View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Theseus

BACHELOR'S THESIS

ELECTRONIC ENGINEERING

NELEKS13

2017

JARMO KIVEKÄS

SPECTRUM MONITORING SYSTEM IMPLEMENTATION USING SOFTWARE-DEFINED RADIO

TURKU AMK
TURKU UNIVERSITY OF APPLIED SCIENCES

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Electronics

2017 | 20

Jarmo Kivekäs

SPECTRUM MONITORING SYSTEM IMPLEMENTATION USING SOFTWARE-DEFINED RADIO

This thesis covers general theory about the applications of radio spectrum occupancy monitoring, and the methods behind it, and describes essential elements of a spectrum monitoring system that was implemented using a National Instruments Universal Radio Peripheral (USRP). The intention for the thesis was to gain deeper understanding of the required components by using the implementation process to aid in identifying challenges. The root causes of the challenges were studied so that their negative impact could be minimized.

The system was implemented using the Python scripting language for high-level functionality, and the GNURaido code libraries were used for signal processing and control of the USRP. Spectrum measurements collected with the system were compared to measurements made with a conventional spectrum analyzer with matching results. The thesis also compares the use of histograms, where power distribution information in the time domain is kept, instead of commonly used spectrograms.

Measurements were done to observe the behavior of phenomena such as CIC roll-off, DC-offset, and the trade-off relations between system characteristics such as frequency resolution, temporal resolution, required computational power, and data set size. The thesis states that the RF front-end of the USRP can be protected from high-power signals by using additional circuitry to sense power. IQ imbalance caused by the front-end can be corrected for by using device-specific empirical calibration measurements.

KEYWORDS:

spectrum monitoring, software-defined radio, digital signal processing

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Elektroniikka

2017 | 20

Jarmo Kivekäs

SPEKTRINMONITOROINTIJÄRJESTELMÄN TOTEUTUS OHJELMISTORADIOLLA

Tämä opinnäytetyö käsittelee radiospektrin monitorointiin liittyvää teoriaa ja kuvailee National Instruments Universal Software Radio Peripheral -ohjelmistoradiolla (USRP) toteutetun spektrinmonitorointijärjestelmän keskeisiä osia ja toimintoja. Työn tarkoituksena oli luoda syvempi ymmärrys tarvittavien komponenttien toiminnasta ja niiden aiheuttamista ongelmista käyttäen järjestelmän toteutusprosessia tukena ongelmien löytämistä varten.

Toteutuksen aikana havaittiin toimintaan vaikuttavia ilmiöitä, joita tutkittiin jotta niiden negatiivinen vaikutus voitiin minimoida. Monitorointijärjestelmän logiikka toteutettiin käyttämällä GNURadio-koodikirjastoa signaalinkäsittelyyn sekä USRP:n ohjaamista varten. Python-skriptikieltä käytettiin korkeamman tason toimintojen toteuttamisessa. Järjestelmällä kerättyä mittaustietoa verrattiin perinteisen spektrianalysaattorin mittaustietoihin, ja todettiin tietojen täsmäävän toistensa kanssa. Työssä verrataan mittaustulosten esittelyä histogrammina, jolloin informaatio tehon jakautumisesta aikatasossa säilyy, toisin kuin tavanomaisessa tehospektrissä.

Työssä selvitettiin, miten ohjelmistoradioiden etuosan epäideaalisuudet aiheuttavat spektrissä näkyviä tasajännitepiikkejä ja kuinka sopimattoman näytetaajuuden valitseminen voi aiheuttaa suuria vääristymiä mitatun spektrin amplitudissa vastaanottimen tekemän uudelleenäytteistämisen jälkeen. Ohjelmistoradiota käytettäessä voidaan spektrin mittauksessa tehdä kompromisseja tarvittavan levytilan, laskentakyvyn, taajuusresoluution ja aikaresoluution välillä, riippuen siitä millaisia ilmiöitä halutaan mitata. Työssä todettiin, että etuosan toimintaan voidaan vaikuttaa laitteen suojelemiseksi suurtehosilta signaaleilta. Etuosan aiheuttamia IQ-näytteistämisen epätasapainoisuuksia voitiin korjata empiirisen kalibroinnin avulla.

AVAINSANAT:

spektrin monitorointi, ohjelmistoradio, digitaalinen signaalinkäsittely

List of abbreviations

CIC	cascaded integrator–comb
DSP	digital signal processing
FFT	fast Fourier transform
FICORA	The Finnish Communications Regulatory Authority
FPGA	field programmable gate array
LNA	low noise amplifier
LO	local oscillator
LSA	Licensed Shared Access
MSPS	mega-samples per second
PHY	physical layer
RBW	resolution bandwidth
RF	radio frequency
SDR	sofware-defined radio
UHD	USRP Hardware Driver
USRP	Universal Software Radio Peripheral

Contents

oo

1 INTRODUCTION

This thesis covers basic aspects of radio spectrum monitoring and some of its applications in modern communication systems.

Spectrum monitoring, in a general sense, involves sensing and interpreting the frequency content of a band of the radio spectrum over time. The complexity of monitoring systems can vary from simple running time-averages of a sensed spectrum to more complex systems that can, for example, decode signaling protocols, store spectrum usage history, and analyze data either in real-time or in post-processing to provide more information.[1][2]

The bulk of work in this thesis is the implementation of a radio spectrum monitoring system consisting of a commercial software-defined radio peripheral and a Linux laptop with custom application logic, post-processing, and data visualization scripts.

2 BACKGROUND

2.1 Spectrum Allocation

Conventionally, bands of the radio spectrum are allocated for use in a particular application, and the rights to transmit on those bands are licensed by a governing body. Licensees may obtain licenses for comparatively long spans of time, during which the allocated spectrum might not be used fully and continuously. Faster data transmission rates and widespread use of radio-based communications means that the efficient use of the available spectrum is increasingly important, as it is a finite resource.[3]

The Finnish Communications Regulatory Authority (FICORA) is the governing body that handles spectrum allocation in Finland. FICORA regulates the use of frequencies 9 kHz – 400 GHz.[4]

The International Telecommunications Union (ITU) is a specialized agency of the United Nations that allocates radio spectrum and satellite orbits. The ITU coordinates with national regulatory authorities to maintain cross-compatible radio regulations globally.

