FUDMA Journal of Sciences (FJS)
ISSN online: 2616-1370
ISSN print: 2645 - 2944
Vol. 4 No. 3, September, 2020, pp 17 – 25
DOI: https://doi.org/10.33003/fjs-2020-0403-167

DEVELOPMENT OF SOFTWARE FOR QUANTUM VOLTAGE SYNTHESISER

Abubakar Y. M.

Department of Physics, Ahmadu Bello University Zaria, Kadua State, Nigeria

*Corresponding Author's Email: myabubakar@abu.edu.ng

ABSTRACT

Signal analysis is a wide area of research, which received considerable attention due to its numerous fields of application. The impact of these fields in technological advancement cannot be overemphasized. No matter how good a signal is acquired, it can still contain some unwanted components (noise) that is capable of undermining its integrity. This work synthesizes signal from a delta-sigma system by designing digital filters LabVIEW program. The designed lowpass and bandpass filters were examined in addition to different filter design algorithms. Based on the analysis, Dolph-chebyshev and Kaiser window filters provide excellent filtering by blocking quantization noise to up to - 40dB for fixed point design and, -60dB for floating point. Despite issues with windowing in filter design, these two algorithms seem to work well with this type of signal. Additionally, this design can be optimised by pushing the filter to its limit.

Keywords: Digital filter, digital signal processing, analogue to digital conversion, quantum voltage detection.

INTRODUCTION

Digital signal processing (DSP) is a key aspect in signal acquisition and transmission process. Signals are normally produced either in digital or analogue form. Several systems such as sensors, produce analogue signals in their output. The signal can be processed in the analogue or digital domain. The analogue signals can be converted to digital by sampling and quantization; a process that is popularly called Analogue-to-Digital conversion. DSP involves manipulating the converted signal and extracting information from it (Kehtarnavaz & Kim, 2011). Converting analogue signal to digital form allows programmability; such that a single hardware may be used for different applications. The only thing that changes is the code in the memory.

DSP is an established technology that recorded applications in a number of areas such as communication, medical imaging, radar, music and oil prospecting to mention a few (Smith, 2003). One of the most important components of the DSP is digital filtering. Filtering refers to removal of unwanted signal (noise) from a sampled or acquired signal (Smith, 2002).

To meet the increasing demands of industry National Physical laboratory (NPL) is developing a voltage source able to produce waveforms directly traceable to fundamental quantum constants. This Quantum Voltage Synthesiser utilises optically coupled pulse driven Josephson junction arrays (JJA's) to generate quantum accurate voltages. The desired waveform is produced by controlling the pulse stream sent to the arrays using Delta-Sigma modulation.

In this work various options of digital filter design are considered. LabVIEW-FPGA (Fast Programmable Gate Array) codes were developed and deployed into PXIe system. This digital filter design forms part of a complete system of quantum voltage detection (QVD) developed at the national physical laboratory (NPL) Teddington, London. This QVD is an innovative way of tracing the definition of Voltage from first principle.

OVERVIEW OF THE EXISTING DELTA-SIGMA SYSTEM AT NPL

Delta-sigma (D-S) modulation involves sampling signal at a very high frequency such that the noise in the signal will be pushed to higher frequency regimes. The D-S system consists of difference amplifier, integrator comparator and digital-to-analogue converter (DAC) as shown in figure 1.

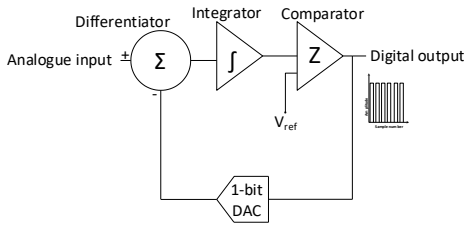

Figure 1: *Schematic diagram of delta-sigma system*

The D-S converter samples a low frequency analogue signal multiple times (over- sampling). Each sample is recorded over a period and averaged. The converter has two sampling rates: the input sampling rate (f_s) and the output data rate f_D (Baker, 2011). The modulator part (which is differential amplifier and integrator) digitizes the analogue signal and reduces its noise at lower frequencies through a process called noise shaping. This operation pushes the noise to higher frequencies outside the band of interest. The dynamic range of D-S as analogue-to-digital converter (ADC) is limited by quantization noise. This is a rounding-off error due to conversion of analogue to digital

signal. As an example, if the output 1 V (analogue) is 01(digital), then 0.75 V (analogue) would still have the output of 01(digital), corresponding to 1 V (analogue input). The difference (i.e. 1 - 0.75=0.25) is called quantization error whose power e_{rms} is represented by (Jarman, 1995)

$$e_{rms} = \frac{1}{q}\int_{-\frac{q}{2}}^{\frac{q}{2}} e^2\, de \quad (1)$$

where q is the quantization interval, e, de or least significant bit (LSB). The next step is removing this noise from the output code using a low pass filter. The noise being pushed to higher frequencies is easily visualised when the conversion is performed in frequency domain or the output data is displayed in frequency domain by using fast Fourier transform (FFT). Here, the choice of filter type including windowing is very critical because, some of the signal (or its components) may be lost in the process (Kehtarnavaz & Kim, 2011).

