UPPSALA
UNIVERSITET

UPTEC F 14017
Examensarbete 30 hp
Maj 2014

Improving efficiency of EMC Immunity Monitoring of RBS using a FPGA based instrument

Viktor Arfwedson

UPPSALA
UNIVERSITET

Teknisk- naturvetenskaplig fakultet
UTH-enheten

Besöksadress:
Ångströmlaboratoriet
Lägerhyddsvägen 1
Hus 4, Plan 0

Postadress:
Box 536
751 21 Uppsala

Telefon:
018 – 471 30 03

Telefax:
018 – 471 30 00

Hemsida:
http://www.teknat.uu.se/student

Abstract

Improving efficiency of EMC Immunity Monitoring of RBS using a FPGA based instrument

Viktor Arfwedson

A test system for Immunity testing of Ericsson's LTE radio base station (RBS) was built using a National Instruments (NI) PXI chassis fitted with a vector signal transceiver (VST) module and a PC controller module. A program made in LabVIEW and MATLAB was run on the controller module in the PXI chassis. The tasks of the program was, on one hand, to control the VST which was used to record the signal emitted from the RBS and on the other hand, to process the signal and determine its quality by acquiring the bits transmitted. Functionality enabling the VST to transmit a given signal was also included in the program. The built system performed up to ninety times faster than the old system but lacked turbo decoding necessary to correctly determine bit error ratio (BER) and block error ratio (BLER). The performance of the system leaves room for adding time consuming processes such as turbo decoding later on and by examining the undecoded bits the signal quality can still be measured. The program handles both 1tx and 2tx signals.

Handledare: Patrik Hellström
Ämnesgranskare: Mikael Sternad
Examinator: Tomas Nyberg
ISSN: 1401-5757, UPTEC F 14017

Improving efficiency of EMC Immunity Monitoring of RBS using a FPGA based instrument

Viktor Arfwedson

May 12, 2014

Special thanks to:

Patrik Hellström

Mikael Sternad

Peter Bengtsson

Örjan Eriksson

Sammanfattning

Ett testsystem för immunitetstestning av Ericssons LTE-radiobasstation (RBS) byggdes med ett PXI-chassi från National Instruments (NI) som bestyckades med en vector signal tranceiver (VST) modul och en PC-controller-modul. Ett program skapat i LabVIEW och MATLAB kördes på controller-modulen i PXI-chassit. Programmets uppgift var dels att kontrollera VST:n vilken användes för att spela in signalen som RBS:en sände ut och dels bearbeta den inspelade signalen och avgöra dess kvalitet genom att ta fram de databitar som skickats. Funktionalitet för att låta VST:n sända en bestämd signal inkluderades även i programmet. Det byggda systemet presterade upp till nittio gånger snabbare än det gamla systemet men saknade turboavkodning, något som är nödvändigt för att beräkna bitfelsratio (BER) och blockfelsratio (BLER) på ett korrekt sätt. Programmets snabbhet lämnar utrymme för att lägga till mer tidskrävande processer som turboavkodning vid ett senare tillfälle och genom att studera de icke avkodade databitarna kan signalens kvalitet fortfarande mätas. Programmet hanterar både 1 tx och 2 tx singaler.

Abstract

A test system for Immunity testing of Ericsson's LTE radio base station (RBS) was built using a National Instruments (NI) PXI chassis fitted with a vector signal transceiver (VST) module and a PC controller module. A program made in LabVIEW and MATLAB was run on the controller module in the PXI chassis. The tasks of the program was, on one hand, to control the VST which was used to record the signal emitted from the RBS and on the other hand, to process the signal and determine its quality by acquiring the bits transmitted. Functionality enabling the VST to transmit a given signal was also included in the program. The built system performed up to ninety times faster than the old system but lacked turbo decoding necessary to correctly determine bit error ratio (BER) and block error ratio (BLER). The performance of the system leaves room for adding time consuming processes such as turbo decoding later on and by examining the undecoded bits the signal quality can still be measured. The program handles both 1tx and 2tx signals.

Contents

Abbreviations

16QAM	16 Quadrature Amplitude Modulation
64QAM	64 Quadrature Amplitude Modulation
ASIC	Application-specific integrated circuit
BER	Bit Error Ratio
BaE	Bit acquisition and Error calculation
BLER	Block Error Ratio
BS	Base Station
DL	Downlink
EMC	Electromagnetic Compatibility
ESD	Electrostatic Discharge
FDD	Frequency-division duplexing
FFT	Fast Fourier Transform
FPGA	Field Programmable Gate Array
HDL	Hardware description language
IFFT	Inverse Fast Fourier Transform
LTE	Long Term Evolution
MIMO	Multiple Input Multiple Output
NI	National Instruments
OFDMA	Orthogonal Frequency Division Multiple Access
OFDM	Orthogonal Frequency Division Multiplexing
PCI	Peripheral Component Interconnect
PDSCH	Physical Downlink Shared Channel
PSS	Primary Synchronization Signal
PXI	PCI eXtensions for Instrumentation
QPSK	Quadrature Phase-Shift Key
RBS	Radio Base Station (Ericsson terminology)
RF	Radio Frequency

RS	Reference Signal
RRU	Remote radio unit
TB	Transport Block
UE	User Equipment
UI	User Interface
UL	Uplink
VST	Vector Signal Tranceiver
VI	Virtual Instrument
VSG	Vector Signal Generator
VSA	Vector Signal Analyzer

Notation and symbols

f_0	Carrier frequency
T_s	Symbol time
Δf	Subcarrier spacing
d	Symbols to be transmitted in one subframe
M_{symb}	Number of PDSCH symbols per subframe
$\mathbf{x}$	Layer mapped symbols
$\mathbf{y}$	Precoded symbols
b	Bit sequence
c	Scramble sequence
$\tilde{b}$	Scrambled bit sequence
A	Matrix containing all symbols in one frame
A_{PDSCH}	Matrix A with zeroes on all non PDSCH resource elements

1 Introduction

Base stations make up the body of a wireless network and via antennas they handle the direct communication with all mobile devices within the network. As more and more of our communication requires a wireless connection increased coverage is of essence. A steady increase in speed and stability is also expected as people buy faster and more advanced phones.

To keep up with growing demands the network infrastructure needs both continuous expansions and upgrades. By creating a more widespread network, coverage can be achieved, by building a more dense network speed and stability can be achieved. Both solutions require more base stations. Higher speed can also be achieved by improving the way the physical resources such as time and bandwidth is utilized. New standards like LTE (Long term evolution) improves the rate at which data can be transferred via wireless connections. Upgrading networks to meet these standards also require the production of new base stations equipped with radios to handle both new and sometimes also older standards.

All in all, new and better base stations are and will be in demand for some time to come and this is what has made Ericsson one of Sweden's largest companies and a big player on the international telecom market with networks handling over 40 percent of world's mobile traffic. One of Ericsson's key products is their base station, the RBS (radio base station), which come in a number of varieties ranging from small units with integrated antennas to larger cabinets containing up to twelve different radios. One of Ericsson's largest line of products is the RBS 6000 series, an example of an RBS 6201 is shown in figure 1. They are a common in many cities although not very noticeable, often placed in basements, on roofs or inside containers in close proximity to the antennas. In order for Ericsson to maintain its position on the market, production needs to be fast and effective. One important step of the production procedure is electromagnetic compatibility (EMC) verification testing which is performed to make sure that the RBS functions properly even under sub optimal radio frequency (RF) conditions. If EMC verification test times can be reduced without affecting the tests reliability there is profit to gain not only by saving in on testing costs but also by speeding up production, thus making the products more competitive.

1.1 Thesis description

This thesis was performed at Ericsson's Integration and Verification (I&V) EMC unit. It focuses on improving the efficiency of the current test system for EMC verification tests for base stations transmitting Frequency-division duplexing (FDD) LTE signals, specifically the ra-

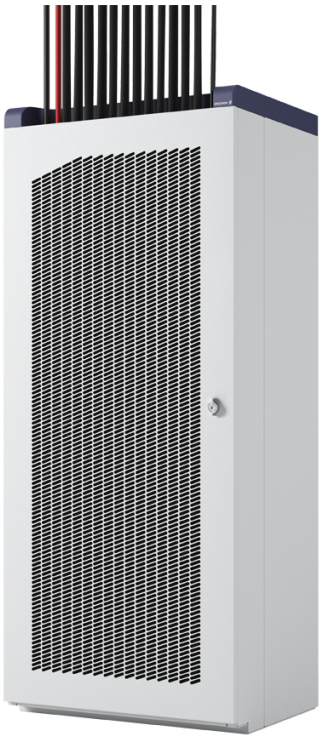

Figure 1: Ericsson base station RBS 6201

diated immunity test case where the RBS immunity to radiated electromagnetic waves is tested . The current test system used for this test case requires a three second "dwell time" at each frequency the RBS is tested for. This is the time during which the system radiates the RBS, records the signal transmitted from the RBS ans analyzes the signal quality by determining the bit error ratio (BER). By using the new vector signal transceiver (VST) from National Instruments (NI) combined with a LabVIEW program to perform the BER tests serveral time consuming steps of the current test system could be avoided. To achieve the desired results I have looked into the possibilities provided by the built in field programmable gate array (FPGA) as well as the advantages of integrating MATLAB code into my program. Much focus has also been put on gaining an understanding of the structure of the FDD LTE signal which of great importance for this thesis.

The result of this thesis was a LabVIEW program using the above mentioned hardware from NI and built to handle the following:

- Generation of Uplink test signal
- Acquisition of Downlink test signal

- Finding the raw undecoded data from the acquired signal and comparing it to known correct bits

The program does not contain proper turbo decoding and therefore lacks the ability to correctly determine BER. To determine the performance of the system tests were made trying to optimize the number of radio frames per second that the system could process.

2 EMC Verification

EMC Verification tests are an important part of finalizing all sorts of electrical products. This is mainly due to two things. First, many countries, including Sweden, have laws regulating the the electromagnetic emissions products are allowed to produce. Within the European Economic Community, for example, we have the CE marking which indicates that a product carrying it does meet certain requirements concerning EMC [1]. Secondly, in order for a product to to be competitive on the market, it must not only perform well under optimal conditions but must also be able to withstand a certain amounts of electromagnetic radiation from its surroundings as such radiation is something to reckon with in most populated parts of the earth. Immunity tests are therefore performed to ensure a products capability to withstand external radiation.