Advances in radio technology allow the implementation of flexible radio systems that reduce underutilization of available RF spectrum[1]. Transmission frequencies, bandwidth, and modulation schemes can be changed in a dynamic way thanks to increased flexibility and accommodate for changes in the available spectrum. Spectrum monitoring is a key technology when considering the use of dynamic spectrum access.[5]

2.2 Applications of Spectrum Monitoring

Spectrum monitoring, or spectrum occupancy measurement, is used to study how effectively a frequency band of interest is used in some geographical area. The level of use is determined based on the proportion of time when the frequencies are in use versus them being unoccupied. Information obtained from spectrum monitoring helps regulatory authorities assess the effectiveness of their current allocations, and plan for future use of the radio spectrum. Spectrum monitoring is also used to improve the accuracy of spectrum usage databases to facilitate sharing of spectrum.[6]

2.2.1 Opportunistic Spectrum access

Opportunistic spectrum access refers to techniques that make it possible for radio systems to use frequencies in a flexible manner, by automatically changing the transmission frequency, modulation method, or the time of transmission depending of were unoccupied spectrum can be found at any given moment.

Licensed Shared Access

Licensed shared access (LSA) is an approach to radio spectrum regulation that allows further use of spectrum that is already allocated to an incumbent user. LSA is based on a framework where the incumbent user, one or more LSA licensees – i.e new users, and the spectrum regulation authority collectively agree on a sharing scheme. The sharing scheme in LSA is controlled in a way that both the incumbent user, as well as the licensee can expect predictable quality-of-service and are protected from interference.[7]

Availability information and spectrum access policies are held centrally in an LSA repository. Spectrum monitoring is a key component for implementing a system such as LSA.

2.3 Spectrum Sensing Methods

Energy detection is commonly used in research applications to determine the utilization of radio frequencies. In a simplistic application, energy detection can be done by digitizing a band of the spectrum using a software defined radio, or a purpose-built spectrum analyzer. A binary decision about whether a particular frequency is in use is made by comparing the received RF energy on that frequency to a fixed threshold value.[3]

Energy detection using a fixed threshold is problematic. Threshold values that are set manually are error prone, and may need re-adjustment depending on the environment in which measurements are done. A threshold that is set too high will cause false negatives when a signal that is present is not strong enough to pass the threshold. Similarly, if the threshold value is too low, false positives may be triggered by noise, whether man-made or otherwise, that exceeds the threshold.[3]

Advanced Methods

Mathematically more complex and compute-intensive methods relying on autocorrelation and correlation distance based algorithms are also used besides energy detection. These techniques have the advantage of producing more reliable results in environments that change over time, as they can account for the changes. Even a spectrum monitoring node located in a fixed geographical location may observe changes in the noise level in the surrounding environment. Changes can occur on different time scales, anything from momentary spurious emissions from radio-based communication systems, differences in man-made noise depending on the time of day or events on longer time spans such as new buildings being constructed in an urban area.[3][1]

To maintain long-lasting spectrum occupancy measurements or a viable network of spectrum monitoring sensors, it is important that monitoring nodes can operate without the intervention of a technician. Needing to constantly update detection thresholds on monitoring sensors is

time-consuming and error prone. Using more intelligent decision-making algorithms increases the quality of the data produced by a measurement campaign.[1]

Sensing applications can be either generalized or designed for a specific transmission type, to monitor the use of a particular radio system. Energy detection, autocorrelation, and correlation distance based system are generalized techniques for determining occupancy. A system-specific monitoring application may be able to provide more useful information about the use of a spectral band compared to a general solution by demodulating and decoding signals to determine eg. The number of timeslots used in a time-domain multiplexed communications network. Having free timeslots available means, that while the band is technically occupied, there is still throughput capacity available in the network.[6]

2.4 Software-Defined Radio

An ideal software-defined radio peripheral is in simple terms a fast analog-to-digital converter (ADC) that's attached to an antenna. SDR peripherals are used to digitize a band of the radio spectrum which is then either processed in real-time, or it can be written to non-volatile storage and processing of data can happen at a later stage.

In an ideal SDR solution, the antenna would be connected directly to the ADC. In actual applications, it is necessary to use an RF front-end. Typical parts of such a front-end include a band select filter, a low-noise amplifier (LNA), and a mixing stage. The signal conditioning done by the front-end is needed to shift the wanted signals to lower frequencies so that they are within the bandwidth limit of the ADC.[8]

3 IMPLEMENTATION OVERVIEW

This section describes the implementation of a radio spectrum monitoring system using an NI USRP as the antenna interface. The application logic of the spectrum monitor was implemented in the Python[9] scripting language by utilizing the open-source GNURadio[10] software suite and adjacent code libraries for DSP algorithms, visualization, and controlling the USRP.

3.1 The Universal Software Radio Peripheral

The USRP is a software-defined radio platform that is designed for research applications, and it is suitable for spectrum sensing applications.[2][11]

The majority of practical work in this thesis was done using a USRP-2932 programmed with the USRP Hardware Driver (UHD) firmware.

The USRP has an FPGA that can be used for simple signal processing. However, due to the small size of the FPGA, it is limited in its capability and cannot be used to implement complex physical layer (PHY) DSP blocks for signal decoding. The FPGA's main purpose is to do resampling and type conversion of the digitized signal as well as handle network communication with the host PC, sample streaming, and control the RF daughterboard.[12][13]

