element on the platform. The Montium is designed to perform computationally intensive DSP algorithms while offering reconfigurability for adaptive algorithms in an energy efficient manner. As the core algorithm in the OFDM based Cognitive Radio and multi-resolution sensing, dynamically reconfigurable FFT is mapped onto the Montium. Due to the high computational complexity, cyclostationary feature detection is a targeted algorithm to be processed on the Montium.

5.5.1 Dynamically reconfigurable FFT on the Montium

Two types of FFTs are considered: reconfigurable radix-2 FFTs and sparse FFTs. Radix-2 FFT may change its size to change the number of subcarriers in OFDM or to support multi-resolution sensing. Cognitive Radio may switch to a sparse FFT when a large number of subcarriers are nullified or to focus on a small portion of the spectrum for fine sensing.

The basic idea to make radix-2 FFTs reconfigurable is to reuse the computation structure and the twiddle factors of larger FFTs for smaller FFTs. An example is shown in figure 5.15 where a 4 point FFT can be reconstructed from an 8 point FFT by skipping the last stage butterfly operations. The same rule applies for reconstructing any smaller size FFTs from a larger FFT. Therefore, in our implementation we load the configuration of a large size FFT during the initialization and switch to small FFTs by adapting a small part of the configuration. In this way the reconfiguration overhead is reduced to a minimum since there is no need to reload the whole configuration during reconfigurations. From the computational structure of the sparse FFT in figure 5.5, we can see that a sparse FFT can re-use the radix-2 FFT configuration in the first part and only add an additional part for the multiplication and recombination of the twiddle factors.

An FFT algorithm consists of a number of stages depending on the size of the FFT. Each stage consists of several butterfly operations. One butterfly operation can be computed in one clock cycle using four Montium ALUs. The butterfly computation is done sequentially on a stage by stage basis. Since the butterfly operation has a repeating pattern, it can be implemented in a loop. A stage requires two sequencer instructions which are looped for the amount of input values that are needed. For the reconfigurable radix-2 FFT, during the initialization stage we load the configuration of the largest FFT needed into the sequencer. Run-time reconfiguration is achieved by altering the configuration memory. Each entrance in this memory defines an instruction and contains a control part which is used to implement a sequencing state machine. By reducing the loop counter values in the sequencer instructions and only using the stages needed, the large FFT can be reconfigured to a small FFT. Switching back to a large FFT can also be done by adjusting the loop counters and using more stages in

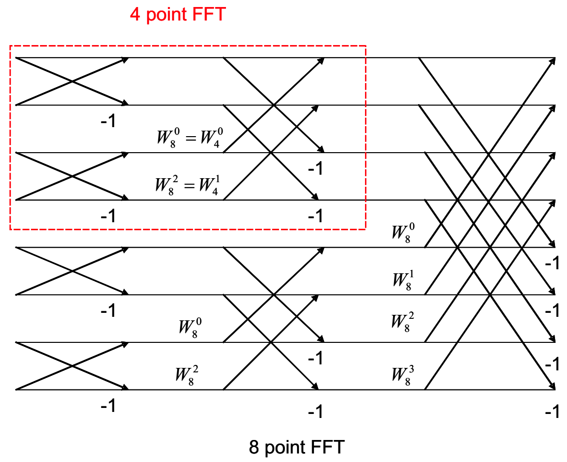


Figure 5.15: An example of reconfigurable radix-2 FFT

the sequencer. Table 5.8 shows a simplified version of what a 16 point FFT sequencer

Seq PC	ALU	AGU	jump	loop	description
[0]	[0]	[3]	[1]	1	Start of stage 1
[1]	[1]	[4]	[2]	8 (4)*	
[2]	[1]	[4]	[3]	1	Start of stage 2
[3]	[0]	[3]	[4] ([7])*	8 (4)*	
[4]	[0]	[3]	[5]	1	Start of stage 3
[5]	[1]	[4]	[6]	8	
[6]	[1]	[4]	[7]	1	Start of stage 4
[7]	[0]	[3]	[8]	8 (4)*	

Table 5.8: A simplified sequencer program for a reconfigurable FFT changing from a 16 point to an 8 point FFT where the adaption has been suggested

program would look like. The numbers below the AGU and ALU column represent indices of specific Address Generation Unit (AGU) or ALU instructions. In order to change this 16 point FFT sequencer program to an 8 point FFT only the loop counter values have to be adjusted and a stage has to be skipped. By adjusting the jump target in sequencer instruction 3 from 4 to 7 and adjusting the loop counters from 8 to 4, an 8 point FFT is created. The values changed are indicated by an asterisk in table 5.8. The method of mapping a sparse FFT onto the Montium has been indicated in [87]. During the input mapping and the recombination stage, the memory addressing is constantly hopping from a position in one block to the same position in another, see figure 5.5 (e.g. from $x_0(0)$ to $x_1(0)$). Each memory of the Montium has an Address Generation Unit (AGU) which can generate the required addressing pattern. The radix-2 FFT configuration can be re-used for each memory block in the first part. In the second part where multiplications with twiddle factors and recombination are done, we use the 5th ALU to calculate the indices to be multiplied and used as a memory address for AGU. This address generation costs 3 extra clock cycles per each nonzero value, which is the efficiency bottleneck of the sparse FFT on the Montium.

Table 5.9 shows the measured execution time of radix-2 FFTs on the Montium run-

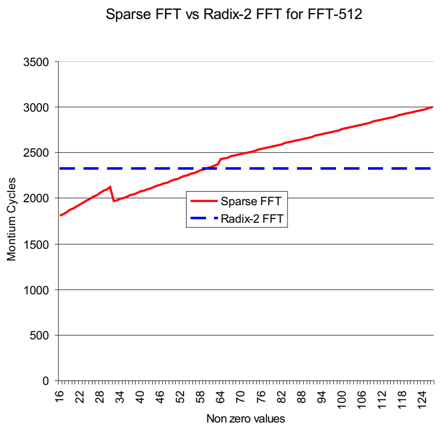


Figure 5.16: The performance of the sparse FFT vs radix-2 FFT for FFT-512 on the Montium

ning at 6.8MHz in comparison with the fixed point implementations on the ARM946E-S running at 86MHz. Although the Montium runs at 12 times lower frequency than the ARM, it still outperforms the ARM by more than 4 times in processing speed. The power consumption of the Montium, in 0.13μm technology, is estimated at 0.577 mW/MHz. Therefore the largest reconfigurable FFT (FFT-512) on the Montium only cost 1.35 μJ which is comparable to an ASIC implementation. Figure 5.16 shows the

