RESTRICTED

FP7-ICT Future Networks
SPECIFIC TARGETTED RESEARCH PROJECT
Project Deliverable

PHYDYAS Doc. Number	PHYDYAS_015
Project Number	ICT - 211887
Project Acronym+Title	PHYDYAS – PHYsical layer for DYnamic AccesS and cognitive radio
Deliverable Nature	Report
Deliverable Number	D9.2
Contractual Delivery Date	July 1st, 2009
Actual Delivery Date	July 29th, 2009
Title of Deliverable	WiMAX simulation results – Lab setup and measurements
Contributing Workpackage	WP9
Project starting date; Duration	01/01/2008; 30 months
Dissemination Level	RE
Author(s)	Frank Schaich (ALUD); Jurgen Vandermot (AGI); Vidar Ringset, Helge Rustad (SINTEF); Montse Najar, Francisco Rubio, Marius Caus (CTTC); Zhao Zhipeng (COMSIS); Mathieu Huchard (LETI)

Abstract:
The simulator that is developed in work package 9, based on the existing WiMAX standard (Worldwide interoperability for Microwave Access) and enhanced with algorithms developed by other work packages to include the FBMC principle has already extensively been used. In section 2 of this deliverable, the results of those simulation tests are shown, analyzed and discussed.
Section 3 is dedicated to the demonstrator setup. According to the milestone at month 18 included in the DoW, a final definition of the advanced laboratory setups is provided. In fact, it was too difficult to elaborate on the advanced setups at the start of the project and, as specified in the DoW, the detailed motivation for the decision will be included in the next deliverable D9.3

Contents

Notations

A, B input streams of the CTC encoder
Y_1, Y_2
W_1, W_2 output streams of the CTC encoder

M overall number of subcarriers, FFT size
K overlapping factor in prototype filter design

m time index at SFB output/AFB input
N_{ovs} oversampling factor

i user index in multiuser cases
U number of users

$x_{k,n}$ observed ideal (without channel) complex sample of subcarrier k in symbol n
$y_{k,n}$ observed channel-distorted complex sample of subcarrier k in symbol n
$s(m)$ transmitted sequence at SFB output in the downlink
$s_i(m)$ transmitted sequence at SFB output of user i in the uplink

P number of paths in the multipath channel model
$P_{i,p}$ path power of path # p in the multipath channel model of user i
P_i transmit power of user i
S_i path-norm factor

$c_{i,p}$ complex gain of the pth path of the channel of user i
$\tau_{i,p}$ delay of the pth path of the channel of user i
m_i path independent time offset due to a failed synchronization
$m_{i,p}$ path dependent time offset due to different path lengths
$\Delta\rho_i(m)$ user specific phase and frequency offset
σ_n^2 channel noise variance

$r_i(m)$ received complex sequence at AFB input at the mobile of user i (downlink)
$r(m)$ received complex sequence at AFB input at the basestation (uplink)

$H_{k,n}$ channel response for subcarrier k and symbol n (assuming flat-fading and time variant case)
$\tilde{H}_{k,n}$ estimate of the channel response for subcarrier k and symbol n
$n_{k,n}$ noise sample disturbing subcarrier k in symbol n
$\tilde{n}_{k,n}$ noise sample disturbing the estimate of the channel response for subcarrier k in symbol n

1 Introduction

The objective of PHYDYAS is to propose a physical layer for future radio systems that is more efficient than present OFDM based solutions and better suited to the new concepts of DASM (Dynamic Access Spectrum Management) and cognitive radio.

With OFDM, part of the signal power leaks into neighboring subcarriers, leading to a higher sensitivity to frequency offsets. To cope with this problem, capacity reducing band guards are needed. The new physical layer proposed in the PHYDYAS Project, is less sensitive to carrier offsets and can use the spectrum more efficiently. Additionally, spectrum sensing in cognitive radio is more reliable due to lesser interference.
To allow for synchronization and to manage the frequency selective behavior of multi path channels, OFDM based systems introduce a so called cyclic prefix to the signal, but with the new physical layer investigated in PHYDYAS there is no need for this cyclic prefix anymore, again increasing the capacity.

To reach these goals a filter bank is introduced into the transmission chain. This filter bank is the heart of the project. A basic solution was provided at the beginning [1], which was and will be further developed within the project. Other aspects of the transmission chain, such as synchronization and initialization, channel estimation and tracking, equalization, demodulation and MIMO (multiple input multiple output) processing need to be addressed. This is done within other work packages and thus this deliverable will not go into detail with respect to these algorithms.

To gain acceptance within the community the physical layer proposed in PHYDYAS needs to be compared with the state of the art with respect to performance and complexity. To get performance measurements, a link level simulator gets developed reproducing the complete wireless transmission chain including the algorithms produced by other work packages that are needed to realize the new physical layer. This simulator is based on IEEE 802.16e [7], a worldwide accepted standard defining the WiMAX air interface for broadband wireless access systems.

The first part of the deliverable presents the simulation results of the new FBMC-based physical layer and compares them with the performance of a standard WiMAX implementation. At this point it shall be emphasized that the principles investigated in PHYDYAS are not limited to WiMAX. With minor modifications they easily can be translated to other OFDM based transmission systems such as LTE (Long Term Evolution).

The second part of the deliverable gives a detailed description of the demonstrator setup and the measurements that could be performed with it. The goal is to setup a demonstrator which resembles the implementation of the simulator. This approach enables cross-checking between measurement results from simulator and demonstrator.
The measurement results should also include the performances of both standard WiMAX and FBMC-based WiMAX implementations for comparison reasons, but these results will be presented in the next deliverables.

2 The Simulator

2.1 Additions to the Simulator

This chapter summarizes the additions to the simulator since the last deliverable of this workpackage [1].

2.1.1 Downlink Frame Preamble

Additional degrees of freedom with respect to the characteristics of the downlink frame preamble are implemented. Up to D9.1 the preamble occupied a single OFDM symbol and followed the guidelines of [7]. New parameters are available in PAR1_CELL_... to define the characteristics of the preamble:

Parameter name	Possible values	Remarks
C1_CELL.Frm. PreambleBoosting	any real positive value	Boosting factor of the Preamble symbols (WiMAX: $2\sqrt{2}$)
C1_CELL.Frm. PreambleType	'WiMAX', 'selfDef'	**WiMAX**: according to [7] **selfDef**: user defined
C1_CELL.Frm. PreambleOQAM	'true', 'false'	'true' → Preamble gets OQAM processed 'false' → Preamble gets not OQAM processed (zero symbols are inserted)
C1_CELL.Frm. NbPreambleSymbs	any positive integer smaller than the number of OFDM symbols of the frame	Number of OFDM symbols used for the preamble
C1_CELL.Frm.Preamble	Matrix containing the preamble	The preamble to be used (only if C1_CELL.Frm.PreambleType='selfDef')

Table 1: Parameters for Preamble Design [1]

Example:
Let's say we want to use a preamble needed for the preamble based channel estimation using the interference approximation method (IAM) proposed in D3.1 of work package 3 [3]:

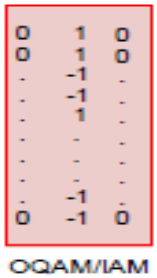

Figure 1: Preamble for channel estimation using the IAM scheme.

Then we have to choose:

```
C1_CELL.Frm.PreambleType = 'selfDef';
C1_CELL.Frm.PreambleOQAM = false;
C1_CELL.Frm.Preamble = zeros(3, 1024);
C1_CELL.Frm.Preamble(2,1:4:1024) = ones(1,1024/4);
C1_CELL.Frm.Preamble(2,2:4:1024) = ones(1,1024/4);
C1_CELL.Frm.Preamble(2,3:4:1024) = -ones(1,1024/4);
C1_CELL.Frm.Preamble(2,4:4:1024) = -ones(1,1024/4);
```

2.1.2 Estimation / Synchronization / Equalization Chain

The main addition to the simulator is the iterative estimation, synchronization and equalization of carrier frequency offset (CFO), fractional time delay (FTD) and channel. Figure 2 depicts the new receiver structure:

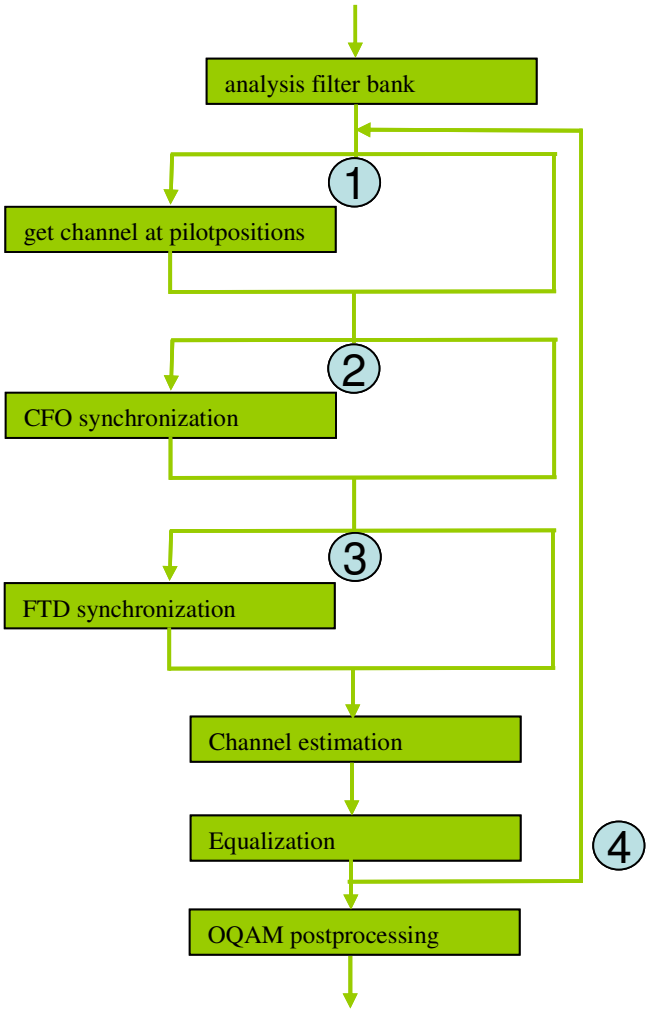

Figure 2: Modified receiver structure.

Once the received signal is transformed by the analysis filter bank, the channel estimates at the pilot positions are collected. With their help the CFO and the FTD are estimated. With the help of 2-dimensional linear interpolation, the channel coefficients at the subcarrier positions of the complete

frame (in downlink) or the burst respectively (in uplink) are approximated. In downlink all pilots of the complete frame are placed and thus usable for channel estimation. In uplink naturally just the pilots placed within the actual user burst can be used, as the others were transmitted by other users. Hence, channel interpolation in downlink is more reliable.
As already mentioned the chain, depicted above, can be used iteratively. Once the channel is equalized the complete procedure can be repeated (4) to improve the estimates.
One option is to use perfect channel knowledge for channel estimation, CFO estimation and/or FTD estimation. In this case additional iterations are useless. Therefore, the bypasses in (1), (2) and (3) are present.
The algorithms used for CFO and FTD estimation are based on the work done in work package 2. For a more detailed description the interested reader is referred to [4] and [5]. The available parameter space is:

Parameter name	Possible values	Remarks
C1_MS_DIGI.FBMC. CFO_correction (DL) C1_BS_DIGI.FBMC. CFO_correction (UL)	'PCI', 'None', 'RegressionEstimation', 'WeightedEstimation'	algorithm to be used for CFO estimation: **PCI**: use perfect channel knowledge **None**: no CFO correction **RegressionEstimation**: linear regression (in time direction) using the estimates at the pilot positions as supporting points **WeightedEstimation**: averaging over the CFO estimates obtained with the help of adjacent pilots
C1_MS_DIGI.FBMC. cfo_recompensation (DL) C1_BS_DIGI.FBMC. cfo_recompensation (UL)	'true', 'false'	Boolean to choose, if the equalizer shall apply amplitude recompensation (3 taps needed)
C1_MS_DIFI.FBMC. FTD_estimation (DL) C1_BS_DIFI.FBMC. FTD_estimation (UL)	'PCI', 'Estimation', 'EstIIC'	algorithm to be used for FTD estimation: **PCI**: use perfect channel knowledge **Estimation**: linear regression (in frequency direction) using the estimates at the pilot positions as supporting points **EstIIC**: iterative estimation based on the interference cancelation approach

Table 2: Parameters for Preamble Design [2]

Parameter name	Possible values	Remarks
C1_MS_DIGI.FBMC. estiic_ftd_iterations (DL) C1_BS_DIGI.FBMC. estiic_ftd_iterations (UL)	positive integer	number of iterations performed, if the iterative interference cancelation approach is applied
C1_MS_DIGI.FBMC. NbSynchIterat (DL) C1_MS_DIGI.FBMC. NbSynchIterat (UL)	positive integer	number of iterations performed covering the complete synchronization and equalization chain

Table 3: Parameters for Preamble Design [3]

The performance of the frequency and time synchronization chain can be measured with the help of the error rates calculated by the simulator. However, to get a closer look to the performance of a single synchronization element two additional output parameters are added. After the simulation has ended they are stored within the simulation directory in '_workspace.mat':

MEAS.est_ftd.BS{iB}.MS{iM}.Zone{iZ}.ant{iA}.NbIterat{iI}.frameSNR contains the matrix of estimated FTD values at antenna number iA of mobile number iM located in zone iZ using the basestation configuration iB after iteration number iI per SNR and per frame.

MEAS.est_cfo.BS{iB}.MS{iM}.Zone{iZ}.ant{iA}.NbIterat{iI}.frameSNR contains the matrix of estimated CFO values at antenna number iA of mobile number iM located in zone iZ using the basestation configuration iB after iteration number iI per SNR and per frame.

2.1.3 Frequency and Time Gaps

Even perfectly synchronized users are causing multi-user interference (in uplink), as different users pass through different channels. Gaps both in time and frequency between the user bursts are to be placed to avoid this. The actual version of the simulator has the capability to introduce such gaps. Following is a short description how to parameterize the simulator.
To introduce a time gap between two users someone has to define two zones, each carrying one of the users. The size and position of the zones within the frame are defined with two parameters (C1_CELL.Zone{iZ}.OfsOsym and C1_CELL.Zone{iZ}.OsymLen). Let's say both zones shall include nine symbols each and a gap of a single symbol shall be present between them. This can be achieved by the following settings:

C1_CELL.Zone{1}.OfsOsym = 0 … Zone 1 starts at the beginning of the UL subframe
C1_CELL.Zone{1}.OsymLen = 9 … and covers 9 symbols

C1_CELL.Zone{2}.OfsOsym = 10 … Zone 2 starts 10 symbols after the beginning of the UL subframe (→ one symbol gap between the zons)
C1_CELL.Zone{2}.OsymLen = 9 … and covers 9 symbols, either.

Such an indirect definition of the time gap is necessary, as the allocation of the user bursts in uplink is 'time first' and not rectangular as in downlink.

The introduction of a frequency gap between users is more straightforward. A new parameter is introduced in PAR3_MS_DATA_...:

Parameter name	Possible values	Remarks
C1_MS_DATA. OfsSubcrr	integer	Frequency offset in number of subcarriers relative to the position of the burst within the zone. **>0** → burst shifted to higher freq. **<0** → burst shifted to lower freq.

Table 4: Parameter to Introduce a Frequency Gap

2.1.4 Planned Additions

Depending on the advances of the respective work packages the planned additions within the next semester are:

- MIMO
  - transmit/receive diversity
  - spatial multiplexing
  - channel estimation for multi antennas
  - synchronization for multi antennas
- Preamble based synchronization
- RF Impairments (see 2.4 RF Impairments Baseband Model)

2.2 Simulation Setups

Next is the description of the simulation setups used within the simulation campaigns. It was agreed that they also will be used for measurements done with the demonstrator.

Both downlink and uplink will be part of the simulations/measurements. For downlink the chosen mode is PUSC, for uplink AMC will be applied. To achieve reasonable results typical burst sizes are to be chosen. The following figures depict the size and placement of the user data. Not shown, but naturally present during simulation and measurement are interfering users placed adjacent to the user of interest.

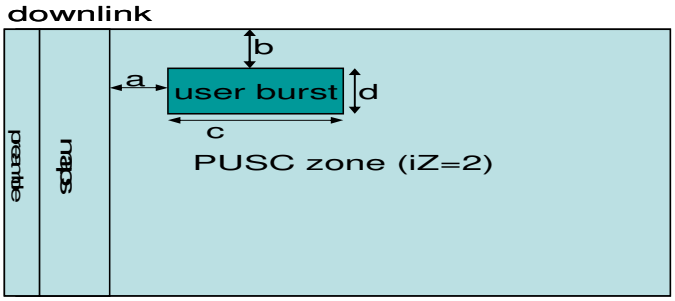

Figure 3: Burst placement downlink

The respective parameter settings are:

Parameter name	chosen value
a = C1_MS_DATA.OfsTslots	1
b = C1_MS_DATA.OfsSchn	3
c = C1_MS_DATA.NbTslots	2
d = C1_MS_DATA.NbSchn	6
C1_MS_DATA.NbSlots	12
C1_MS_DATA.PacketSize	64 (Bytes)

Table 5: Burst Placement Downlink

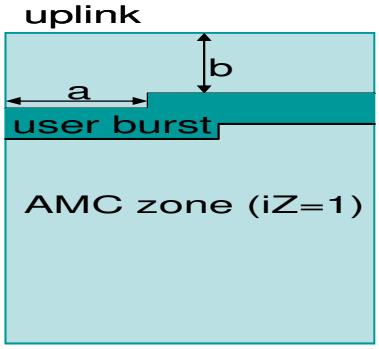

Figure 4: Burst placement uplink

The respective parameter settings are:

Parameter name	chosen value
a = C1_MS_DATA.OfsTslots	2
b = C1_MS_DATA.OfsSchn	2
C1_MS_DATA.NbSlots	11
C1_MS_DATA.PacketSize	64 (Bytes)

Table 6: Burst Placement Uplink

The common simulation parameters are, if not specified differently:

Parameter name	chosen value
number of subcarriers	1024
carrier frequency	2.5 GHz
bandwidth	10 MHz
sampling rate	11.2 MHz
subcarrier spacing	10.94 kHz
filter bank	CNAM solution
overlapping factor	4
pilot boost	2.5 dB
pilot method	auxiliary pilots
excess ratio	1
raised cosine ratio	0.1
coding scheme	CTC
code rate	1/2
number of turbo iterations	4

Table 7: Common Parameters

2.3 Simulation Results

To asses the performance of a system under investigation three different measures will be used:

- Spectral efficiency
- Required SNR @ PER = 0.01
- 3 dB tolerance for desynchronization (CFO/FTD)

The spectral efficiency is a measure with which the system under investigation can be classified in terms of achievable throughput under the given circumstances.
The required SNR @ PER = 0.01 is the signal-to-noise ratio that has to be guaranteed at the receiver to ensure a packet error rate of 1% (mean values). 1% is typically the target error rate if only the physical layer is considered. Higher layer concepts (e.g. packet retransmission in the IP layer) guarantee a reasonable user experience.
The 3 dB tolerance reflects the range a specific parameter may span in which the required SNR @ PER = 0.01 does not increase more than 3 dB. This way the requirements to that parameter can be seen, if the link budget includes a 3 dB tolerance dedicated to this parameter (examples for such a parameter are CFO and FTD).

2.3.1 Spectral Efficiency

Within this chapter the spectral efficiency of a WiMAX-like system applying FBMC is compared to a true WiMAX system. Both uplink and downlink are looked at. For downlink PUSC permutation is used, in uplink AMC is applied. One symbol is used as Preamble. The WiMAX system thus has 46 multicarrier symbols usable for data transmission (including the maps). Due to the absence of a cyclic prefix, if FBMC is used, 52 multicarrier signals are usable there (preamble already excluded). In the following the distribution of the symbols is as follows:

downlink subframe	WiMAX: 31 FBMC: 35
uplink subframe	WiMAX: 15 FBMC: 17

Table 8: Subframe Dimensions

The number of subcarriers per multicarrier symbol available for data transmission (including pilots) within the WiMAX system is 840/864 (DL PUSC/UL AMC). If FBMC is applied, 912 subcarriers can be used for data transmission without increasing the out-of-band interference [4].

2.3.1.1 Definition of Spectral Efficiency

The spectral efficiency is calculated as follows:

$$c = c_{\max} r R_T R_G R_p$$

with

$$c_{\max} = R_{CTC} ld(B)$$

$$r = 1 - PER$$

c_{max} is the maximal achievable spectral efficiency depending on the modulation order B and the code rate R_{CTC}. We have based the normalized throughput r on the packet error rate (PER), i.e. a Bit is treated as erroneous if any of the Bits within its packet (64 Bytes) is received erroneous. R_T reflects the efficiency in time direction, i.e. it is the ratio of time duration used for data transmission and the time duration of the complete subframe:

$$R_T = \frac{N_S T_S}{T_{SF}}$$

N_S is the number of multicarrier symbols within the subframe used for data transmission (This value may be no integer to include the need of time gaps between users.), T_s the useful symbol length for data transmission. T_{SF} is the subframe length.
$R_G R_P$ reflects the efficiency in frequency direction, i.e. the overhead due to the guard bands and the pilot symbols is accounted for here.

$$R_G = \frac{N_{SC} - N_G - 1}{N_{SC}}$$
$$R_P = \frac{N_{SC} - N_G - 1 - N_P}{N_{SC} - N_G - 1}$$

N_{SC} is the number of subcarriers per multicarrier symbol, N_G the number of subcarriers used as guard and N_P the number of pilots per multi carrier symbol.

2.3.1.2 Spectral Efficiency in Downlink (PUSC)

The following figures compare the spectral efficiency of the WiMAX reference and the FBMC system. Up to now perfect channel knowledge and perfect synchronization are assumed. Thus the results are upper bounds. Further investigations will take real channel estimation and synchronization impairments into account. The increased efficiency due to the usage of the filter bank is apparent.

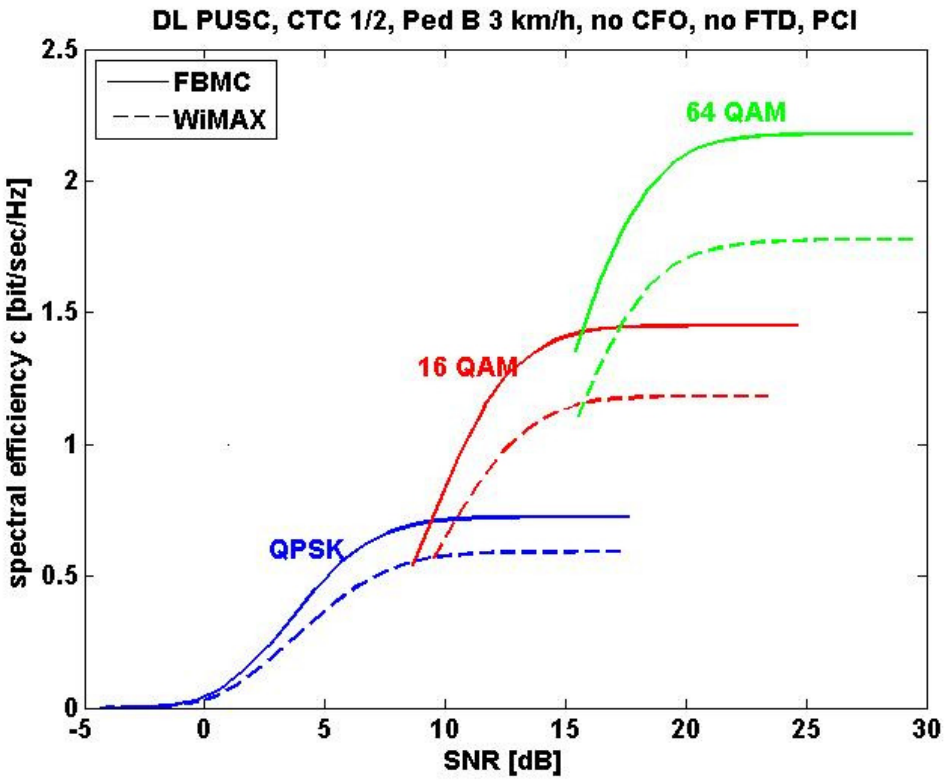

Figure 5: Spectral efficiency, perfect synchronization (Ped B 3 km/h, DL PUSC)

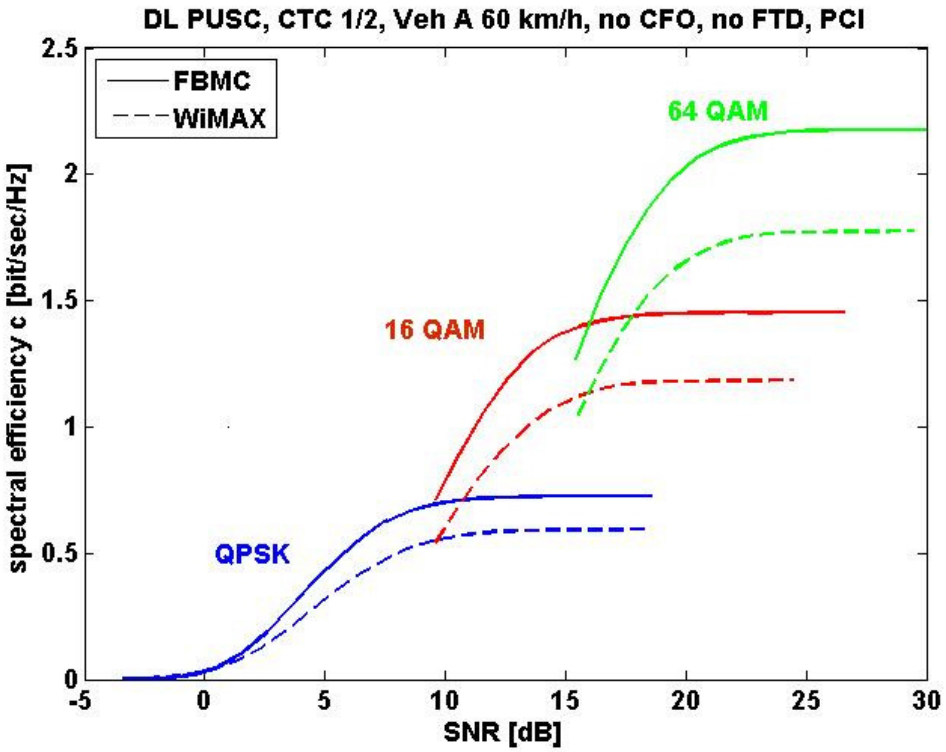

Figure 6: Spectral efficiency, perfect synchronization (Veh A 60 km/h, DL PUSC)

2.3.1.3 Spectral Efficiency in Uplink (AMC)

The uplink signals of different users pass through different channels and thus adjacent subcarriers (adjacent with respect to their placement within the frame) are not orthogonal any more at the base station, severe interference occurs. Thus, in uplink spectral efficiency will be slightly reduced, as guards are necessary to separate the users. **Error! Reference source not found.** depicts the single user case, thus no subcarriers are left empty as guards. Up to now perfect channel knowledge is assumed.