3.2 Control Flow

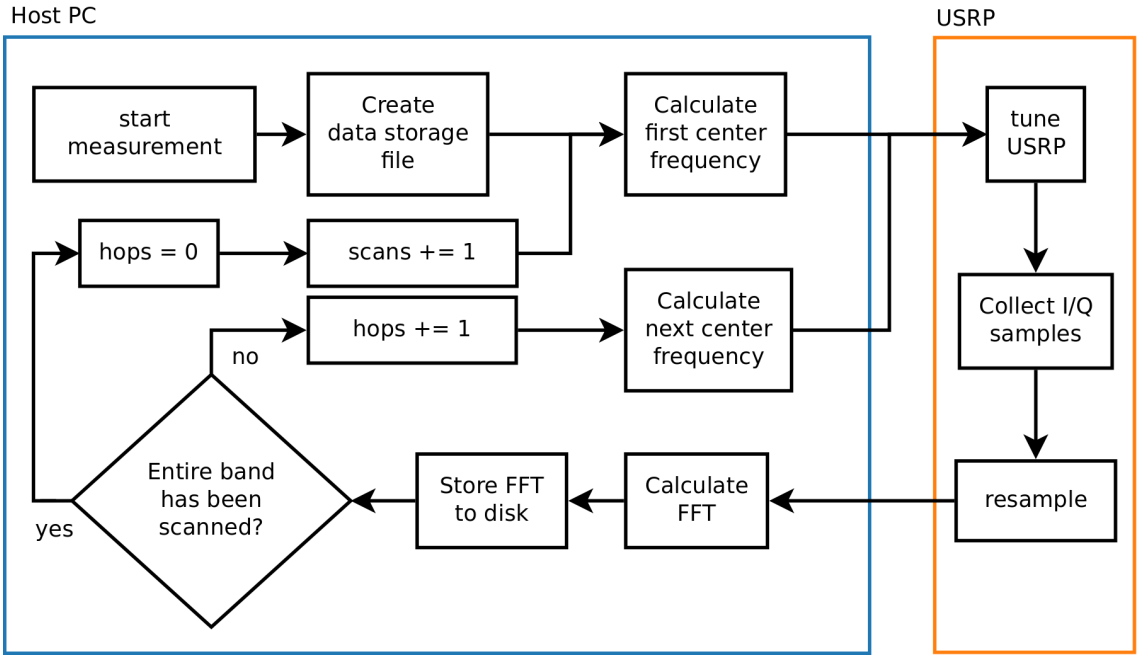

Figure 1: Overview of the system control flow.

An overview of the monitoring system's control flow is presented in Figure 1. The USRP's

center frequency is incremented at regular intervals to complete full scans of a wide band of the spectrum. An FFT is calculated and stored for each hop during the scan.

The difference in the center frequency of each consecutive hop is slightly less than the width of each FFT. This overlap between FFTs can be used to improve data quality by discarding some of the lowest and highest frequency FFT bins, which may suffer from roll-off.

A new identical scan is started every time a scan of the entire band of interest is completed. The application keeps scanning until it is interrupted by a user.

3.3 Data Model

The scan data is represented as a tabular data structure where each row represents a single FFT bin.

The columns in the data structure are

1. a timestamp with millisecond accuracy
2. the center frequency to which the USRP was tuned
3. frequency of the FFT bin
4. magnitude of the FFT bin's power
5. a counter value for which hop of a scan the FFT bin belongs to, and
6. a counter value for which scan iteration the hop belongs to.

Representing the data in the described manner allows for easy manipulation of the data with existing tools at the cost of increased data set size due to redundancy.

Developing a more storage-efficient data model is beyond the scope of work for this thesis.

3.4 Data storage on disk

The collected spectrum data can be stored long-term in plain text files as comma separated tabular data in .csv files. This makes it easy to import the data into a large variety of applications for post-processing.

Alternatively, the data can be stored in a more compact way using Python's native pickle storage or compressed text files. This is for collecting data over long periods of time without being constrained as much by storage space limitations. The redundant representation of the data makes it compressible, with compressed files being commonly $1/5$ of the original size.

4 MEASUREMENTS AND OBSERVED PHENOMENA

4.1 Available Sample Rates

The sample rate that is chosen impacts the speed of scanning and the available frequency resolution.

The USRP is able to stream complex samples over its Gigabit Ethernet interface at rates of up to 50 MSPS at an 8-bit resolution and 25 MSPS at a 16-bit resolution. The resolution of the 16-bit samples is 14 bits in practice, which is the accuracy of the ADCs used for sample acquisition. The acquisition ADCs have a sample rate of 100 MSPS, the available sample rates are limited by the throughput capacity of the Ethernet host PC interface.[13]

An 8-bit sample refers to a sampling scheme where 8 bits are used to each of the in-phase (I) and quadrature (Q) samples, making the I/Q sample pair a total of 16 bits in size. Similarly, an I/Q pair of 16-bit samples is 32 bits in size.

The USRP and GNURadio ecosystems for signal processing primarily use I/Q-sampling when representing waveforms digitally. The Nyquist frequency for complex sampling is equal to the complex sample rate. In this context, passband bandwidth is often shown as the same value as the signal sample rate. In fact, passband bandwidth is often referred to as the sample rate.

4.2 CIC Roll-Off

Cascaded integrator-comb filters, CIC filters for short, are a class of hardware-efficient finite response filters that are used for decimation and interpolation of a signal.[14]

The USRP's integrated FPGA processes samples at 100 MSPS from the antenna ADC. The samples are downsampled to a lower sample rate to transfer them over the Gigabit Ethernet interface to the computer using a CIC filter.

The chosen sample rate has a significant impact on the quality of the scan data. Choosing an inappropriate sample rate will cause the data to have CIC roll-off artifacts from the filter that is involved in the down sampling.

The input sample rate to output sample rate ratio of the conversion needs to be even in order to avoid CIC roll-off:

$$\frac{rate_{in}}{rate_{out}} \mod 2 = 0$$

The CIC roll-off is at its worst when the ratio is odd.

CIC Roll-Off Measurements

The measurements presented in figure 2 show the manifestation of CIC roll-off at two distinct sample rates. The resampling is done from 100 MSPS, making the rate ratios are 100 MSPS / 20 MSPS = 5 (odd) and 100 MSPS / 25 MSPS = 4 (even)

The sample rates 20 MSPS and 25 MSPS were chosen to show the most extreme and least extreme cases of CIC roll-off while still maintaining the highest available sample rate, in this case, using 16-bit samples. Choosing a high sample rate allows for measuring a wider band of the spectrum at once, which is desirable in the context of the spectrum monitoring application presented in this thesis.