FFT size	512	256	128	64
Montium (ms)	0.35	0.15	0.07	0.03
ARM (ms)	1.31	0.59	0.27	0.12

Table 5.9: The execution time of radix-2 FFTs on the Montium tile processor and the ARM on the BCVP

number of clock cycles of the sparse FFT vs. the radix-2 FFT for 512 samples on the Montium as a function of the amount of non-zero subcarriers. Due to the address generation bottleneck, the final sparse FFT implementation on the Montium turns out not as efficient as the estimation in [87]. As shown in figure 5.16, the complexity of the sparse FFT exceeds the radix-2 FFT when there are more than 64 non-zero outputs. However, it is still more efficient than the radix-2 FFT when a large number of subcarriers is switched off (e.g. 480 out of 512). An additional benefit of the sparse FFT is that only the non-zero outputs are sent through the on-chip communication architecture. Therefore it results in less communication cost which is considered to be the bottleneck for smaller feature sized SoCs in the future. Table 5.10 shows the costs of the reconfiguration for both radix-2 FFT and sparse FFT (less than 64 non-zeros) in terms of bytes. When using reconfigurable code, a large configuration is sent to the Montium at initialization and changing FFT size involves small pieces of reconfigurable code. When using static code, changing FFT size requires a complete reconfiguration. Dynamic reconfiguration is about 10 times more efficient than reloading the complete static configuration.

	reconfigurable code	static code
Initialization	1792	0
to 512-point	104	1138
to 256-point	82	1054
to 128-point	78	970
to 64-point	80	886
to 512-sparse (less than 64 nonzeros)	78	–

Table 5.10: Bytes that need to be sent for reconfiguration

Task	#cycles
multiply accumulate	12192
read data	381
FFT	1040
reshuffling	256
initialisation	127
total	13996

Table 5.11: Number of processor cycles

5.5.2 Cyclostationary feature detection on the Montium

Due to its high computational complexity, cyclostationary feature detection is the exact type of application that the Montium is targeted. In [95], we proposed a two-step methodology to analyse the mapping of cyclostationary feature detection. In the first step, the tasks to be executed by each core are determined in a structured way using techniques known from the design of array processors. In the second step, the implementation of tasks on a processing core is analysed.

The quantitative analysis of mapping CFD onto multiple Montium cores is based on the simulation of an application where 256-point spectra are analysed. Both f and a range from -63 to +63 which implies that $P = 127$ and $F = 127$. The platform used within the AAF project consists of 4 Montium cores ($Q = 4$). The number of tasks to be executed by one Montium core is therefore smaller than or equal to 32 ($T = 32$). The number of memory locations needed for storing the results, when accumulating over n, equals $T \cdot F = 32 \cdot 127 < 4K$ complex values or less than 8K real values. The total memory capacity of the Montium memories M01 to M08 equals 8K words of 16 bits. So, for dynamic ranges smaller than 96 dB, the Montium memories are sufficiently large. The communication shift registers are mapped onto memories M09 and M10. Each memory contains 32 complex values. From each memory a value is read every clockcycle. The read-address is generated by an Address Generation Unit (AGU) which accompanies each memory.

We simulated the tasks for a single Montium tile with the Montium simulator. Table 5.11 gives an overview of the number of processor cycles required for the different tasks. The total number of complex multiply accumulate operations equals $T \cdot F = 4064$. Simulations show that a multiply-accumulate requires three clockcycles, so the required number of clock cycles equals 12192. For each 32 multiply accumulate operations, 3 additional clockcycles are needed to read data which leads to 381 additional clockcyles. The DSCF is based on a 256-point spectrum which can be calculated by one Montium in 1040 clockcycles. The reshuffling of the conjugated values is done in 256 clockcycles and initially loading the Montium with data requires 127 clockcycles. The total number of clockcycles for 1 integration step in the calculation of a DSCF then equals 13996. The maximum clockspeed of a Montium core equals 100 MHz and therefore the time required for the calculation of one integration step in the calculation of the DSCF equals 139.96 μs. To analyse 256 samples takes approximately 140 μs. If all samples of a stream are analysed in blocks of 256 samples, an analysed bandwidth of approximately 915 kHz is realised. A single Montium occupies approximately 2 mm^2 using the Philips 0.13 μm CMOS12 process technology. A platform consisting of 4 Montium processors will occupy approximately 8 mm^2. Typical power consumption of a Montium processor is estimated to be 500 μW/MHz. When running on 100 MHz, this results for 4 Montium tiles in 200 mW. In this evaluation we already used the scalability property of the application, the mapping and the platform. The analysed bandwidth, chip area and power consumption scale linearly with the number of Montium processors. This property can be used to estimate performance of other platform configurations.

5.5.3 Publications with respect to the topic

- Zhang, Q. and Walters, K.H.G. and Kokkeler, A.B.J. and Smit, G.J.M., “Dynamically Reconfigurable FFTs for Cognitive Radio on a Multiprocessor Platform,” In: *International Conference on Engineering of Reconfigurable Systems and Algorithms*, 2008, to appear.

- Kokkeler, A.B.J. and Smit, G.J.M. and Krol, T. and Kuper, J., “Cyclostationary Feature Detection on a tiled-SoC,” In: *DATE2007 Proceedings*, 16-20 April 2007, Nice, France. pp. 171-176. European Design and Automation Association. ISBN 978-3-9810801-2-4

- Walters, K.H.G. *Cognitive Radio on a reconfigurable platform*, Master Thesis, University of Twente Master’s thesis, Univ. of Twente.

5.6 Conclusions

In this chapter, we proposed an MPSoC architecture to support adaptive baseband processing of Cognitive Radio. The key element on this platform is a home grown coarse grain reconfigurable processor call the Montium. The Montium tile processor offers reconfigurability in an energy efficient manner.

We investigated three DSP algorithms particularly interesting for Cognitive Radio, namely reconfigurable sparse FFT, discrete cyclostationary feature detection and filter bank. The sparse FFT is proposed as a novel alternative to normal FFT for OFDM based Cognitive Radio where a large number of subcarriers are deactivated. Based on this sparse FFT, we also proposed a novel energy based multi-resolution spectrum sensing method which enable Cognitive Radio to focus on a small part of interested spectrum in a finer resolution with low computational cost. Due to its high computational complexity, the DCFD has been analyzed and mapped onto the MPSoC platform. We proposed an oversampled filter bank multicarrier for Cognitive Radio based on the generalized DFT implementation.

A system level design method, call task transaction level interface (TTL), has been proposed for mapping algorithms onto the MPSoC platform. The method is used for build system level application of Cognitive Radio for MPSoC and also for system level profile generation. The generated profile can help to evaluate the efficiency of algorithms and provide information for the subsequent run-time mapping.