2.1 Immunity testing

This section attemts to give an overview of the immunity test specification in [4]. The immunity tests of Ericsson's RBS 6000 series are performed in an RF anechoic chamber and during specific environmental conditions. The specification refer to things such as temperature, humidity and power supply variations. The tests consist of four test cases:

- Electrostatic Discharge (ESD) – ESD's are applied to multiple parts of the RBS including the case, each of the modules making up the internals of the RBS and cable connectors. Connector pins however are excluded and may not receive ESD's.

- Radiated Immunity – The RBS, placed on an insulated support above ground, is exposed to radiation at all four sides. These tests cover a large frequency spectrum by dividing it into a discrete number of frequencies and testing each of them separately. This process of stepping through a large number of frequencies also makes this test case the most time consuming.

- Electrical fast transients/bursts – Bursts are applied to the cables within and connected to the RBS. Depending on the type

of cable either a coupling/decoupling network or a capacitive coupling clamp is used.

- Surge – A surge, or a voltage spike, is applied to the cables connected to the RBS. The point where the surge is applied to the telecom ports cables is either at a distance of 20 meters from the RBS or as far as possible. For power suppliy cables there is no specified required distance.

During these test cases, a communication link that has been established between the test equipment and the RBS is evaluated by measuring throughput for the Uplink (UL) and throughput as well as Bit Error Ratio (BER) for the Downlink (DL). The Downlink, which is the main focus fork this work, uses a Test Model E-TM 1.1 signal described i table 1. The meaning of this will be described further in section 4). Measurements can also be made by looping the signal. This means that the bits from the downlink signal are sent back without checking the BER.

The performance criteria for both UL and DL are a throughput of $\geq$95% and no loss of user control functions and stored data. For test concerning transient phenomena the throughput may occasionally drop beneath 95% as long as the communication link is maintained. If looping is used the performance criteria is $\geq$90% throughput covering both UL and DL.

Table 1: Specifications of the E-TM 1.1 signal used for immunity testing.

Bandwidth	5 MHz
Modulation scheme	QPSK
Cyclic prefix	Normal
Data	pn9 (2216 bits)

2.2 Current testbed

The current testbed used by Ericsson for performing Immunity Verification tests, shown i figure 2, mainly consists of three components, an instrument with built in signal analyzer and signal generator and a computer. The signal analyzer is used to record and locally store portions of the signal sent from the RBS while the computer handles the following tasks:

- set up and control of the signal analyzer
- reading the recorded data from the internal memory of the signal analyzer

- evaluating Downlink BER and throughput, calculations are performed in proprietary instrument specific code
- collecting Uplink throughput data from the RBS
- running and updating the user interface (UI) shown in figure 3.

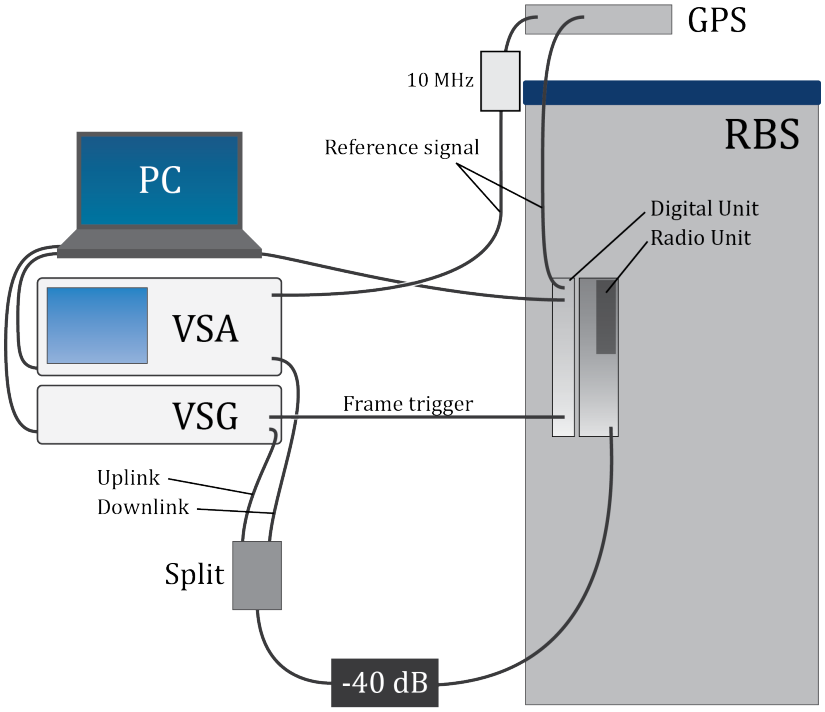

Figure 2: Current testbed for immunity verification using a Vector Signal Analyzer (VSA) and a Vector Signal Generator (VSG).

2.2.1 Dwell time

The dwell time represents a minimum of time that the test system needs to remain (dwell) at a certain frequency during immunity testing. The existence of a dwell time is necessary for the link to be able to transmit enough data to enable throughput and BER calculations. In every capture of Downlink data, one frame needs to be present in order to calculate throughput and BER, (see section 4.3). However due to the process of the having the signal analyzer locally store the recorded signal before the computer can read and evaluate it, the dwell time is significantly increased. For the current testbed this means an increase from 1 second to at least 3 seconds. The 1 second minimum dwell time is due to limitations of the system at the test facility

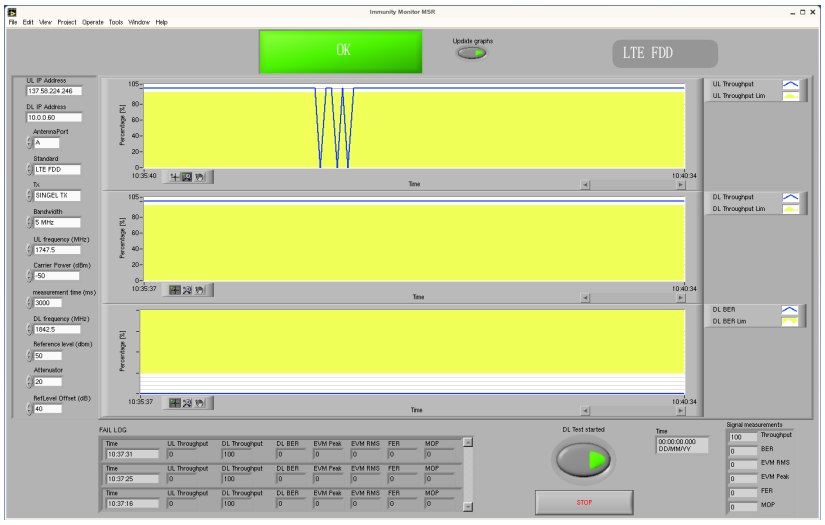

Figure 3: Current testbed user interface, center part showing graphs of the Uplink (UL) and Downlink (DL) throughput as well as the (Bit Error Ratio) BER over time. The left panel shows a number of test settings that can be set by the user prior to the test startup.

2.3 Planned testbed

In order to minimize dwell time a new testbed is planned. Similar to the current testbed the planned testbed will also consist of two main components, one of which is the same computer used in the current testbed. The second component, replacing analyzer and the generator, is a National Instruments PCI eXtensions for Instrumentation (NI PXI) Chassis equipped with a National Instruments Vector Signal Transceiver (NI VST) module and a controller which is basically a small PC. Figure 4 shows the planned testbed. The controller runs all the software needed to set up and control the the modules in the chassis, it can also use the field programmable gate array (FPGA) in in the VST for some calculations. The NI PXI, controller and VST module are presented more thoroughly in section 3.

2.3.1 Improvements

The principal difference from currently used testbed is that all the mathematical calculations needed to evaluate the recorded signal are now to be performed on either the controller or the FPGA in the VST instead of on the computer. This will eliminate the need to write and read data from a hard drive and the goal is to reduce the dwell time this way.

The new testbed will also feature updates to the computer software allowing the use of both the old and the new analyzing hardware without changing end user experience.

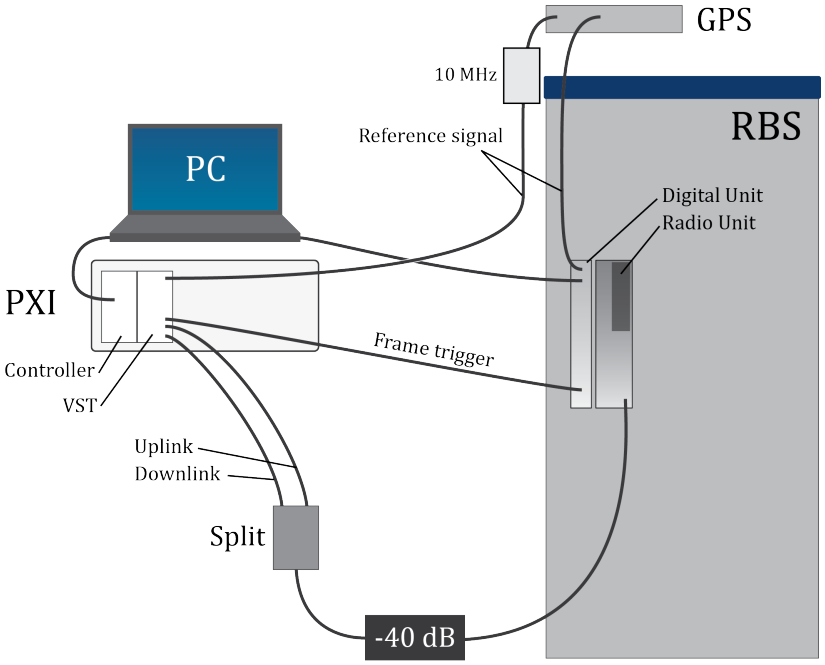

Figure 4: Planned testbed for immunity verification using a PXI (PCI eXtensions for Instrumentation) chassis equipped with a Vector Signal Tranceiver (VST). The VST is handling both the Uplink signal generation and the Downlink signal analysis.