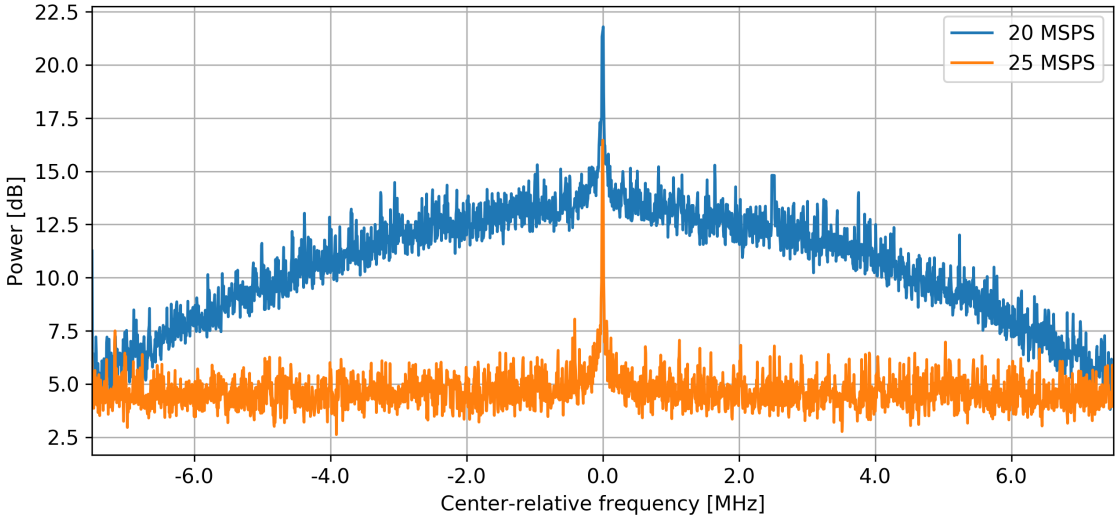

Figure 2: Sample rates chosen to maximize (20 MSPS) and minimize (25 MSPS) the effect of CIC roll-off.

Measurement configuration:

- radio peripheral: USRP-2932
- host interface: Gigabit Ethernet
- FPGA DSP rate: 100 MSPS
- I/Q sample rate: 25 MSPS, 20 MSPS
- I/Q sample depth: 16-bit
- load: 50 ohm RF terminator

The carriers visible in the center of plotted spectra in figure 2 are DC-offset artifacts caused by phenomenon unrelated to CIC roll-off.

The measurements verify what should be there in theory, is also observable in practice.

4.3 Frequency Resolution

Sample rate and FFT size (bin count) determine the greatest available resolution in the frequency domain.

The frequency resolution is given by

$$resolution\ [\text{Hz}] = \frac{samplerate\ [\text{Hz}]}{FFTsize}$$

.

That is to say, by capturing a narrower band of the spectrum, it is possible to achieve more granular frequency resolution with the same amount of computation.

Increasing the number of bins in an FFT increases the amount of computation required. It is possible to save the raw I/Q samples to disk, and compute the large FFTs in a post-processing step where real-time computation is not required. In this case, a likely bottleneck will be the amount of aivailable storage space. The lowest sample rate supported by `uhd_rx_cfile` is approximately 0.2 MSPS, which will produce close to 0.8 MB of data per second when using 16-bit samples. The maximum sample rate 25 MSPS produces 100 MB data per second.

GNURadio and Baudline[15] both require the FFT sizes to be powers of two (2^n), due to the algorithms used.

Major factors limiting sample rate are the rate of the SDR peripheral's ADC, throughput available for transferring samples to the host PC, and the computational load that has to occur in real-time on the host PC.

Frequency Resolution Measurement

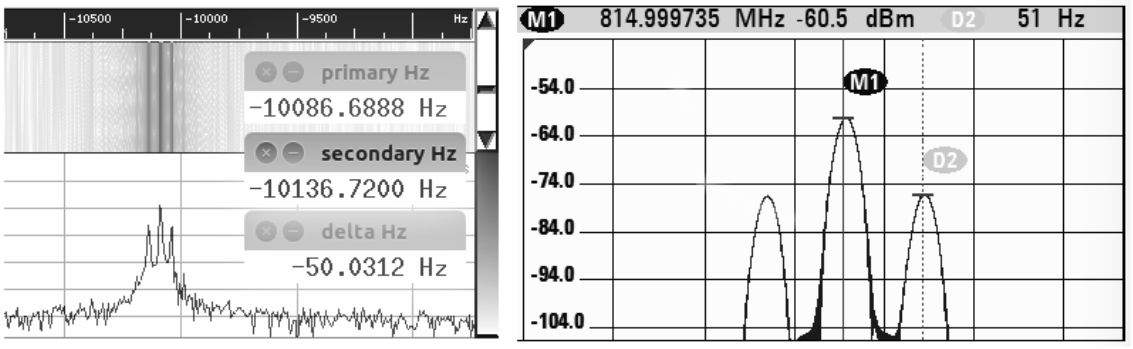

Figure 3: Baudline used to distinguish between 50 Hz peaks (left). Comparisons done with an FSH4 spectrum analyzer (right).

Raw I/Q samples were recorded with a USRP into a file using the `uhd_rx_cfile` program that is part of the USRP + GNURadio ecosystem. The samples were recorded at 0.2 MSPS, a rounded sample rate close to the lowest supported sample rate, which is 0.195312 MSPS.

The recorded files do not contain any metadata about sample types or sample rates, only the raw sample values. The following settings were used to read the recorded files into baudline:

- decompression: off

- initial byte: 0
- sample rate: custom, 200000 samples per second
- channels: 2, quadrature, flip complex
- decode format: 32-bit float, little endian
- normalization: 1.0
- transform size: 65536

The absolute minimum resolution in the frequency domain with the given configuration is

$$200000 \text{ Hz}/65536 \approx 3.05 \text{ Hz}$$

.

Figure 3 shows that Baudline is capable of distinguishing between distinct peaks 50 Hz apart. 20 Hz and 15 Hz gaps could be also observed during the test, although at times this required choosing a different windowing mode for the FFT, and even then the results were not always consistent. Sine wave peaks too close to each other start melding together into one peak in the FFT due to spectral leakage, where the power of a sine wave is distributed among neighboring frequency bins.

The effects of spectral leakage can be influenced by choosing different windowing methods. Different windowing methods can be used for example depending on whether it is more important to have narrow peaks or accurate amplitude information. A Hann window is often used since it has little spectral leakage, and good frequency resolution.[16]

While the FSH4 spectrum analyzer and the USRP are both capable of measuring the spectrum of the test signal, the main difference comes in temporal resolution. The FSH4 used 7.8 seconds to obtain a single measurement of a 570 Hz span of spectrum phase imbalance FFTs can be computed for each individual new sample when recording I/Q samples with an SDR peripheral such as the USRP. In theory, this means the temporal resolution at which FFTs can be obtained in this example is 1/200000 Hz = 5 µs.

4.4 Noise Floor

Figure 4 shows a measurement of the relation of an FFT's noise floor as a function of the FFT bin count. The bin count of an FFT acts in a similar way as resolution bandwidth (RBW) in analog spectrum analyzers. The measurement conforms with what was expected, the noise floor drops by 3 dB when the number of bins is doubled. Doubling the bin count results in the bandwidth of each individual bin to be half of what it was earlier. Half the bandwidth means the power of the measured noise is cut in half as well, a 3 dB difference.