Chapter 6

Executive Summary

by André Kokkeler

The design of a network, datalink and physical layer for Cognitive Radio is a 'wide and open' area of research. Within the AAF project, a first 'map' of this area has been drawn from two different perspectives. A first perspective is to look at Cognitive Radio from an OSI layered model point of view. The AAF project covered network- and datalink-layer aspects of Cognitive Radio in workpackage 4 (WP4). Physical layer aspects have been addressed in WP3: spectrum sensing and medium access techniques have been explored for their suitability for Cognitive Radio. The second perspective is to consider Cognitive Radio from an executing platform point of view. Since Cognitive Radio is expected to find its use in both stationary and mobile applications, non-functional characterstics are evenly important as functional characteristics. The two most important non-functional charateristics are flexibility (reconfigurability) and power consumption (computational efficiency).

Since research from both perspectives was conducted in parallel within the AAF project, it was not possible to use identical algorithms for both the functional evaluation and the platform evaluation. In general, for functional evaluation, new algorithms have been developed and for the platform evaluation, existing algorithms have been evaluated and at most adapted to increase computational efficiency.

6.1 Functional evaluation of network- and datalink layer

Concerning the evaluation of the network- and datalink layer of an Opportunistic Spectrum Access (OSA) network, two important issues are the selection and effects of a Transport Control Protocol (TCP) over OSA links and the design of a medium access control (MAC) protocol.

TCP over OSA links

The most importante versions of TCP that are currently in use are: Linux New Reno, Linux Vegas, NS New Reno and NS Sack. We have investigated the performance of these TCP flavors in an OSA environment. The investigated TCP stacks can achieve better than 95% efficiency on OSA links with widely varying characteristics, under a very wide range of network configurations, if a large (but not unrealistically so) buffer is available at the Base Station, and the receiver employs Selective Acknowledgments. We have also seen that TCPs have trouble adapting to even brief reductions in capacity, if end-to-end delay is large. This implies that the probability of false alarm, a parameter of the OSA link's Primary User (PU) detection process which indicates the probability that a PU is detected while not present, may have a larger effect on throughput than is apparent from theoretical analysis of TCP's *steady state* behavior.

Design of a MAC protocol
From an extensive literature study we distilled five important features:

- Bootstrapping. During bootstrapping a secundary user decides which channels can be used for communication. There are basically two options: a third party provides the information on channel usage or the secundary user has to find out itself which channels are in use by a primary user.

- Control channel. For the exchange of control information between secundary users, a control channel needs to be present. Four different options have been identified: a dedicated control channel (DDC), a hopping control channel (HCC), a split phase control channel (SPCC) and a multiple rendez-vous control channel (MRCC).

- Scanning. The aim of scanning is to detect primary user activity. The quality of scanning can, in general, be improved if the scanning period is lengthened.

- Radio front-end. Depending on the MAC protocol, several functions can be executed in parallel: sending and receiving data and spectrum sensing. Either several front-ends should be present or time-multiplexed use of front-ends is required.

- Interference Management Policies. Harmful interference to primary users cannot be avoided since their presence cannot be be detected with 100 percent certainty. The maximum level of interference is typically specified through interference policies (IPs). Three major policy classes are identified: time based, power based and collision based.

Many proposed solutions do not cover crucial elements of a proper OSA MAC protocol design. Since the operating conditions of OSA networks are typically unknown during the design time phase, the bootstrapping procedure to setup the network before communication is very important. However, it is omitted in many of the protocol designs. In case of OSA networking, this bootstrapping cannot be considered to be a one-time effort at the start of the communication network, so it is crucial to make it as efficient as possible and embed it in the MAC protocol design. Also, the required scanning for the presence of a PU is sometimes omitted in the protocol design or performance analysis. More importantly, the specification of policies to regulate the coexistence with PUs is often described very vaguely or even fully omitted. It can be concluded that although many individual contributions can be found, it is important to assess how these subtasks can be integrated together into a complete solution to be able to fully assess the expected QoS of OSA networks.

The organization of the Control Channel is a very important aspect of multichannel OSA networking. Although the solutions proposed in literature always assume the availability of a fixed channel for control information exchange (for DCC this channel is only for control, in SFCC the channel is also used for data), this is not necessarily optimal. Especially in the case that there are a lot of possible channels to use, a fixed control channel easily becomes the bottleneck. Also, when no channel can be assumed to be free from PU activity, it is best to spread the control exchanges over different channels as much as possible. As a result, MRCC outperforms DCC, HCC and SFCC over a broad range of PU traffic conditions.

No solutions found so far in literature assess the QoS given to the secondary network in detail. This is however very crucial since the introduction of OSA networks only makes sense if a sufficient level of QoS can be expected. We have studied the delay and throughput performance of a broad range of OSA designs as a function of PU activity. Also, the fundamental trade-off between PU QoS and SU QoS has been assesed. The more freedom is given to the SU to access the channel, the more capacity it can use and the better its performance. However, more freedom to the SU means less guarantees for the PU and the success of OSA networking will depend on how well we can optimize this trade-off with a given policy.

6.2 Functional evaluation of the Physical Layer

6.2.1 Medium Access Techniques

The most important characteristic of a suitable medium access technique for Cognitive Radio is that it should be ’frequency agile’. This means that the technique should be able to flexibly change, in a controlable way, the spectral usage such that parts of the operating frequency band of the Cognitive Radio user are not contaminated with power. We have investigated three medium access techniques:

- Orthogonal Frequency Division Multiplexing (OFDM)
- Transform Domain Communication System (TDCS)
- Wavelet Packet MultiCarrier Modulation (WPMCM)

OFDM is considered as a major candidate modulation technique for Cognitive Radio mainly because of its ability to control spectral usage in a controllable way by switching on or off carriers (adaptive bit loading). However, due to spectral leakage, power of a single switched-on carrier is not only concentrated in the assigned frequency bin but spreads over a larger frequency range. One of the key research issues within the AAF project has been to reduce this spectral leakage. By means of windowing, spectral leakage can be reduced. A better-than-raised-cosine (BTRC) window results in the lowest sidelobes. This technique, in combination with carrier deactivation can be used to reach acceptable spectral leakage levels. A second isssue is adaptive bit loading. ’Good’ carriers can be loaded with more bits than carriers that suffer from severe channel conditions. Estimating the channel accurately is therefore of high importance. Obtaining information about the channel from a Wiener filter in combination with a Fisher carrierwise bit loading algorithm was found to result in the lowest bit error rate. To establish accurate channel estimates in case not all carriers are used, carefull distribution of pilot symbols over the remaining carriers is important. A hexagonal pattern in combination with the concept of virtual pilots shows a large increase in performance compared to fixed pilot patterns.