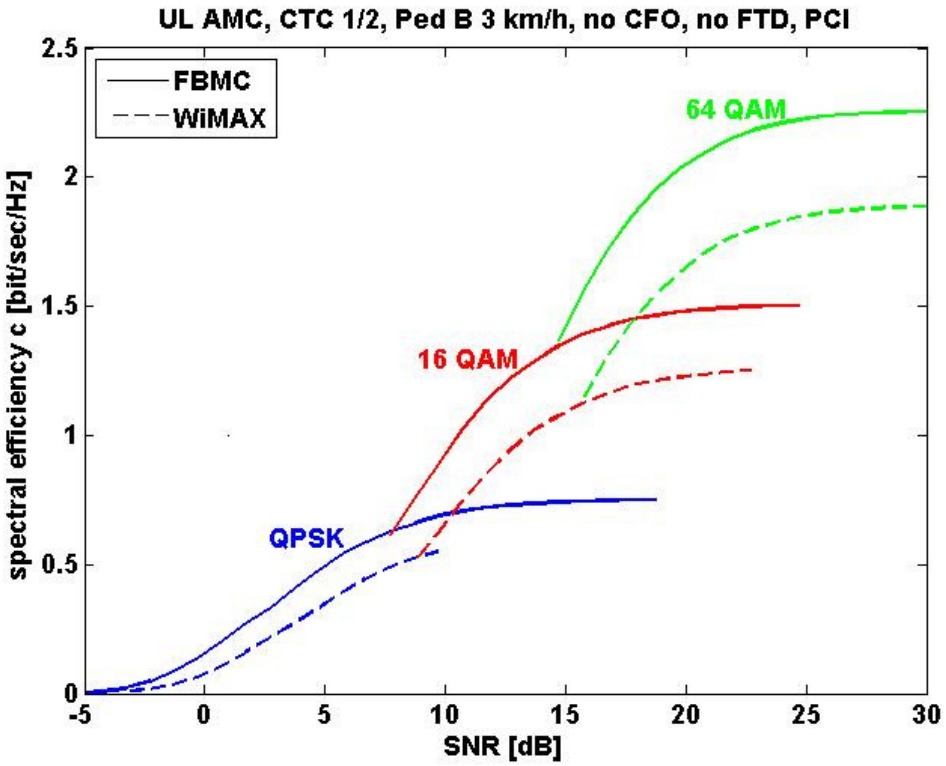

Figure 7: Spectral efficiency, single user case, perfect synchronization (Ped B 3 km/h, UL AMC)

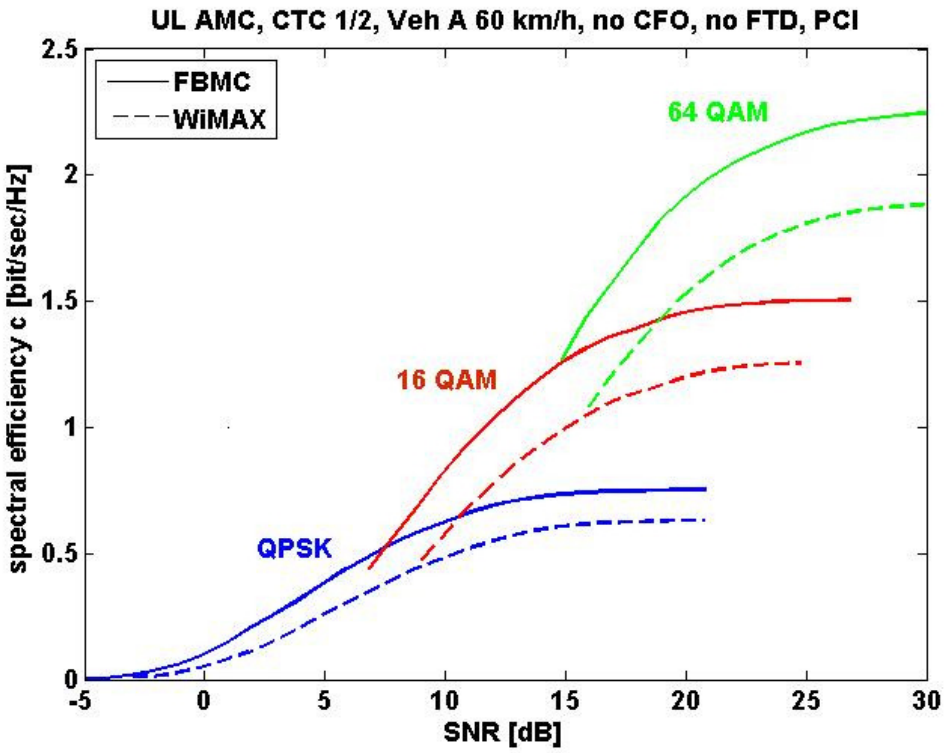

Figure 8: Spectral efficiency, single user case, perfect synchronization (Veh A 60 km/h, UL AMC)

These results mark the upper bound. The more users are active the lower efficiency will get due to the guards. The next deliverable of work package 9 will include further results taking the multi user case into account. Additionally the impact of channel estimation and CFO/FTD will be taken into account.

2.3.2 Synchronization

Synchronization is a major topic in PHYDYAS, visible by the fact that a complete work package (2) is dedicated to this topic. Within this work package diverse algorithms are described and their performance assessed and compared. To be able to concentrate on the synchronization algorithms the system models used there are partly simplified. E.g. no error correction scheme is applied and the PUSC permutation scheme is not fully implemented. One of the tasks of work package 9 is to gather the algorithms and include them into the WiMAX frame work. After this inclusion several simulation campaigns were performed to assess the performance. The following sections contain the results. At this point it shall be emphasized, that the synchronization issue is not yet closed. It will be further tackled in future campaigns. The results will be part of later documents.

UL AMC is applied. It is assumed, that large enough time and frequency gaps are used. Thus, no multi user interference is present. Performance decreasing factors caused by the offsets are self-interference and phase distortion.

For different modulation orders and diverse channels the required SNR to achieve a PER of 1% is shown within the following figures. The blue horizontal line within the figures displays a 3 dB offset relative to the perfectly synchronized system. Both, compensation with perfect channel knowledge (PCI) and without compensation at all is shown (upper and lower bound of the achievable performance). Additionally compensation with estimation based on scattered pilots is drawn (CFO and FTD estimation based on linear regression, channel estimation based on the pilots and linear interpolation).
At the end of the subsections the 3 dB tolerances are depicted. CFO/FTD combinations lying on the lines within these diagrams are resulting exactly to a 3 dB deviation in PER performance at PER=1%. Thus, these are the limits of acceptable desynchronization, if the link budget spends 3 dB to it.

2.3.2.1 QPSK

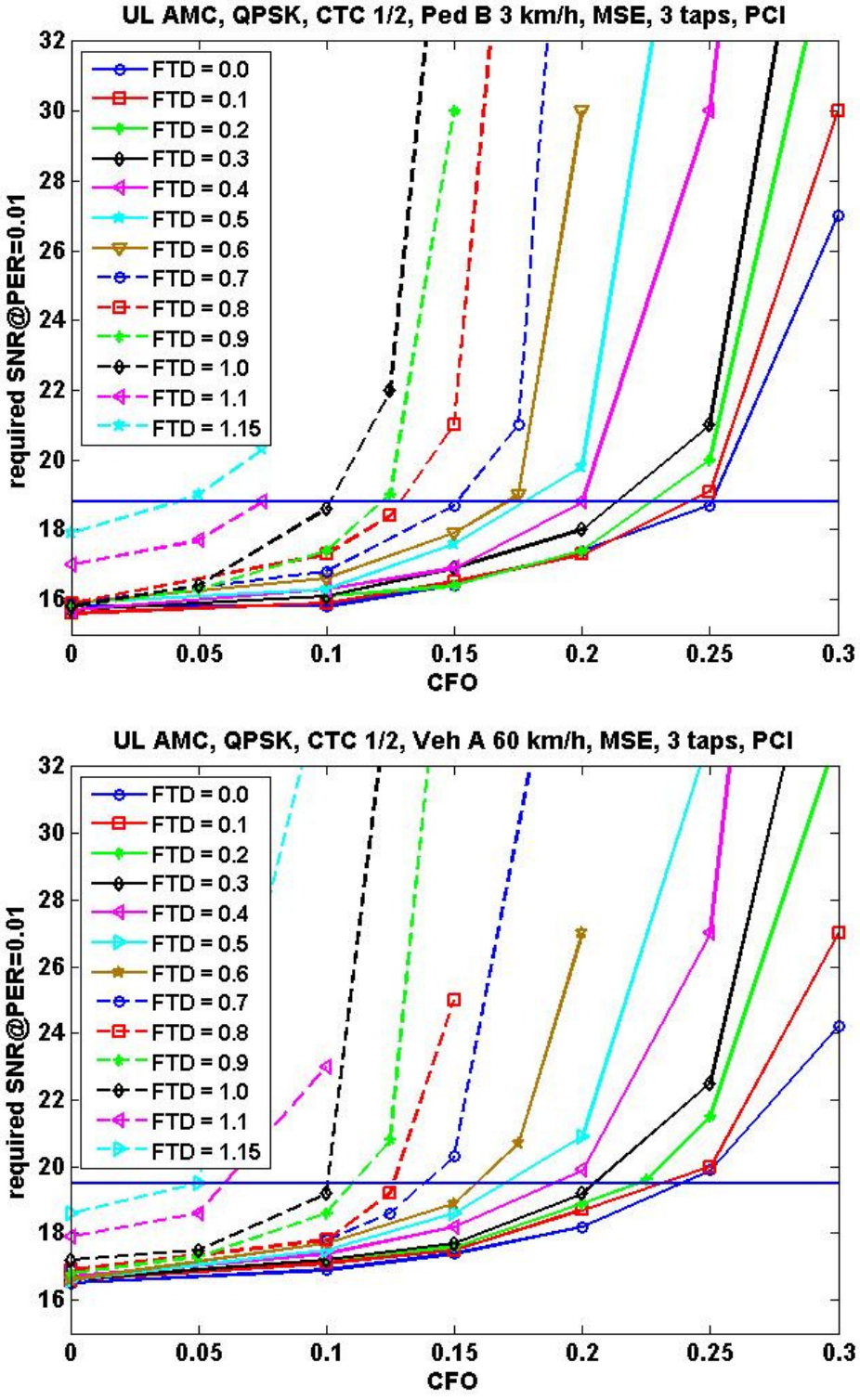

Figure 9: Required SNR to achieve PER = 1% with perfect channel information (upper bound)

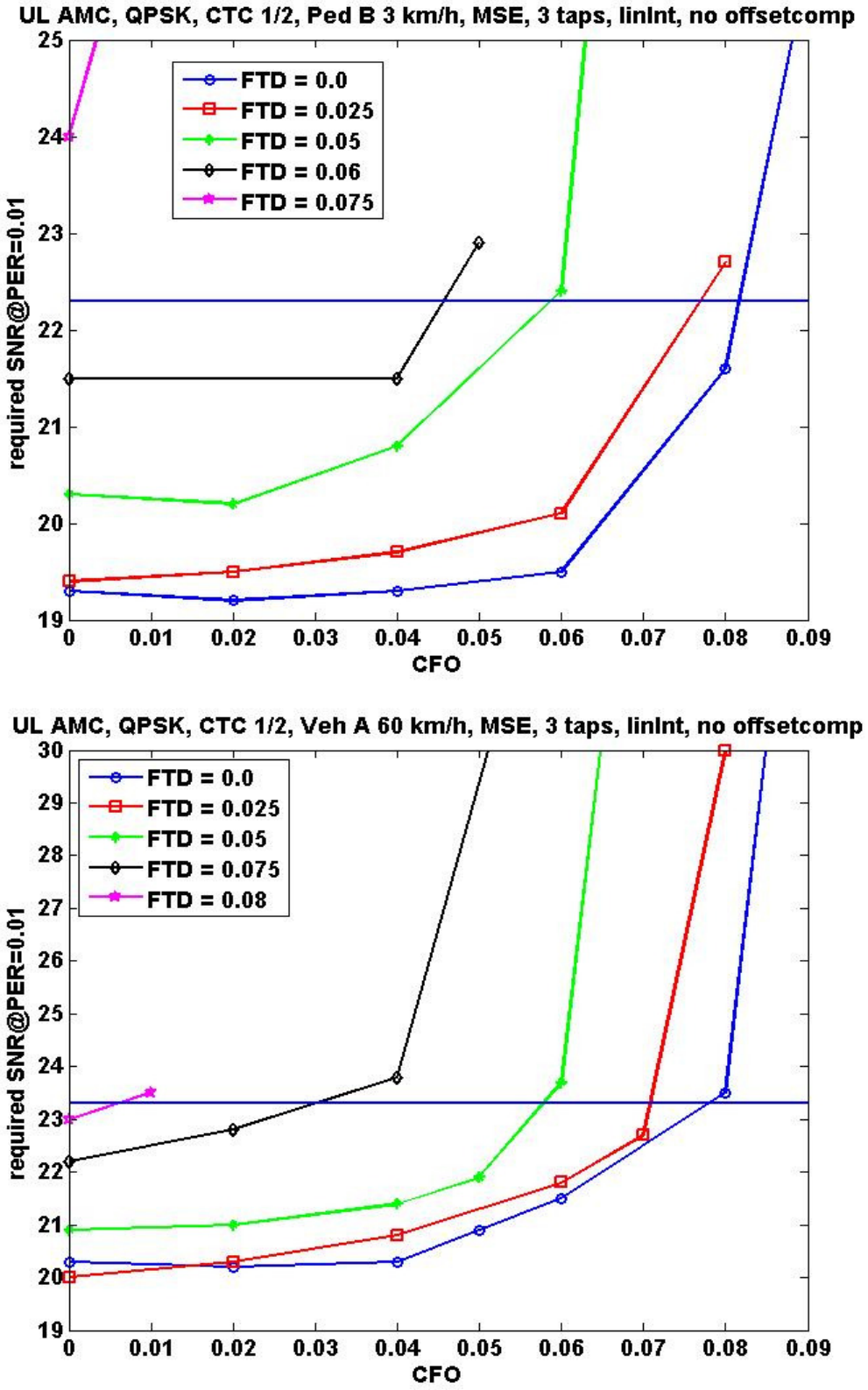

Figure 10: Required SNR to achieve PER = 1% without offset compensation (lower bound)

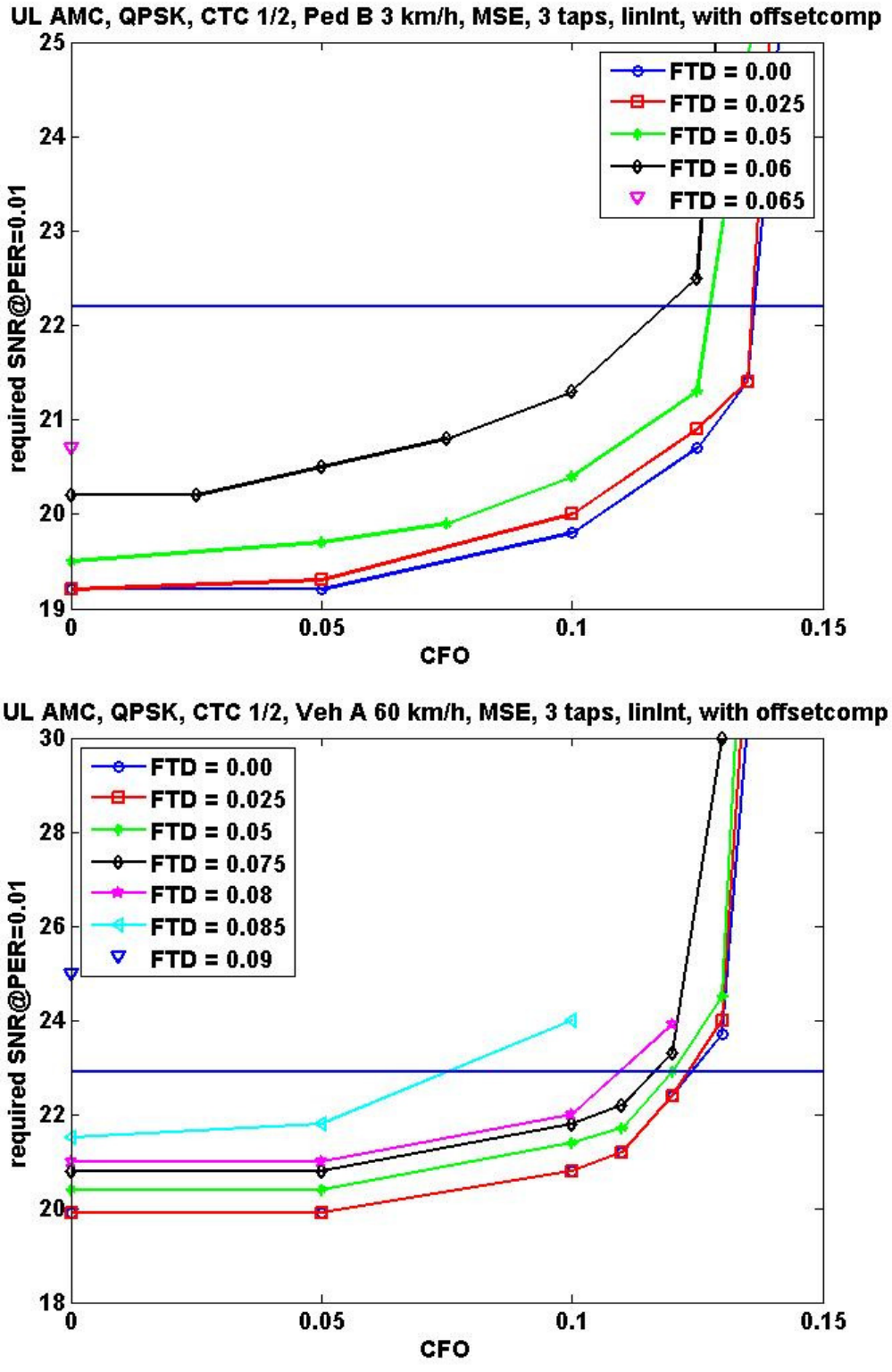

Figure 11: Required SNR to achieve PER = 1% with offset compensation based on scattered pilots

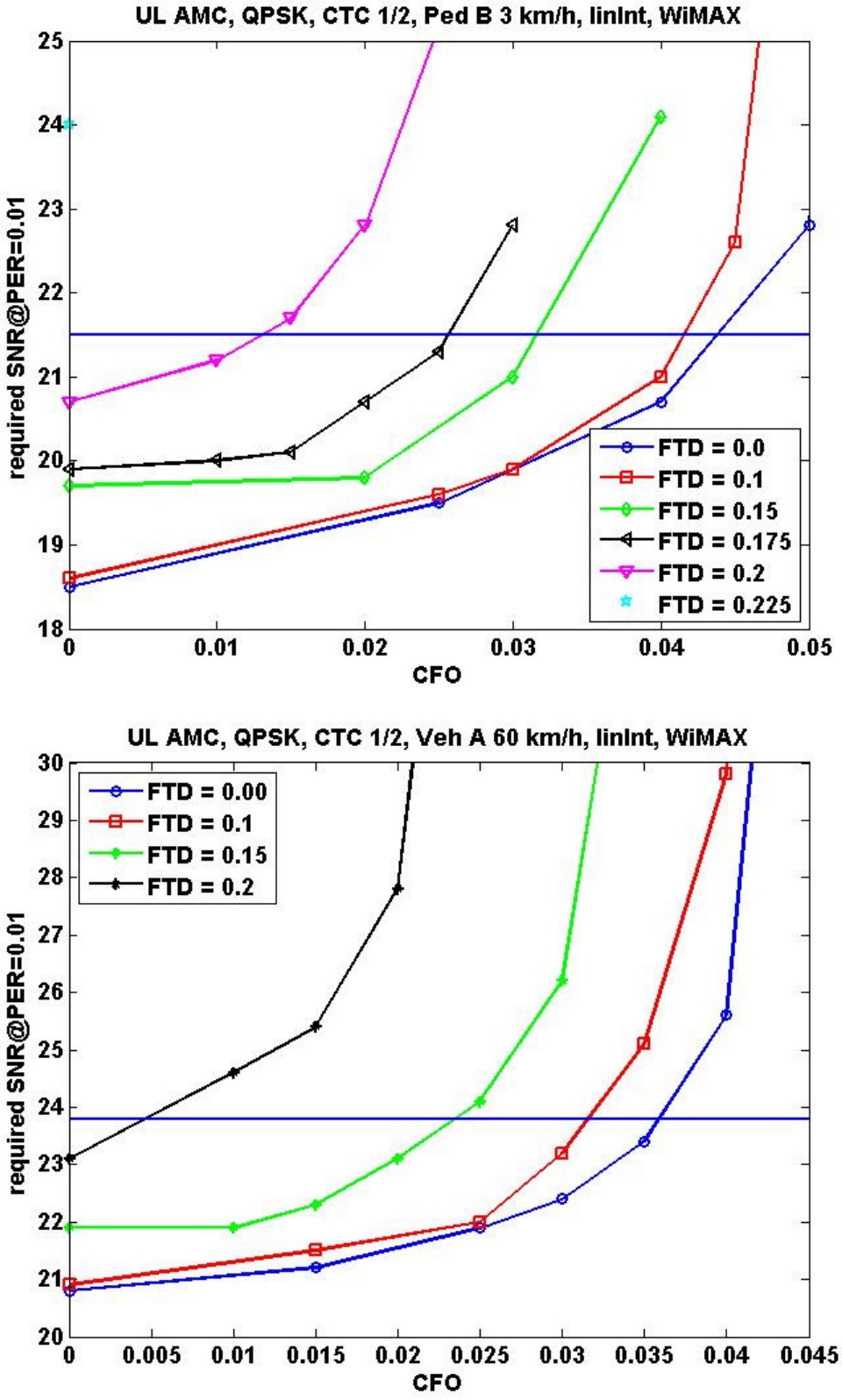

Figure 12: Required SNR to achieve PER = 1% with the reference WiMAX system

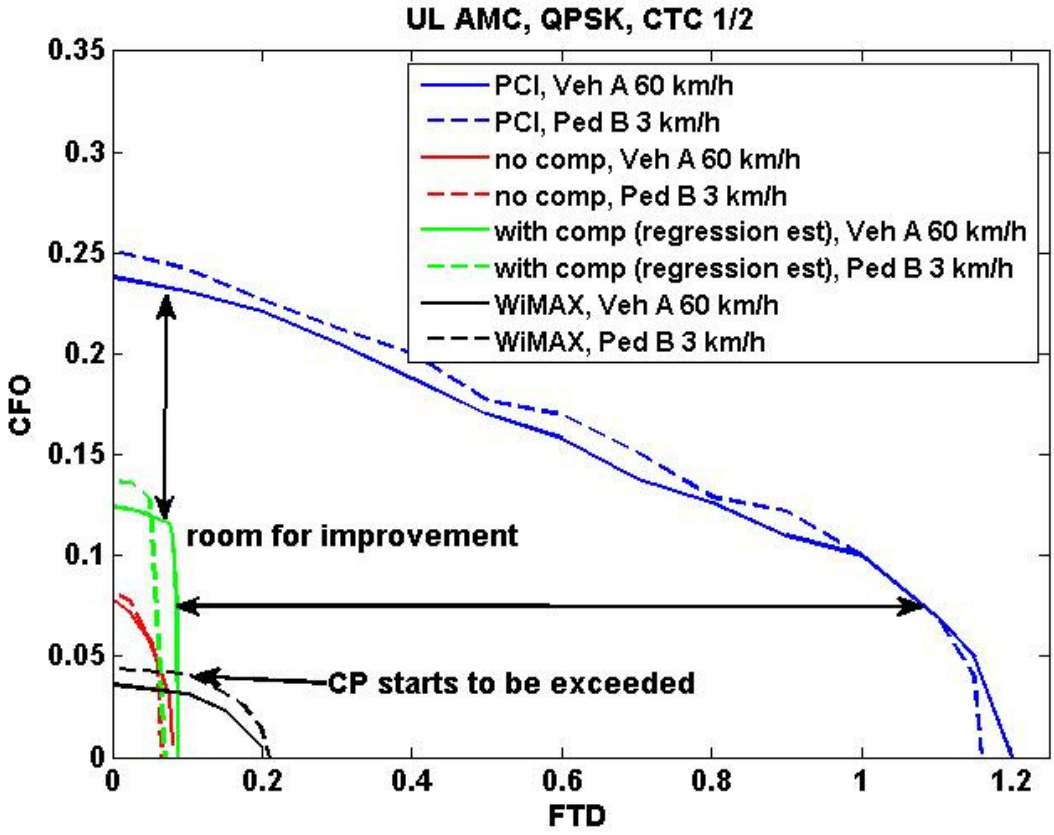

Figure 13: CFO/FTD tolerances with 3 dB link budget dedicated to desynchronization

Conclusions:

- A system experiencing a channel similar to Veh A 60 km/h is more sensitive to CFO and less sensitive to FTD than a system working under conditions similar to Ped B 3 km/h. This is due to the higher Doppler and the lower delay spread, respectively (more on this within the section dedicated to 64 QAM).
- The compensation range of the FTD is limited by the distance of the pilots in frequency direction, as the estimation gets ambiguous due to the ambiguity of the phase. Thus, the small gain compared to the uncompensated system.
- Degradation due to FTD gets significant once the CP starts to be exceeded (WiMAX system).
- As expected the system applying the filter bank is less sensitive to CFO, due to the reduced out-of-band interference. On the other side the sensitivity to FTD is higher, as with the given pilot spacing the FBMC system cannot reach the range guaranteed by the CP. This may change with improved estimation schemes.