3 NI PXI

The NI PXI is a modular instrumentation platform, meaning it works as a platform on which the user can combine a number of instrumentation modules all according the users preferences. The PXI concept was originally introduced by National Instruments in 1997 and they are still among the leading companies providing PXI solutions. PXI modules differ from conventional instruments in that they do not include power supply, cooling, memory and processors for running software or displays. Power supply and cooling is provided by the chassis and and a computer for handling software and displays (i.e. the controller) can be either included into the chassis as a module or connected to the the chassis via a thunderbolt cable. Compared to the regular cables used to connect instruments the PCI slots connecting the modules to the chassis also have a high bandwidth. These things combined makes the use of a PXI as a base for different kinds of tests, measurement or system controls

- cost efficient
- space efficient

- power efficient
- fast.

If only NI software and hardware, both modules and chassis, is used the PXI solution is also guaranteed to be out of box compatible. This guarantee does not apply to PXI modules provided by other manufacturers.

3.1 LabVIEW

The system-design platform provided by NI to set up a PXI based solution is called LabVIEW. It is also a development environment for National Instruments own programming language G, which is a graphical dataflow language. Programming in LabVIEW is done with so called Virtual Instruments (VIs) which are presented with a front panel and a block diagram. The front panel is a customizable user interface with input fields and components such as graphs for displaying data and results. The block diagram contains all the underlying logic that handles user input and other data. Execution of the code is determined by the order in which the programmer connects (wires) function nodes to each other in a block diagram as function nodes only executes when all its connected inputs are populated with data. There are essentially four different types of components that make up a block diagram:

- Controls – Holds the value specified by the user before and/or during execution. Can represent both simple data types such as integers or more complex clusters of different data types. Controls are shown as fields or selectors on the front panel.
- Constants – Holds the value specified by the programmer. Can also be used to instantiate variables which will be handed a value during the execution of the block diagram.
- Function Nodes – Performs operations on the data at its inputs. Operations vary from simple tasks such as addition to advanced operations such as Fast Fourier Transforms (FFTs).
- Indicators – Receives and presents data, usually after the Function Nodes have performed the preferred manipulations of input data. Usually shown as programmatically populated fields or graphs on the front panel.

All of these components are shown in figure 5. The front panel representation of this VI is shown i figure 6.

In addition to being executed on its own a VI can also be saved as a subVI in which case the controls represents inputs and indicators represent outputs. This allows for the VI to be inserted into the block

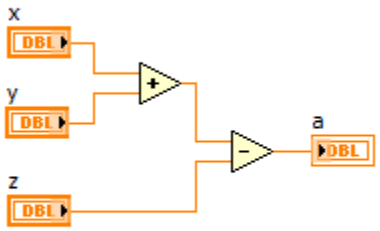

Figure 5: A VI block diagram with controls, function nodes and an indicator.

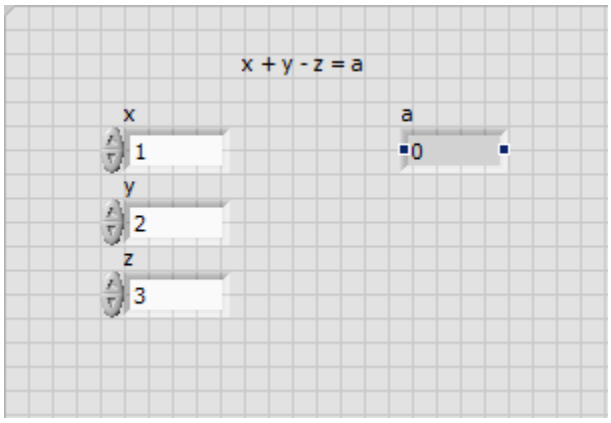

Figure 6: The front panelof the VI in figure 5.

diagram of other VIs thus becoming a subVI. This allows for a nested code structure where subVIs may contain other subVIs in an unlimited number of steps.

Another common part part of the block diagram is the property node which is usually linked to some measuring hardware and can be read continuously for its measurement values. Continuous analysis like that is often performed within a "while loop" that both collects measurement data and updates the graphs on the front panel. Normally the loop runs until a soft button on the front panel is clicked.

3.1.1 Parallel execution

Unlike many text based programming languages which execute linearly row by row, G allows for simple and intuitive parallel programming since the only thing determining when a VI or Function Node will

execute is when all the connected inputs have received valid data. This makes it easy to determine what parts of the code are independent from other parts and a computer with a multi-core processor will then run the different and independent parts of the code simultaneously on different cores if possible.

3.1.2 Mathscript

For developers familiar with MATLAB National Instruments provides a plug-in for LabVIEW called Mathscript that makes it possible to insert MATLAB scripts directly into the block diagram. This is done by placing a Mathscript node as a block in the block diagram and writing MATLAB code in the node. Code written in the node will execute row by row. To transfer data into or out from the node, inputs and outputs named equally to the variables used in the code can be added to the edges of the node border as seen in figure 7.

Mathscript handles most of the functions that exist in MATLAB but also allows the developer to add its own subscripts as m-files in a designated folder in the project. These subscripts may then be called as usual by the code in the Mathscript node.

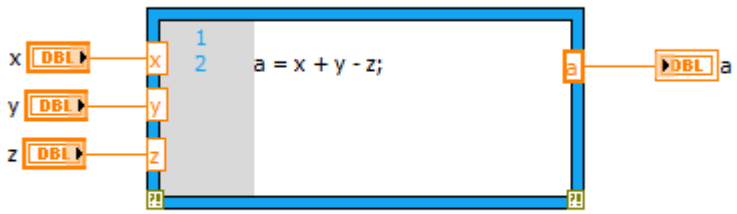

Figure 7: Block diagram with a Mathscript node performing the same task as the block diagram in figure 5.

3.2 NI VST

The Vector Signal Tranceiver (VST) is an instrument combining both a Vector Signal Generator (VSG) and a Vector Signal Analyzer (VSA) using an integrated FPGA-chip for real-time signal processing and control. It is designed as a PXI module to be fitted into a PXI chassis and controlled using the chassis controller module. While there are VSGs and VSAs available as separate PXI modules the VST is often the better choice when both are needed, especially when communication time between the two is to be minimized. The FPGA-chip in

the VST can also be programmed by the user to customize the signal processing of the VST as well as to perform other tasks more suitable more for an FPGA.

3.2.1 FPGA

An FPGA is an integrated circuit built to allow the customer to re-program and configure it in the field, therefore the name "field-programmable". Compared to statically configured integrated circuits such as the Application-specific integrated circuit (ASIC) the FPGA is slower, larger and less power efficient. The larger size also makes FPGAs more expensive when produced in larger batches. However, the ability of re-programming gives the FPGA a great advantage towards the ASIC when used for prototype development and testing systems as such systems often follow the premisses

- required ability of continuous updates
- small production numbers
- no specific limits in size and power consumption.

An ASIC may however replace the FPGA in the finished product to reduce cost and increase efficiency if the product is intended for mass production.

The programming of an FPGA is normally made using a hardware description language (HDL) such as VHDL but NI also provides the possibility to program their integrated FPGAs in G using certain FPGA-specific libraries in LabVIEW.

4 LTE

LTE is the latest technology in mobile communications available on the market. The technology is developed by the 3rd Generation Partnership Project (3GPP) and consist of an open standard to which manufacturers of radio infrastructure and mobile devices shall conform in order for it to work properly. It is, however, left to the manufacturing companies to determine whether their products follow the standard properly [6].

4.1 OFDMA

Orthogonal Frequency Division Multiple Access (OFDMA), a multi-user version of Orthogonal Frequency Division Multiplexing (OFDM), is the modulation scheme used for downlink in LTE [6]. OFDM is an implementation of multicarrier modulation which is when data is transmitted on multiple subchannels simultaneously, each subchannel having a different carrier frequency. OFDMA is simply OFDM with

the ability to allocate different subcarriers to different users and thus allowing multiple access to the channel.

In an OFDM signal the frequency of each subchannel consists of a common carrier frequency, f_0, shared by each subchannel and a subcarrier frequency, Δf_n unique for each individual subchannel [5] are added together. Normally when data is transmitted to a user it is done by transmitting multiple symbols over over different subchannels. A symbol is essentially a sine wave with a fixed phase and amplitude being transmitted over a specific period of time, this time is refereed to as the symbol length. When modulated and shown in the frequency domain this kind of signal will produce a sinc function as shown in the upper plot of figure 8. It can be mathematically shown by Fourier transforming a square wave that the distance between the main peak and the closest zero is determined by the symbol length. They depend on each other according to the relation

$$\Delta f = 1/T_s, \tag{1}$$

where Δf is the the distance from the main peak to the first null in the frequency domain and T_s is the symbol length. The following nulls occur at distances of multiples of Δf.

What makes OFDM orthogonal is the spacing between the subcarriers and the symbol length which is configured in a way that all subcarriers main peaks line up perfectly with the nulls of the other subcarriers, a shown in the lower plot of figure 8 , thus making the subcarriers orthogonal. This would ideally make all subcarriers immune to interference from other subcarriers. [6]. In an LTE signal the subcarrier spacing, Δf , is 15 kHz giving a symbol length, T_s, of 66.7 µs.

4.2 Signal generation

In digital communication a symbol is often represented by a point or a vector in the complex plane. When represented this way the distance from origo is determined by the amplitude of the symbol signal and the angle is determined by the phase shift of the symbol signal.

The simplest modulation scheme supported by LTE is quadrature phase-shift key (QPSK). It allows four different symbols, each representing two bits, to be transmitted. These symbols are placed at $(1, 1)$, $(-1, 1)$, $(1, -1)$ and $(-1, -1)$, e.g. only the phase shift is different as suggested by the name. An example of a system with four subcarriers sending an QPSK signal is shown in figure 9 . Here each subcarrier is sending a different symbol which means the phase shift of each wave is different but with no change in amplitude. The frequencies of the subcarriers are also spaced by Δf to make them orthogonal, this gives them different wavelengths.