The Transform Domain Communication System (TDCS) has been researched concerning its capabilities for Cognitive Radio. TDCS as is, has too low a data rate for Cognitive Radio and for that reason, TDCS has been extended. An additional data input has been created and the data is used to modulate the carriers that generate the time signal before modulation. This way, the data rate can be increased or equivalently, less signal power can be used for equal Bit Error Rates (BERs). TDCS with embedded 256 PSK symbols requires 9 dB less SNR to have the same BER as conventional TDCS.

As a third option, Wavelet Packet Multicarrier Modulation (WPMCM) has been investigated. In this technique, the conventional Fourier based complex exponential carriers within OFDM are replaced with orthonormal wavelet packet bases. For both systems (OFDM and WPMCM), carriers need to be switched off to avoid jamming licensed users. WPMCM needs more carriers to be switched off than OFDM but for an equal number of carriers switched off, WPMCM outperforms OFDM.

MIMO (Multiple Input Multiple Output) has attracted a lot of attention in the last few years as a technique to enhance communication capacity in a (highly) scattering environment. As such, it can be used for Cognitive Radio as well. One of the classical MIMO techniques is V-BLAST. Its use has been evaluated in conjunction with the three access techniques used within AAF. Both TDCS and WPMCM outperform OFDM in case of MIMO.

6.2.2 Spectrum Scanning

To avoid interference, a cognitive radio should sense the spectrum for the presence of primary users. Only if a spectral band is unused, a cognitive radio may temporary reuse it as a secondary user. To attain a sufficient level of certainty about the absence of a primary user, large technical challenges have to be solved. To focus research on physical layer aspects of spectrum sensing we developed and implemented the Cognitive Radio Verification Platform (CRVP). The goal of the CRVP is to demonstrate physical layer issues involved in spectrum sensing. It consists of a PC application that combines several system level simulations with a graphical user interface. The CRVP is designed as a hybrid between simulation and implementation. In simulation mode, signals are synthetically generated from baseband models of various modulation schemes. Also propagation and analog front-end impairments are simulated. In implementation mode, the CRVP can connect to a Universal Software Radio Peripheral (USRP) or receiver hardware. With such hardware, live radio signals can be measured and analyzed at run time or stored to disk for offline analysis. Various spectrum estimation methods where studied and implemented on the CRVP under which the Welch averaged periodogram method and cyclostationary feature detection. Also modulation type recognition and feature extraction methods have been studied. Cyclostationary feature detection is generally considered to be superior to energy detection in low, or even negative SNR regimes. However, no solid evidence is available that this technique is useful to improve spectrum sensing for cognitive radio. Simulation and analysis of these algorithms performed in the AAF project show that indeed cyclostationary features can be measured as predicted by theory. However, it was also observed that the computational load of the algorithms is too high for current day mobile processors. It is expected however, that future mobile processors have enough processing power for these algorithms. Besides the study on sensing and detection algorithms, also system level aspects were addressed. Radio spectrum monitoring data, collected by Agentschap Telecom, was analyzed and measurements with a spectrum analyzer have been made. From these measurements we obtained an overview of current 'white space' and typical usage of frequency bands. Furthermore, we identified that certain bands can be classified as 'gray space'. Such bands contain important primary users, but because of bursty signal patterns, spreading codes, and low transmit power they may appear as white space in superficial measurements. Examples are satellite downlinks and uplink bands from trunked radio systems.

6.3 Platform evaluation

The platform evaluation for Cognitive Radio basically consist of four steps:

- Identification and specification of the Cognitive Radio application to be mapped onto a platform
- Platform definition
- High level modeling and partitioning of the application
- Mapping of the application

Specification

At the begining of the AAF project, OFDM was considered as a major candicate medium access technique for Cognitive Radio. Therefore an OFDM based Cognitive Radio was specified using adaptive bit loading and power loading. Note that, by loading no bits onto a carrier, that carrier is effectively switched off. Bit allocation vectors are assumed to be communicated to both a transmitter and receiver via a Common Control Channel. A crucial functionality required for Cognitive Radio, is spectrum sensing. Basically three different methods had been proposed for use within Cognitive Radio: power detection, matched filter and cyclostationary feature detection. The power detection method has no sub-noise detection capabilities and a matched filter is not considered to be feasible for Cognitive Radio. For that reason, an improved power detection method and Spectrum Sensing by means of cyclostationary feature detection is specified for the platform evaluation.

Platform definition

The targeted platform for Cognitive Radio is a heterogeneous reconfigurable multiproccesor System-on-Chip (MP-SoC). This MP-SoC is a tiled architecture where tiles can be various processing elements including General Purpose Processors (GPPs), Field Programmable Gate Arrays (FPGAs), Application Specific Integrated Circuits (ASICs) and Domain Specific Reconfigurable Hardware. The tiles of an MP-SoC are interconnected by means of a Network-on-Chip. The different types of processing elements are used for different types of processing. Within the AAF project, we aimed at implementation of computational kernels like the FFT onto Domain Specific Reconfigurable Hardware. More specifically we aimed at implementation onto the Montium, developed at the University of Twente. The Montium is tailored towards the signal processing domain. It is flexible enough to adapt to different algorithms with good energy efficiency.

High level modeling and partitioning

Parameterizable OFDM has been modeled using a Task Transaction Level (TTL) interface. By means of this interface, an application can be split explicitely into computational kernels and communication between the kernels. The computational kernels become ’tasks’ which communicate via ports. This separation into computation and communication is a first step towards implementation. The biggest advantages of using TTL are:

- A first step towards implementation can be made at an abstract level where low level details can be ignored, increasing productivity of the designer.
- The partitioned model is executable so functional correctness can be checked.
- The TTL environment can generate high level profile information in terms of computation workload and communication workload.

A TTL model of parameterizable OFDM has been realized and simulated for correctness. An overview of the computation workload, which is determined automatically by TTL, confirmes that the FFT is the most computationally intensive part of an OFDM transmitter or receiver. Furthermore, the indicated computational workload of the different parts can be used to estimate execution times on a final platform. For the parameterizable OFDM application using 512 carriers, we concluded that a single Montium processor offers sufficient raw computing performance (the total execution time was estimated to be 960 μs while the available time is 3.125 ms). Using the indicated computational workload, first estimates of energy consumption can be given as well.