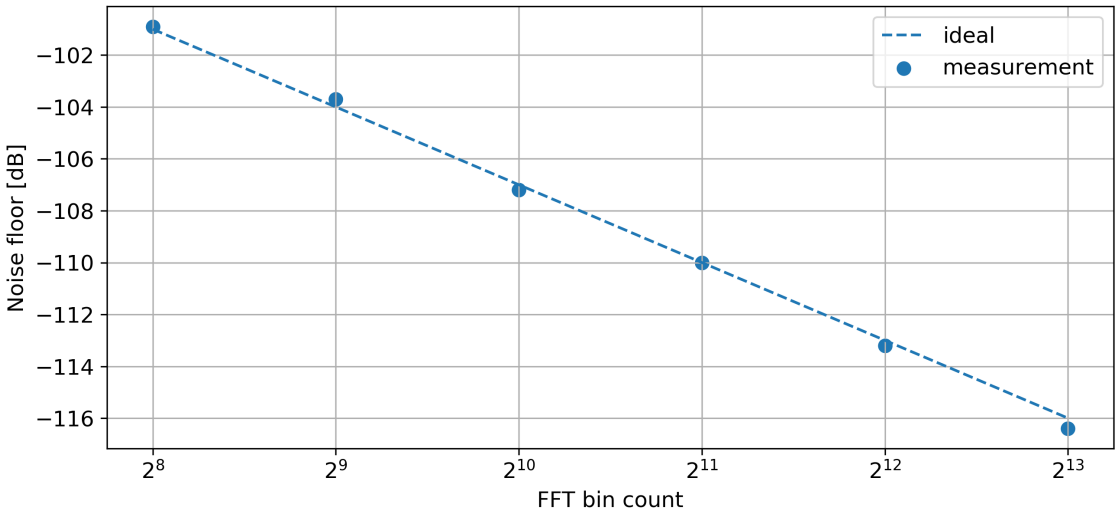

Figure 4: The level of the noise floor can be lowered by increasing the number of FFT bins.

4.5 Visualization & Interpretation

The spectrum monitoring system presented in this thesis includes a visualization mechanism for showing histograms of spectrum usage over time. As the name implies, histograms can be used to display the distribution of measured data points over time in a single graph.

All the measurements presented in this section were done with a D470-860FN1 antenna made by Aerial OY. The antenna's bandwidth is 470 – 860 MHz. The measurements were done indoors at the Turku University of Applied Sciences radio laboratory.

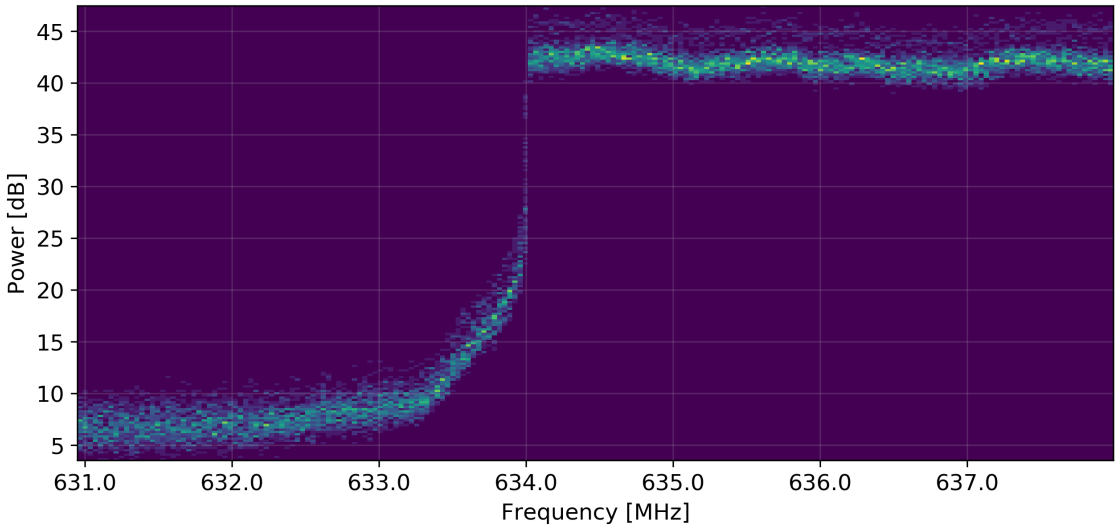

Figure 5: The edge of a high-throughput digital signal without downtime.

Figure 5 shows the spectrum histogram at the low-frequency edge of a Digital Video Broadcast (DVB) signal. The histogram shows no data points at the noise floor's level (around 7 dB)

above frequencies of the DVB signal's lower edge at 634 MHz, which means the signal was likely to be present 100% of the time. It is not guaranteed that the signal was present at all times, as there is a delay between each time the spectrum is measured, making it possible for a signal to not be present for a short while and return before the next measurement is made. Shortening the delay, therefore increasing the temporal resolution, is foremost a tradeoff in data set size and required computational power.

Increasing the temporal resolution will increase the number of measurements, hence increasing data size. The digitized signal waveform has to be processed on the host PC before completing each measurement. More computation power helps with increasing temporal resolution by speeding up the waveform processing.

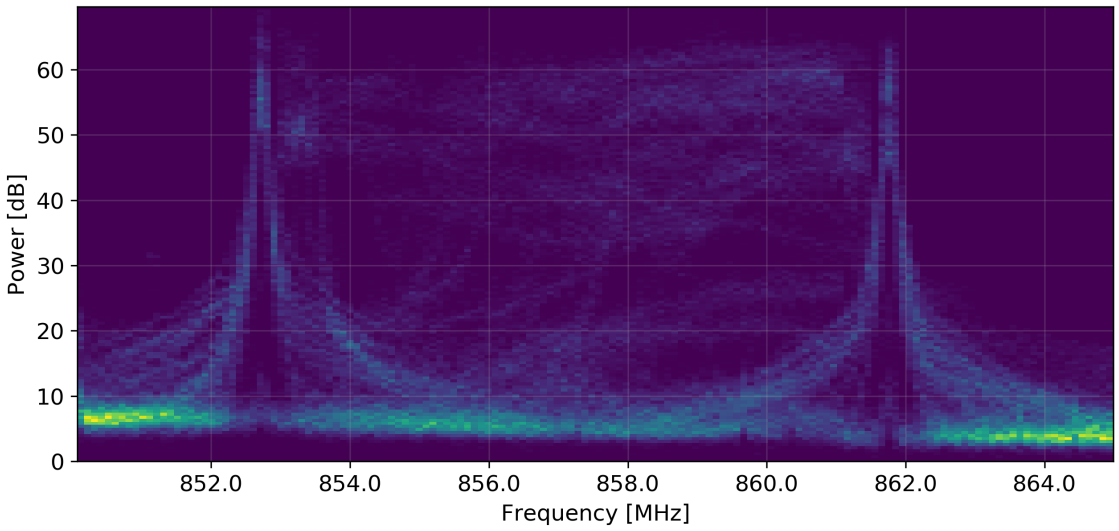

Figure 6: A mobile communications band not in use 100% of the time.