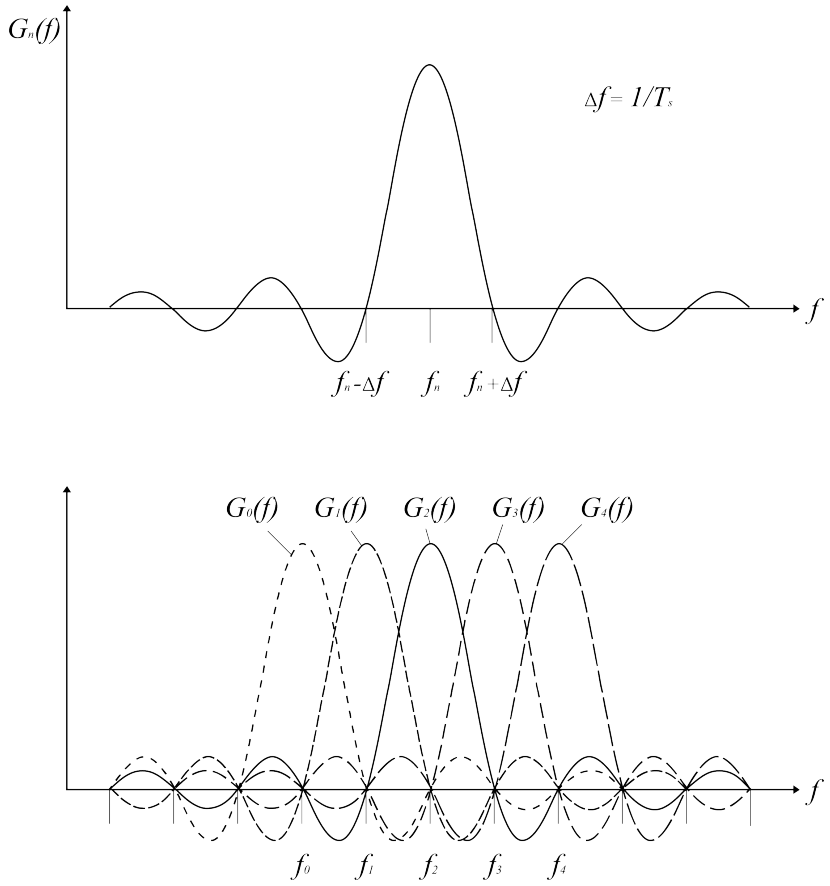

Figure 8: A plot of a sinc function and an plot of multiple orthogonal sinc functions.

A simplified model of the signal generation procedure is shown in figure 10. The first step is to define the subcarriers in the frequency spectrum by the vector of the symbol that the subcarrier shall transmit, this is referred to as mapping. After mapping the symbols to the subcarriers they have a cyclic prefix attached and then each subcarrier is converted to the time domain using an inverse fast Fourier transform (IFFT). The resulting wave, like the one shown on the lower half of the figure 9, is a time dependent complex combination of the four different subcarrier waves that can be transmitted through the air. The model of the signal analysis procedure used for downlink, see figure 11, is similar to uplink but opposite in order using the fast Fourier transform (FFT) to convert the signal to frequency domain.

4.3 Downlink signal structure

To describe the structure of the signal in LTE the easiest way is to represent the signal with a two dimensional matrix having the fre-

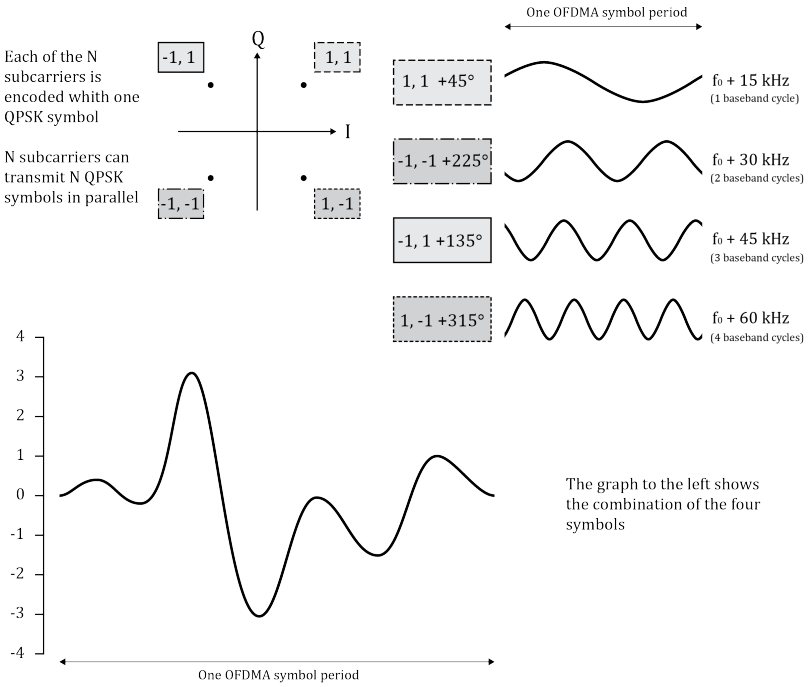

Figure 9: Description of QPSK signal generation. This figure is based on figure 2.2-5 in [6].

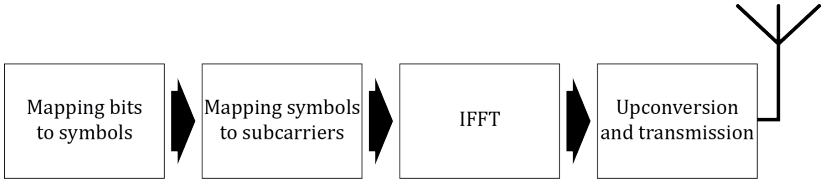

Figure 10: A simplified model of the signal generation procedure.

quency domain on one axis and the time domain on the other axis. In the frequency domain the signal is divided into sections 15 kHz wide, the subcarrier spacing Δf, each representing a subcarrier. In figure 12 the frequency domain is on the y axis. As an example of the dimensions the most common bandwidth for immunity testing, 5 MHz (of which 4.5 MHz is used for transmission), has 300 subcarriers. LTE does however support up to 1600 subcarriers (20 MHz bandwidth).

In the time domain the signal structure is more complex. The smallest element in this domain is the symbol as mentioned earlier in OFDMA. Each symbol then have a cyclic prefix attached to the start to eliminate inter symbol interference due to delay spread. This combination of symbol and cyclic prefix will still be referred to as a symbol to simplify future explanations. The symbols are then grouped as follows:

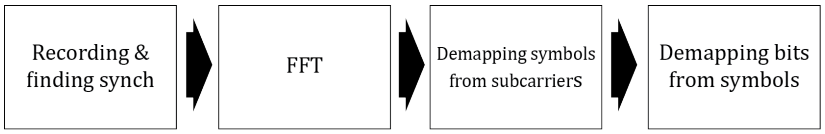

Figure 11: Signal analysis procedure for Downlink.

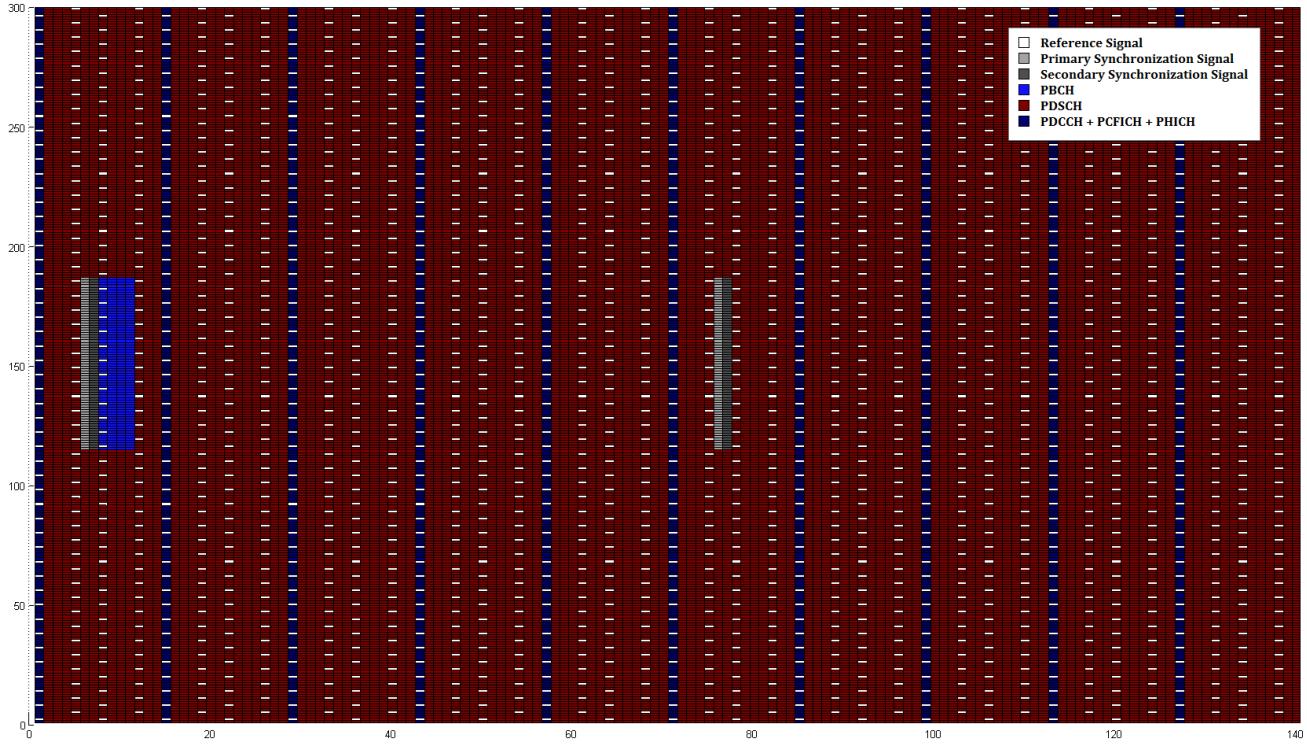

Figure 12: Figure showing the FDD LTE signal structure with frequency domain on the y axis and time domain on the x axis.

1. seven symbols makes up one slot
2. two slots makes up one subframe
3. 10 subframes makes up on frame.

This way of structuring the signal in the time domain is shown in figure 13. The frame is the largest entity in the time domain and consist of 140 symbols, it lasts exactly 10 ms. On figure 12 the entire length of the x-axis corresponds to one frame.