Mapping of the application

Within the AAF project, we investigated the mapping of Adaptive OFDM and Spectrum sensing onto an MP-SoC, more specifically onto the Montium. As already was concluded from the high-level modeling stage, the FFT part of OFDM is the most computationally complex part and we therefore concentrated on efficient implementation of the FFT on the Montium reconfigurable processor. Ordinary mapping of an FFT onto the Montium was already known at the start of the project. However, in case of Cognitive Radio, many carriers might be switched off which implies that many computations within the FFT are useless. To reduce power consumption, mechanisms to reduce the number of useless calculations have led to the implementation of the sparse-FFT. If less than half of the carriers are actually used, the sparse-FFT gives a higher computational efficiency. The minimum complexity of the sparse-FFT equals half of the complexity of a 'conventional' FFT. Because carrier usage is anticipated to be very agile, the sparse-FFT has to be reconfigured for different carriers very efficiently (fast against low power). When using the reconfigurable Montium processor, a large configuration is sent to the processor at initialization and changing the positions of used carriers involves small pieces of reconfigurable code. The effect is that for example, a sparse-FFT of 512 carriers at most, and less than 64 used carriers can be reconfigured with 78 bytes. In this case reconfiguration of a 100 MHz Montium will take less than 1 μs.

Anticipating on the fact that 'normal' OFDM might not give sufficient suppression of signals in adjacent channels used by the primary users, we developed a computationally efficient oversampled filter bank multicarrier system for Cognitive Radio. At the moment of writing the white paper, no results for this system are available yet.

A second computationally intensive part of Cognitive Radio is the Spectrum sensing part. Two different alternatives have been investigated: Multi-resolution spectrum sensing and cyclostationary feature detection. Multi-resolution spectrum sensing is based on the dynamically reconfigurable FFT as described above. The total bandwidth is first sensed using coarse resolution followed by a fine resolution sensing performed on only a portion of the interesting bands for Cognitive Radio. Because the energy contents of carriers that are not of interest are not calculated power is saved by avoiding unnecessary computations. If less than 25 percent of the total bandwidth requires fine resolution sensing, our proposed multi resolution sensing approach is more efficient that using a large FFT with equivalent resolution.

Using an FFT to detect the presence of signals is usually refered to as power detection. Because of the limited detection capabilities of power detection, it is doubtfull whether it is really applicable for Cognitive Radio. An alternative is to use cyclostationary feature detection. By means of cyclostationary feature detection, correlation between signals at different carriers can be detected. The implementation of cyclostationary feature detection onto an MP-SoC has been investigated. When using spectra consisting of 256-points (obtained after an 256 point FFT), the correlation step within cyclostationary feature detection will take approximately 140 μs using a platform with 4 Montiums which implies that a band of approximately 1 MHz can be fully analysed. As a reference, calculating a 256 point spectrum by means of an FFT on a single Montium takes approximately 10 μs.

6.4 Conclusions

The conclusion of the research within the AAF project concerning the network-, datalink and physical layer are:

- Modern real world TCP stacks can achieve high efficiency on OSA links.
- A dedicated control channel is not always the right solution for OSA networks.
- OFDM is considered as the strong modulation mode for OFDM but a Transform Domain Communication System and Wavelet Packet Modulation are adequate alternatives.

- MIMO is effective for Cognitive Radio.
- Measurements acknowledge the general opinion that spectrum is scarsely used.
- Cyclostationary Feature Detection has shown to detect signals that cannot be detected by power detection using real-life measurements.
- A MultiProcessor System-on-Chip platform has shown to be a suitable platform for Cognitive Radio based on OFDM giving sufficient flexibility against acceptable power consumption.

Appendix A

List of Abbreviations

AGU	address generation unit
ALU	arithmetic and logic unit
ASIC	application specific integrated circuit
AWGN	additive white gaussian noise
BCVP	basic concept verification platform
BDP	bandwidth delay product
BER	bit error rate
BPSK	binary phase shift keying
BTRC	better than raised cosine
CCSK	cyclic code shift keying
CDMA	code division multiple access
CFD	cyclostationary feature detection
CFWCN	cognitive functionality wireless communications node
CM	configuration manager
CR	cognitive radio
CTF	channel transfer function
CWSE	coded weak subcarrier excision
DCFD	discrete cyclostationary feature detection
DCC	dedicated control channel
DFT	discrete Fourier transform
DSRH	domain specific reconfigurable hardware
DSP	digital signal processing
ESM	exclusive spectrum management
FDM	frequency division multiplexing
FFT	fast Fourier transform
FMT	filtered multitone
FPGA	field programmable gate array
GI	guard interval
GPP	general purpose processor
GSM	global system for mobile communications
HCC	hopping control channel
HSM	hierarchical spectrum management
IFFT	inverse fast Fourier transform

IP	intellectual property
IRCTR	international research center for telecommunications and radar
ISI	intersymbol interference
LU	license user
MAC	medium access control
MCM	multicarrier modulation
MIMO	multiple input multiple output
MPSoC	multiprocessor system on chip
MRCC	multiple rendezvous control channel
NoC	network on chip
OFDM	orthogonal frequency division multiplexing
OSA	opportunistic spectrum access
OSFB	oversampled filter bank
PAM	pulse amplitude modulation
PSD	power spectral density
PU	primary user
QAM	quadrature amplitude modulation
QoS	quality of service
RFE	radio front end
RU	rental user
SBLA	simple blockwise loading algorithm
SC	spectrum commons
SDR	Software Defined Radio
SINR	signal to noise and interference ratio
SNR	signal to noise ratio
SoC	system on chip
SPCC	split phase control channel
SU	secondary user
TCP	transport control protocol
TDCS	transform domain communication system
TDD	time division duplex
TDM	time division multiplexing
TTL	task transaction level
UWB	ultra wide band
V-BLAST	vertical bell layered space time
WCET	worst case execution time
WPMCM	wavelet packet multicarrier modulation

Bibliography

[1] M. Cave, C. Doyle, and W. Webb. *Essentials of Modern Spectrum Management*. Cambridge University press, 2007.

[2] Joseph Mitola III. Cognitive radio for flexible mobile multimedia communications. In *Proc. IEEE MOMUC'99*, San Diego, CA, November 15–17, 1999.

[3] R. Venkatesha Prasad, Przemysław Pawełczak, James Hoffmeyer, and Steven Berger. Cognitive functionality in next generation wireless networks: Standardization efforts. *IEEE Commun. Mag.*, 46(5):72–78, April 2007.

[4] Qing Zhao and Brian M. Sadler. A survey of dynamic spectrum access: Signal processing, networking, and regulatory policy. *IEEE Signal Processing Mag.*, 24(3):79–89, May 2007.