2.3.2.2 16QAM

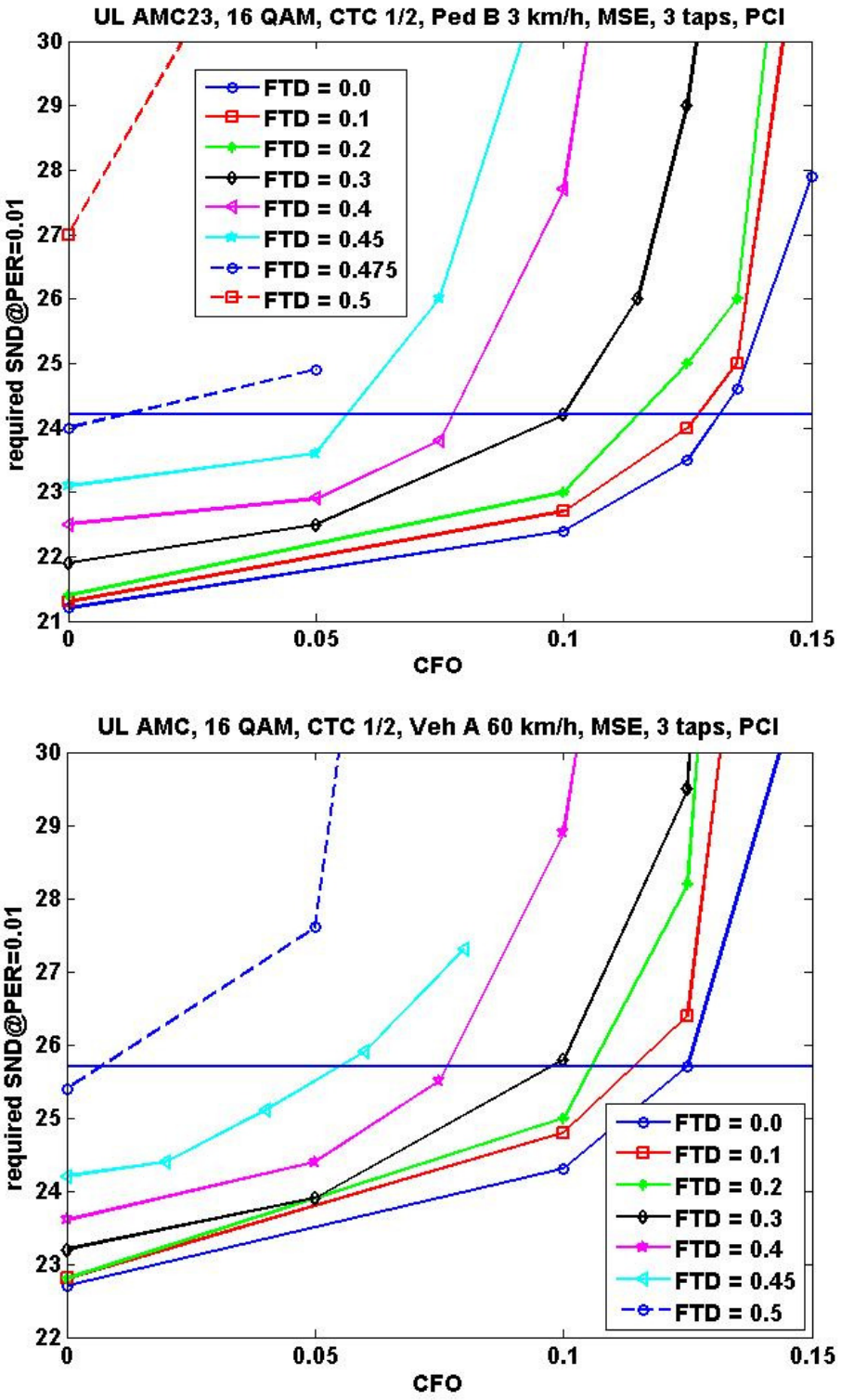

Figure 14: Required SNR to achieve PER = 1% with perfect channel information (upper bound)

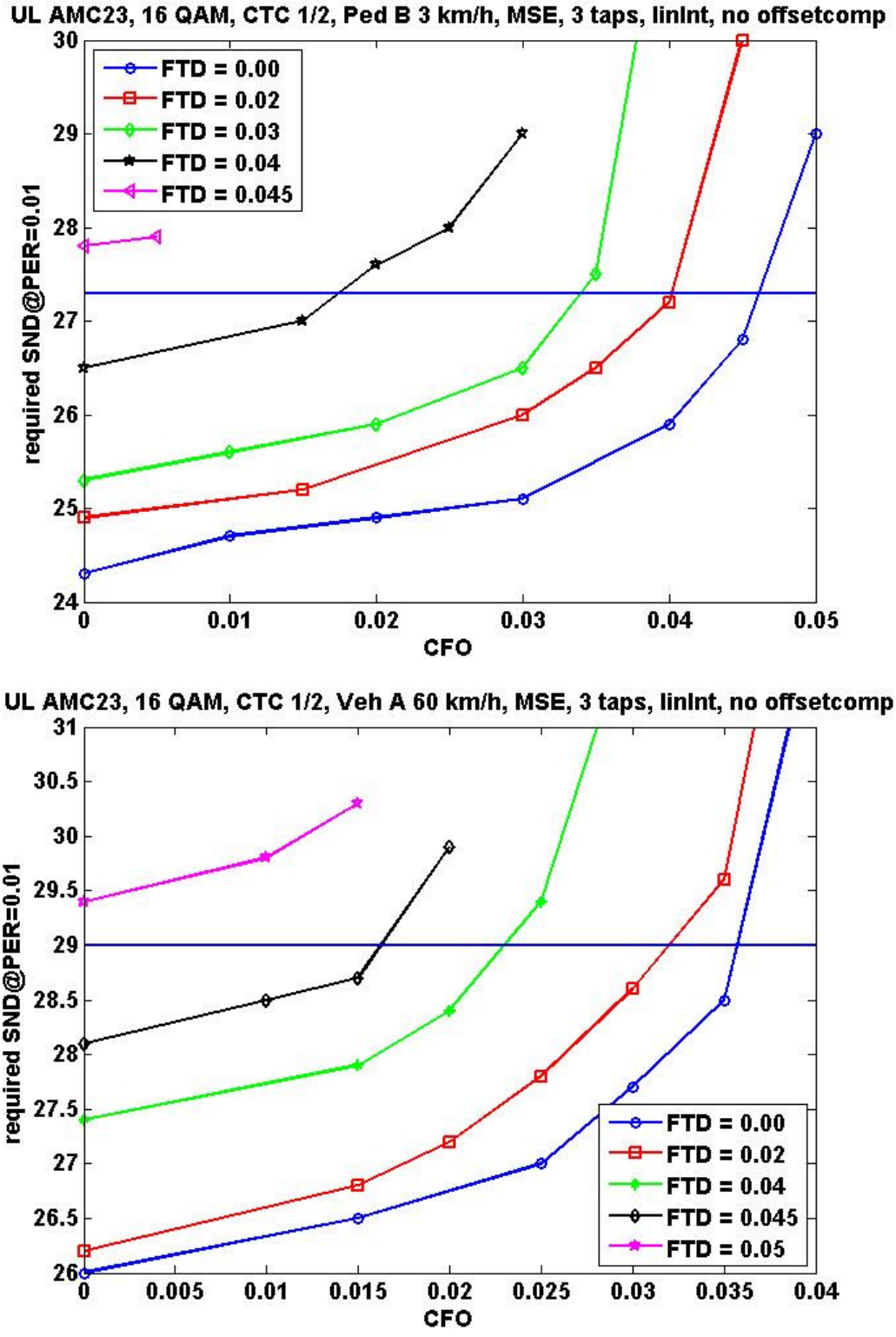

Figure 15: Required SNR to achieve PER = 1% without offset compensation (lower bound)

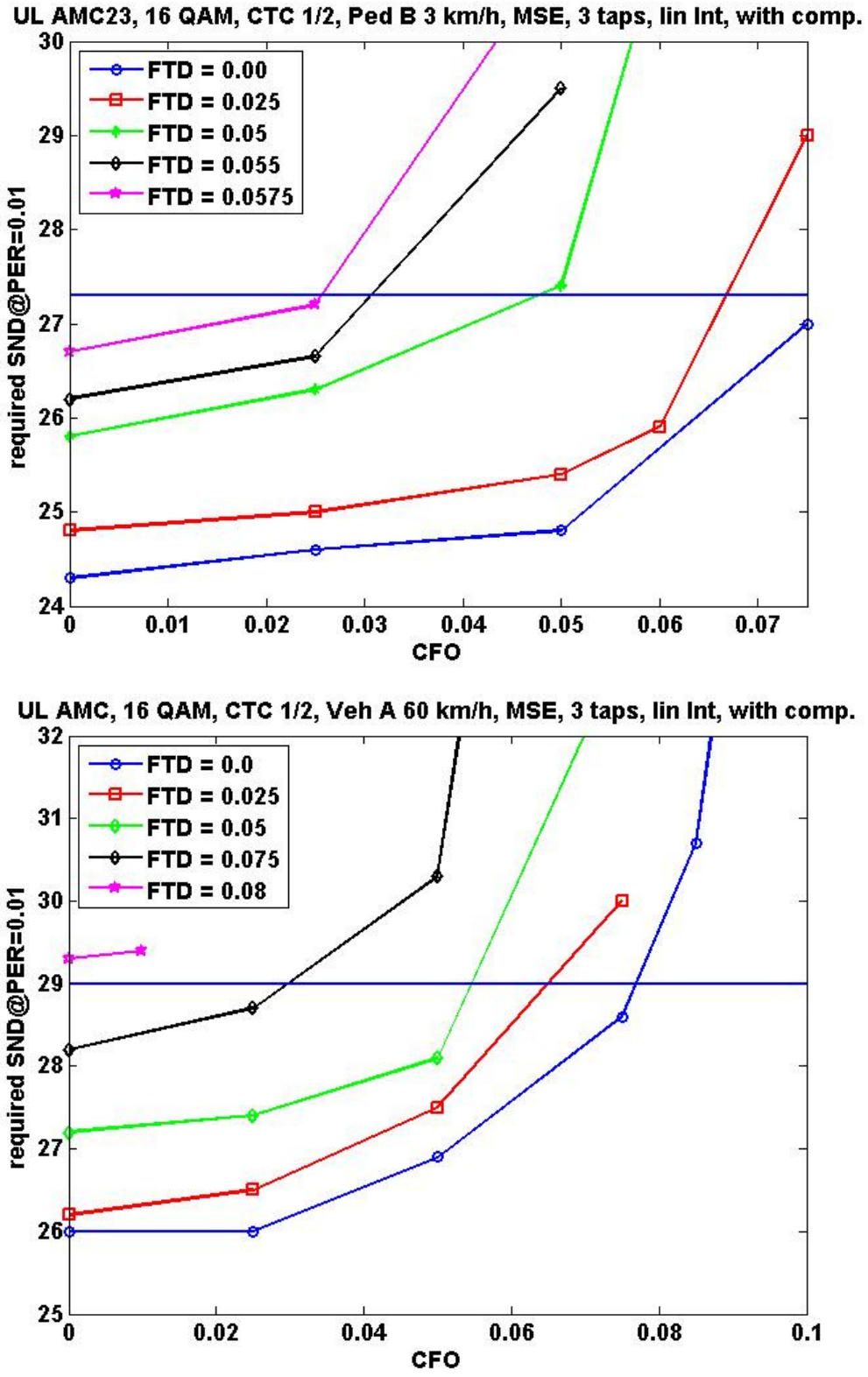

Figure 16: Required SNR to achieve PER = 1% with offset compensation based on scattered pilots

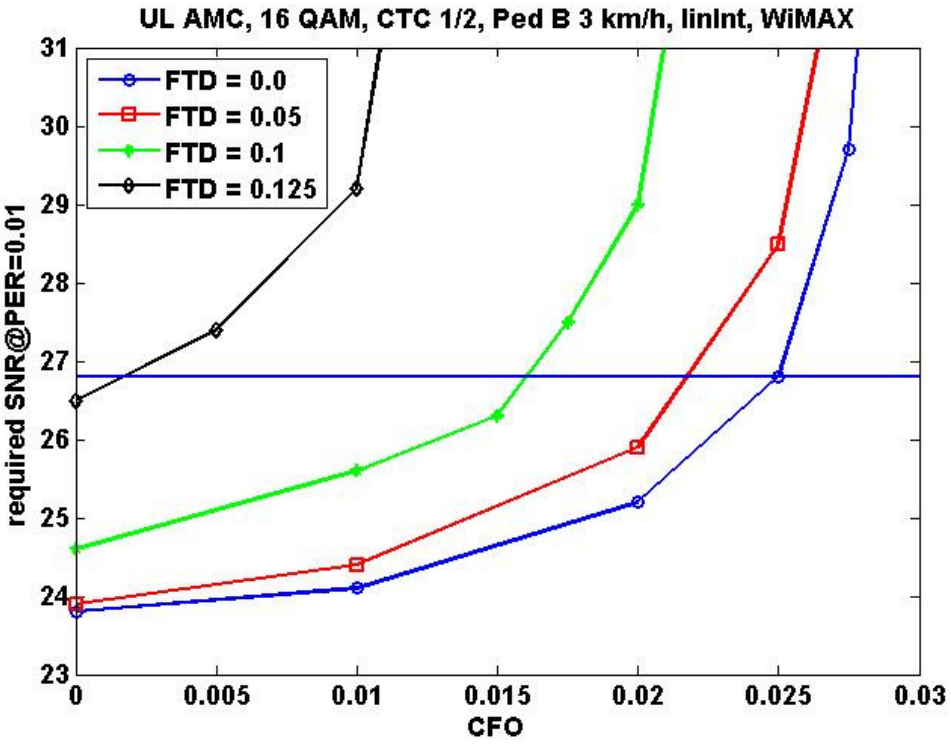

Figure 17: Required SNR to achieve PER = 1% with the reference WiMAX system

With the given setup the WiMAX system could not successfully deliver PER = 1% in case of Veh A 60 km/h and 16 QAM for any SNR, even with perfect synchronization. Due to the high Doppler (in the range of 1% of the subcarrier spacing) already with perfect synchronization the carrier frequency and with it the signal degeneration varies too much.

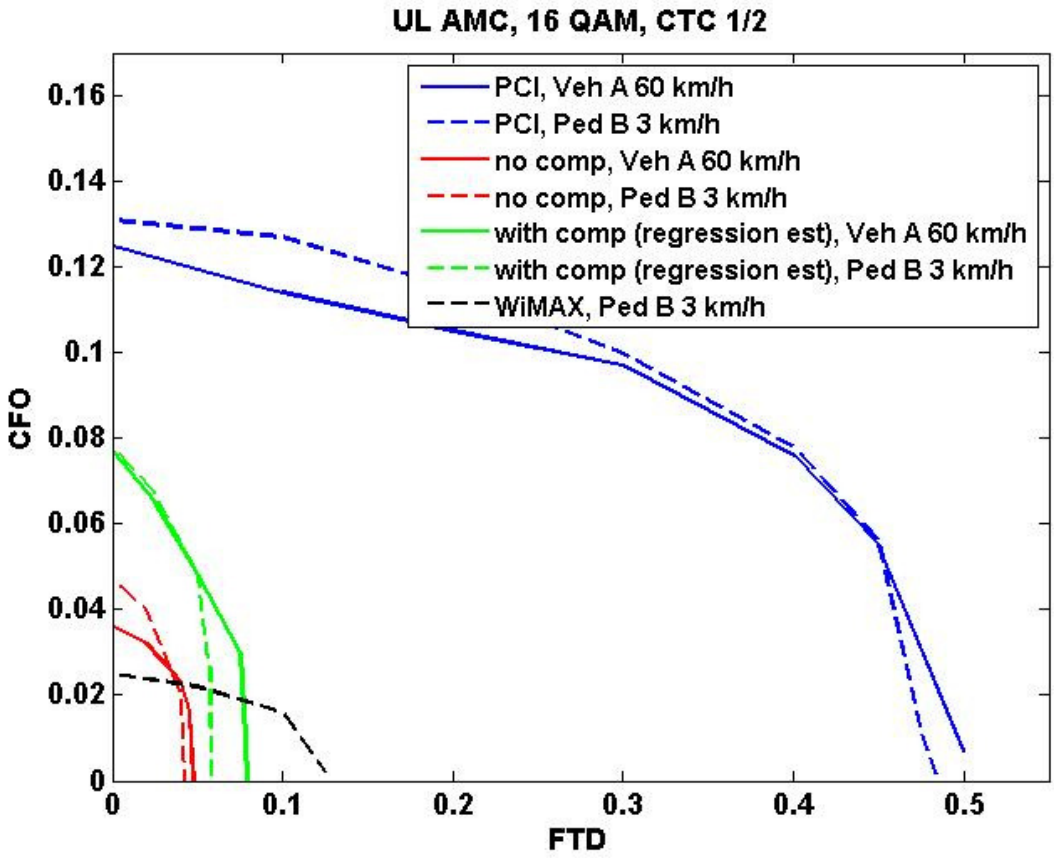

Figure 18: CFO/FTD tolerances with 3 dB link budget dedicated to desynchronization

Similar conclusions can be drawn as with QPSK. The gains due to the FTD compensation are a bit higher as 16 QAM is more sensitive to FTD as QPSK. Thus the performance of the uncompensated system with respect to the FTD sensitivity not already reaches the limit of the compensated one due to the pilot spacing.

2.3.2.3 64QAM

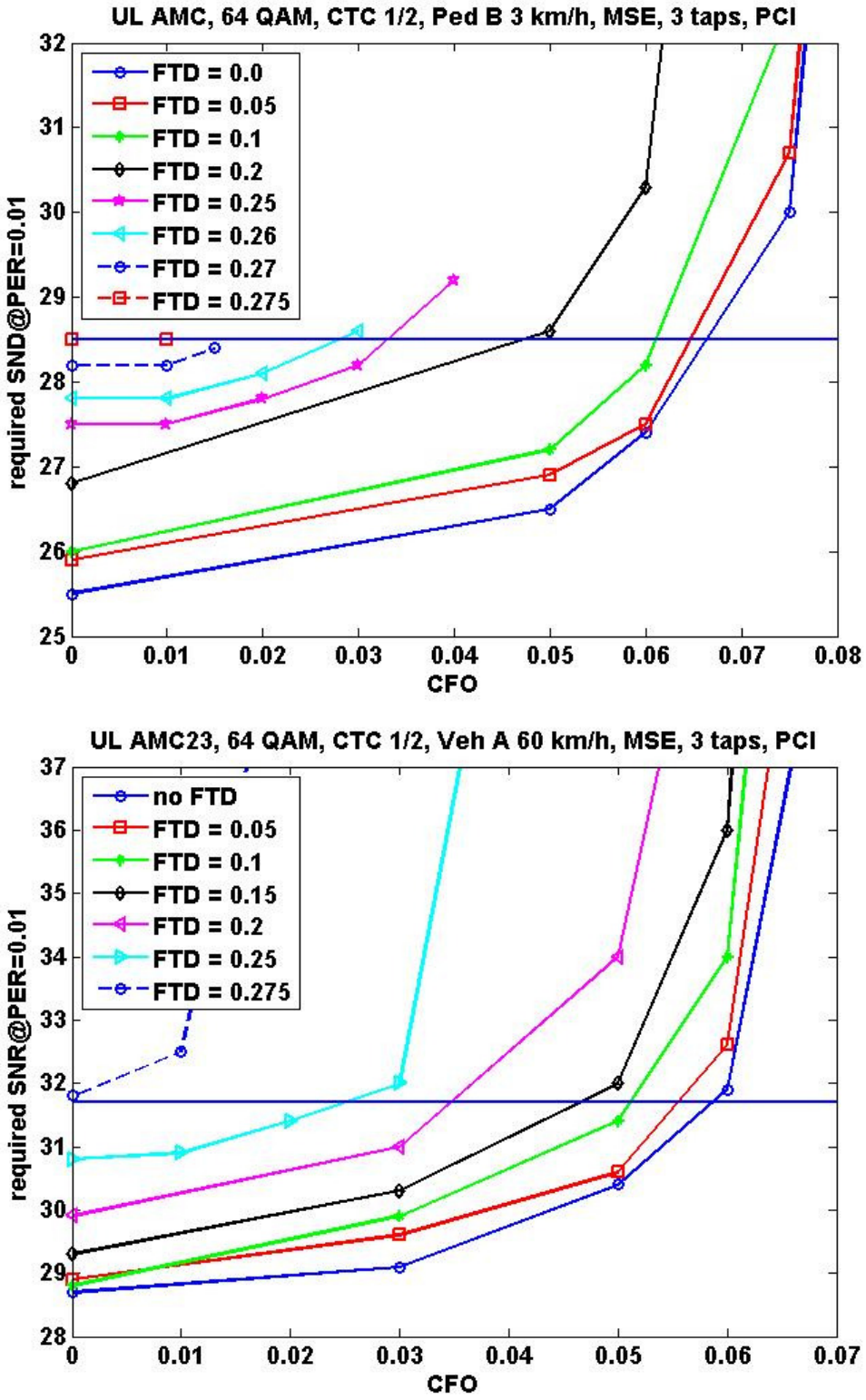

Figure 19: Required SNR to achieve PER = 1% with perfect channel information (upper bound)