4.3.1 Resource block

The smallest unit in the signal structure is the resource element, it represents one symbol in the time domain and one subcarrier in the frequency domain. A resource block consist of the combined signal on

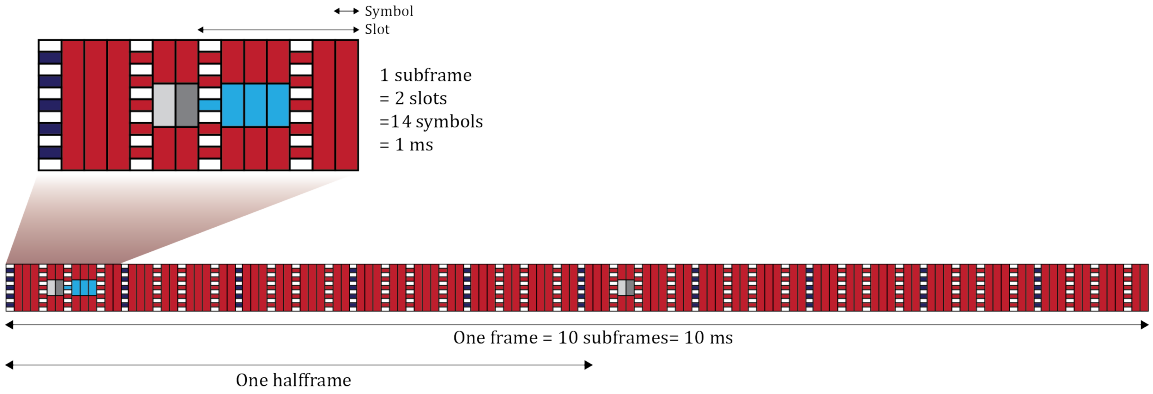

Figure 13: Time domain structure of FDD LTE signal.

twelve subcarriers over 0.5 seconds of time, an interval equal to one slot or seven symbols. The resource block is the smallest unit that can be scheduled for transmission [6].

4.3.2 Signal components

As shown in figure 12 different parts of the signal are allocated for different kinds of information. This is a reoccurring pattern and every frame is structured in the same way. In the case where only one user is being assigned all the resources and all channel info is known, such is the case during immunity testing, there are primarily three parts of the signal to take into account:

- primary synchronization signal (PSS)
- reference signal (RS)
- physical downlink shared channel (PDSCH).

PSS: The primary synchronization signal, shown as the light grey field in figure 12, always occupies the central 62 subcarriers of the channel and is transmitted twice every frame. It is designed to be detected by all types of user equipment (UE) in order to synchronize with the channel and can be used to find coarse frequency and phase errors at the UE side. These errors can then be compensated for by altering the received signal. To allow detection the PSS subcarriers are modulated using a Zadoff-Chu sequence known by the UE. This sequence vary depending on the identity of the cell in which the user is positioned to avoid synchronizing with neighboring cells. Figure 14 shows an example of the Zadoff-Chu sequence constellation plot for cell one.

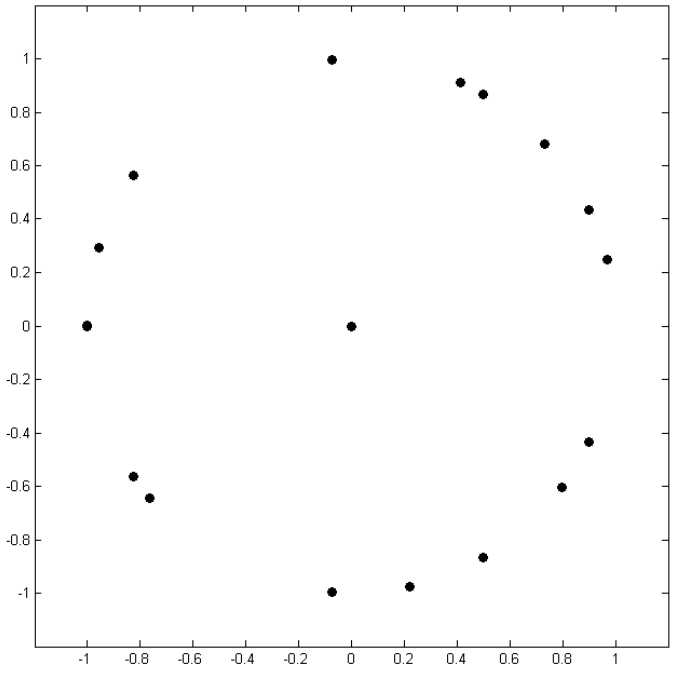

Figure 14: Zadoff-Chu sequence constellation plot for cell one.

RS: To perform a more fine tuning of the signal in terms of phase and frequency the UE utilizes the reference signal. As shown in figure 12, the RS is uniformly allocated across the entire signal structure, this allows the UE to estimate phase and frequency errors that are more representative for the entire channel. In this work the fact that the RS differs over the signal in a predefined pattern is used to determine whether the PSS initially found is the one positioned in the first or the eleventh slot. This is crucial for the the test system to know since the number of resource elements in the slots vary depending on the slots position within the frame. Trying to find data from a different resource elements than are present will cause the system to display incorrect results.

PDSCH: The physical downlink shared channel (PDSCH) is the channel that carries the traffic data to the UE. This is naturally also the channel that makes up most most of the elements in signal structure which figure 12 also shows. On a 5 MHz band for example the number of PDSCH resource elements per subframe vary between 3330-3750 out of 4200 depending on subframe index. The PDSCH supports QPSK, which is the modulation scheme used for immunity testing

as well as 16 and 64 quadrature amplitude modulation (16QAM and 64QAM respectively).

4.3.3 Multiple Antenna Systems

LTE supports systems using multiple antennas, sometimes referred to as multiple input multiple output (MIMO) systems [6]. At Ericsson an RBS designed to use multiple antennas is normally referred to as having 2tx (two antennas) or 4tx (four antennas) capability.

For immunity testing, the use of multiple antenna systems means testing on an RBS with multiple radio outputs, for example a Remote radio unit (RRU). In such a system each radio output transmits a different signal both compared to each other and the one transmitted in 1tx (regular one-antenna systems). Differences between 1tx and 2tx are the following:

RS: In a 2tx signal there are twice as many resource elements allocated for the RS compared to 1tx. As can be seen in figure 15, positions of the RS on port 0 (the first antenna port) are the same as for the 1tx case while the RS on port 1 (second antenna port) is placed orthogonally in both frequency and time [6]. A 2tx signal sent on port 0 can not allocate any data on on the resource elements intended for the port 1 RS and vice versa. This must be taken into account while collecting data since it affects the number of PDSCH resource elements.

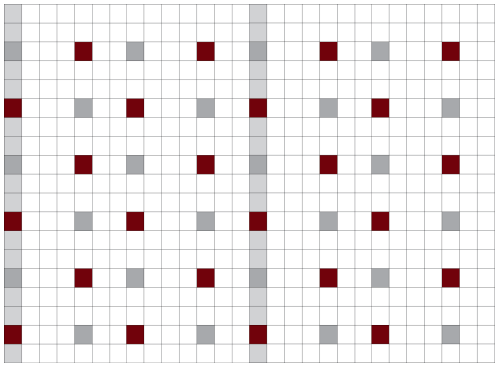

Figure 15: Port 0 reference signal (red) positioning in a 2tx signal. Dark grey areas represent non-allocated resource elements where port 1 reference signal is positioned.

Layer mapping: As shown in figure 16 there are two extra steps in the signal generation model when 2tx is utilized. The first step, layer mapping, takes the array of symbols to be transmitted, d of length M_{symb}, and reshapes it into a matrix $\mathbf{x}$ given by

$$\mathbf{x} = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} d(2k) \\ d(2k+1) \end{bmatrix}, \; k = 0, 1, 2, \ldots, M_{symb}/2 - 1. \qquad (2)$$

This forms the two layers required for the 2tx precoding. In a 4tx system four layers need to be formed.

Precoding: This is the second step added to the signal generation model shown in figure 16. Since both antennas will transmit the same data there is a risk of channel fading affecting the same symbols in both signals. To avoid this it is made sure that the antennas are sending the same symbols on different subcarriers and making one of them orthogonal to the other. This is done by creating two new arrays of symbols, y_0 for antenna port 0 and y_1 for antenna port 1. Both are of size M_{symb}. Symbol array $\mathbf{y}$ is defined by

$$\mathbf{y} = \begin{bmatrix} y_0(2k) & y_0(2k+1) \\ y_1(2k) & y_1(2k+1) \end{bmatrix} = \begin{bmatrix} x_0(k) & x_1(k) \\ -x_1^{\star}(k) & x_0^{\star}(k) \end{bmatrix}, k = 0, 1, 2, \ldots, M_{symb}/2-1. \qquad (3)$$

This is also known as Alamouti space-time, decribed in [5]. By looking at equation 2 and equation 3 it can be seen that y_0 is equal to d, in this case just like it is for 1tx.

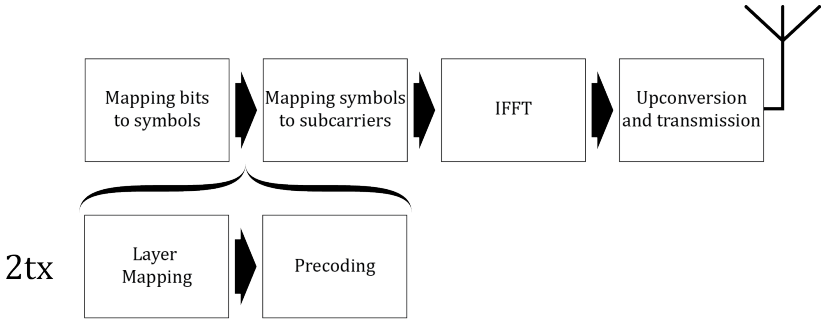

Figure 16: Signal generation model for a 2tx signal.

4.4 Downlink transport channel

In order for a data bit to become mapped to a symbol several steps of bit processing needs to be performed. This processing is done in the transport channel. Figure 17 shows the downlink transport channel, scrambling and the signal generation model from figure 16 which combined make up the entire process from bit to air transmitted wave in LTE.