[5] Bhaskar Sardar and Debashis Saha. A survey of TCP enhancements for last-hop wireless networks. *IEEE Communications Surveys & Tutorials*, 8(3):20–34, 2006.

[6] K. Fall and S. Floyd. Simulation-based comparisons of Tahoe, Reno and SACK TCP. *ACM Computer Communication Review*, 26(3):5–21, 1996.

[7] S. Floyd and T. Henderson. The NewReno modification to TCP's fast recovery algorithm. Request for Comments (Experimental) 2582, Internet Engineering Task Force, April 1999.

[8] L.S. Brakmo and L.L. Peterson. TCP Vegas: end to end congestion avoidance on a global internet. *IEEE J. Select. Areas Commun.*, 13(8):1465–1480, October 1995.

[9] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow. TCP selective acknowledgment options. Request for Comments (Proposed Standard) 2018, Internet Engineering Task Force, October 1996.

[10] A. Lo, G.J. Heijenk, and I.G.M.M. Niemegeers. On the performance of TCP Vegas over UMTS/WCDMA channels with large round-trip time variations. In *Proc. 9th IFIP TC6 International Conference on Personal Wireless Communications*, Lecture Notes in Computer Science 3260, pages 330–342. Springer Verlag, September 2004.

[11] David X. Wei and Pei Cao. NS-2 TCP-linux: An NS-2 TCP implementation with congestion control algorithms from linux. In *Proc. ACM Workshop on NS-2 (ValueTools 2006)*. ACM press, 2006.

[12] S. Sengupta, S. Brahma, M. Chatterjee, and S.N. Shankar. Enhancements to cognitive radio based IEEE 802.22 air-interface. In *Proc. IEEE ICC'07*, Glasgow, Scotland, June 24–28, 2007.

[13] Gunter Auer, Harald Haas, and Peter Omiyi. Interference aware medium access for dynamic spectrum sharing. In *Proc. IEEE DySPAN'07*, Dublin, Ireland, April 17–20, 2007.

[14] Xin Liu and Zhi Ding. ESCAPE: A channel evacuation protocol for spectrum-agile networks. In *Proc. IEEE DySPAN'07*, Dublin, Ireland, April 17–20, 2007.

[15] Carlos Cordeiro and Kiran Challapali. C-MAC: A cognitive mac protocl for multi-channel wireless networks. In *Proc. IEEE DySPAN'07*, Dublin, Ireland, April 17–20, 2007.

[16] Michael Timmers, Antoine Dejonghe, Liesbet Van der Perre, and Francky Catthoor. A distributed multichannel MAC protocol for cognitive radio networks with primary user recognition. In *Proc. IEEE CrownCom'07*, Florida, USA, August 1–3, 2007.

[17] Noun Choi, Maulin Patel, and S. Venkatesan. A full duplex multi-channel MAC protocol for multi-hop cognitive radio networks. In *Proc. IEEE CrownCom'06*, Mykonos Island, Greece, June 8–10, 2006.

[18] Tao Shu, Shuguang Cui, and Marwan Krunz. Medium access control for multi-channel parallel transmission in cognitive radio networks. In *Proc. IEEE GLOBECOM'06*, San Francisco, CA, Nov. 27 – Dec. 1, 2006.

[19] Srivatsan Sankaranarayanan, Panagiotis Papadimitratos, and Amitabh Mishra. A bandwidth sharing approach to improve licensed spectrum utilization. *IEEE Commun. Mag.*, 43(12):S10–S14, December 2005.

[20] Liangping Ma, Xiaofeng Han, and Chien-Chung Shen. Dynamic open spectrum sharing MAC protocol for wireless ad hoc networks. In *Proc. IEEE DySPAN'05*, Baltimore, MA, November 8–11, 2005.

[21] Juncheng Jia, Qian Zhang, and Xuemin Shen. HC-MAC: A hardware-constrained cognitive MAC for efficient spectrum management. *IEEE J. Select. Areas Commun.*, 26(1):106–117, January 2008.

[22] Hang Su and Xi Zhang. Cross-layer based opportunistic MAC protocols for QoS provisioning over cognitive radio wireless networks. *IEEE J. Select. Areas Commun.*, 26(1):118–129, January 2008.

[23] Liangping Ma, Chien-Chung Shen, and Bo Ryu. Single-radio adaptive channel algorithm for spectrum agile wireless ad hoc networks. In *Proc. IEEE DySPAN'07*, Dublin, Ireland, April 17–20, 2007.

[24] Qing Zhao, Lang Tong, Ananthram Swami, and Yunxia Chen. Decentralized cognitive MAC for opportunistic spectrum access in ad hoc networks: A POMDP framework. *IEEE J. Select. Areas Commun.*, 25(3):589–600, April 2007.

[25] Jeonghoon Mo, Hoi-Sheung Wilson So, and Jean Walrand. Comparison of multi-channel MAC protocols. *IEEE Trans. Mobile Comput.*, 7(1):50–65, January 2008.

[26] Tevfik Yucek and Huseyin Arslan. A survey of spectrum sensing algorithms for cognitive radio applications. *IEEE Commun. Surv. and Tut.*, April 2007. accepted for publication.

[27] Fadel F. Digham, Mohamed-Slim Alouini, and Marvin K. Simon. On the energy detection of unknown signals over fading channels. In *Proc. IEEE ICC'03*, Anchorage, AK, May 11–15, 2003.

[28] R. Prasad. *OFDM for Wireless Communications Systems*. Artech House, Norwood, 2004.

[29] P. S. Chow et al. A Practical Discrete Multitone Transveiver Loading Algorithm for Data Transmission over Spectrally Shaped Channels. In *IEEE Transactions On Communications No.2*, volume 43, pages 773–775, February 1995.

[30] R. F. H. Fischer and J. B. Huber. A New Loading Algorithm For Discrete Multitone Transmission. In *IEEE Proc. GLOBECOM '96*, pages 724–728, November 1996.

[31] A. N. Barreto and S. Furrer. Adaptive Bit Loading for Wireless OFDM Systems. In *Proc. IEEE Int. Sym. Personal, Indoor, Mobile Radio Commun.*, volume 2, pages 88–92, 2001.

[32] M. Lei and P. Zhang. Subband Bit and Power Loading for Adaptive OFDM. In *Proc. 58th IEEE Vehicular Technology Conference (VTC)*, volume 3, pages 1482–1486, October 2003.

[33] R. Gruenheid et al. A Blockwise Loading Algorithm for The Adaptive Modulation Technique in OFDM Systems. In *Proc. Veh. Technol. conf.*, volume 2, pages 948–951, May 2001.