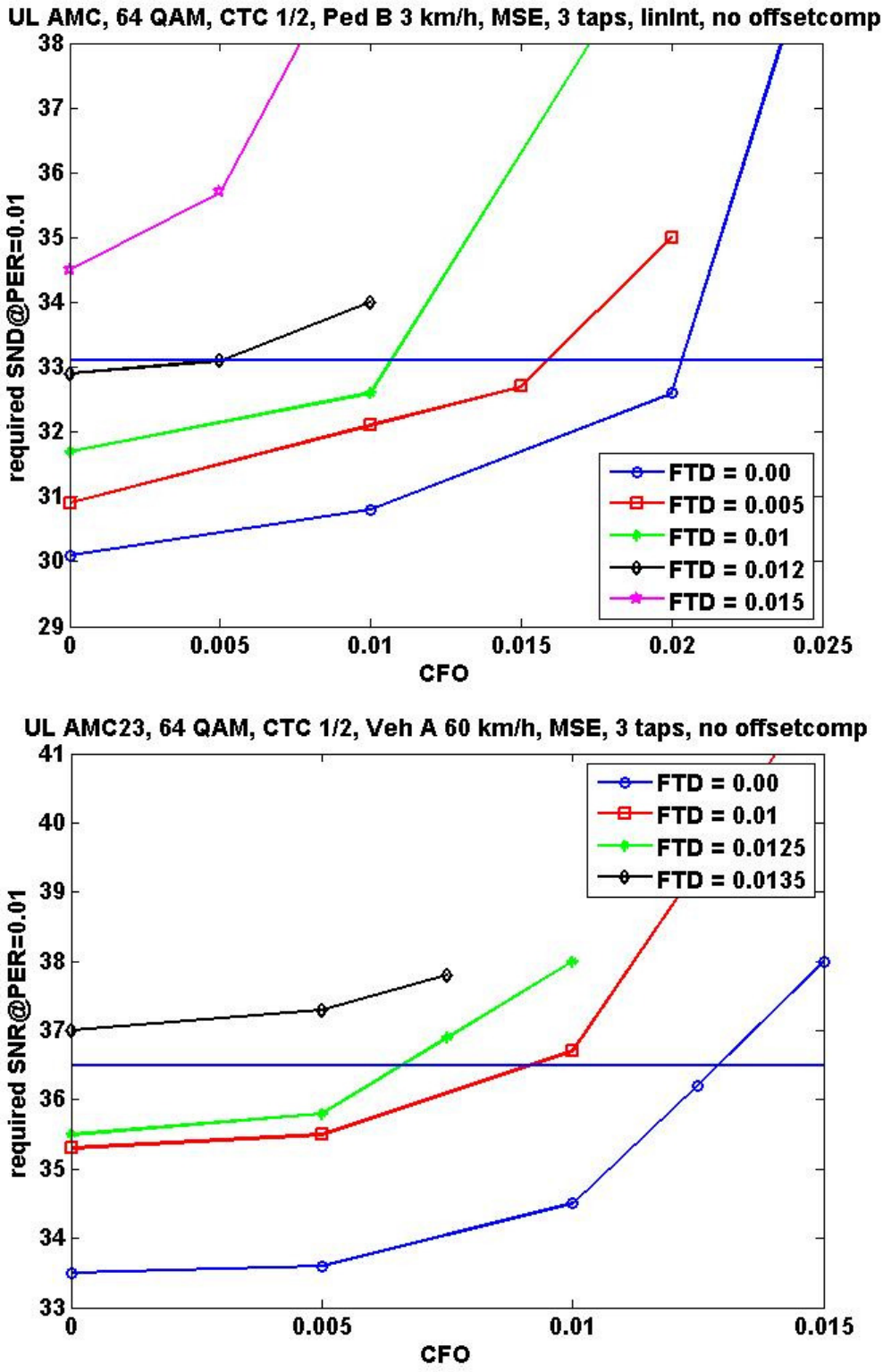

Figure 20: Required SNR to achieve PER = 1% without offset compensation (lower bound)

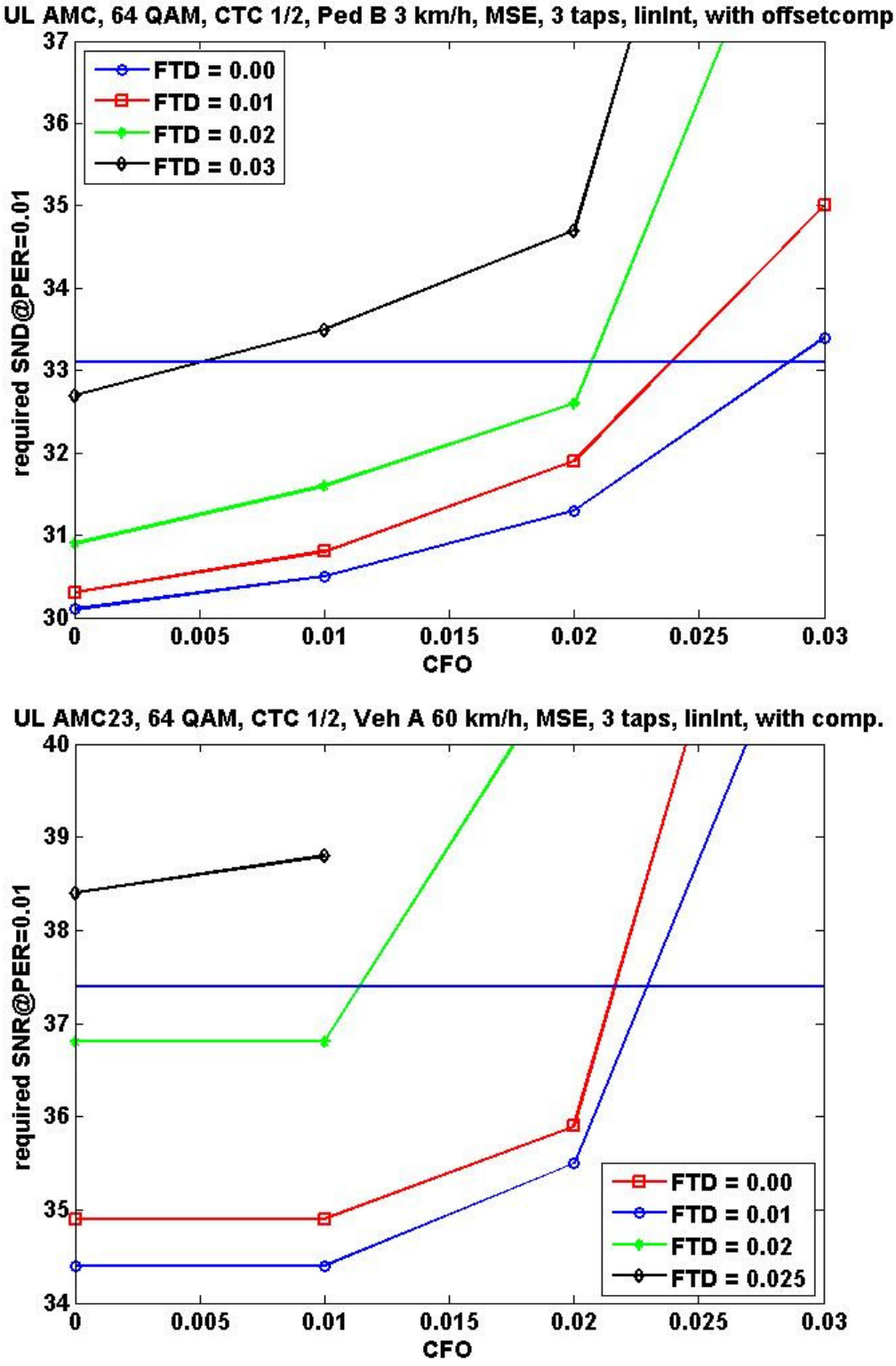

Figure 21: Required SNR to achieve PER = 1% with offset compensation based on scattered pilots

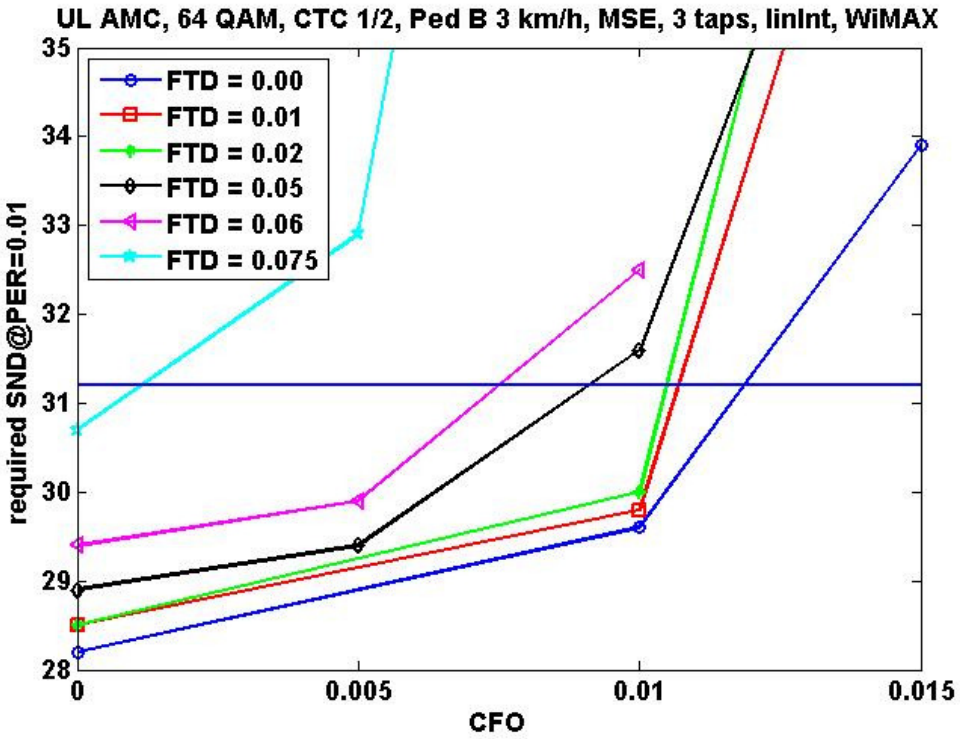

Figure 22: Required SNR to achieve PER = 1% with the reference WiMAX system

Again as with 16 QAM the given setup could not decode the data successfully in case of the WiMAX setup and Veh A 60 km/h, even with perfect synchronization. A look into the figure depicting the CFO/FTD sensitivities confirms this: with 64 QAM the degradation due to Doppler gets significant (around 1% of the subcarrier spacing at 60 km/h). This is visible with the fact that even without additional CFO (beside Doppler) Veh A 60 km/h cannot outperform Ped B 3 km/h. This was different for lower modulation orders.

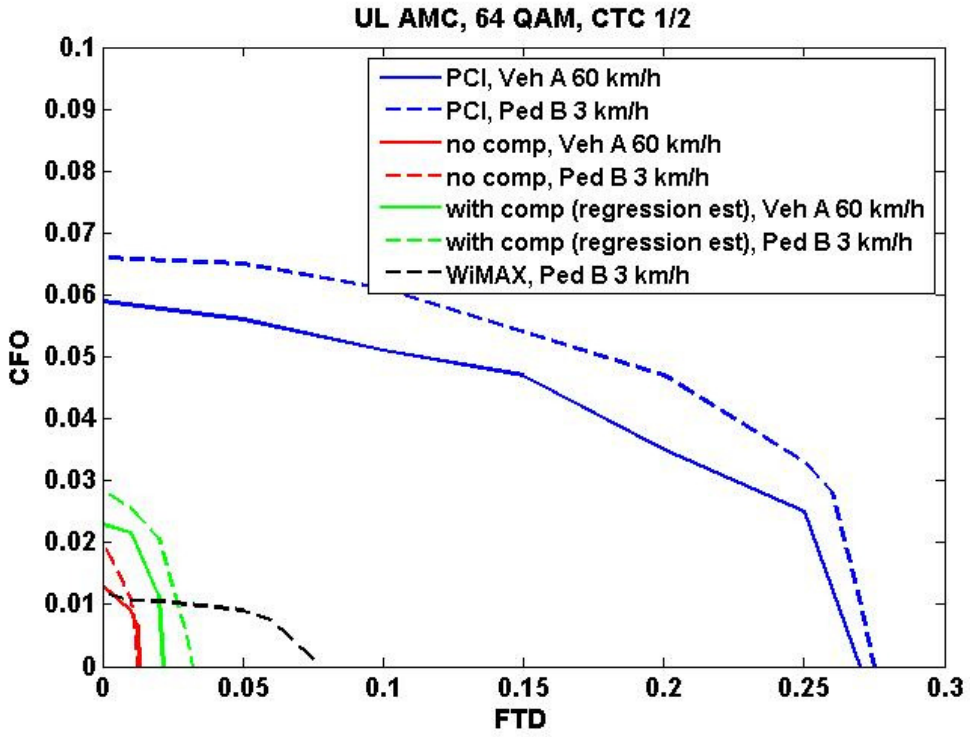

Figure 23: CFO/FTD tolerances with 3 dB link budget dedicated to desynchronization

Again (naturally) a system having a channel similar to Veh A 60 km/h is more sensitive to CFO than in the case of Ped B 3 km/h. Naturally the higher Doppler is partly causing this, however, not exclusively. Ped B has a higher delay spread and thus a lower coherence bandwidth. With respect to channel estimation/equalization worse results are to be expected. However, with respect to the diversity gain achievable with forward error coding, where the data is spread over a frequency range the smaller coherence bandwidth is helpful. In this case deep fades are covering less symbols leading to smaller error bursts. This is performance increasing as convolutional turbo coding is sensitive to long error bursts. The following figure separates these two effects:

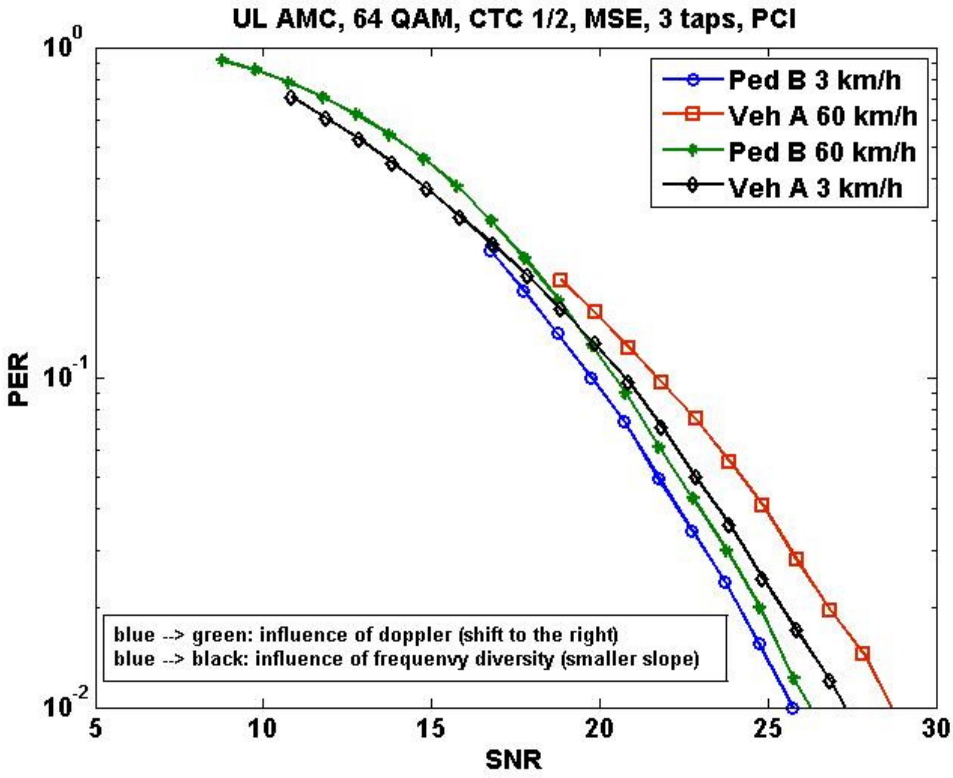

Figure 24: Impact of Doppler and coherence bandwidth

By comparing the black curve (diamond) and the blue one (circles) the impact of the coherence bandwidth is visible. The slope of the latter one is steeper hinting the higher diversity gain (the comparison of red (rectangles) and green (stars) illustrates the same). The impact of the Doppler can be seen by comparing the blue line (circles) and the green one (stars) or alternatively black (diamond) and red (rectangles). Here the impact is a shift to higher SNR values, the slope is not changing. This behaviour perfectly fits the deviation to be expected (inter carrier interference and phase distortion) due to higher Doppler spread. The impact to and by the channel estimation/equalization is excluded by the use of perfect channel information (PCI).

2.4 RF Impairments Baseband Model

As a multicarrier modulation scheme, FBMC shares many of the characteristics of OFDM. From the RF point of view, the aspects commonly studied includes peak-to-average power ratio and sensitivity to non linearity's, sensitivity to inter-carrier (ICI) and inter-symbol interferences (ISI), carrier frequency offset (CFO) and spectral dynamic range (SDR) and out-of-band attenuation.
The main difference of FBMC over OFDM holds in the OQAM pre/post processing and the filtering which is performed in the time domain. The advantage of these additional processes is a much increased spectral selectivity well adapted to applications such as cognitive radio or frequency division multiple accesses at the expense of an increased latency. From the RF architecture point of view, the impairments occurring due to the imperfections of the analog components might have a different effect on an FBMC system compared to OFDM. This is either because the OQAM or filter-banks modify how the impairment impacts the signal or because the expected performances of the system are not the same. An example of the first kind is the fact that the filter can result in inter symbol interferences in the presence of impairments in the channel, and an example of the second kind is the spectral dynamic range of the overall system that have need to be improved in order to exploit the full possibilities inherent to the FBMC waveform.
As a consequence, in order to implement FBMC as a viable alternative to OFDM, the aspects related to RF impairments have to be studied. This is done in the frame of PHYDYAS in WP8 and WP9.
The purpose of this contribution is to present the RF model blocks that will be integrated in the PHYDYAS simulator and that will be used as a basis for RF impairments studies.

2.4.1 Sources of Impairment

Common sources of RF impairment have already been discussed in D5.1 in section 4.2. and thus for a detailed description of these sources, we refer to this section. The list of possible sources is given below.

Sources of impairment:

- DC Offset
- Phase Noise
- Spurious Effects
- IQ Mismatch
- Frequency Offset
- Non-linearities
- Power Amplifier
- ADC Impairments due to clipping or quantisation noise

2.4.2 Model Blocks

The various imperfections have been isolated and a baseband model has been developed in Matlab for each source of impairment.
Even if major sources of imperfection can be pointed out and isolated, as the classical TX/Rx architecture shows in Figure 25, it is relevant, whenever possible, to study imperfections as global consequences in a black box approach. This allows reaching more general conclusions which are

not limited to specific architectures. For instance, although non-linearity of the transmitter mainly results from the power amplifier, non-linearity may also occur in other components such as mixers and it all adds up.

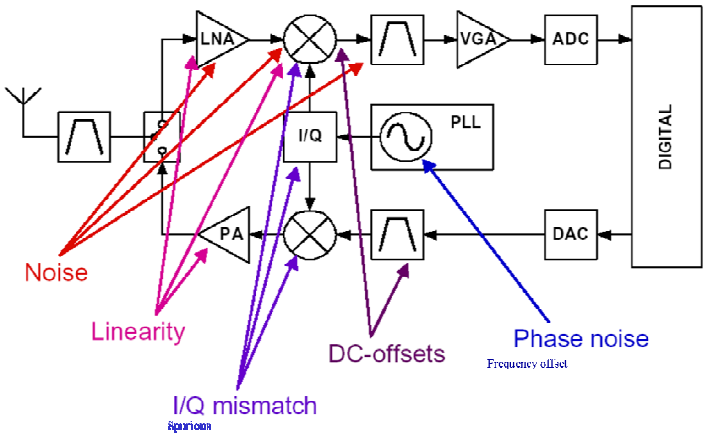

Figure 25: RF Impairment Sources in a Receiver

Each source of imperfection is implemented in Matlab by a function. Those functions can then be assembled to model various RF architectures. This section aims at describing the relationship between the input and output base band signals of each block ends.

2.4.2.1 DC Offset

DC offsets can be divided into a static part and a dynamic part:

$$V_s(t) = V_e(t) + Statique_{DC} + Dynamique_{DC} \qquad (2.1)$$

The static DC offset is applied during the whole simulation while the dynamic DC offset can be applied during a certain number of symbols.

2.4.2.2 Phase Noise

The local oscillator introduced a random phase rotation $\Delta\Phi(t)$ to the received signal when it is down converted. Phase noise broadens oscillator spectrum.

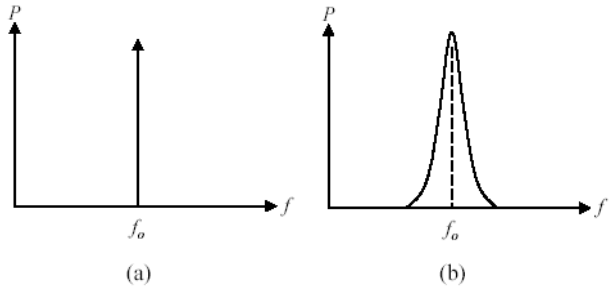

Figure 26: Output Frequency Spectrum, Ideal (a) and with Phase Noise (b)

Free Running Oscillator

The phase noise power spectral density is commonly modelled as a Lorentzian spectrum. This behaviour can be asymptotically modelled by a Wiener process, whose spectrum decreasing as a function of $\frac{1}{f^2}$, and a noise floor.

In the time domain, the variance of the Wiener phase noise is a linear function of the time:

$$\sigma_{\Delta\Phi}{}^2(t) = 2Dt \qquad (2.2)$$

where D is the diffusion factor of the free running oscillator. Thus $\Delta\Phi(t)$ is modelized as a normal gaussian random variable which variance increases linearly with the time. In order to integrate the diffusion factor in simulations, it must be scaled to the sampling frequency:

$$D_{Ts}(s) = \frac{D}{f_s} \qquad (2.3)$$

The Wiener phase noise is computed iteratively as follows:

$$\Delta\Phi_1(n+1) = \Delta\Phi_1(n) + X(n) \text{ where } X(n)\sim N(0, 2DTs). \qquad (2.4)$$

The noise floor is modelled by a white Gaussian noise of variance σ^2.

The total phase noise is:

$$\Delta\Phi(n) = \Delta\Phi_1(n) + randn(\sigma^2) \qquad (2.5)$$

Finally, the baseband signal is impaired by the following relationship:

$$V_s(nT_s) = V_e(nT_s) \cdot e^{\Delta\Phi(n)} \qquad (2.6)$$

Controlled Oscillator

When the oscillator is integrated in a PLL, its Wiener phase noise component is filtered. The transfer function *H* of the PLL is equivalent to a high pass filter. Let's note w_n and ξ the PLL cut-off frequency and resonance coefficient. The first and second order PLL Laplace transfer functions are defined by the underneath expressions:

$$H_1(p) = \frac{p}{p + w_n}$$
$$H_2(p) = \frac{p^2}{p^2 + 2p.\xi.w_n + w_n{}^2} \qquad (2.7)$$

Temporal PLL filter coefficients are eventually obtained by carrying out a z-transform. PLL filtering is achieved the same way as channel filtering.

Phase Noise Spurious

The phase comparator of the PLL generates parasitic harmonics. They introduce a frequency modulation of the pulsation at the RF signal output of the oscillator:

$$w(t) = w_0 + \Delta w.\cos(w_m t) \qquad (2.8)$$

Where w_0 is the pulsation of the fundamental of the oscillator and w_m are the pulsation of the harmonics generated by the PLL. It has to be noted that in this equation, the phase noise is not taken into account. Contributions in the phase noise are additive. Random phase noise and spurious can therefore be treated separately.

The instantaneous phase at the output of the oscillator is therefore:

$$\Delta\Phi_2(t) = w_0 t + m.\sin(w_m t) + \varphi_0 \qquad (2.9)$$

where $m = \frac{\Delta w}{w_m}$ is defined as the modulation index.

Usually, phase noise spurious are characterized and measured by the power of the harmonics in the phase noise of the signal at the output of the PLL. The question is therefore how to relate the modulation index m to the power of the phase spurious. The instantaneous phase obtained above is introduced in the signal produced at the output of the PLL:

$$\begin{aligned} v(t) &= \cos(\Delta\Phi_2(t)) \\ &= \cos(w_0 t + \varphi_0)\cos(m.\sin(w_m t)) - \sin(w_0 t + \varphi_0)\sin(m.\sin(w_m t)) \end{aligned} \qquad (2.10)$$

Using Bessel functions of the first kind, it can be written as follows:

$$\begin{aligned} v(t) = \; & J_0(m)\cos(w_0 t) + J_1(m)\left[\cos((w_0 - w_m)t) - \cos((w_0 + w_m)t)\right] \\ & + J_2(m)\left[\cos((w_0 - 2w_m)t) - \cos((w_0 + 2w_m)t)\right] + \ldots \end{aligned} \qquad (2.11)$$

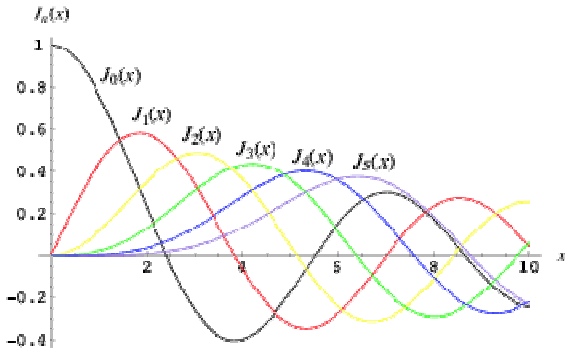

Figure 27: First Order Bessel Function

The phase spurious is thus completely defined by the power of its different harmonics ($k.w_k$) normalized by the fundamental:

$$P_{dBc}(k) = 20\log\left(\frac{J_k(m)}{J_0(m)}\right) \qquad (2.12)$$

For practical values ($m < 1$), the phase spurious can be reduced to a unique harmonic. Moreover, for small m, the power ratio $P_{dBc}(k)$ is well approximated by a linear function of the modulation index.