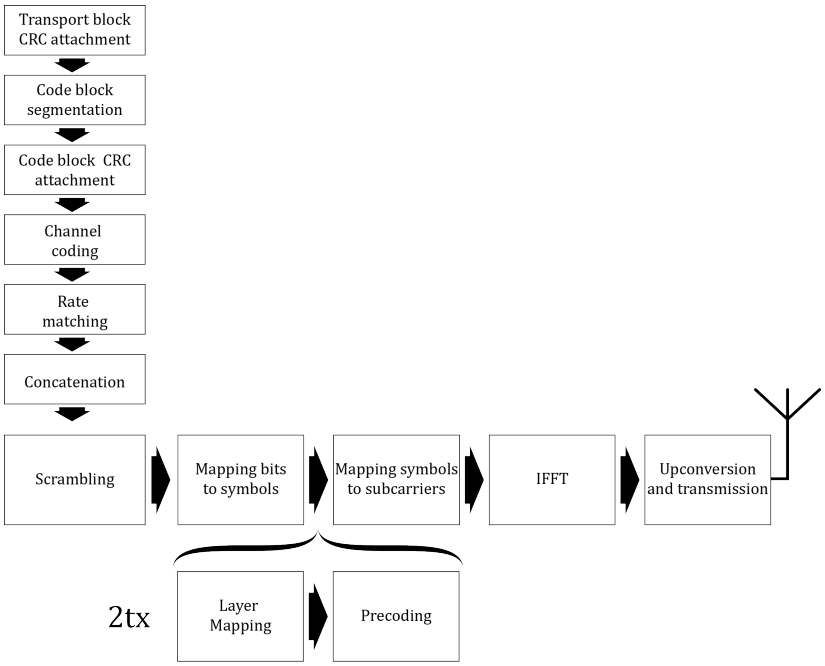

Figure 17: Transport channel, scrambling and signal generation model.

Figure 18 shows the transport channel and examples of how it affects the information bits.

The first step of preparing a binary file for wireless transmission is to split it into portions of a desired length called transport blocks (TB), the size of the transport block is defined in [3]. The downlink transport channel then works as follows.

Transport block CRC attachment: A cyclic redundancy check (CRC) suffix is attached to the end of the TB. The CRC is a sequence created by using cyclic generator polynomial on the entire TB and will be used to determine whether a received TB is intact or if it needs to be transmitted again [6].

Code block segmentation: Depending on the bandwidth of the channel the number of bits that can be transmitted in one subframe is different. If the size of the TB is larger than the maximum number of bits allowed in a subframe it will become divided into code blocks (CB). Otherwise the transport block simply becomes a code block.

Code block CRC attachment: A CRC is attached to the code block. The CB CRC is created using a different polynomial from the TB CRC but it has the same purpose.

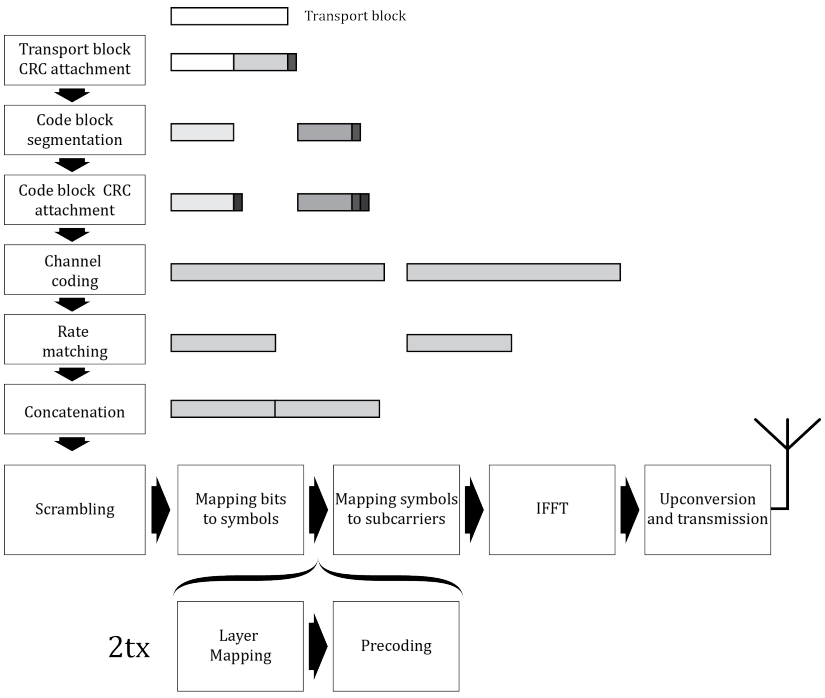

Figure 18: Transport channel, scrambling and signal generation model with data block sizes described for each transport channel step.

Channel coding: Each CB is encoded using the rate 1/3 parallel concatenated turbo encoder shown in figure 19. This reduces the bit error probability at low signal to noise ratios by expanding the CB size by three times, introducing new bits derived from the original CB bits [5].

Rate matching: The three output blocks from the encoder output each have extra dummy bits attached to them in order for them to completely fill up a matrix with a given width. They are then individually interleaved by reorganizing the matrix columns in a defined pattern, rotating the matrix and combining the rows into a single array block. The second and third block is combined by alternating the bits and and then attached to the end the first block. The resulting block is stored in a circular buffer and the dummy bits are removed

Concatenation: Bits from the circular buffer are selected from a given index until the number of selected bits matches the number bits that can be transmitted in the next subframe.

Scrambling: The last thing to do with the bits before mapping them to the symbols is to make sure the ones and zeroes are evenly

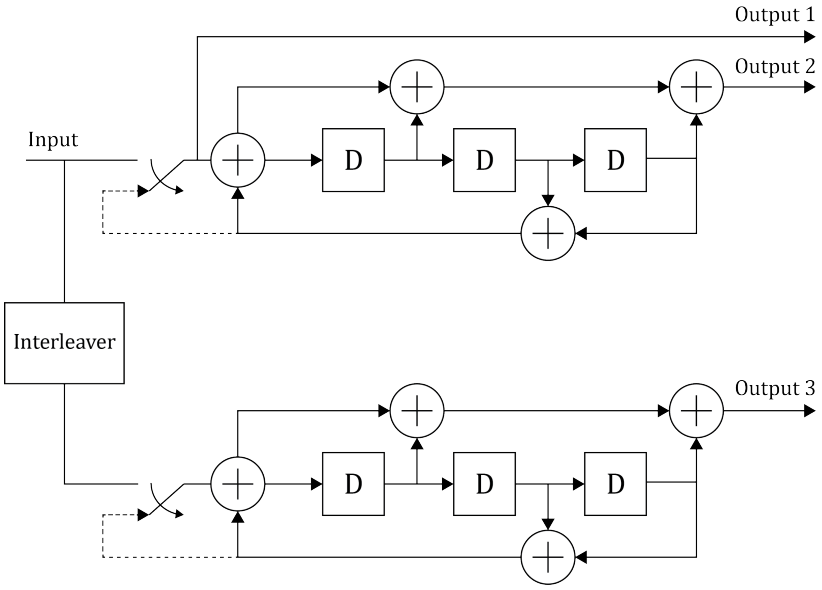

Figure 19: 1/3 parallel concatenated turbo encoder.

distributed. This will in turn lead to an even distribution of the symbols meaning that there will be approximately the same number of constellation dots in each quadrant which gives a more robust signal. This is done by altering the bits in a way called scrambling. The scrambled bit sequence is given by

$$\tilde{b}(q) = (b(q) + c(q))mod2 \tag{4}$$

where b is the initial sequence and c is a pseudo-random bit sequence defined in [2].

4.5 Data Acquisition

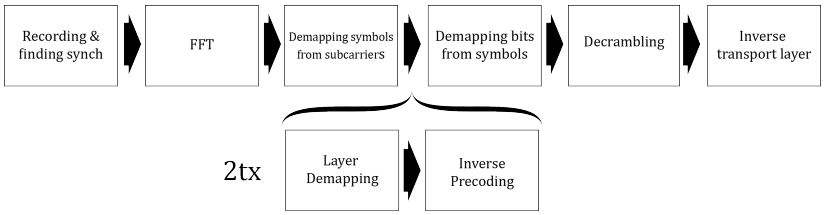

Figure 20: Data acquisition model.

The main task of this thesis was to acquire the data bits from a time domain complex signal sent from an RBS. In practice this means reversing the process in figure 17 by building a system performing the

process shown in figure 20. Following steps describes how this was done.

Signal recording: A program made with LabVIEW is run on the PXI Chassis Controller , the main VI front panel is shown in figure 21. It uses the VST and the VST specific library of VIs to record sequences of the downlink signal with regular intervals. The recorded sequences are 22 ms long, which is the equivalent to 2.2 frames. This is done to ensure at least one complete frame within the recorded sequence.

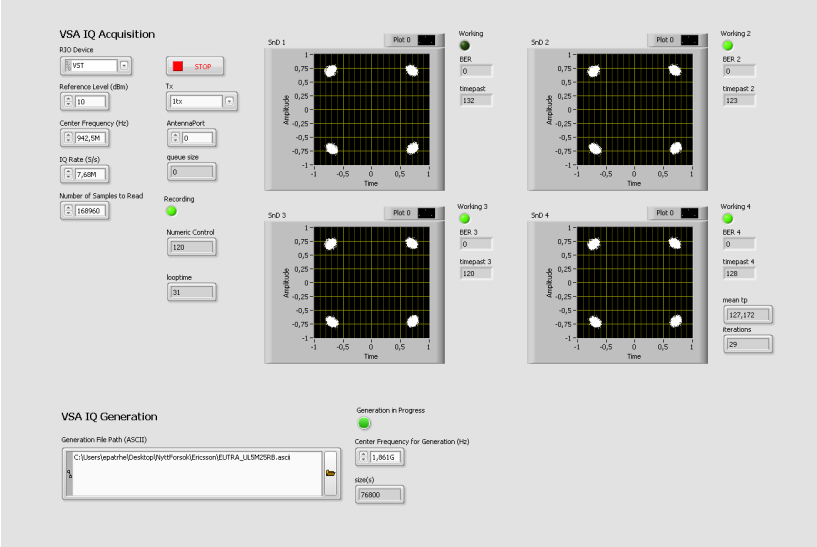

Figure 21: The front panel of the main VI.

Frame synchronization: The position of the first PSS in the recorded sequence is located by calculating the cross correlation between a programmatically created PSS sequence and the recorded sequence. That position becomes the assumed starting position of a complete frame. As shown in figure 12 there are two PSSs in each frame and they are not positioned at the exact start of each half-frame. If the number of samples before the PSS is not enough it means the frame with the first detected PSS cannot be complete. If that is the case the assumed starting position of the frame will be adjusted by 5 ms, one half-frame.