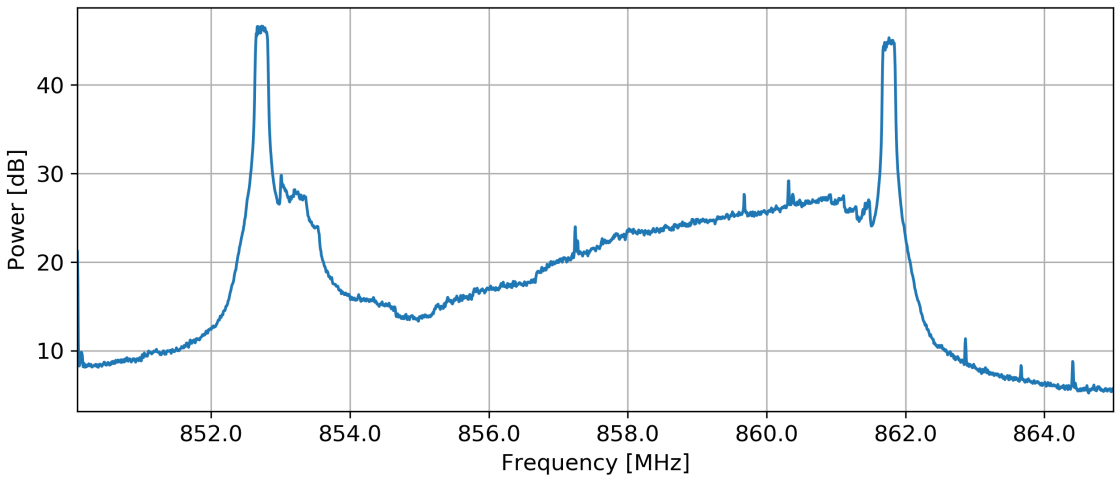

Figure 7: Time-average of the measurements shown as a histogram in figure 6.

Figure 6 shows a measurement of a band of spectrum used for mobile communication. The

power distribution shows a visible noise floor within the allocated frequency band, which indicates that the frequencies are not constantly in use and there are periods of time where no signals are present. Figure 7 shows the same spectrum measurement represented as a time-average plot of the power. The partial absence of signals in this example is not clearly visible in the type of running time-average typically found in traditional spectrum analyzers. The intermittent use of the band shows up as lower average power, but the band still seems constantly occupied over time.

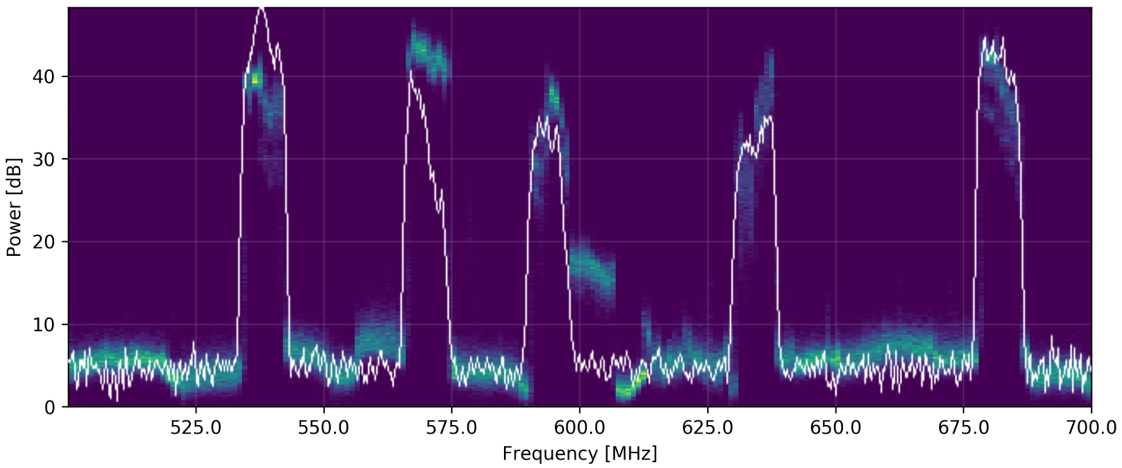

Figure 8: 225 MHz span of spectrum stiched from multiple measurements.

Figure 8 showcases a wide-band data set which was obtained by scanning the band and stitching several consecutive measurements into one dataset. The white line figure 8 is a comparison measurement done using a Rhode & Schwarz FSH-4 spectrum analyzer. The comparison shows that the spectrum measured by the USRP does not have excessive amounts of interference created internally by the USRP hardware, as the measured spectrums are similar.

It should be noted that the comparison measurements are not from the exact some point in time, as both devices measure the spectrum in different ways, and neither can obtain a snapshot of the band from a single moment in time.

5 THE RF FRONT-END

5.1 Protecting The Radio Peripheral

An inherent quality of radio spectrum monitoring applications is that receiver equipment needs to be able to cope with very high-power transmissions in order to have a robust system. When tuning the monitoring system to a certain band, one can not be certain of what signals will be present at those frequencies.

In spectrum monitoring, it is often desirable to be able to record the presence of signals regardless of their power. A general monitoring application may not have given standardized specifications to conform to in the same manner as a traditional receiver purpose-built for a specific communications application. A purpose-built receiver doesn't necessarily need to work with signal strengths that are too low or too high to conform with the expected signal strengths given in the application's standard. Out-of-spec signals can be rejected and ignored.

Simplified measurement of the RF energy present on a band can be done on an attenuated version of the signal to set the receiver's gain to an appropriate level and protect the monitoring system from damage caused by high-power signals.