Hence, it can be computed easily as a function of the phase spurious power:

$$m \approx 10^{\frac{P_{dBc}+6}{20}} \qquad (2.13)$$

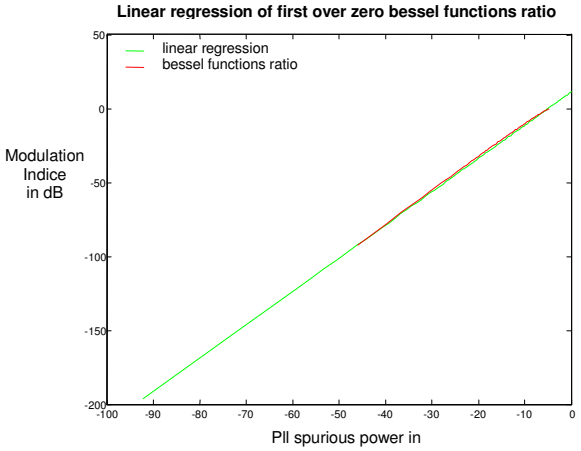

Figure 28: Spurious Phase Noise Power Ratio

The phase spurious is then:

$$\Delta\Phi_2(n) = m.\sin(n.w_m) \qquad (2.14)$$

Where w_m is the PLL pulsation normalized by the sampling frequency. Eventually, the total phase noise is the sum of all the components:

$$\Delta\Phi(n) = \Delta\Phi_1(n) + \Delta\Phi_2(n) + randn(\sigma^2) \qquad (2.15)$$

A Cartesian to polar conversion is performed in order to evaluate the phase and module of each received sample. Then the phase noise is added to the original phase, and a polar to Cartesian conversion is done to get the output signal.

And the baseband signal is given by:

$$V_s(nT_s) = V_e(nT_s) \cdot e^{\Delta\Phi(n)} \qquad (2.16)$$

2.4.2.3 Spurious

Parasitic signals are generated by the different active components in the receiver. A harmonic with a given frequency and amplitude is added to the received signal.

2.4.2.4 IQ Mismatch

Figure 29 presents a model of the IQ mismatch impairment.

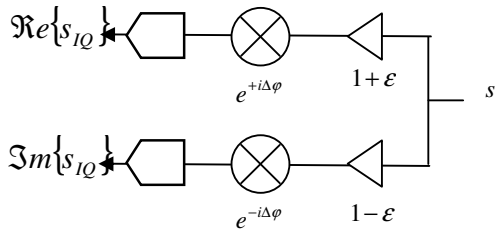

Figure 29: IQ Mismatch Model

The expressions of the real and imaginary parts (I and Q) of the baseband signal are given by:

$$\begin{aligned}\mathrm{Re}\{s_{IQ}(t)\} &= (1+\varepsilon)\cos\Delta\Phi\,\mathrm{Re}\{s(t)\} - (1+\varepsilon)\sin\Delta\Phi\,\mathrm{Im}\{s(t)\}\\ \mathrm{Im}\{s_{IQ}(t)\} &= (1-\varepsilon)\cos\Delta\Phi\,\mathrm{Im}\{s(t)\} - (1-\varepsilon)\sin\Delta\Phi\,\mathrm{Re}\{s(t)\}\end{aligned} \quad (2.17)$$

with ε and $\Delta\phi$ are respectively the amplitude and phase imbalances.
Two complex number α and β related to the imbalance parameters ε and $\Delta\phi$ are introduced by:

$$\begin{cases}\alpha = \cos(\Delta\phi) + j\cdot\varepsilon\cdot\sin(\Delta\phi)\\ \beta = \varepsilon\cdot\cos(\Delta\phi) - j\cdot\sin(\Delta\phi)\end{cases} \quad (2.18)$$

The impact of the IQ imbalance acts as a linear combination of the non impaired signal and the complex conjugate of the non impaired signal:

$$V_s(t) = \alpha\cdot V_e(t) + \beta\cdot V_e(t)^* \quad (2.19)$$

In the RF impairments model, α and β are calculated thanks to gain and phase mismatches parameters. Then, the signal is impaired according to (1.19).

2.4.2.5 Frequency Offset

Frequency offset between emitter and receiver frequency synthesizer are performed by a simple phase rotator:

$$V_s(t) = V_{e(t)}.e^{j2\pi f_{offset}t} \quad (2.20)$$

2.4.2.6 Non-linearities

Front-End RF non-linearities come from all the active components at the receiver (Low Noises Amplifier, mixer, etc…). These impairments are modelled by a polynomial function whose coefficients are computed thanks to inter-modulation (IM) product values:

$$V_s(t) = \sum_{k=1}^{M} a_k V_e(t)^k \qquad (2.21)$$

For a 2-tones signal:

$$V_e(t) = A\cos(w1t) + A\cos(w2t) \qquad (2.22)$$

Intermodulation products appear when developing the third order polynomial $a_1 x + a_3 x^3$:

$$\begin{aligned} V_s(t) = \; & V\left(a_1 + \frac{9}{4}a_3A^2\right)A\cos(w1t) + \left(a_1 + \frac{9}{4}a_3A^2\right)A\cos(w2t) \\ & + \frac{3}{4}a_3A^3\cos(2w1 - w2) + \frac{3}{4}a_3A^3\cos(2w2 - w1) \end{aligned} \qquad (2.23)$$

The third order interception point (IP3) amplitude is computed thanks to polynomial coefficients:

$$A_{IP3} = \sqrt{\frac{4}{3}\left|\frac{a_1}{a_3}\right|} \qquad (2.24)$$

For $a_1 = 1$, we get :

$$a3 = \frac{4}{3A_{IP3}{}^2} \qquad (2.25)$$

With further developments, we also have:

$$a_5 = \frac{8}{5A_{IP5}{}^4} \qquad (2.26)$$

The parameter IP3 is determinate as followed. Let's note $A_{IP3,s}$ the output IP3 amplitude, $A_{IP3,e}$ the input IP3 amplitude, A_s the output signal amplitude and A_e the input signal amplitude. We assume that output over input amplitude ratio are quite similar for inter modulation products and for signals. Then:

$$A_{IP3,s} \approx A_{IP3,e}\frac{A_s}{A_e} \qquad (2.27)$$

So, we get:

$$a_3 = \frac{4Gain^2}{3A_{IP3,s}{}^2} \qquad (2.28)$$

A_{IP3} is obtained when cascading the IP3 of the different reception chain components thanks to the Norton formula.

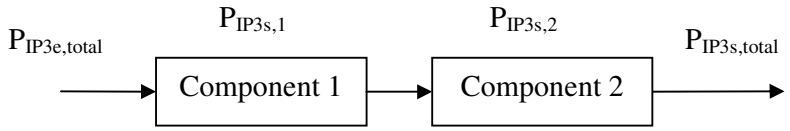

Figure 30: Diagram of the Norton Formula

$$P_{IP3,total} = P_{IP3s,2} - 10\log\left(1 + \frac{1}{G_2}\frac{P_{IP3s,2}}{P_{IP3s,1}}\right) \quad (2.29)$$

where G_2 is the gain of component 2.

In the RF impairments model, only IM3 and IM5 are taken into account. Polynomial coefficients a_3 and a_5 are computed thanks to fifth-order odd polynomial coefficients.

2.4.2.7 Power Amplifier

Power amplifiers (PA) are modeled by their AMAM and AMPM characteristics. Non linearity's of the power amplifier are taken into account by the implementation of a Rapp's SSPA and TWT models.

The Solid State High Power Amplifier (SSPA) model is represented by its AMAM and AMPM characteristics related amplitude and phase of the ouput signal as a function of the amplitude of the input signal.

The SSPA's AMAM characteristic is given by:

$$g_{sspa}(A(t)) = v \cdot \frac{A}{\left(1 + \left(\frac{A}{A_0}\right)^{2p}\right)^{\frac{1}{2p}}} \quad (2.30)$$

With p > 0, A0 > 0, the parameters of the models that can be interpreted as the saturation level and the smoothness of the transition between linear and saturation regions.

The SSPA model assumes no relationship between input amplitude and output phase. Its AMPM characteristic is therefore null.

The TWT model is expressed by it AMAM characteristic:

$$g_{twt}(A(t)) = \frac{\alpha_A A}{\left(1 + \beta_A A^2\right)} \quad (2.31)$$

And by its AMPM characteristic :

$$\Phi_{twt}(A(t)) = \frac{\alpha_\Phi A^2}{\left(1+\beta_\Phi A^2\right)} \qquad (2.32)$$

α_A, β_A ,α_ϕ and β_ϕ being the parameter of the model.
Finally, assuming that the normalized input baseband signal is break down into its amplitude and phase:

$$V_e(t) = A(t).e^{j\phi(t)} \qquad (2.33)$$

The output signal is expressed by:

$$V_s(t) = g_x\left[A(t)\right].e^{j(\phi_x(t)+\Phi[A(t)])} \qquad (2.34)$$

Subscript *x* refers to the choice of the model, namely SSPA or TWT.

In the current implementation, the input signal is scaled by the input back-off (IBO) parameter relatively to normalized AMAM and AMPM characteristics (i.e. small signal gain and saturation power are unity).

2.4.2.8 Analogue to Digital Conversion Impairments

If the signal has a zero mean, we define an amplitude functional zone over [-V;V]. The signal is clipped and then quantified. The quantification error corresponds to the integer part rounding:

$$V_s(nT_s) = E\left[\frac{V_e(nT_s)+V}{2V} * (2^N - 1)\right]\left(\frac{2V}{2^N - 1}\right) - V \qquad (2.35)$$

where V_e is the signal value to be coded and *N* the number of quantification bits. Sampling jitter has not been modelled yet.

2.4.2.9 Implementation in the Phydyas Simulation Chain

Front-end blocs have been integrated in Matlab. An overview of the simulation blocks and how they are parameterized is depicted in Figure 31.

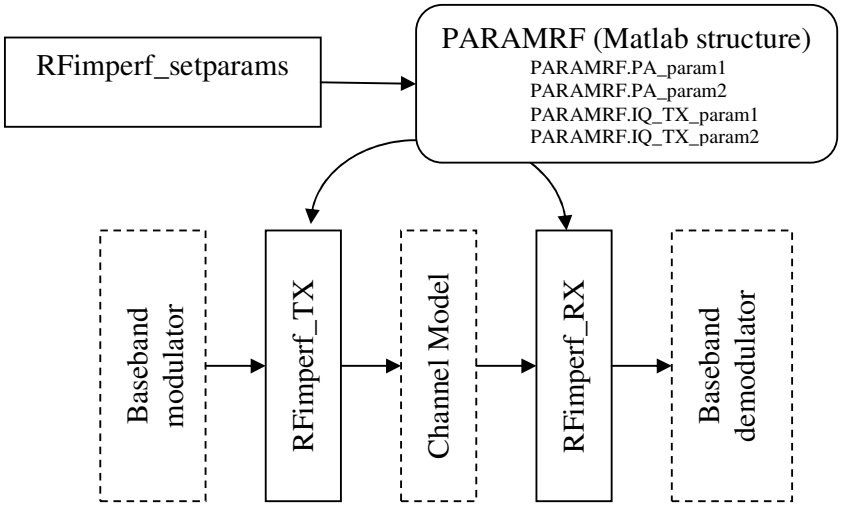

Figure 31: Global Architecture of the Simulation Chain

Sub-functions modelling each imperfection are grouped in two blocks *RFimper_TX* and *RFimper_RX*, one for the TX and one for the RX front-end. Different versions of the two blocks can be assembled in order to fit to particularities in a given RF architectures.

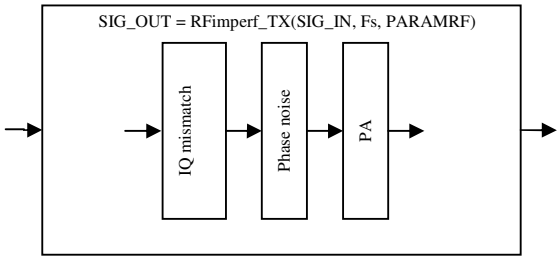

Figure 32: RFimper_TX

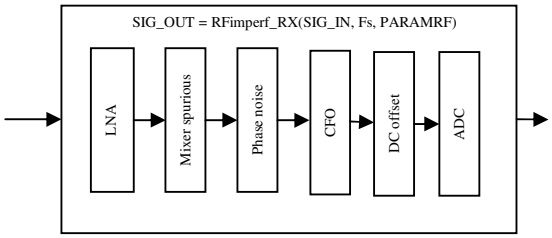

Figure 33: RFimper_RX

They are parameterized by a structure containing all the parameters required for the embedded models named *PARAMRF*. The structure is initialized to default value by the function *RFimperf_setparams*. Then, when the functions are implemented in a given simulation chain, the parameter in the *PARAMRF* structure can be overloaded as required by scripts or parameter sweep depending on the context of the ongoing study. A table with all parameters can be found below.

Parameter	Description	Unit
Function_RFimperf_RX	Function handle defining version of the block to use	
Function_RFimperf_TX	Function handle defining version of the block to use	
RX_ADC_Input_Power	Input power	dBm
RX_ADC_Nb_bits	Number of bit of quantization	
RX_ADConv	Activate block ? (0/1)	
RX_AntTX_G	Antenna Gain	dB
RX_CFO	Activate block ? (0/1)	
RX_CFO_delta_f	Offset carrier	Hz
RX_DCFilter	Activate block ? (0/1)	

RX_DCO	Activate block ? (0/1)	
RX_DC_offset	DC Offset	dBm
RX_IQ_delta_gain_dB		dB
RX_IQ_delta_phase_deg		degree
RX_IQ_mismatch	Activate block ? (0/1)	
RX_LNA	Activate block ? (0/1)	
RX_LNA_Gain	Gain	dB
RX_LNA_IIP3	IIP3	dBm
RX_LNA_IIP5	IIP5	dBm
RX_LO_IF_Isol	IF isolation	dB
RX_LO_Power	LO Power	dBm
RX_Mixer	Activate block ? (0/1)	
RX_Mixer_Gain	Gain	dB
RX_Mixer_IIP3	IIP3	dBm
RX_NF	Receiver Noise Factor	dB
RX_PhaseNoise	Activate block ? (0/1)	
RX_PhaseNoise_FRef	Frequency offset reference for phase noise measurement	
RX_PhaseNoise_PSD_1MHZ	Phase noise at given frequency offset	dBc/Hz
RX_PhaseNoise_pll_cutoff	PLL cutoff frequency	Hz
RX_PhaseNoise_wpn	Phase Noise	dBc/Hz
RX_Split_Loss	Split losses due to I/Q branches split	dB
TX_PA	Activate block ? (0/1)	
TX_PA_A0	SSPA Model parameter	
TX_PA_AMAM_model	Model for AMAM (SSPA,TWT,Simple,Spline)	
TX_PA_AMPM_model	Model for AMPM (SSPA,TWT,Simple,Spline)	
TX_PA_IBO	Input Back-Off at PA input	
TX_PA_TWTAM_Pinsat_dBm	Input saturation power for AMAM TWT model	dBm
TX_PA_TWTPM_Pinsat_dBm	Input saturation power for AMPM TWT model	dBm
TX_PA_alpha_A	TWT model parameter	
TX_PA_alpha_phi	TWT model parameter	
TX_PA_beta_A	TWT model parameter	
TX_PA_beta_phi	TWT model parameter	
TX_PA_limiter_Pinsat_dBm	SSPA Model parameter	
TX_PA_p	SSPA Model parameter	
TX_PA_pp_amp	Spline Model parameter	
TX_PA_pp_phase	Spline Model parameter	
TX_PA_renormalize	Debug purpose	
TX_PA_splineAM_Pinmin_dBm	Spline Model parameter	
TX_PA_splineAM_Pinsat_dBm	Spline Model parameter	
TX_PA_splinePM_Pinmin_dBm	Spline Model parameter	
TX_PA_splinePM_Pinsat_dBm	Spline Model parameter	
TX_PA_verbose	Debug purpose	
TX_IQ_delta_gain_dB	Gain	dB
TX_IQ_delta_phase_deg	Phase	degree
TX_IQ_mismatch	Activate block ? (0/1)	
TX_PhaseNoise	Activate block ? (0/1)	
TX_PhaseNoise_FRef	Frequency offset reference for phase noise measurement	
TX_PhaseNoise_PSD_1MHZ	Phase noise at given frequency offset	dBc/Hz
TX_PhaseNoise_pll_cutoff	PLL cutoff frequency	Hz
TX_PhaseNoise_wpn	Phase Noise	dBc/Hz

Table 9: RF Impairment Parameters

3 OFDM-FBMC Demonstrator Setup

One of the objectives of the laboratory setup is to test, measure and demonstrate the advantages of an FBMC-based approach as compared to an OFDM-based approach. It will try to bring additional insight on top of the simulation results. Realistic performance expectations can only be checked if we can evaluate the impact of large time scale channel effects in a repeatable way.
The figure below shows the demonstrator setup that will be build for the Phydyas project.

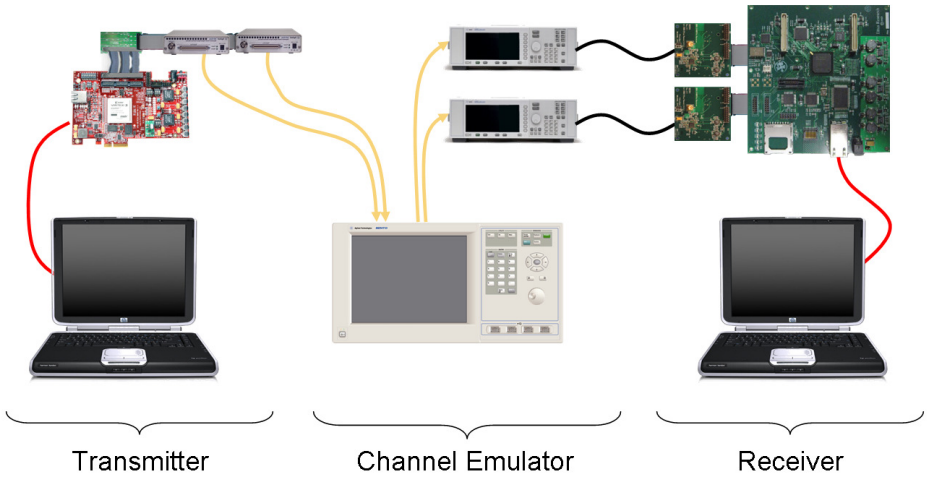

Figure 34: Demonstration Setup

3.1 WiMAX OFDM/FBMC Transmitter

The test transmitter will together with the fading simulator and test receiver make near real-time measurements of FBMC performance possible. It will also allow comparisons between WiMAX and FBMC performance.
Another interesting feature is the possibility to feed the receiver-part of the simulator with data generated with this transmitter.

SINTEF is responsible for the test transmitter design and implementation. This chapter describes the specifications and architecture of the test transmitter. Although some parts are already up and running, other parts still remain to be designed in detail. For this reason the level of detail differs between the modules.

3.1.1 Physical Architecture

The test transmitter is implemented on an FPGA board (HTG-V5-DDR3-PCIE) from www.hitechglobal.com. It will connect to a PC through a USB port for configuration and control. It will connect to the fading simulator through a LVDS based flat cable interface. Two instances of the transmitter can run simultaneously on the FPGA board using two different interfaces to the fading simulator. The two transmitter instances can be coordinated

for MIMO simulation or one can be used for data while the second transmit a disturbing neighbour station signal.
The FPGA board's main component is a Virtex-5 SX 95 FPGA from Xilinx. This is a large FPGA well suited to our design. It's being programmed in VHDL. Xilinx ISE development tools for Windows PC are used. The generated program is installed in a PROM on the FPGA board using the Xilinx tools. No special tools are needed to operate the transmitter.

Device	Configurable Logic Blocks (CLBs)			DSP48E Slices(2)	Block RAM Blocks			CMTs(4)	PCI Express Endpoint Blocks	Ethernet MAC Blocks	Maximum RocketIO GTP Transceivers(5)	Total I/O Banks(7)	Max User I/O(6)
	Array (Row x Col)	Virtex-5 Slices(1)	Max Distributed RAM (Kb)		18 Kb(3)	36 Kb	Max (Kb)						
XC5VSX95T	160 x 46	14,720	1,520	640	488	244	8,784	6	1	4	16	19	640

Table 10: FPGA Resources (Virtex-5 FPGA User Guide)

The board was chosen for its large FPGA and because it has connectors with many digital IO lines. It has many other resources like SRAM, DRAM, PCI Express interface, Gigabit serial interfaces and Ethernet interface but these will not be used in our design.

To control and configure the board, a PC will communicate with the board using a serial port over USB. The serial port runs at 115 kbaud which is sufficient for configuration and control.

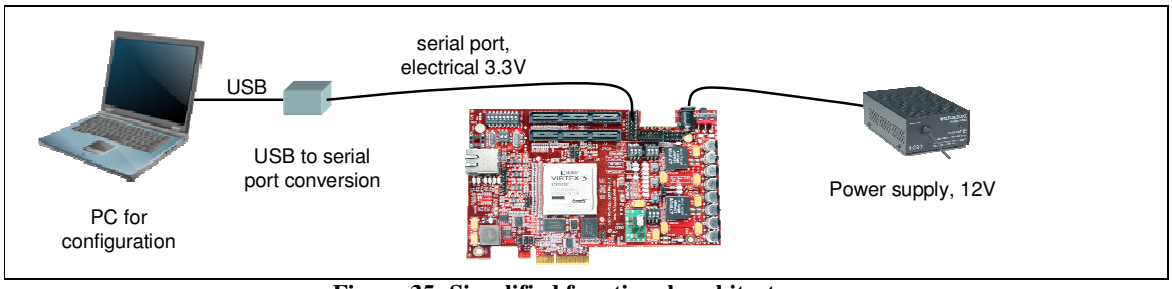

Figure 35: Simplified functional architecture

To support MIMO with two channels two separate interfaces to the fading emulator are needed. Each interface uses a flat cable and has one clock line and 16 single ended data lines plus ground lines. The clock comes from the fading simulator at 8 times the sample rate. The data lines alternate between 16 bit I and Q values.

The FPGA board with the interconnection boards will be mounted in a suitable cabinet to avoid damage to the boards and cables.

If the double use of interface boards causes too much noise a direct interconnect cable will be made.

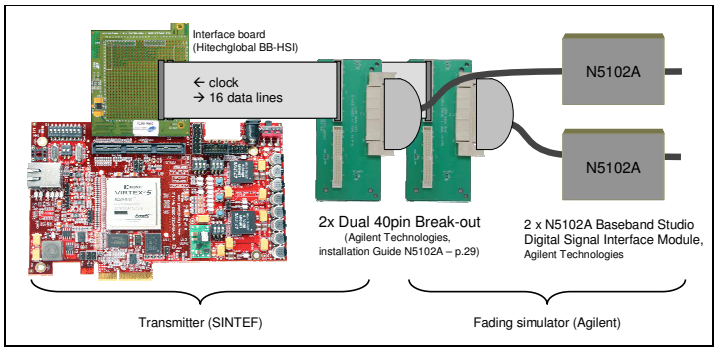

Figure 36: Interface to the Channel Emulator

3.1.2 Functional Architecture

The transmitter implements the physical layer only. Therefore the data source for the data part of the transmitted bursts is a PN generator. As the transmitter will only simulate one burst per frame with encoded data there is only one PN generator.
The coding and interleaving follows the WiMAX standard.

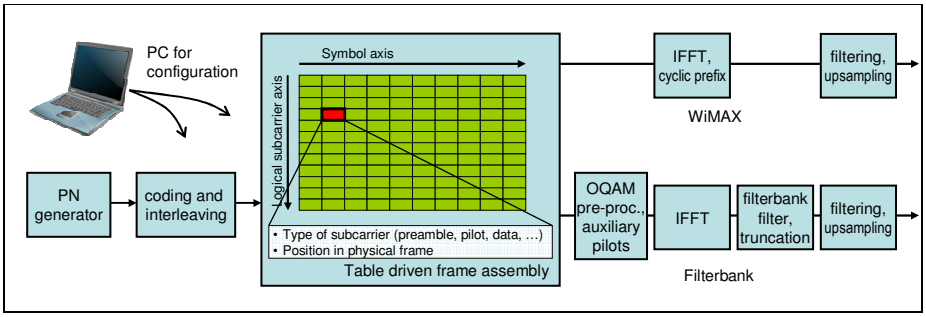

Figure 37: Simplified Functional Architecture

The frame assembly uses a table driven approach that can be configured to generate almost any configuration of frame (uplink or downlink) for WiMAX and filterbank modulation. There is one table element per subcarrier and symbol in the frame. This table element defines the encoding for that subcarrier. The subcarrier can be either an encoded burst data subcarrier or a random data subcarrier with selectable encoding (QPSK-64QAM), a preamble subcarrier, a pilot or a silent subcarrier. Gain can be set independently for each subcarrier. The table also encodes the logical to physical subcarrier mapping and can implement almost any conceivable mapping.

After the frame assembly comes the FFT and the upsampling and filtering. These stages will differ between the WiMAX and filterbank implementations.

For each scenario that will be implemented a set of configuration files will be prepared manually on the PC side. At the beginning of the test these files are loaded onto the FPGA board. Once the test has started the FPGA board runs autonomously.

3.1.3 Features and Limitations

The following features are supported by the architecture of the transmitter. All these features will not necessarily be exploited in the scenarios defined for the tests.