As described in section 4.3.2 the RS is different for different slots. By comparing the RS at the assumed starting position with the known RSs for the first and the eleventh slot, the slots at which the PSS is allocated, it can be determined if the assumed starting position is at the correct slot. If the result of the comparison indicates that it is incorrect the assumed starting position will be adjusted by another 5 ms, otherwise it will be left unchanged. The starting position of a

complete frame is now located and this is the frame from which the data will be acquired.

Adjust for Phase and Frequency Error: By comparing a known complex time domain signals with parts of the recorded signal that are expected to be the same as the known signal, differences in frequency and phase between the two can be determined. This is done in three steps:

1. Divide the known correct signal with the recorded signal sample by sample.
2. Calculate the angle of each element in the resulting sequence from the previous step, this will give the difference in angle between the two initial signals.
3. Perform a first order polynomial fit on the angle sequence.

The initial difference in angle between the two signals represents a constant phase error. This value is given by the order zero term in the polynomial fit result vector. The linear increase or decrease in angular difference between the two signals over time represents a constant frequency error. This value is given by the first order term in the polynomial fit result vector. The signals used to estimate phase and frequency errors are:

- Primary Synchronization Signal
- Parts of the Reference Signal
- Cyclic Prefix

When phase and frequency errors are estimated the recorded signal is adjusted to compensate for these.

FFT: The time domain signal is divided into 140 sequences, one for each time domain symbol. Each of the 140 sequences have its cyclic prefix removed and the sequences are individually transformed to the frequency domain using the FFT. The transform produces a vector of which the 300 values closest to the middle, the values representing the 300 subcarriers, are kept. The result is a 300 by 140 matrix, A, similar to the one shown in figure 12.

Demapping: To get the symbols containing the elements of A are multiplied with the elements of a binary matrix of the same dimensions. This binary matrix is created so that all the resource elements corresponding to the PDSCH are equal to one while the rest are zero. The resulting matrix, A_{PDSCH}, now contains only the data resource elements from A.

In the test case used for immunity testing one block of data is sent every subframe, this means that the subframes must be analyzed individually. To prepare for this A_{PDSCH} is divided into ten matrices of size 300 by 14, each representing one subframe. Each subframe matrix is reshaped into a vector and all elements with the value zero are removed. Remaining are ten vectors containing only the data symbols for each subframe.

Demodulate: In a system sending a QPSK signal getting the raw bits from the demapped symbols is just a question of determining in which quadrant the symbol constellation dots, see section 9, are in.

Inverse layer mapping and precoding (only 2tx): In a 2tx system the orthogonalization and swapping of subcarrier, described in section 4.3.3, must be taken into account if measurements are performed on port 1. In the test system built for this thesis work this part in incorporated into the demodulation part by demodulating the symbols two and two. Before determining the bits the symbols exchange places and are conjugated and multiplied by -1 according to formula 3.

Descrambling: By performing the same scrambling operation as the one used on the downlink transport channel at the RBS, described in 4.4, the raw bit array becomes descrambled.

Inverted transport channel: The final step of the data acquisition is to perform the transport channel backwards step by step. This incorporates a number matrix and vector operations, all of which are directly derived from each step of the transport channel described in section 4.4. The only exception to this is the channel decoding which work in a different way.

During throughput tests a bit sequence consisting of only zeroes can be used. Because the turbo encoder used for downlink, see figure 19, will produce a sequence consisting of only zeros when fed with a zero only sequence the channel decoding step will not be necessary during data acquisition in order to get an indication of the signal condition. Such a test will not however take in to account the error correcting effect of the turbo decoder. In this report bits that have not been processed by a turbo decoder will be referred to as raw data bits or raw bits.

5 Results

The main result of this thesis work is a program made in LabVIEW that with some modification can be integrated into the EMC verifi-

cation testbed used at Ericsson today. Figure 22 shows the main VI block diagram and points out the parts it is made up by.

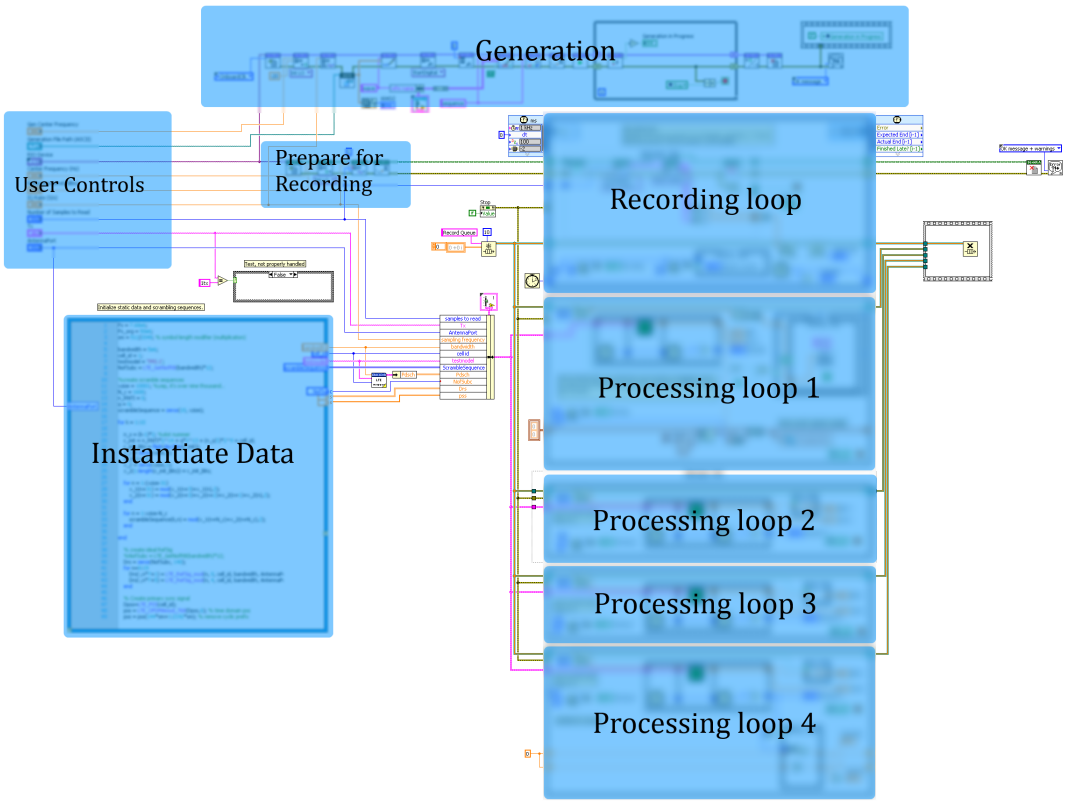

Figure 22: Description of the main VI block diagram.

The program performs the following tasks:

- Reading in constant objects and values together with the user settings. These are stored locally and used as input for the other VIs in the program.
- Generating a continuous uplink signal from a file containing a waveform sequence in ASCII format. This is meant for uplink tests and the signal is transmitted to the RBSs antenna port at the same time as downlink tests are performed.
- Recording sequences of the downlink signal repeatedly with a given time interval between every recordings. The sample rate, number of samples, center frequency and reference level can be set by the user. The recorded sequences are stored locally in a queue for processing.

- Simultaneously running multiple instances of the VI performing raw bit acquisition and error calculation on the queued recordings. This VI is described further in section 5.1.
- Displaying test data such as constellation dots and raw bit errors.

The program runs continuously until the user decides to stop it. When the program stops the queue memory is freed and the sessions for signal generation and signal recording are closed.

5.1 Bit acquisition and error calculation (BaE) VI

The block diagram of the VI performing the raw bit acquisition and error calculation is shown in figure 23. The colored fields describe the different parts of the VI. Its inputs are a recorded sequence of the signal and a cluster of static data types with different values. Having the data cluster as an input allows for the static data to only be instantiated once instead of having to instantiate it every time the BaE VI runs. This would slow down the process. The VI outputs are the constellation dots of the signal and the number of erroneous raw bits in the signal. As can be seen in the figure a Mathscript Node takes up most of the space, this is also representative of the proportions between G-code and MATLAB code in the VI. Much of the MATLAB code lies in subscripts that are called from the main script. These are stored in the program as m-files in, MATLABS own format.