This added level of protection may come at the expense of performance, as sensing the power of a band and setting the receiver gain before tuning the radio peripheral to the band in question will take some amount time. This time can accumulate if the system is used to scan a wide band of spectrum by constantly re-tuning the radio peripheral, making each full scan take longer.

5.2 DC-Offset

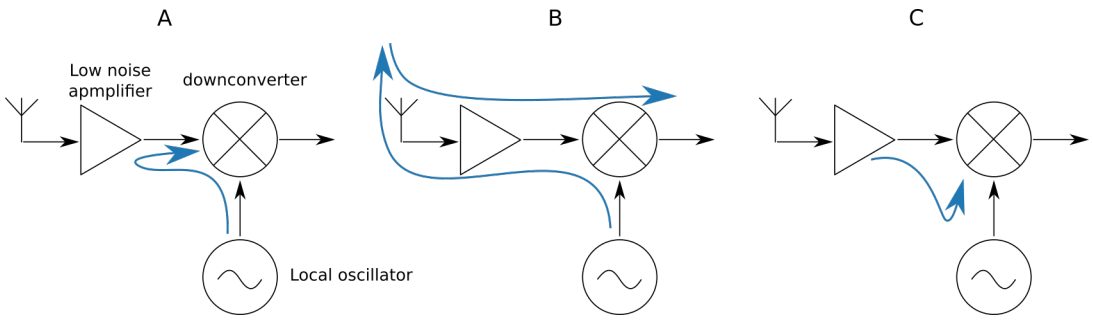

Figure 9: Mechanisms for DC-offset: A) LO leakage, B) LO re-radiation, C) in-band interference.

It's common to see an interference artifact at the center of the band captured by a software defined radio. The interference is a DC-offset caused by the direct-conversion receiver in the RF front-end which downmixes signals to the baseband before digitizing the signal. This phenomenon was observed with different software-defined radio peripherals including the NI USRP-2932, and two different commercial DVB-T -tuner style radios.

Strong local signals or the receivers own local oscillator (LO) can self-mix with itself down to zero-IF, which causes the DC-offset.

LO leakage, LO re-radiation, and having strong in-band interference are the main mechanisms causing DC-offset. The LO is a relatively strong signal in order to accomplish mixing in the downconverter. The LO signal can leak through unintended paths into the LNA in the in the front-end, where it reflects back and is fed into the downconverter where it is mixed with itself and causes a DC signal in the downconverter's output. The interference can be even stronger if the LO leaks into the LNA's input and is therefore amplified before self-mixing.[8]

The LO signal can unintentionally end up radiating from the receiver's antenna and end up reflecting the signal from the environment back into the receiver's RF front-end. Fading and multipath propagation can cause the reflected LO signal's strength to change quickly, causing the DC-offset's level to vary over time.[8]

Strong external interference at the LO's frequency can also cause DC-offset. The radiated LO signal of a similar nearby receiver is one example of a source for such interference. Interference can also be caused by nonlinearities in the receiver chain before mixing.[8]

DC-offset is caused in the SDR peripheral's RF front-end and is hardware-dependent. The root causes for DC-offset cannot be corrected by choosing different digitizing parameters in the way eg. CIC roll-off can be, although they can be corrected for using DSP after the fact.

Impact On Data Quality

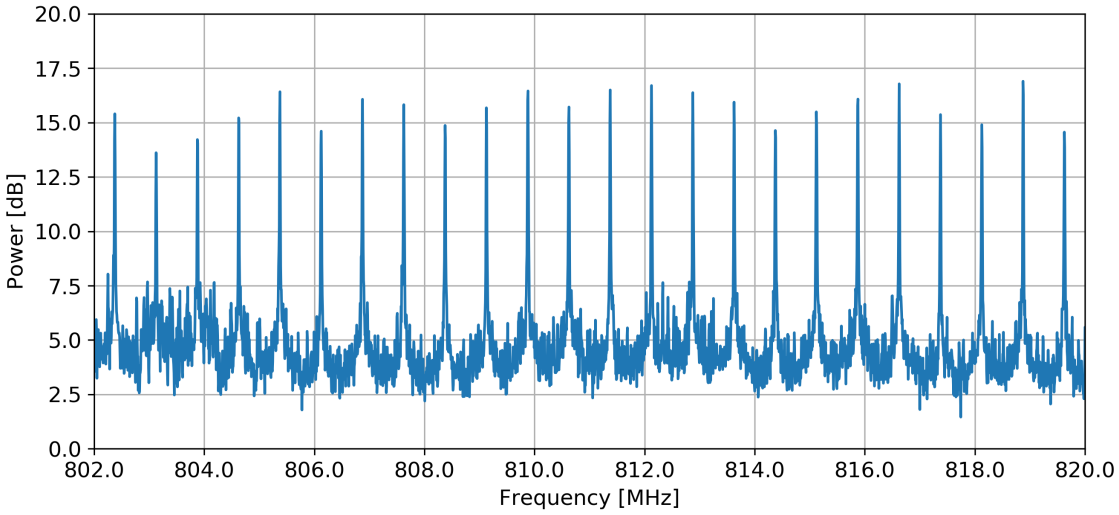

Figure 10: DC-offset can cause severe data quality issues.

Figure 10 showcases how DC offset can cause data quality issues. The spectrum shown in the figure was measured with a 50 ohm RF terminator as the load on the USRP's antenna connector. Each of the peaks in the spectrum are caused by DC-offset at different hop of the scan.

The bandwidth of each measurement hop and the size of the FFT were chosen in a way to make the impact on data quality more severe. Using a narrow passband bandwidth and FFT's with a small bin count results in tightly spaced DC-offset artifacts that have a wide peak.