The QVD system at NPL includes field programmable gate array (FPGA) and Josephson's junction arrays (JJAs) as depicted in figure 2.

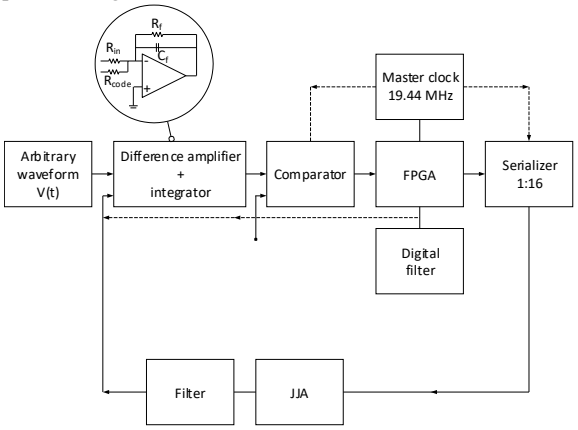

Figure 2: *Schematic diagram of delta-sigma system at NPL. The broken lines show the digital code fed directly into the D-S electronics*

These components are connected to the D-S first order converter to form a feedback loop. The master clock synchronizes between FPGA read/write and the serializer. Comparator level is set by the V_{ref} and can be latched at a rate synchronized with the master clock (Ireland et al., 2016).

DIGITAL FILTER DESIGN FUNDAMENTALS

Filters are normally used for signal separation or restoration. The former refers to separating a noisy signal from the signal of interest while, the latter imply improving a distorted signal. This correction can be performed by using analogue or digital filter. Analogue filters are fast and cheap with a large dynamic range in that they are constructed using electronic components such as resistors and capacitors. Digital filters on the other hand, have superior performance which normally shifts the emphasis on the limitations of input signals and other issues regarding their processing (Jackson, 1996).

Basically, the sense of sight and hearing in humans work via the detection of waves. The characteristics (such as amplitude and frequency) of these waves are normally used to extract vital information from the wave. Digital signal processing (DSP) converts samples (amplitude of an analogue signal at an instant in time) to numerical values, which are stored and transmitted or processed (Analog Devices, 1995). Some of the advantages of DSP include high signal to noise ratio (SNR) and modification of digital hardware without changing any circuit component (National Instruments, 2019).

In DSP, analogue signals entered into the ADC and exits at the DAC. The signal may be processed in time domain or converted to a frequency domain before processing (Isen, 2009) as shown in figure 3.

Commonly, time domain processing involves filtering a signal. In continuous mode, it is performed using combined electronic components mathematically manipulated using a discrete time processing system. This system comprises of registers. On the other hand, frequency domain processing takes a block of such as operational amplifiers, transistors, inductors, capacitors and resistors (Smith, 2003). In discrete mode, the signal sequence is time domain samples and evaluates the frequency content. Transformation from frequency to time or vice-versa is achieved via Discrete Fourier Transform (DFT); which is implemented using Fast Fourier Transform (FFT).