1. The transmitter will generate data based on PN sequences. It will code, modulate and transmit this data in real time.

2. The transmitter will support UPLINK and DOWNLINK (uplink has highest priority on implementation).

3. The transmitter will support PUSC and AMC23.
    - Limitation: Only one zone is supported at a time, using either PUSC or AMC23.

4. The transmitter will support ONE fully encoded burst with PN sequence data meant for bit/block error counting in the receiver. This burst can have any legal form, most sizes and be anywhere in the frame. It can also be placed in the FCH and DL-MAP positions.
    - The bit stream from the PN generator is chopped into blocks the size of the current FEC block. It is up to the receiver whether to synchronize to the PN sequence per FEC block, per burst/frame or keep synchronization between frames. (PN sequence TBD)
    - Limitation: The transmitter will ONLY SUPPORT CC (convolutional coding) and CTC (convolutional turbo codes).
    - Limitation: The transmitter will ONLY SUPPORT burst sizes that results in a whole number of maximum size FEC coding blocks - for ½ rate QPSK multiples of 6 slots, for higher levels of coding multiples of fewer slots.

5. Outside of the encoded burst the transmitter will support sending nothing, random symbol data values or pilots programmed per subcarrier/symbol. This can be used to simulate data around the fully encoded burst from the same or another transmitter module.

6. The transmitter will support sending of preamble on downlink.

7. The pilot modulation values are defined by a PN sequence. In the downlink this sequence repeats from frame to frame and is fully supported and programmable. In the uplink the initialization of the PN sequence depends on the 4 LSB of the frame number. Two alternatives will be supported:
    - A PN sequence that repeats from frame to frame as in downlink. This is a slight break of the standard but eliminates the need for communication of frame number to the receiver and the need for the receiver to do frame number dependant demodulation.
    - PN sequence initialization based on frame number (4 bits). As the frame number is normally transferred in the downlink FCH which is not implemented another solution must be found. We can convey the frame number to the receiver as a second *frame start pulse* that appears every 16-th frame when the frame number is zero (lower priority in implementation).

8. The transmitter architecture is designed to support any pilot placement pattern, not just those specified in the WIMAX standards. The architecture is also designed to support any channel allocation and logical to physical channel permutation within or outside the standards as long as they repeat from frame to frame and match the limitations imposed by the coding algorithms.

9. The transmitter will support the MIMO variants with two transmit antennas. Two instances of the transmitter are synchronized and generate signals for one antenna each.

- No MIMO related feedback from the receiver will be supported. MIMO configuration that could have been a result of feedback in a real system can be configured statically before a test (for beamforming etc).

10. The transmitter may if the receiver requires it deliver a synchronization *frame start pulse* to the receiver once in each frame. This pulse can be dynamically programmed to appear anywhere in the frame. The placing of the pulse can be arranged to compensate for delays in the fading simulator, to simulate synchronization mismatch, to start sampling in the receiver etc. Physical interface needs to be defined but probably a twisted pair wire will be used.

11. The transmitter is a physical layer simulator only. It will not transmit valid content in the FCH and DL-MAP fields. Such information must be delivered to the receiver offline.

3.1.4 Details of the Functional Architecture

The test transmitter is designed with a data flow architecture where there is one single data flow common for WiMAX and filterbank transmission. Changes between WiMAX and filterbank modulation which in some cases can occur dynamically are handled by changing operating mode within the modules in the data flow. In most modules WiMAX can be considered as a subset of filterbank.

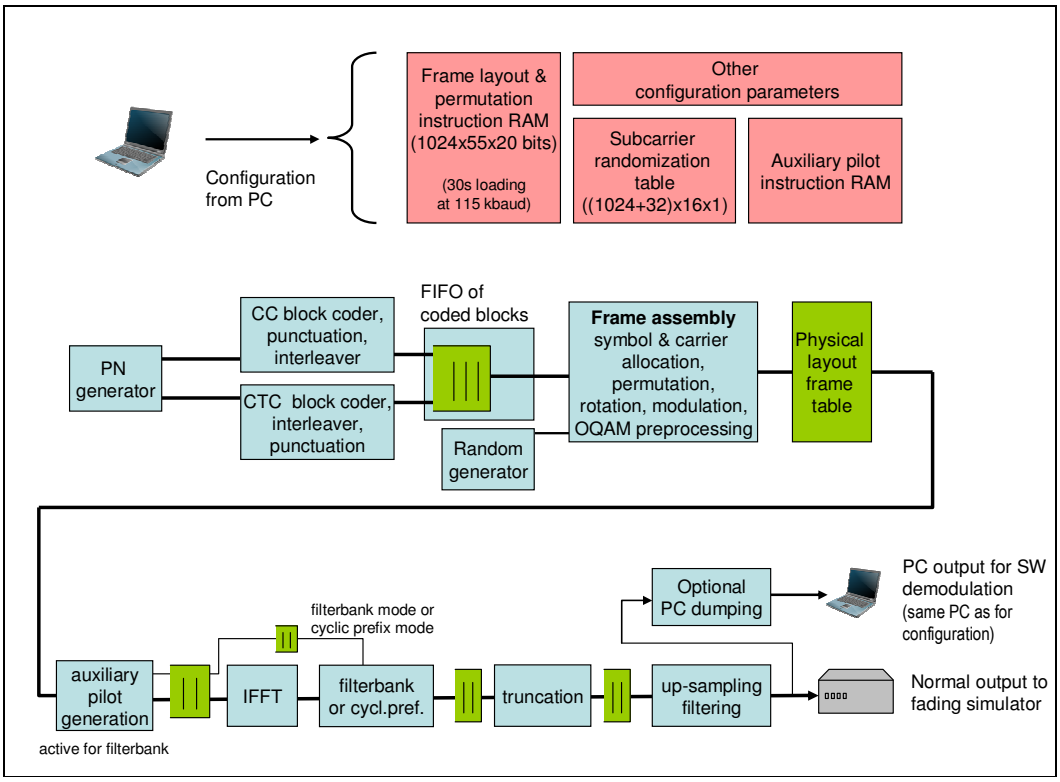

Figure 38: Functional Architecture of the Transmitter

Some modules are block oriented while others operate on a sample by sample basis. The modules are interconnected by FIFOs and RAM blocks with data handshaking. All modules operate on an internal FPGA clock (250 MHz) and can produce data at a higher rate than the

nominal. The data rate is constrained at the output end by the reading clock from the fading simulator.
In the following chapters the modules are described in more detail.

3.1.4.1 Data Generation

The data to be sent in the encoded bursts is generated using a PN generator. The polynomial can be selected at compile time and must be agreed upon.

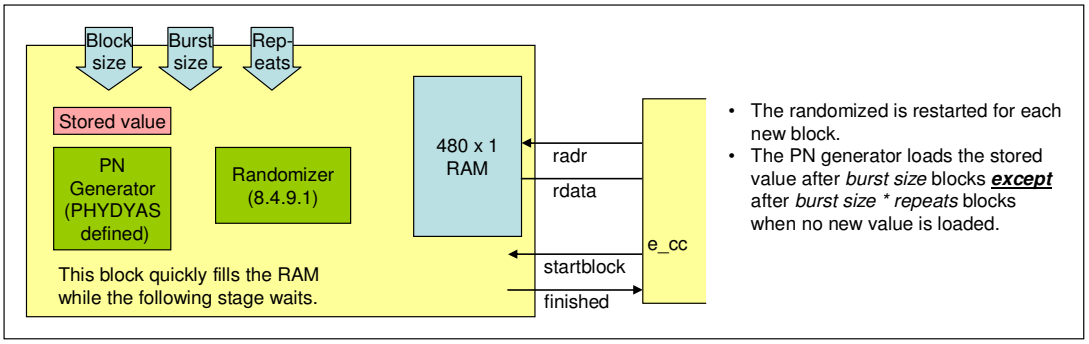

Figure 39: Data Generator and Randomizer

Normally the PN sequence runs continuously from transmitter start. The transmitter can be configured to repeat the data content of a frame several times before advancing the PN generator further. This feature can be used to simulate the effects of HARC.
After the PN generator follows the randomizer that operates according to the WiMAX standard.

Limitations

- A burst can only consist of a number of complete blocks of the same size.
- Only one burst with fully encoded random data can be sent in a frame.

3.1.4.2 CC Encoding and Interleaving

This module reads a randomized data block from the previous stage and then performs CC encoding, puncturing and interleaving before writing the data to a FIFO in the next block in the pipeline.

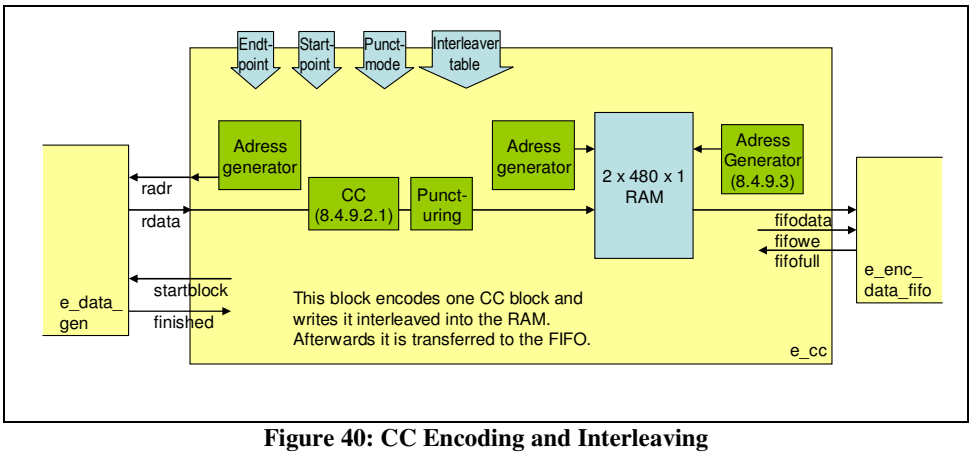

Figure 40: CC Encoding and Interleaving

The module tries to keep the output FIFO full by cycling through these steps:

- Command the preceding stage to generate a block of data in its output RAM and wait until it has finished.
- Read the data from the RAM while doing CC encoding and puncturing. The output data is written sequentially to the internal RAM. Because of "tail-biting" in the CC encoding some data is read twice from the preceding stage.
- Read the data in interleaved order from the internal RAM while writing it to the output FIFO. Wait until there is space in the FIFO for each bit written.

Limitations

- All blocks must be of the same size using the same puncturing and interleaving table.

3.1.4.3 Frame Assembly and Modulation

This module operates like a programmable computer. It reads instructions from an instruction RAM that is prepared offline and loaded before the transmitter is started. Most instructions generate one encoded symbol in the output RAM.

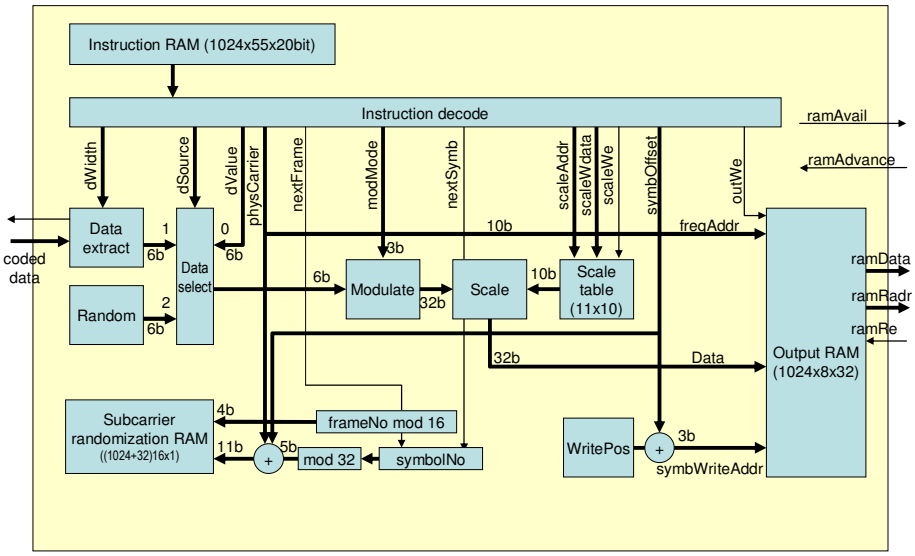

Figure 41: Frame Assembly and Modulation

Typically the processing of one instruction:

- selects data from either the stream of encoded bursts, a random generator or from the instruction itself;
- generates a modulated symbol according to the modulation method specified in the instruction (BPSK, QPSK, …);
- scales the amplitude of the modulated symbol;
- if filterbank mode converts the symbol to two OQAM symbols;
- writes the output symbol(s) to its permutated position in the output RAM.

The instructions are processed in logical frame ordering, frequency first while data is written to the output RAM in permutated order. The permutation position is part of the instruction. There is one exception to this rule. Since input data is read in encoded block order while blocks can span up to three symbols in time up to three output symbols may be generated at the same time.

3.1.4.4 Auxiliary Pilot Generation

This module changes value of auxiliary pilots based on the surrounding sample values. In WiMAX mode it simply transfers samples from its input RAM to its output queue. Deliverable 2.1 describes two sizes of the window affecting each auxiliary pilot (11 and 17 samples). The transmitter will support both, but only without recursive calculation. This means that in uplink PUSC only the smallest window can be implemented.

Implementation:

1. This module reads OQAM values from frame assembly output RAM. It must do addressing since it needs to read the same samples several times and in out-of-sequence order.
2. Weighting coefficients for auxiliary pilot calculation are stored in a table. The surrounding samples are always read in the same order, so this table is always accessed in the same order (0:upper).
   If the weighting values are all 0, no calculation on the pilots is done.
3. The algorithm is implemented in a simple processor that can handle both WiMAX and filterbank operation with the following instructions:

NEXT_INPUT_SYMBOL	Step to next OQAM symbol. Implies handshake with preceding and following processing step. Clears subcarrier counter.
ZERO	Pass a zero sample instead of current subcarrier to the output queue. Advance subcarrier counter.
PASS_FILTERBANK	Pass current subcarrier to output queue unmodified. Advance subcarrier counter.
PASS_WIMAX	Pass current subcarrier to output queue unmodified. Advance subcarrier counter two steps (skip the second OQAM sample).
CLEAR_ALU	Clears accumulator and resets coefficient address counter.
SCALE OFFSET_F, OFFSET_T OFFSET_F: -1,0,1 OFFSET_T: -4,-3,-2,-1,0,1,2,3,4	Reads sample with offset from current subcarrier in time and frequency direction. Scales this value with coefficient from coefficient table. Adds result to accumulator. Advances coefficient address counter.
SCALE_ZERO	Same as Scale but uses zero instead of reading a sample. Used when SCALE would read a value outside allowed frame area.
PASS_RESULT	Pass result from ALU accumulator to output queue. Advance subcarrier counter. Do CLEAR_ALU.
MODE_CHANGE WiMAX/filterbank	Changes the mode output flag which controls the operating mode of the filterbank filter.
NEXT_FRAME	Restart program.

4. The module runs a program specific for the current scenario and “knows” the type of the current sample by counting symbols/samples from the beginning of time.

5. RAM reading takes many clocks. Instruction execution must therefore be pipelined.
6. Program size equal to frame size. Overhead due to extra instructions for each contributing sample to each auxiliary pilot.

Note also that we can bypass this entire module so that there's no latency introduced if there's no calculation on the pilots needed.

3.1.4.5 IFFT

The IFFT is based on FFT (actually inverse FFT) IP cores from the Xilinx core generator.

The filterbank version of the FFT is identical to the WiMAX version except that it has to run twice as fast.

3.1.4.6 Filterbank Filter and Cyclic Prefix Insertion

The filterbank filter needs to process 1024 channels in parallel. The data rate is rather low however so one multiply/add module can process all 1024 channels serially. A special multichannel filter will be designed in VHDL for this filtering. The figure above shows a preliminary sketch of the implementation.

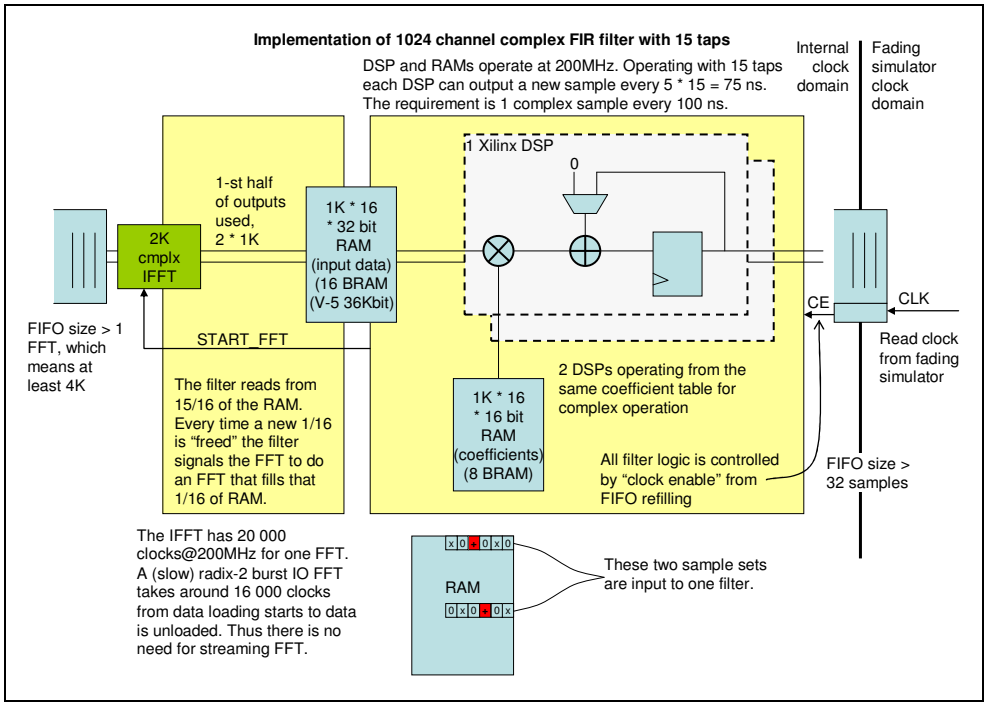

Figure 42: Multichannel Filterbank Filter

In WiMAX mode the operation of the filter is slightly changed so that data passes through unfiltered and cyclic prefix is inserted by a change of the read order from the internal RAM.

3.1.4.7 Tile/Burst Weighting/Truncation (Filterbank only)

This module allows weighting/truncation (fading in and out) of each tile or burst. It is applied when tiles from different transmitters are concatenated without gap in the time direction. In

this case the outermost half-symbols contain only auxiliary pilots (no data) and the truncating reduces overlap problems at the border. It can also truncate and/or fade the beginning and end of longer bursts.

Implementation:

1. Works on sequential sample stream.
2. Runs a program specific for the current scenario. Detects borders by counting samples from time zero.
3. Fading curve stored as table common to pre- and post-tale.
4. Simple programmable fading processor with the following instructions:

DISCARD N	Discard N samples (do not pass them to output queue).
ZEROES N	Pass N samples with scaling factor "0".
ONES N	Pass N samples with scaling factor "1".
UPSCALE N	Pass N samples with scaling factors from "scaling_table(0:N)".
DOWNSCALE N	Pass N samples with scaling factors from "scaling_table(N:0)".
LOOP	Restart program.

3.1.4.8 Upsampling, Filtering and Interpolation

The generated data stream is up-sampled by a factor of four to a mean sample rate of 89.62048 MHz. The fading emulator is programmed to generate a reading clock of this frequency and reads samples continuously from the output FIFO. The processing chain produces data non-continuously at a higher rate constrained by the internal operating clock and algorithmic delays.

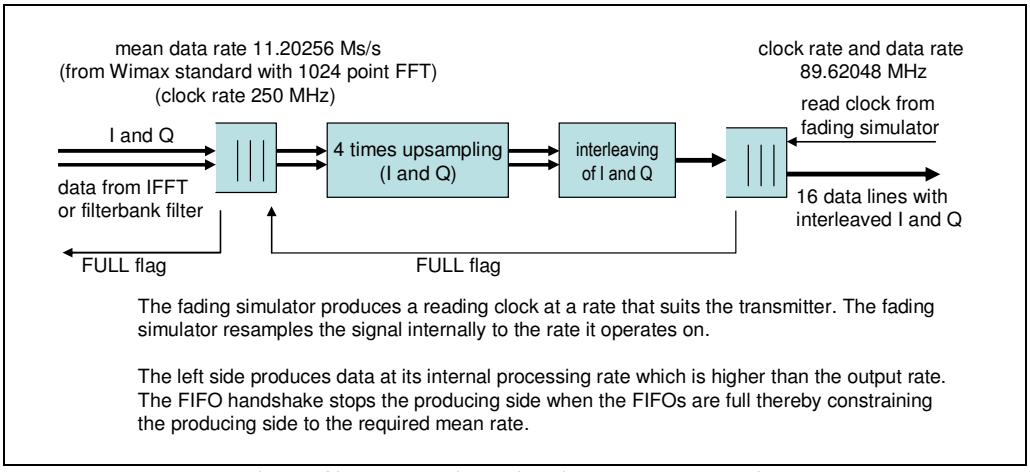

Figure 43: Upsampling, Filtering and Interpolation

3.2 Channel Emulator

Multiple Input Multiple Output (MIMO) technology holds the promise of higher data rates with increased spectral efficiency. Due to the potential improvement in system performance and advances in digital signal processing, many wireless systems, including IEEE 802.16e-based Mobile WiMAX™, have adopted the use of MIMO and multiple antenna technologies. These commercial wireless systems operate in high multipath environments and it is the benefit of multipath that provides the performance improvement when using multiple antenna configurations. While MIMO offers the potential for increased signal robustness and capacity improvement when operating in rich multipath environments, developing and testing MIMO components and systems requires advanced channel emulation tools that are easily configured and provide an accurate representation of realistic wireless channels and conditions.
This chapter describes how these complex channels can be emulated using commercially available instrumentation such as the Agilent N5106A PXB MIMO Receiver Tester, which will be referred to as the PXB.

3.2.1 Channel Correlation effects on MIMO Performance

For wireless communication systems, the wireless channel is the key factor that determines system performance. Channel effects, such as path loss and multipath fading, result in the attenuation of the signal amplitude at the receiver. Multipath may also induce inter-symbol interference if the delay spread is longer than the cyclic prefix in an OFDM signal. Spatial diversity and spatial multiplexing have been shown, both theoretically and experimentally, to substantially improve performance and overcome the undesired effects of multipath but only if the spatial dimension is properly configured to leverage the richness of the multipath environment.
Diversity gain achievable using Space Time Coding is dependent on the channel diversity order. Only when the channels between each transmit-receive antenna pair fade independently will the channel diversity order be equal to the product of the number of transmit and receive antennas. Alternately, if the channels between transmit-receive antenna pairs are highly correlated, then the achievable diversity gain is very limited. Low correlation channels are also required in spatial multiplexing MIMO applications. The different spatial signal streams can be well separated only under favorable channel conditions. This often requires proper positioning of transmit and receive antennas in order to provide low channel-to-channel correlations between the antenna pairs.
As a measurement example, Figure 44 shows the 2x2 MIMO channel coefficients, $h00$, $h10$, $h01$, and $h11$, for two different fading channels, one with relatively high channel-to-channel correlations and the other with low correlations. These measurements were made using an Agilent dual-channel 89600-series vector signal analyzer (VSA) on a WiMAX OFDMA signal that was faded using the PXB. The plot on the upper left shows the four channel coefficients as a function of subcarrier frequency for the high correlation case. It can be observed that the magnitude of the coefficients have a similar frequency response resulting from the high degree of correlation between some of the paths. The lower plot displays the measured constellation for the demodulated symbols which shows a high level of signal corruption. As a comparison, the figure on the upper right shows the coefficients for low channel-to-channel correlations. In this case, the frequency responses of the coefficients are dissimilar, resulting in an improvement in the MIMO symbol recovery, as shown by the measured constellation in the lower right of

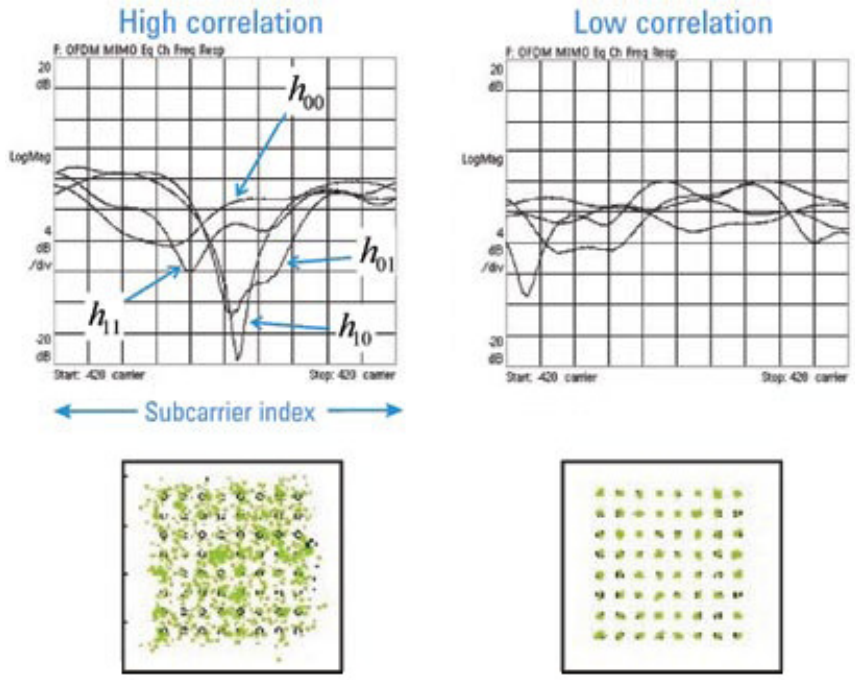

Figure 44: 2x2 MIMO Channel Coefficients and Constellations

3.2.2 Challenges in Emulating MIMO Channels

Testing MIMO receivers and systems under realistic channel environments can often be challenging due to the large number of transmit-receive channel combinations. For example, in a 2x2 MIMO configuration, using two separate SISO channel emulators is not adequate to model the four separate channels that exist between the pairs of transmit and receive antennas. In addition, SISO channel emulators do not provide any correlation between channels, which was previously shown to be an important characteristic when testing system performance. Testing directly in a "real" wireless environment is not an effective method, especially during the design and validation stages, as the channel is very sensitive, not controllable, and not repeatable. Also, testing in a real channel is not practical when different environments are required and when mobility testing is also necessary.