[34] R. Gruenheid et al. Adaptive Modulation for the HIPERLAN/2 Air Interface. In *Proc. 5th International OFDM Workshop*, Hamburg,Germany, September 2000.

[35] P. Hoeher et al. Pilot Symbol Aided Channel Estimation in Time and Frequency. In *Proc. IEEE Global Conference (GLOBECOM)*, pages 90–96, Phoenix,USA, April 1997.

[36] P. Hoeher et al. Two Dimensional Pilot Symbol Aided Channel Estimation by Wiener Filtering. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1845–1848, ,, 1997.

[37] T. Weiss et al. Mutual Interference in OFDM-based Spectrum Pooling Systems. In *IEEE Vehicular Technology Conference (VTC Spring)*, Milan, Italy, May 2004.

[38] T. Weiss et al. Spectrum Pooling - An Innovative Strategy for the Enhancement of Spectrum Efficiency. In *IEEE Communications Magazine, no.3*, volume 42, Milan, Italy, March 2004.

[39] T. Hunziker and D. Dahlhaus. Adaptive OFDM Transmission for Broadband Fixed Wireless Access Systems. In *Proc 6'th International OFDM Workshop (InOWo)*, pages 5–1 – 5–4, Hamburg,Germany, 2001.

[40] T. Hunziker and D. Dahlhaus. Optimal Power Adaptation for OFDM Systems with Ideal Bit Interleaving and Hard Decision Decoding. In *Proc. IEEE International Conference on Communications (ICC)*, pages 3392 – 3397, Alaska,USA, May 2003.

[41] C. S. Park and K. B. Lee. Transmit Power Allocation for BER Performance Improvement in Multicarrier Systems. In *Proc. IEEE Transactions On Communications, no. 10*, volume 52, pages 1658–1663, October 2004.

[42] I. Dagres and A. Polydoros. Mode Adaptation Combined with Power Allocation for Guaranteed QoS Constraints in COFDM Transceiver. In *Proc. WPMC*, pages 726–729, Aalborg,Denmark, September 2005.

[43] K. Kaemarungsi and P. Krishnamurthy. On the Use of Adaptive OFDM to Preserve Energy in Ad Hoc Wireless Networks. In *Proc. 13th MPRG / Virginia Tech Symposium on Wireless Personal Communications*, USA, June 2003.

[44] B.S. Krongold et al. Computationally Efficient Optimal Power Allocation Algorithms for Multicarrier Communications Systems. In *IEEE Transactions on Communications, no.1*, volume 48, pages 23–27, January 2000.

[45] H. Nikookar and R. Prasad. Wave shaping of Multicarrier Signal for Data Transmission Over Wireless Channels. In *Proc. IEEE 6th ICUPC*, pages 173–177, 1997.

[46] P. Tan and N. C. Beaulieu. Reduced ICI in OFDM Systems Using the Better than Raised Cosine Pulse. In *IEEE Communications Letter, no.3*, volume 8, March 2004.

[47] A. Assalini and A. M. Tonello. Improved Nyquist Pulses. In *IEEE Communications Letter, no.2*, volume 8, pages 87–89, 2004.

[48] C. Muschalik. Improving an OFDM Reception Using an Adaptive Nyquist Windowing. In *IEEE Transactions on Consumer Electronics, no.3*, volume 42, pages 259–269, 1996.

[49] K. Witrisal. *OFDM Air Interface Design for Multimedia Communications*. PhD thesis, Delft University of Technology, Delft, The Netherlands, April 2002.

[50] H. Nikookar and R. Prasad. Optimal Waveform Design for Multicarrier Transmission over a Multipath Channel. In *Proc. 47th IEEE Vehicular Technology Conference (VTC)*, pages 1812–1816, Arizona,USA, May 1997.

[51] F. Harris and E. Kjeldsen. A Novel Interpolated Tree Orthogonal Multiplexing (ITOM) Scheme with Compact Time-Frequency Localization: an Introduction and Comparison to Wavelet Filter Banks and Polyphase Filter Banks. In *Proc. Software Defined Radio Technical Conference and Product Exposition*, Orlando,USA, November 2006.

[52] I. Rashad. Optimum Pilot Pattern Design in Wiener Filter Channel Estimation for OFDM Based Cognitive Radio. Master's thesis, Delft University of Technology, July 2007.

[53] D. E. Dudgeon and R. M. Mersereau. *Multidimensional Signal PRocessing*. Prentice Hall, 1983.

[54] M. J. Fernandez-Gettino Garcia et al. Efficient Pilot Patterns for Channel Estimation in OFDM Systems over HF Channels. In *Proc. IEEE Vehicular Technology Conference)*, Amsterdam,The Netherlands, September 1999.

[55] F. Hoeksema et al. A Node Architecture for Disaster Relief Networking. In *1st IEEE International Symposiumon new Frontiers in Dynamic Spectrum Access Networks(DySPAN)*, pages 577–594, Baltimore,USA, Nov 2005.

[56] ETSI. *Digital Cellular Telecommunications System (Phase 2+). Technical Specification ETS 300 910 GSM 05.05, ver. 5.3.1*, 1997.

[57] I. Budiarjo, Q. Zhang, M. Heskamp, and F.W. Hoeksema. D3.31 Radio system: First phase verification and advances in architecture and algorithms. Technical Report, Adaptive Ad-hoc Freeband (AAF), Netherland, 1-August-2006.

[58] V.D. Chakravarthy et al. Cognitive Radio - An Adaptive Waveform with Spectral Sharing Capability. In *Proc. IEEE Wireless Communications and Networking*, pages 724–729, March 2005.

[59] V.D. Chakravarthy et al. TDCS, OFDM, and MC-CDMA : A Brief Tutorial. In *IEEE Communications Magazine, Issue 9*, volume 43, pages S11–S16, Sept 2005.

[60] M. Benedetto and G. Giancola. *Understanding Ultra Wide Band Radio Fundamentals*. Prentice Hall, 2004.

[61] G.M. Dillard et al. Cyclic Code Shift Keying : A Low Probability of Intercept Communication Technique. In *Proc. IEEE Trans. on Aerospace and Electronic Systems, no.3*, volume 39, pages 786–798, July 2003.

[62] A. Jamin and P. Mahoenen. Wavelet Packet Modulation for Wireless Communications. In *Wireless Communications and Mobile Computing magazine*, volume 5, pages 123–137. Wiley Interscience, December 2004.