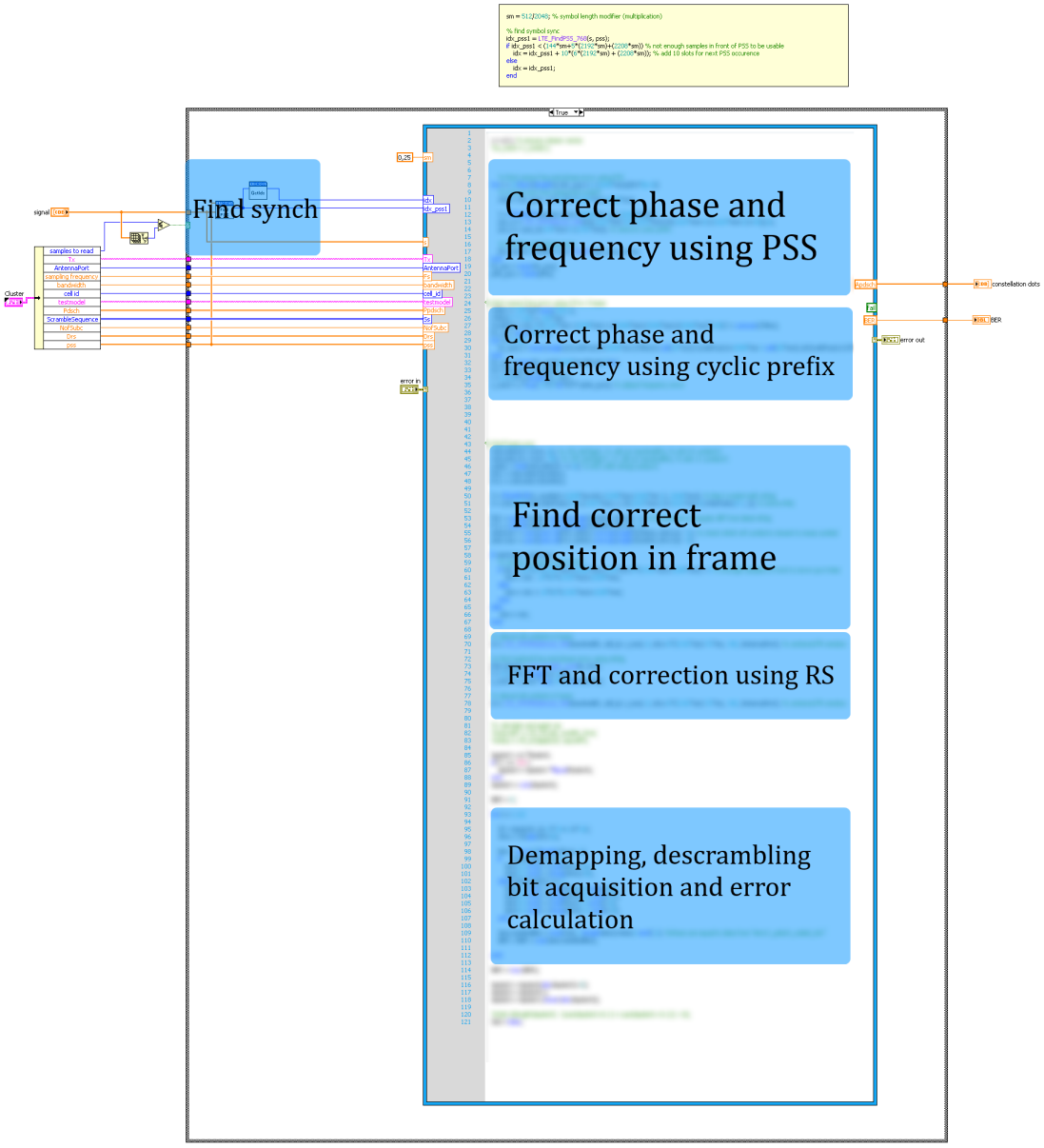

Figure 23: Block diagram of the VI performing the raw bit acquisition and error calculations.

5.1.1 Turbo Decoding

There is no turbo decoding in the BaE VI because the complexity of creating a turbo decoder from scratch required more time than was available. Instead the system now uses zero only sequences transmitted from the base station. This means that the bits that are about

to be decoded consists only of zeroes and can still be examined for errors. This lack of turbo decoding is discussed further in conclusions and open issues, section 6.1.

5.1.2 G-code Version

All of the MATLAB code in the BaE VI has been rewritten into G-code with the exception for one subscript. The block diagram of the BaE VI using this version of the code is shown in figure 24 and gives a more comprehensive picture of the program workflow.

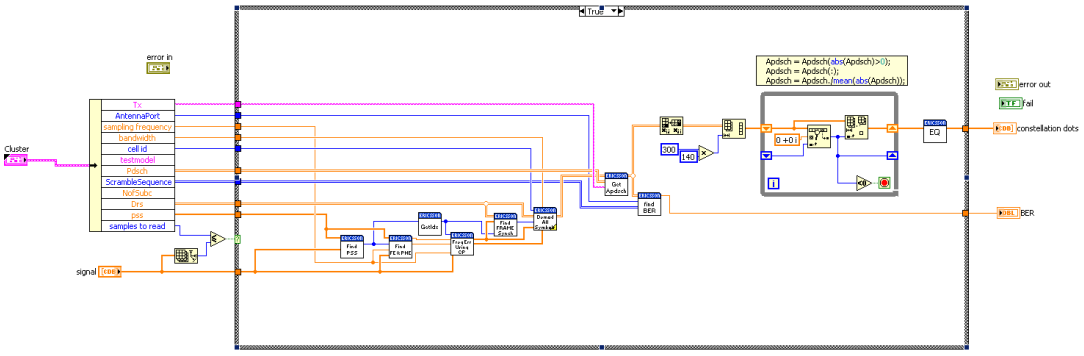

Figure 24: Block diagram of G-code version of the BaE VI.

5.2 Performance

5.2.1 BaE VI execution

Results tests performed on the two versions of the BaE VI are shown in table 2. The time shown is the mean execution time of the BaE VI at 500 executions. The MATLAB version is approximately 60 % faster than the G-code version.

Table 2: Mean execution times for the BaE VI over 500 executions.

BaE VI verison	Mean execution time
MATLAB	87.1 ms
G-code	225.5 ms

5.2.2 Parallel execution

The program was run at three different recording rates with four parallel loops each executing one instance of the BaE VI. For these tests the faster MATLAB version of the BaE VI was used. Table 3 show

the results of the runs. The execution for each of the instances of the BaE VI time appears to be approximately 20 to 30 percent slower when parallel execution is used. This means that the overall execution rate can substantially increased by utilizing parallel execution. Even at the lowest speed the frame processing rate falls way beyond the allowed one frame per second mentioned in section 2.2.1. The frame process rate 30.3 per second corresponds to a 33 ms exection time for the recording loop, shown in figure 22. Faster execution times for the recording loop was not possible to achieve in this program.

Table 3: Relation between the number of records prcessed per second, the recording loop time and BaE execution time.

Frames processed (per second)	Recording loop execution time	Mean BaE execution time (500 executions)
20	50 ms	105.5 ms
25	40 ms	114.0 ms
30.3	33 ms	105.5 ms

6 Conclusions and open issues

6.1 Performance

The program can acquire raw undecoded bits and detect errors at a very high rate as compared to the current test system. Tests show that frames can be processed up to thirty times faster than currently required. This gives a margin that allows future additions of time consuming processes into the program.

Although turbo decoding has yet to be implemented the use of a zero sequence may be sufficient to determine the status of the downlink signal. This is due to the binary character of the EMC verification meaning it mostly either has full throughput or it has zero throughput.

Using statistical data of the turbo decoder's correcting abilities its effect may also be simulated. This would be done by modifying the error percentage of the raw bits to produce an estimation of the real BER. I see this as a last resort and it should be done with caution.

6.2 Turbo decoding implementation

The implementation of turbo decoding is looked into as this paper is written. The manufacturer of the FPGA built into the VST, Xilinx, also provides a toolkit which contains an LTE turbo decoder among other things. This tool is built to run on the FPGA which in theory would make it fast. Adding it to the program would of course mean slowing it down a bit as it adds an extra step och bit processing to

the workflow but as mentioned earlier there is a large margin to work with.

Due to licensing problems the Xilinx LTE turbo decoder has not yet been tested but myself and people at NI are working on a solution.

6.3 NI PXI Chassis & VST

The use of NI's PXI Chassis, controller and VST combined proved very useful for this type of system. It was fast enough and easy to work on using LabVIEW. Everything worked "out of the box" with software provided by NI. It remains however to see how how the VST it performs in tests where a finalized version of the thesis test program is compared to the old testbed. The VST also makes implementation of looping a possibility as this requires both a signal analyzer and generator connected by fast system.

6.4 Mathscript

Using Mathscript helped saving a lot of time when programming the the more calculation heavy parts. With the exception for a few functions MATLAB code could be directly ported into LabVIEW whithout any problems. The only downside was that editing a large Mathscript node like the one shown in figure 23 was very slow and often resulted in LabVIEW crashing due to lack of memory. Debugging a Matscript node was also quite complicated and I recommend doing the programming in MATLAB before moving the code to LabVIEW.

6.5 Optimization

A few thing could be done to optimize the code further. Something that has shown to slow down a LabVIEW program is over-use of indicators such as graphs on the front panel. By removing those the performance of the program should be improved. If the program is to be included into the current testbed these things will be removed as they will not fill a purpose anymore.

6.5.1 Combinations of MATLAB and G-code

As of now there are only two versions of the BaE VI, the fastest version combining mostly MATLAB code with some G-code and the slower version with almost no MATLAB code. Previous to the faster MATLAB heavy version there was one version with only MATLAB code. This one was slower and was removed due to that. This does however shows that there are possibilities of improving performance by finding the right combination of MATLAB and G code. How this would be done is not intuitive right now and I suspect it's a trial and error work.

6.5.2 Run more things on the FPGA

Having the FPGA perform tasks now performed by the CPU in the controller could speed up the program both by utilizing the speed of the FPGA better and freeing the CPU allowing it to perform other tasks that are now delayed. The challenge in this will be to determine what parts of the code is suitable for moving to the FPGA.

6.6 Implement looping

By combining the generation part with the rest of the program that ability of performing looped tests could be achieved. This would require the implementation of a turbo decoder. Depending on the rate at which frames need to be transmitted on the uplink the downlink bit acquisition might not be fast enough. One solution to this would be to repeatedly transmit the last received frame on the uplink until a new downlink frame has been processed and prepared for retransmission.

6.7 Inclusion into the testbed

Including the program into the current testbed is the main future goal. To do this it needs to be made sure that it works properly. This requires further testing, primarily by comparing the performance of this program to the program currently used focusing on the ability to measure the correct BER. This also requires turbo decoding to be implemented.

If it is to be included some restructuring also needs to take place. The program together with the test equipment (in this case the PXI) is to be seen as a virtual device by the testbed. This will allow the tester to choose between the new and the old test program without changing the user experience (except for the test speed). In principle this means that the program needs to be able to handle the same input and produce the same output as the old program thus making them interchangeable. How this is to be done is something that will be looked further into.

6.7.1 Shorten test times

If the new program works and is included in the testbed the next step of optimization would be to examine the 1 second dwell time limitation of the system at the test facility. Lowering that time by any percentage could mean lowering entire immunity test time by almost as much. This would be desirable in an economic point of view.

References

[1] Directive 93/68/EEC. Internet: http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31993L0068:en:HTML, 1993. Retrieved February 8, 2014, from http://eur-lex.europa.eu.

[2] *3GPP TS 36.211 Chapter 5.3.1.* 3GPP, 2013.

[3] *3GPP TS 36.213 Chapter 7.1.7.2.* 3GPP, 2013.

[4] *Verification Specification for EMC Tests, RBS 6000.* Ericsson, Kista, 2013.

[5] Andrea Goldsmith. *Wireless Communications.* Camebridge University Press, 2005.

[6] Moray Rumney. *LTE and the Evolution to 4G Wireless.* Wiley, 2013.