Creating realistic MIMO channels using software tools is another option but is often time-consuming and produces results that are not real-time. For example, after creating the channel fading coefficients in software, the convolution of these coefficients with the transmitted signals is a relatively long process preventing real-time performance. In some types of software-based test systems, the modulated data and faded signals are used to create complex I/Q waveforms. These waveforms are downloaded into the memory of an arbitrary waveform generator (ARB) for playback. There are many software tools that can accelerate the creation of faded waveforms, such as Agilent Signal Studio, Mathworks MATLAB™ and Agilent Advanced Design System (ADS), but these tools are often limited to traditional fading models. In addition, the arbitrary waveform generators have limited playback memory resulting in relatively short waveforms that repeat over time. Therefore, specialized instrumentation that emulates realistic MIMO channels provides the best solution for these challenging test conditions.

A channel emulator, such as the PXB, that replicates real-world MIMO conditions using powerful digital signal processing technology will provide a quick path for troubleshooting

advanced radio components and systems. The channel emulator also has the advantages that it can generate realistic fading scenarios including path and channel correlations.
The PXB provides up to 8 faders useful for testing and troubleshooting up to 4x2 MIMO systems. Figure 45 shows a simplified configuration diagram for testing a 2x2 MIMO transmitter and receiver using the PXB. It's internal faders can be independently configured with a standards-compliant fading model, such as a WiMAX ITU Pedestrian B, or custom configured model using a variety of path and fading conditions.

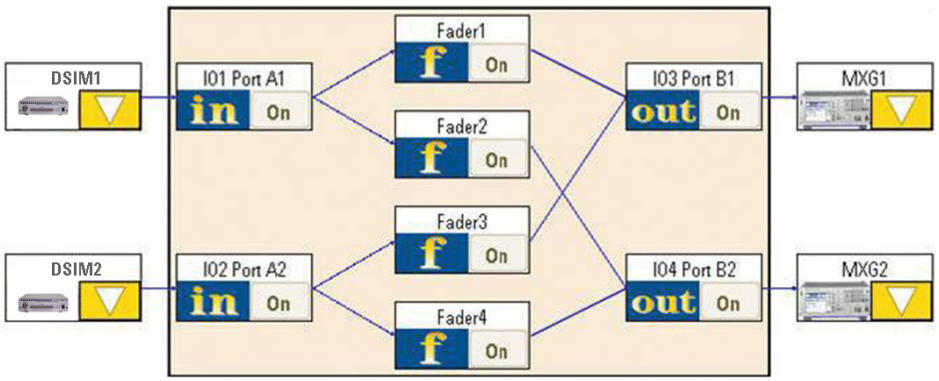

Figure 45: PXB 2x2 MIMO Block diagram

3.2.3 Channel Variations

A signal propagating through a wireless channel arrives at the destination along a number of different paths, referred to as multipath. These paths arise from scattering, reflection and diffraction of the radiated energy by objects in the environment or refraction in the medium. Variations in the received signal power are due to several effects: mean propagation (path) loss, macroscopic (large scale or "slow") fading and microscopic (small scale or "fast") fading, time-varying fading due to Doppler
The mean propagation loss is range dependent and results from absorption by water and foliage and the effect of ground reflection. Macroscopic fading results from the shadowing effect by buildings and natural features. Microscopic fading results from the constructive and destructive combination of multipath and is also known as fast fading since amplitude fluctuations are rapid when compared to macroscopic fading.

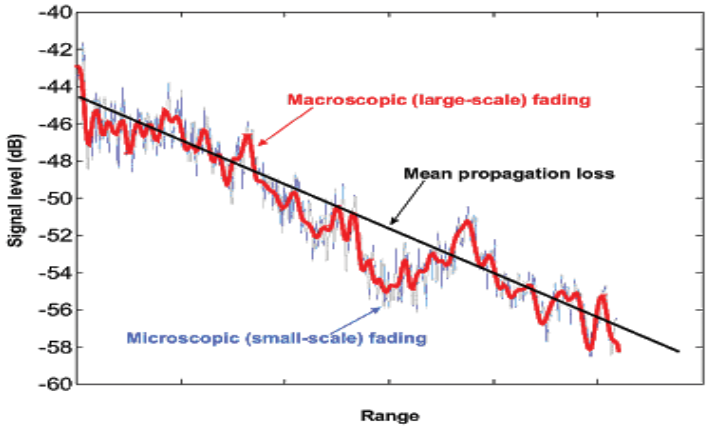

Figure 46: Signal Power Fluctuation vss. Range

3.2.3.1 Mean Propagation Loss

The overall mean loss in signal strength as a function of distance will follow a $1/d^n$ law, where d is the distance between the transmitter and the receiver and n is the slope index ranging from a value of 2 to 6 depending on the environment. For example, in free space, n = 2 resulting in a 20 dB/decade slope. In a terrestrial environment, a typical value of n = 4 results in a 40 dB/decade signal loss as a function of distance. In this terrestrial setting, changing the distance from 100m to 1000m (one decade) would result in an average signal drop of 40 dB.

3.2.3.2 Macroscopic (slow) Fading

Macroscopic or slow fading is determined by a local mean of the received signal over a distance of approximately 20 wavelengths. The macroscopic fading distribution is influenced by antenna heights, the operating frequency and the specific type of environment.

$$f(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-(x-\mu)^2/\sigma^2}$$

In the above equation, x (in dB), is a random variable representing the large scale signal power level fluctuation. The variables, μ and σ, are the mean and standard deviation of x, respectively. Both μ and σ are expressed in dB. The mean value, μ, is equal to the mean propagation loss discussed in the previously. The standard deviation, σ, may have values as high as 8 dB for some urban environments.

3.2.3.3 Microscopic (fast) Fading

Rapid changes in received signal strength may occur when the distance is varied by approximately one-half wavelength, thus giving this characteristic the name “fast” fading. When examining the fading statistics in the received power over a relatively short distance of approximately 20 wavelengths, the in-phase (I) and quadrature (Q) components of the superimposed signal can be modeled as an independent zero-mean Gaussian process. This model assumes that the number of scattered components is very large and independent. The voltage amplitude envelope of this received signal would then have a Rayleigh distribution with a PDF given by

$$f(x) = \begin{cases} \frac{x}{\sigma^2} e^{-x^2/2\sigma^2} & x \geq 0 \\ 0 & x < 0 \end{cases}$$

where x is a random variable taken here as the received voltage amplitude and σ is the standard deviation. A similar response would also be found for a stationary subscriber as a function of time due to the relative motion of scatterers in the local vicinity of the subscriber. The relative change in power level between a peak to null is typically 15-20 dB but can be as high as 50 dB under some channel conditions.

3.2.3.4 Doppler Spectrum

Another type of variation is due to scatter or the relative motion between transmitter and receiver. This time-varying fading results in a spread in the frequency domain response or Doppler spectrum. The maximum Doppler frequency, $f_{d,max}$, is related to the relative velocity by the following equation.

$$f_{d,\max} = f_c \frac{v}{c}$$

where v is the velocity of the mobile, f_c is the carrier frequency (Hz) and c is the constant for the speed of light. The spectral spreading of the pure tone would cover the range of $f_c \pm f_{d,max}$. Assuming uniformly distributed scattering around a mobile terminal, there is an equal probability that the multipath signal is received with an arrival angle anywhere within the range from 0 to 360 degrees. In this case, the theoretical Rayleigh Doppler power spectrum would exhibit the classical "U-shape".
Rician fading is formed by the sum of a Rayleigh distributed signal and a LOS signal.

The PXB provides several types of selectable spectrum shapes for accurate modeling of various multipath channels. Figure 47 shows the 4 most commonly used models.

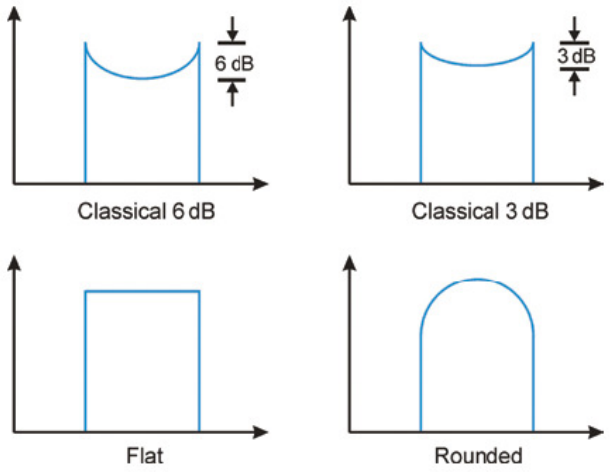

Figure 47: Fading Power Spectrum Shapes

3.2.3.5 Angle Spread

The types of variation mentioned above can accurately represent the multipath effects in a SISO system, but the shortcoming of these models is that they don't include spatial effects introduced by antenna position, antenna polarization, angle of departure (AoD) and angle of arrival (AoA).
If for example 2 transmit antennas are very close to one another, then the AoA and AoD would be very similar and a high fading correlation may exist between the antenna pairs, reducing the MIMO performance. Therefore it is important for any MIMO channel emulator to include a model for spatial effects and resulting channel correlations for antenna pairs.

Rather than modeling each AoD and AoA, a rich multipath environment can be achieved by including the spread of the AoDs and AoAs referred to as “angle spread”.
The angle spread for a typical Base Station (BS) is very narrow due to the fact that most scatterers are positioned far from the BS antennas. In contrast, the Mobile Station (MS) contains a large number of local scatterers surrounding the MS and thus resulting in a very wide angle spread, shown in . If the BS antennas are placed physically close together, the narrow angle spread will result in high channel correlation.
Fortunately, a BS often has the area to place its antennas far apart reducing the channel correlations. For MS with large angle spread, the antennas could be placed closer together while maintaining low channel correlations.

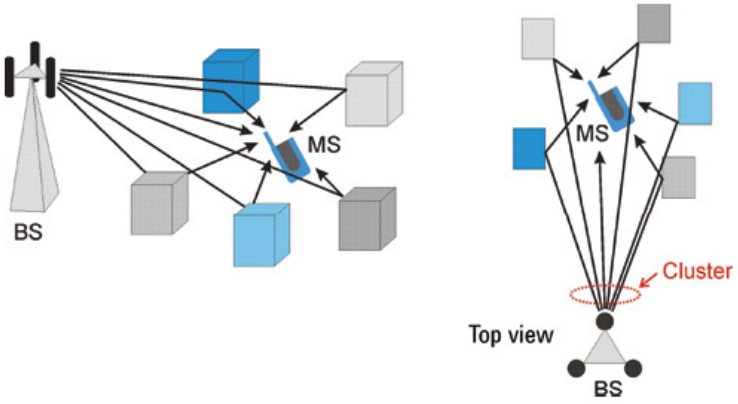

Figure 48: Angle Spread as Function of Antenna Placement

The PXB can be used to define the cluster angles at the transmitter and the receiver for each active path in the MIMO channel model. With the same ease, antenna gain, polarization and spacing can be entered into the PXB as well.
Figure 49 shows the PXB user interface for entering spatial parameters including antenna type and spacing.

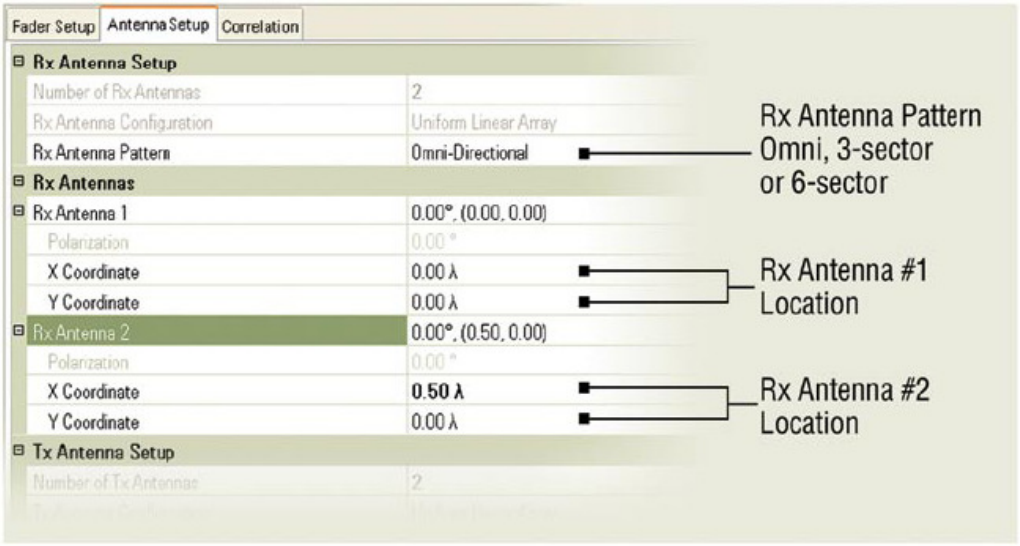

Figure 49: PXB Antenna Parameter Setup Screen

Configuring custom correlation matrices targeted for a specific application can be a lengthy process. Fortunately, the PXB provides a set of pre-defined MIMO channel models based on the specifications of several wireless standards including Mobile WiMAX and LTE. The instrument is configured using a simple menu structure for selecting the channel model. In addition, custom definitions may be created and saved.

3.3 Receiver

In this chapter, an overview of the proposed Phydyas physical layer concept demonstrator in the context of CTTC contribution to work package WP9 is presented. The aim of this document is to specify the physical architecture and the interfaces that have been chosen to implement the test receiver.

The major objectives of the work package are the development of a flexible prototype providing a rapid testing environment for Phydyas physical layer concept. In order to devise the most suited design, we have first delved into the following:

- Study of the computational complexity of the signal processing algorithms defining the physical layer of the Phydyas proposal.
- Evaluation of reusability of CTTC GEDOMIS® testbed for MIMO-OFDM wireless transmissions that are compliant to the standard Worldwide Interoperability for Microwave Access (WIMAX).

3.3.1 Preliminary Requirement Analysis

Since the algorithms that will run on the prototype will be based on the Matlab version of the simulator, we first have analysed the computational complexity. Specifically we have made a rough calculation of the number of sums and multiplications and the number of load and store operations. The results lead us to state that platforms with embedded DSP could not work in real-time. Moreover, benefiting from our experience in the CTTC GEDOMIS® testbed, it is also hard to state that real-time could be attained with embedded FPGA platforms. Since the aim is to focus on proving the concept, real-time will be relaxed to near real-time. Moreover, taking into account that the prototype has to be as much flexible as possible, hardware approaches have also been discarded in favour of software-based solutions.

Therefore, keeping in mind the previous limitations, the solution will be based on a software defined radio (SDR) based platform. In SDR based concept demonstrators, the idea is to integrate in software as many of the radio functionalities as possible, leaving a reduced hardware platform the role of controlling the remaining radio features. For current literature on this approach, please see, e.g., [9], [10], [11].

3.3.2 Proposed Demonstration Platform

We propose the development of a flexible demonstration platform for wireless research and experimentation which uses real channels, and which is highly customizable, i.e., it can be easily tailored to the Phydyas concept for demonstration purposes. Essentially, the demonstrator consists of a flexible RF front-end and a general purpose machine with a software based PHY implementation. In particular, the hardware subsystem is based on the Universal Software Radio Peripheral (USRP) board by Ettus Research LLC, attached via a USB link to a General Purpose x86 Processor (GPP). Furthermore, the Phydyas PHY layer concept is implemented in C++ and wrapped in GNU Radio on GNU/Linux over the GPP (the implementation of the Link layer or MAC layer is not considered). More specifically, a high-level implementation of the PHY algorithms is achieved via a streaming processing approach: blocks defined by Phydyas signal processing algorithms are regarded as nodes connected in a data stream flow graph. Moreover, Simulink-like model-based design and visualization capabilities can be provided by using Python with the SWIG wrapper.

There exists a vast literature on related SDR projects following the previous conception (for a published example, see for instance [12].

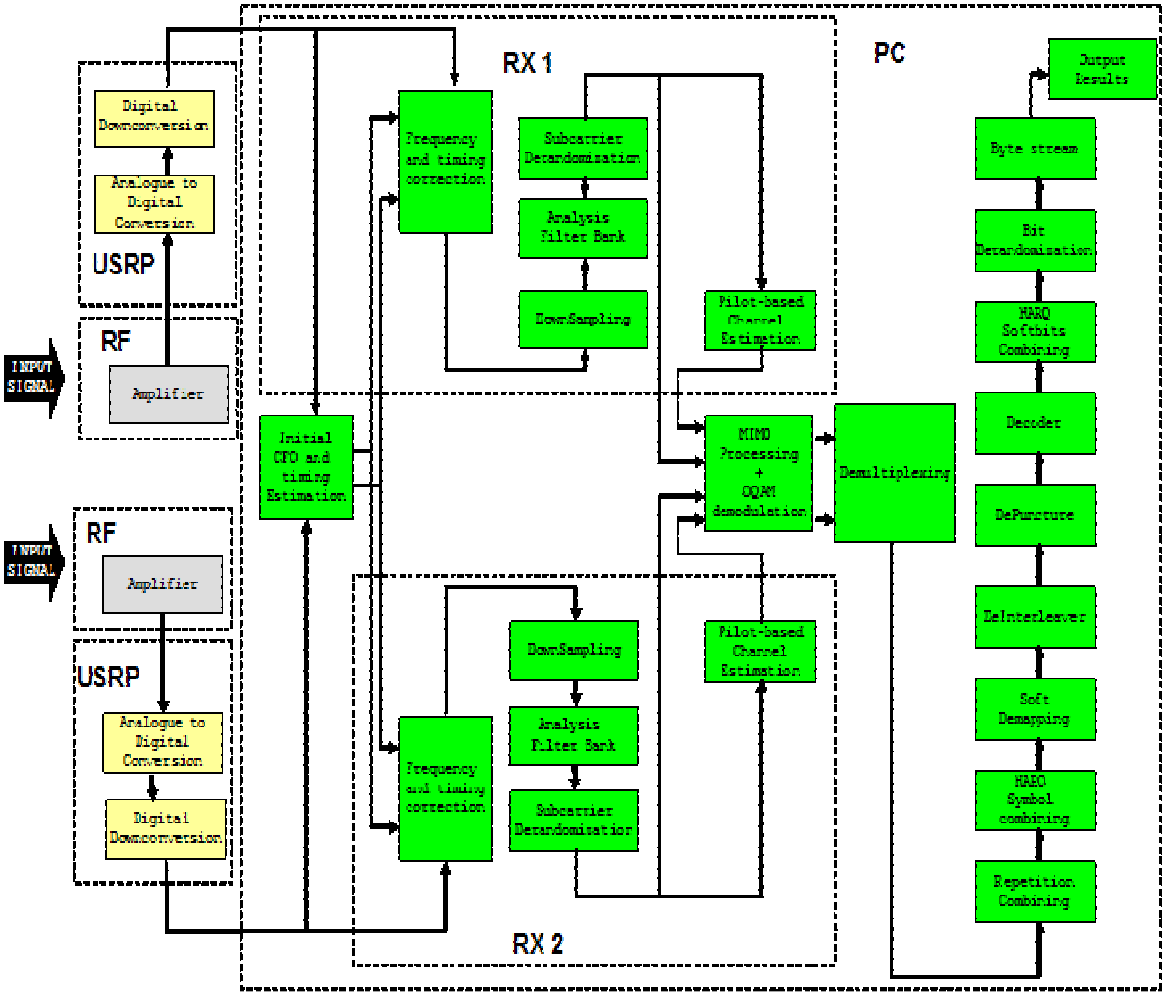

Figure 50: Receiver Diagram

3.3.3 System Architecture

An SDR based demonstration platform has been chosen to develop the Phydyas concept prototype. According to this approach, hardware problems are turned into software problems, making the front-end as general as possible and delegating to the software subsystem the implementation of the actual radio functionalities. Although there are several SDR solutions available on the market, we have focused on the solution offered by Ettus Research LLC [13] which is very suited due to its flexible and entirely open source design, as it works within GNU Radio framework [14].

In order to gain some insight into the proposed approach, we have divided the architecture into two subsystems: the hardware-defined subsystem and the software-defined subsystem. In Figure 2 we have depicted the block diagram of the receiver to clarify which elements belong to each subsystem. In particular, this figure corresponds exactly with the solution that will be used as a MIMO 2x2 receiver.

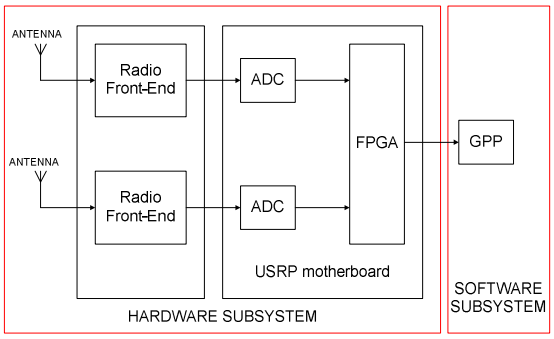

Figure 51: Block Diagram of the SDR Platform

3.3.3.1 Sampling Requirements

Based on the bandwidth and the frequency specifications of the OFDMA signal as described in chapter 3.5, the key parameters for sampling the signal without loss are identified.

RF Frequency band	2.5 GHz
RF Bandwidth	10 MHz
Baseband Bandwidth	5 MHz
Number of antennas	2

Table 11: Bandwidth and Frequency Specifications

Since the hardware platform USRP is expansion card based, two sampling techniques are available:

Baseband (BB) Sampling

In the BB sampling technique each antenna channel needs two analogue-to-digital converter (ADC) to perform the sampling operation of an Inphase (I) and Quadrature (Q) baseband signal, coming from the RF front-end. The down conversion stage, assuming a super heterodyne downconverter, need to obtain in the last step an analogue baseband signal with two local oscillator signals, 90 degrees phase shifted one to other.
The GPP processor expects to receive complex samples.

Intermediate Frequency (IF) Sampling

In the IF sampling on the other hand, the RF signal is downconverted to an intermediate frequency, obtaining a pass band analogue signal. Only one channel ADC is needed.
The GPP processor expects to receive real samples. Since the software defined receiver needs the BB signal, the conversion from IF to BB is performed on the GPP processor implementing a Digital Down Converter (DDC) stage.

Resources (each antenna)	BB Sampling	IF Sampling
Number of ADC	2	1
Minimum Sampling frequency	10 MHz	(5 MHz + IF) x 2
Sample type	Complex	Real
Sample size (8 bits per sample)	16 bits	8 bits
Sample data rate for 2 antennas	20 Mbytes x 2	(5 Mbytes + IF) x 4

Table 12: Comparing BB and IF Sampling

3.3.3.2 USRP 1.0 Option

The USRP 1.0 platform has four 64 MSPS ADCs, and is designed to work using the BB sampling technique but is also capable of perform IF sampling.
The FPGA implements a DDC converter stage which has a decimation phase with a minimum decimation rate of 4. The maximum sampling rate obtained at the GPP processor side is 16 MSPS per ADC.
The samples are transferred to the GPP using the USB 2.0 bus. Since the USB has a stable transfer rate of 32 Mbytes, a limitation of the sampling rate is present.
Two antennas BB sampling will require decimating the digital sampled signal by 8, obtaining four channels of 8 MSPS per channel in order to not rise above the USB transfer rate bottleneck. According to the Nyquist criterion, the maximum available bandwidth in this configuration will be 4 MHz.

3.3.3.3 USRP 2.0 Option

The USRP 2.0 platform has two 100 MSPS ADCs, and is designed to work using the BB sampling technique for one antenna and is capable of perform IF sampling for two antennas.
A Xilinx Spartan 3 2000 FPGA implements a flexible DDC converter and gives enough space and processing power to add a custom firmware written in VHDL or Verilog. In addition, the USRP 2.0 platform has an onboard user RAM of 1 Mbyte.
The samples are transferred to the GPP using the Gigabit Ethernet bus, four times faster than USB 2.0 bus.
Taking into account that the receiver has two antennas, the BB sampling technique is not available without using two synchronized USRP 2.0 boards.
IF sampling technique is the most suitable sampling technique for the USRP hardware, including the possibility of perform the final conversion from IF to BB internally inside the FPGA.