[63] B. G. Negash and H. Nikookar. Wavelet Based Multicarrier Transmission over Multipath Wireless Channel. In *IEE Electronic Letters, no. 21*, volume 36, pages 1787–1788, Oct 2000.

[64] C.V. Bouwel et al. Wavelet Packet Based Multicarrier Modulation. In *Proc. IEEE Benelux Symposium on Communications and Vehicular Technology (SCVT)*, Leuven,Belgium, Oct 2000.

[65] A.Jensen and A.L. Cour-Harbo. *Ripples in Mathematics: The Discrete Wavelet Transform*. Springer, 2001.

[66] O. Rioul and P. Duhamel. A Remez Exchange Algorithm for Orthonormal Wavelets. In *Proc. IEEE Trans. Circuits Systems - II, no. 8*, volume 41, pages 550–560, Aug 1994.

[67] M.K. Lakshmanan, I. Budiarjo and H. Nikookar. Maximally Frequency Selective Wavelet Packets Based Multi Carrier Modulation Scheme for Cognitive Radio Systems. In *Proc. 50th IEEE Global Communications Conference (GLOBECOM)*, Washington D.C., USA, November 2007.

[68] P.P. Vaidyanathan. *Multirate Systems and Filter Banks*. Prentice-Hall, Englewood Cliffs, New Jersey, 1993.

[69] M. Vetterli and I. Kovacevic. *Wavelets and Subband Coding*. Prentice-Hall, Englewood Cliffs, New Jersey, 1995.

[70] P.W. Wolniansky et al. V-BLAST: An Architecture for Realizing Very High Data Rates over the Rich Scattering Wireless Channel. In *Proc. ISSE*, pages 295–300, Pisa,Italy, October 1998.

[71] E.R. de Lima et al. Performance Enhancements in OFDM-WLAN Systems Using MIMO Access Techniques. In *Proc. International Worskshop on Telecommunications*, Santa Rita do Sapucai,Brazil, August 2004.

[72] J. Mitola III and Jr. Maguire, G.Q. Cognitive radio: making software radios more personal. *Personal Communications, IEEE [see also IEEE Wireless Communications]*, 6(4):13–18, Aug 1999.

[73] D. Cabric, S.M. Mishra, and R.W. Brodersen. Implementation issues in spectrum sensing for cognitive radios. *Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers, 2004*, 1:772–776 Vol.1, Nov. 2004.

[74] http://itpp.sourceforge.net/.

[75] A. Papoulis. *Probability, Random variables, and Stochastic Processes*. McGraw-Hill, third edition, 1991.

[76] P. Welch. The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio and Electroacoustics*, 15(2):70–73, Jun 1967.

[77] M. Heskamp and C. H. Slump. Spectrum sensing and carrier frequency estimation for cognitive radio. *Fifth Karlsruhe Workshop on Software Radios*, pages 83–88, March 2008.

[78] Facilitating opportunities for flexible, efficient, and reliable spectrum use employing cognitive radio technologies. Notice of Proposed Rulemaking 03-322, FCC, December 2003. ET Docket No. 03-108.

[79] W. A. Gardner. *Cyclostationarity in Communications and Signal Processing*. IEEE Press, 1994.

[80] M. Heskamp and C. H. Slump. Extracting cyclostationary features from single carrier signals. *Proceedings of the 29th Symposium on Information Theory in the Benelux*, pages 45–52, May 2008.

[81] A. Tkachenko, A.D. Cabric, and R.W. Brodersen. Cyclostationary feature detector experiments using reconfigurable bee2. *2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, 2007. DySPAN 2007*, pages 216–219, April 2007.

[82] Paul M. Heysters. *Coarse-Grained Reconfigurable Processors; Flexibility meets Efficiency*. PhD thesis, University of Twente, September 2004. ISBN 90-365-2076-2.

[83] Paul M. Heysters and Gerard J.M. Smit. Mapping of dsp algorithms on the montium architecture. In *Proceedings of Reconfigurable Achitecture Workshop*, 2003.

[84] Gerard K. Rauwerda, Gerard J.M. Smit, and Werner Brugger. Implementing an adaptive viterbi algorithm in coarse-grained reconfigurable hardware. In *Proceedings of the International Conference on Engineering of Reconfigurable Systems and Algorithms*, 2005.

[85] Arnaud Rivaton, Jérôme Quevremont, Qiwei Zhang, Pascal T. Wolkotte, and Gerard J.M. Smit. Implementing non power-of-two ffts on coarse-grain reconfigurable architectures. In *Proceedings of the International Symposium on System-on-Chip (SoC 2005)*, 2005.

[86] Q. Zhang, A.B.J. Kokkeler, and G.J.M. Smit. A reconfigurable radio architecture for cognitive radio in emergency networks. In *In Proceeding of European Conference on Wireless Technology*, Sep. 2006.

[87] Q. Zhang, A.B.J. Kokkeler, and G.J.M. Smit. An efficient fft for ofdm based cognitive radio on a reconfigurable architecture. In *In Proceeding of IEEE ICC Cognet Workshop*, Jun 2007.

[88] Henrik V. Sorensen and Sidney Burrus. Efficient computation of the dft with only a subset of input or output points. *IEEE Trans. on Signal Processing*, Mar, 1993.

[89] G. Cherubini, E. Eleftheriou, S. Olcer, and J.M. Cioffi. Filter bank modulation techniques for very high speed digital subscriber lines. *IEEE Commun. Mag.*, May, 2000.

[90] R. W. Brodersen D. Cabric, S. M. Mishra. Implementation issues in spectrum sensing for cognitive radios. In *Asilomar Conference on Signals, Systems, and Computers*, 2004.

[91] Pieter van der Wolf et al. Design and programming of embedded multiprocessors: An interface-centric approach. In *Proceedings of the CODES+ISSS*, 2004.

[92] Jeffrey Kang et al. An interface for the design and implementation of dynamic applications on multi-processor architectures. In *Proceedings of ESTImedia*, 2005.

[93] F.W. Hoeksema, M. Heskamp, R. Schiphorst, and C.H. Slump. A node architecture for disaster relief networking. 1^{st} *IEEE International Symposium on new Frontiers in Dynamic Spectrum Access Networks(DySPAN'05)*, pages 577–584, November 2005.

[94] Lodewijk Smit et al. Run-time mapping of applications to a heterogeneous reconfigurable tiled system on chip architecture. *Proceedings of the International Conference on Field Programmable Technology*, Dec. 2004.

[95] A.B.J. Kokkeler, G.J.M. Smit, T. Krol, and J. Kuper. Cyclostationary feature detection on a tiled-soc. In *Proceedings of DATE 2007*, Apr. 2007.