5.3 IQ Imbalance

The analog RF front-end in the USRP is an IQ receiver. An ideal IQ receiver has two identical signal paths after the mixing stage, one for the in-phase (I) signal, and the other for the quadrature (Q) signal. In practice, the signal paths have slight differences that are inherent to the manufacturing process of electronic circuits, which create IQ imbalance. Propagation delays in each signal path can cause the I and Q signals to not reach the sampling ADCs at the same time, making the phase-offset of the sampled signal something other than 90 degrees. The differences in gain of the signal paths also cause IQ imbalance.[17]

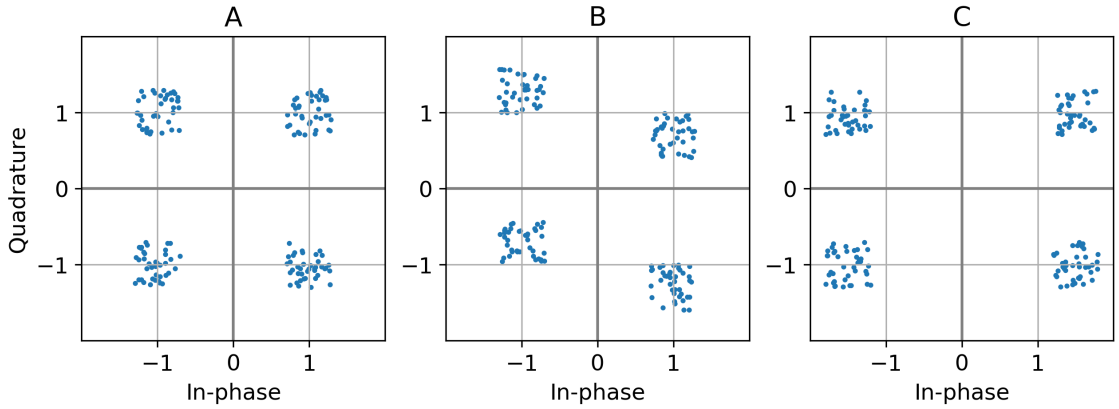

Figure 11: A) No IQ imbalance B) IQ phase imbalance C) IQ amplitude imbalance.

The manifestation of imbalances in the IQ signal paths can be visualized by showing how they impact a signal constellation plot. Figure 11 shows three constellations, constellation A does is a clean constellation without any IQ imbalance, constellation B is vertically skewed due to phase imbalance, and constellation C is stretched horizontally due to amplitude imbalance. The constellation is skewed or stretched either horizontally or vertically depending which signal path has longer propagation delay or higher gain.[17]

IQ imbalance is hardware dependent and will vary between individual circuit boards due to component variance. Operating temperature and signal frequency also have an effect on IQ imbalance.[18]

The USRP Hardware Driver (UHD) comes bundled with a calibration utility that can be used to correct for imbalances in the receiver's signal path. The calibration requires no additional hardware, it relies on known signals leaking from the transmitter to the receiver path. The signal leaked into the receiver path is compared to the known baseline, and correction parameters are stored as a function of signal frequency in the USRP's FPGA.[19]

Since the IQ imbalances are temperature-dependent, the USRP should be left powered on for a while before calibration, so that components reach their operating temperature. The temperature of the environment should also be taken into account.

6 CONCLUSION

The initial goal of the work done for this thesis was to develop a radio spectrum monitoring system using a National Instruments USRP. The implemented spectrum monitoring system was tested, and implements the intended features. Much of the benefit in the work done was not in the implementation itself but in the process thereof: in researching available tools, figuring out how to integrate them, and learning how software-define radio ecosystems work on a lower abstraction level. Running into issues and unexpected measurement results during the development process prompted tangential projects into researching what configuration parameter or circuit is the root cause behind some phenomenon. Much of that knowledge is applicable in other software-defined radio and general RF system even beyond the context of spectrum monitoring.

References

[1] S. Grönroos, *Efficient and Low-Cost Software Defined Radio on Commodity Hardware*. PhD thesis, Åbo Akademi University, 2016.

[2] National Instruments, "Spectrum Monitoring With NI USRPs." http://www.ni.com/white-paper/13882/en, 2015. [Online; accessed 16.5.2017].

[3] S. Subramaniam, H. Reyes, and N. Kaabouch, "Spectrum occupancy measurement: An autocorrelation based scanning technique using usrp," in *2015 IEEE 16th Annual Wireless and Microwave Technology Conference (WAMICON)*, pp. 1–5, April 2015.

[4] Finnish Communications Regulatory Authority, "Radio frequency regulation," 2015.

[5] M. Zennaro, E. Pietrosemoli, A. Bagula, and S. Nleya, "On the relevance of using affordable tools for white spaces identification," in *2012 IEEE 8th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp. 606–611, Oct 2012.

[6] M. Höyhtyä, A. Mämmelä, M. Eskola, M. Matinmikko, J. Kalliovaara, J. Ojaniemi, J. Suutala, R. Ekman, R. Bacchus, and D. Roberson, "Spectrum occupancy measurements: A survey and use of interference maps," *IEEE Communications Surveys Tutorials*, vol. 18, pp. 2386–2414, Fourthquarter 2016.

[7] VTT Technical Research Centre of Finland, "Licensed shared access (lsa)." http://core.willab.fi/?q=system/files/1406%20MAE1.pdf. [Online; accessed 9.12.2016].

[8] R. Svitek and S. Raman, "DC Offsets in Direct-Conversion Receivers: Characterization and Implications," in *IEEE microwave magazine*, 2015.

[9] "Python." https://www.python.org. [Online; accessed 10.5.2017].

[10] "Gnuradio, the free & open source software defined radio ecosystem." https://www.gnuradio.org. [Online; accessed 10.5.2017].

[11] L. Angrisani, D. Capriglione, G. Cerro, L. Ferrigno, and G. Miele, "Employment of software defined radios for dual-use in distributed spectrum monitoring system," in *2016 IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI)*, pp. 1–5, Sept 2016.

[12] National Instruments, "questions about GNU radio," 2013.

[13] Ettus Research, "USRP N200-210 Datasheet." https://www.ettus.com/content/files/07495_Ettus_N200-210_DS_Flyer_HR_1.pdf, 2012. [Online; accessed 26.2.2017].

[14] M. Donadio, "CIC Filter Introduction." https://dspguru.com/files/cic.pdf, 2000. [Online; accessed 10.5.2017].

[15] "Baudline." http://baudline.com. [Online; accessed 10.5.2017].

[16] National Instruments, "Understanding FFTs and Windowing." http://download.ni.com/evaluation/pxi/Understanding%20FFTs%20and%20Windowing.pdf, 2016. [Online; accessed 24.5.2017].

[17] National Instruments, "Sources of Error in IQ Based RF Signal Generation." http://www.ni.com/tutorial/5657/en/, 2016. [Online; accessed 3.6.2017].

[18] Ettus Research, "USRP Hardware Driver." https://kb.ettus.com/UHD, 2017. [Online; accessed 29.6.2017].

[19] Ettus Research, "USRP Hardware Driver and USRP Manual." http://files.ettus.com/manual/index.html, 2017. [Online; accessed 30.6.2017].