3.3.3.4 USRP 1.0 and 2.0 Comparison

Key features of each option are gathered in the table below.

Parameter	USRP 1.0	USRP 2.0
Number of ADC	4	2
Sampling frequency	64 MHz	100 MHz
Maximum transfer rate to GPP	32 Mbytes	125 Mbytes
Maximum BB bandwidth on GPP	8 MHz (2 ADC) or 4 MHz (4 ADC)	50 MHz
FPGA	Altera cyclone	Xilinx Spartan 3
Available RAM on board	NONE	1 Megabyte
BB sampling	YES	NO
IF sampling	YES	YES

Table 13: Comparing USRP1.0 and USPR2.0

3.3.3.5 RF Frontend Requirements

Finally we will evaluate the constraints that the front end has to attain depending on the sampling rate technique. In the BB sampling, the two associated ADCs for each channel have to be fed with inphase and quadrature signals. As a result, the front end has to be able to provide baseband outputs.
In constrast, the requirements for the frontend when the IF sampling rate is considered are different. The signal provided is centered at IF instead of 0Hz, so the signals at the output are

pass band. Nevertheless, to fulfil the Nyquist criterion the IF values can not exceed the values gathered in the table below.

Parameter	USRP 1.0	USRP 2.0
Sampling frequency	64 MHz	100 MHz
Maximum IF for 10MHz bandwidth	27 MHz	45 MHz

Table 14: Maximum IF Values

3.3.4 Proposed Hardware Subsystem

According to the block diagram shown in Figure 51, the hardware-defined subsystem consists of two RF front-ends and one Universal Software Radio Peripheral, which is the motherboard. In the previous section we have tackle in depth the advantages and disadvantages of using either the USRP 1.0 or the USRP 2.0. Since most of the algorithms of the other workpackages have been simulated for 10MHz and 10.94KHz bandwidth and carrier spacing respectively, we have reach the conclusion that these values shouldn't be modified.
In consequence, USRP 1.0 and BB sampling have to be discarded in favour of USRP 2.0 and IF sampling.
Next it is explained with more detail the specific solutions we have been chosen to implement the hardware subsystem keeping in mind the values gathered in section 3.3.3

3.3.4.1 RF Front-end

The RF front-end is a radio frequency tuner system with programmable gain control that act as an intermediate frequency (IF) interface. In order to make a good use of the resources that CTTC have at their disposal, the proposed front-end to implement the test receiver is the ADV-3000T offered by Mercury Computer Systems [15]. The range of input RF frequencies is between 20MHz and 3.2GHz so that the 2.5GHz chosen in Phydyas are covered. The digital down conversion relies on a high performance digital tuning synthesizer, so that at the input of the ADCs we can have IF frequencies varying between 3MHz and 40MHz. Moreover, the IF filters allow to feed the ADCs of the mainboard with signals up to 65MHz of bandwidth. As a result, the 10MHz bandwidth proposed in Phydyas are met. More important is that all the signals downconverted are fully coherent so that MIMO processing is allowed.

3.3.4.2 Baseband Mainboard

As a mainboard we have leant towards the Universal Software Radio Peripheral (USRP2. The advantage of this mainboard is that general purpose processors (GPP) are turned into flexible SDR platforms. The USRP2 consists of two high-speed ADCs and one Xilinx Spartan 3-2000 FPGA. The ADCs are connected to the radio front-ends, while the FPGA is connected to the GPP by means of a Gigabit Ethernet Interface. The main principle behind the USRP2 is that the digital radio tasks are divided between the internal FPGA and the external host CPU. The high speed general purpose processing, like down conversion, decimation, and interpolation are performed in the FPGA, while demodulation, are performed at the GPP.
More details on the two main subsystems are given next.

ADC
The USRP contains four 14 bits ADCs. The sampling rate is 100 Msamples per second, therefore a pass band signal with bandwidth up to 50 MHz may be digitized. The board is also provided with a Programmable Gain Amplifier before the ADCs, in order to use the whole available input range in case of weak signals.

FPGA
The FPGA is connected to the ADCs and the Gigabit Ethernet chipset. In the receiving path, the analog signal is converted by the ADC in 14 bit samples and then passed to the FPGA for further processing. Here, the signal is decimated by a factor specified by the user and might be converted to the baseband. Data is then passed to the Gigabit Ethernet controller for transmission to the GPP.

3.3.4.3 Receiver Sensitivity

According to WIMAX standard **[15]**, the receiver sensitivity should be better than the values listed in Table 15. The minimum power is computed according to the required SNR which in turns is the minimum that allow achieving a BER of 10^{-3} for uncoded signals.

Modullations	Minimum power (dBm)	SNR (dB)	Noise Power (dBm)
QPSK	-83.2	9.8	-93
16-QAM	-76.2	16.8	-93
64-QAM	-70	23.0	-93

Table 15: Min. Received Power at 7dB Noise Figure

However, these results have been computed assuming 7dB and 3dB for the noise figure and the implementation loss respectively. Since the figure noise in our hardware subsystem is 10dB, the noise power will be higher. As a result, to achieve the same performance as the receiver assumed in Table 15, the minimum power has to be increased. In Table 16 are gathered the values that our proposed solution would have to outperform under the same conditions as in Table 15.

Modullations	Minimum power (dBm)	SNR (dB)	Noise Power (dBm)
QPSK	-80.2	9.8	-90
16-QAM	-73.2	16.8	-90
64-QAM	-67	23.0	-90

Table 16: Min. Received Power at 10dB Noise Figure

3.3.5 Software-defined Implementation

After the signal has been processed by the USRP FPGA, the stream of bits finally flows through the USB connection. As a result, the entire processing will run on a general purpose x86 processor with Linux operating system. Regarding the language programming, we have taken into account that baseband processing heavily relies on matrix algebra. Therefore, for the sake of an efficient algorithm implementation C++ has been chosen, as most of the high-quality blocks that we could benefit from are written in this language. Most of the signal processing can rely on the following libraries IT++ [17], GNU Radio [14], Boost [16] and SPUC [18]. Although they cover lots of classes and routines, 2-D interpolation is not considered. Efficient and appropriately documented routines written in C++ can be found in

the websites of the CGAL [19] and QHULL [20] libraries. By means of these routines, we can interpolate the channel estimates in the pilot positions to the rest of positions efficiently.

High-level implementation of the signal processing algorithms is achieved by resorting to a flow graph representation of the data transmission link. More specifically, the vertices of the flow graph are signal processing blocks written in C++, and the edges represent the data flow between them. Signal processing blocks can be either synchronous or asynchronous with different data formats (float, short, chars …). On the other hand, the complete radio implementation can also be coded using a scripting language like Python, such that a model based design similar to those obtained with Simulink is obtained by means of a graphical interface termed GNU Radio Companion.

3.4 MIMO Decoder

COMSIS engages to provide a 2x2 MIMO decoder for the use in both OFDM-based and FBMC-based MIMO system. We develop at the first step a 2x2 decoder, RxV1.2h MIMO decoder, which will be used in the OFDM-MIMO case. At present, most blocks have been developed and we are in phase of validation.
This chapter will give the descriptions of the computations performed by the RxV1.2h 2x2 MIMO decoder. This decoder realizes Maximum Likelihood Hard Output decoding for MIMO system with Rx=2 and both N_{SS} =1 and N_{SS}=2 are supported.

3.4.1 Decoder Architecture and Algorithm

The general architecture is illustrated below:

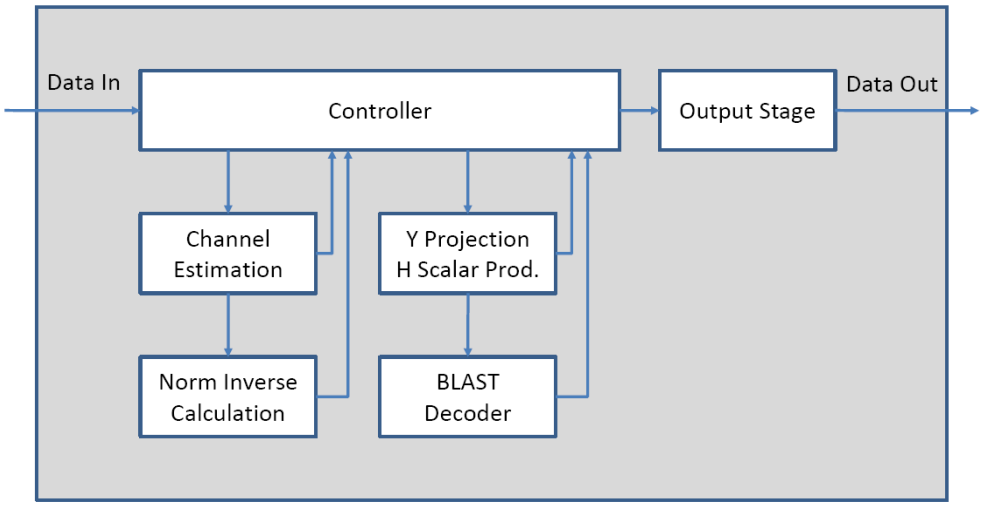

Figure 52: RxV1.2h MIMO Decoder Arcxhitecture

The data is processed by the Y Projection and H Scalar Product block, before being decoded. The controller block operates in two modes: for the data with N_{SS}=1, it hard-decodes the received data via the MRC (inside the controller); for the data with N_{SS}=2, the dedicated BLAST Decoder block will be used for ML decoding.

3.4.1.1 Channel Estimation

The channel estimation gives the channel matrix, denoted by H, which is a complex matrix defined as follows:
For N_{SS}=1,

$$H = \begin{bmatrix} h_{1,1} & 0 \\ h_{2,1} & 0 \end{bmatrix}$$

For N_{SS}=2,

$$H = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix}$$

Where $h_{i,j}$ is the complex channel coefficient at a given data subcarrier between the Tx antenna j and the Rx antenna i.

3.4.1.2 Data In

The data in represents the stream of received word, denoted by Y which is a complex vector of dimension 2:

$$Y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

Where y_i is the complex signal received by the Rx antenna i.

3.4.1.3 Norm Inverse Calculation

The Norm Inverse Calculation block receives the estimated 2 or 4 channel coefficients. From the channel matrix, both column squared-norms N_1 and N_2 are calculated and compared. If N_1 is greater than N_2, then the columns are swapped.

$$N_1 = |h_{11}|^2 + |h_{21}|^2 \qquad N_2 = |h_{21}|^2 + |h_{22}|^2$$

An output signal SWAP keeps track of whether the swap occurred or not: this signal is one bit variable defined as follow:

$$SWAP = \begin{cases} 0 & \text{if } N_2 > N_1 \\ 1 & \text{if } N_2 \le N_1 \end{cases}$$

The greater norm is then passed to the inverse calculator in order to get:

$$N_{max}^{-1} = \frac{1}{N_{2-SWAP}}$$

Both the norm inverse and the swap indicator are sent back to the controller. Note: Due to the precision issue caused by the division, the norm inverse format is floating point, with range limitation.

3.4.1.4 Controller

The Controller receives the coding mode, which takes value from 0 to 3, indicates which QAM constellation is used for decoding and the value of N_{BPSC} which is detailed in the following table:

Coding Mode	0	1	2	3
Constellation	BPSK	QPSK	16-QAM	64-QAM
N_{BPSC}	1	2	4	6

Table 17: Coding Mode and Constellation Correspondence

This value will be communicated with Output Stage and BLAST Decoder block.
The constellation normalization factor, denoted by K, depends on the coding mode. Controller block sends K and the inverse of the constellation normalization factor, denoted by K^{-1}, to the Y Projection and H Scalar Product block. K and K^{-1} are both real positive constants which are provided according to the following table.

Coding Mode	0	1	2	3
K	8/7	$2/\sqrt{2}$	$4/\sqrt{10}$	$8/\sqrt{42}$
K^{-1}	7/8	$\sqrt{2}/2$	$\sqrt{10}/4$	$\sqrt{42}/8$

Table 18: Constellation Normalization Factor

Note: A QAM symbols can be written s = K.x with -1<Re(x)<1 and -1<Im(x)<1.
This treatment simplifies the notation of Re(x) and Im(x) candidates. The possible values of Re(x) and Im(x) are shown in the following table as a function of coding mode/constellation.

Coding Mode	Constellation	Re(x)	Im(x)
0	BPSK	-7/8, 7/8	0
1	QPSK	-1/2, 1/2	-1/2, 1/2
2	16-QAM	-3/4, -1/4, 1/4, 3/4	-3/4, -1/4, 1/4, 3/4
3	64-QAM	-7/8, -5/8, -3/8, -1/8, 1/8, 3/8, 5/8, 7/8	-7/8, -5/8, -3/8, -1/8, 1/8, 3/8, 5/8, 7/8

Table 19: Coding Mode and QAM Values

3.4.1.5 Y Projection and H Scalar Product

The Y Projection block and H Scalar Product block use the following input signals: the received symbol Y, the corresponding channel coefficients H, the norm inverse N_{max}^{-1} , the inverse of the normalization factor K^{-1} and the SWAP parameter.
Three calculations are processed with this block:

1. z, the received symbol Y projected by the channel coefficients and corrected by the scaling factor K^{-1}. The calculation equation is described as below:

$$z = N_{max}^{-1} \cdot K^{-1} \cdot \left[h_{1,2-swap}^{*} \quad h_{2,2-swap}^{*} \right] Y$$

2. P, the scalar product of the channel coefficient matrix. The calculation equation is :

$$P = N_{max}^{-1} \cdot (h_{1,1+SWAP} \cdot h_{1,2-SWAP}^{*} + h_{2,1+SWAP} \cdot h_{2,2-SWAP}^{*})$$

   By construction, we have |P|<1; thus -1<Re(P)<1 and -1<Re(P)<1.

3. H', the channel coefficients scaled by factor K:

$$H' = K \cdot \begin{bmatrix} h_{1,1+SWAP} & h_{1,2-SWAP} \\ h_{2,1+SWAP} & h_{2,2-SWAP} \end{bmatrix}$$

z is sent to both Controller for MRC decoding and the BLAST Decoder block, while P and H' are sent only to the BLAST Decoder.

3.4.1.6 BLAST Decoder

The Blast decoder browses an entire space-time layer, deriving for each possible coordinate the closest corresponding coordinate in the second layer (by orthogonal projection). Then the Euclidean metric between received symbol and each computed point is calculated. By providing the candidate pair (x_1, x_2), this metric is defined as:

$$M(x_1, x_2) = \left| y_1 - h'_{1,1} x_1 - h'_{1,2} x_2 \right|^2 + \left| y_1 - h'_{2,1} x_1 - h'_{2,2} x_2 \right|^2$$

The point featuring the smallest metric of all is elected to be the decoded codeword.

3.4.1.7 Output Stage

The output module outputs N_{SS} x N_{BPSC} bits. These bits match with the Gray Code of the N_{SS} QAM symbols decoded by the RxV1.2h ML decoder. By convention, the first N_{BPSC} bits match with the QAM symbol of the first spatial stream.

3.4.1.8 Algorithm Overview

The whole decoding algorithm is composed of 2 parts: the channel processing and the received word processing.

1. Channel processing (when channel coefficients are ready):
    - Determine K and K^{-1};
    - Compute N_1 and N_2;
    - Determine SWAP;
    - Compute N_{max}^{-1};
    - If N_{SS}=1, compute P and H'.
2. Word processing (when data are ready)
    - Compute z;
    - If N_{SS}=1, round Re(z) and Im(z) to get the closest QAM point x and output the binary Gray code of x;
    - If N_{SS}=2, for all the possible value of x_1 in the QAM constellation:
        - Compute $t=z-P.x_1$;
        - Round Re(z) and Im(z) to get the closest QAM point x_2;
        - Compute the metric $M(x_1, x_2)$;
    - Select the couple (x_1, x_2) which possesses the minimum metric;
    - If SWAP, swap x_1 and x_2;
    - Output the binary Gray code of x_1 and then the binary Gray code of x_2.

3.5 Test Scenarios

For benchmark purposes as well as to gain acceptance within the wireless community, the concept demonstrator of the filter bank multicarrier modulation principle is based on the standard WIMAX. Since the uplink has highest priority in the implementation, only the uplink will be supported by the test receiver in an initial design stage. Next, after the communication uplink has been successfully tested, the prototype will be tentatively enhanced with downlink capabilities such that both uplink and downlink can be used indistinctly. It is worth mentioning that subcarrier permutation is different in each link. In the uplink, AMC23 will be considered, whereas PUSC will be the choice for the downlink. The set of physical layer functionalities that are to be considered are listed next:

- Timing and synchronization constitutes the first processing function which is to be performed in the receiver. Using the training sequence transmitted in the preamble, the receiver can determine when the frame starts and can correct the offsets in the carrier frequency [21].

- The analysis filter bank allows transforming the received time signals into the frequency domain [21].

- The channel estimation blocks provide the channel estimates required for the demodulation of the transmitted data [22].

- The MIMO processing module combines the received data in both antennas in order to exploit the diversity to increase the link reliability [23].

By the time this report has been written, preamble-based synchronization and MIMO decoding algorithms have not been yet chosen.

3.5.1 AMC Uplink Scenario

Parameters:

- Frequency band: 2.5GHz
- Bandwidth: 10MHz
- Frame length: 5ms
- Number of subcarriers (FFT size): 1024
- Carrier Spacing: 10.94KHz
- Encoding: Convolutional Coding, rate ½
- Modulation: QPSK
- Segmentation: No
- Number of symbols per frame: 47
- Uplink zone: Symbol numbers 35 to 46
- Allocation: 30 slots
- Allocation start: logical slot 20

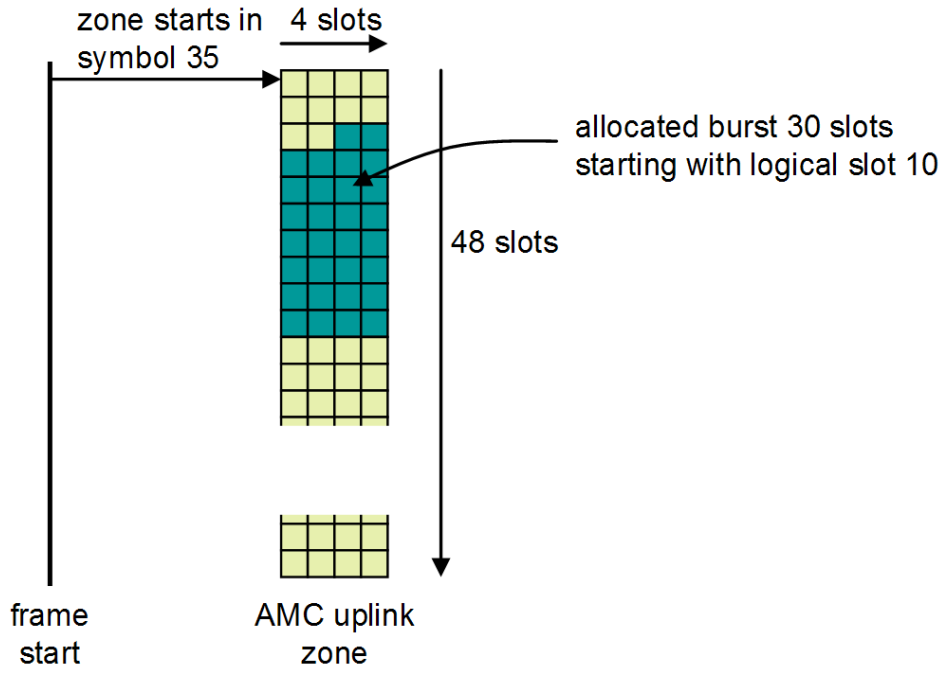

Figure 53: Uplink AMC Scenario

3.5.2 PUSC Downlink Scenario

Parameters:

- Subcarriers: 1024
- Encoding: Convolutional Coding, rate ½
- Modulation: QPSK
- Segmentation: No
- Preamble: Sequence 0
- Downlink zone: Symbol numbers 1 to 13
- Allocation: rectangular 4x3 slots (Freq. x Time)
- Allocation start: slot 3 (Freq.), slot 2 (Time)
- Data in surrounding subcarriers: Random

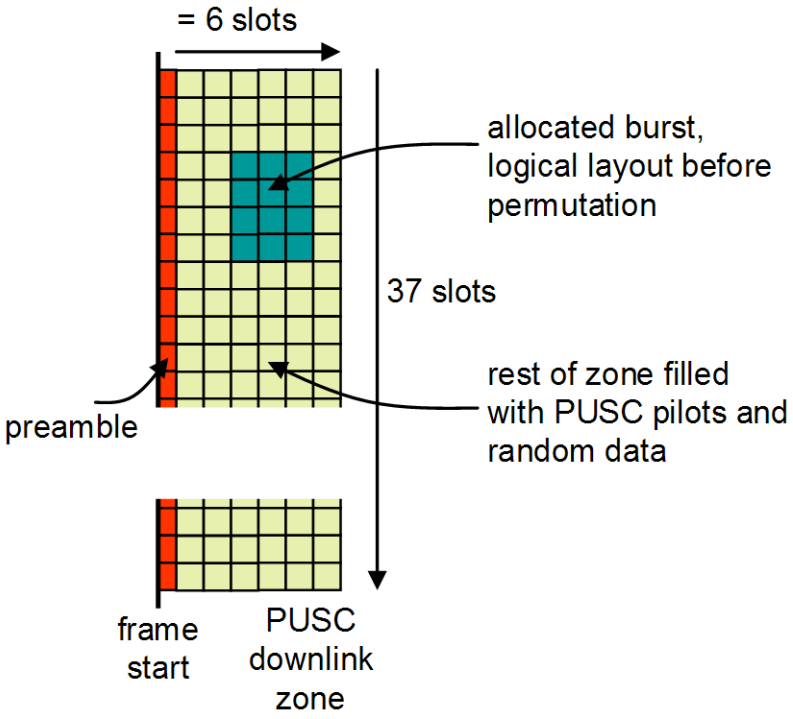

Figure 54: Downlink PUSC Scenario

4 References

[1] F. Schaich, "WiMAX FBMC-OFDM comparison scenarios", ICT-211887, PHYDYAS deliverable D9.1a, January 2009.

[2] IEEE 802.16e-2005, "Air Interface for Fixed and Mobile Broadband Wireless Access Systems", February 2006.

[3] J. Louveaux, L. Baltar, D. Waldhauser, M. Renfors, M. Tanda, C. Bader, E. Kofidis, "Equalization and demodulation in the receiver (single antenna)", ICT-211887 PHYDYAS deliverable D3.1.

[4] M. Tanda, T. Fusco, M. Renfors, J. Louveaux, M. Bellanger, "Data-aided synchronization and initialization (single antenne)", ICT-211887, PHYDYAS deliverable D2.1, July 2008.

[5] M. Tanda, M. Renfors, J. Louveaux, M. Bellanger, "Synchronization and initialization with single antenna. Blind Techniques.", ICT-211887, PHYDYAS deliverable D2.2, January 2009.

[6] M. Bellanger, "PHYDYAS reference filter bank," PHYDYAS internal report.

[7] IEEE 802.16e-2005, "Air Interface for Fixed and Mobile Broadband Wireless Access Systems", February 2006.

[8] WiMAX Forum, "Mobile System Profile Release 1.0 (Revision 1.6.1: 2008-04-01)", April 2008.

[9] M. Rupp, C. Mehlführer, S. Caban, R. Langwieser, L.W. Mayer, and A.L. Scholtz, Testbeds and Rapid Prototyping in Wireless System Design. *EURASIP Newsletter*, 17(3), pp. 32–50, September 2006.

[10] R. Schiphorst, F.W. Hoeksema, and C.H. Slump, A Real-Time GPP Software-Defined Radio Testbed for the Physical Layer of Wireless Standards, *EURASIP Journal on Applied Signal Processing* (16), 2005, pp. 2664–2672.

[11] R. Farrell, M. Sánchez, and G. Corley, Software-Defined Radio Demonstrators: An Example and Future Trends, *International Journal of Digital Multimedia Broadcasting*, 2009.

[12] K. Zheng, L. Huang, G. Li, H. Cao, W. Wang, and M. Dohler, Beyond 3G Evolution, *IEEE Vehicular Technology Magazine*, June 2008, pp. 30-36.

[13] Ettus Research LLC Website, http://www.ettus.com

[14] GNU Radio Official Website, http://gnuradio.org

[15] Mercury Computer Systems website. http://www.mc.com

[16] Boost Homepage, http://www.boost.org

[17] IT++ Homepage, http://itpp.sourceforge.net

[18] Signal processing using C++ Documentation Website, http://spuc.sourceforge.net

[19] CGAL Homepage, http://www.cgal.org

[20] QHULL Homepage, http://www.qhull.org

[21] ICT Phydyas Project Delivrable D2.2

[22] ICT Phydyas Project Delivrable D3.1

[23] ICT Phydyas Project Delivrable D4.1