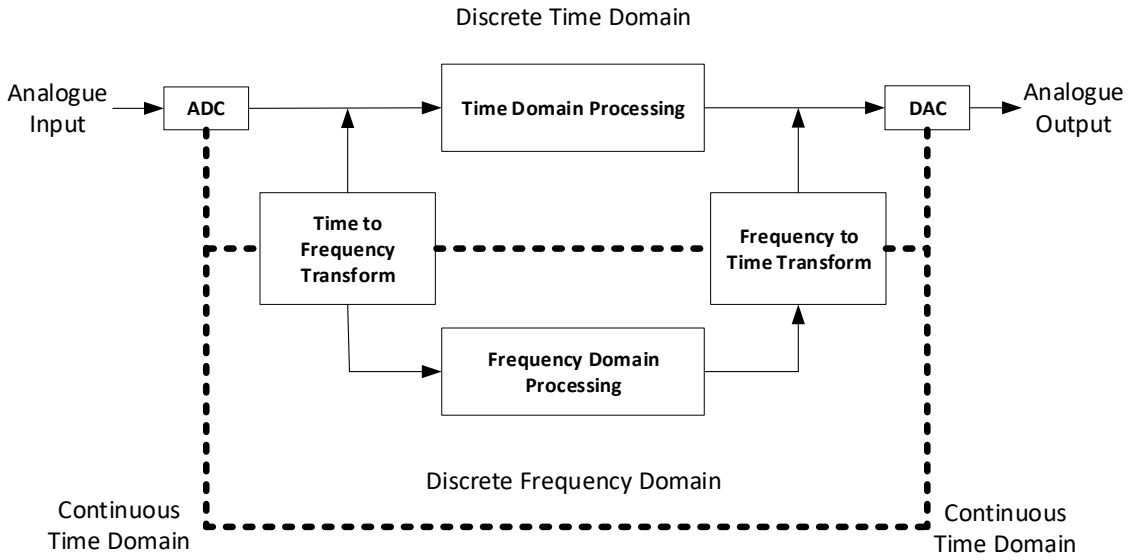

Figure 3: *Working principle of and conceptual overview of digital signal processing* (Isen, 2009).

Some of the terms used in the DSP include:

- Impulse response: is the output of a system when it is fed in with an impulse
- Step response: is the output of a system when the input is a step. Step response is integral of impulse response.
- Frequency response: it can be obtained by taking the FFT of impulse response
- Time domain signal: signal sampled at regular intervals of time
- Frequency domain signal: signal sampled at regular intervals of space

LabVIEW METHODOLOGICAL APPROACH

Apart from choosing the best option for filter design algorithm, other factors to consider are the nature of data to be filtered and the noise in the data. The data acquired may be time or frequency domain. Some filters can easily differentiate between signals of different frequencies (frequency sensitive) while, others cannot (time sensitive filters). For random noise, the best filter is moving average, which is time domain filter (Agilent Technologies, 2008). If the noise is frequency dependent (i.e. sitting in a particular frequency range), the choice will be a frequency domain filter even if the data acquired is time domain. It is worth noting at this point that, best filter in frequency domain often is the worst filter in time domain and vice-versa (Smith, 2002). LabVIEW design approach normally starts with classical design and end with a fixed-point filter using digital filter design tool kit (DFD tool kit). This is the suggested method by National Instruments (National Instruments, 2009). The stages are as follows:

i. **Selecting a filter structure:** This is achieved by placing a DFD classical design.vi on the block diagram from Functions→DFD Design→DFD classical design.vi. Band specifications can be adjusted on the front panel of the virtual instrument (vi). Additional feature is added which allow changing the filter structure using convert structure vi from DFD→conversion→convert structure (see figure 4).

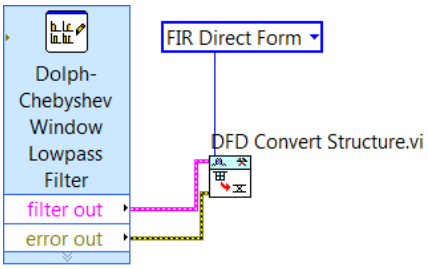

Figure 4: *Block diagram for the first step in designing fixed point filter from classical filter design express vi*

There are more than 20 structures available in this vi ranging from finite impulse response (FIR) to infinite impulse response (IIR) filter.

ii. **Scaling Filter Coefficients:** Consecutive sequence of adders (e.g. $x_1[n] + x_2[n]$) in a filter, are treated as accumulators. These accumulators may use different data ranges, but LabVIEW only provides one sum quantizer. Therefore, it is better to scale all the filter coefficients before quantizing them. In order to do this the DFD scale filter vi is used; which is available in DFD→conversion.

iii. **Quantizing a Floating-Point (FP) Filter:** In practical situations where the filter design is intended to be used in programmable circuits such as FPGA, FP filter cannot be used because,

FPGA only accepts integer-based algorithm. In other to quantize the filter, each value of FP coefficients is approximated to a fixed point (FXP) value. DFD Quantize Coef.vi is the function used for this operation located at DFD→Fixed point tools. For easy or advanced instance option one need to specify other parameters such as coefficients a and b and appropriate gain processing target. Except if the computer is connected to a national instruments (NI) Reconfigurable I/O (RIO) target, the gain processing input is set to host, so that the coefficients are stored on the computer.

iv. **Analysing a Fixed-Point Filter:** At this level it is recommended to analyse the filter parameters in order to study the agreement (or deviation) between FP and FXP filters. The filter analysis vi function is used for this. Filter analysis vi is located at DFD→Analysis.

v. **Creating a Fixed-Point Filter Model:** As soon as it is realised that the FXP filter design has met the requirements, DFD fixed point modelling vi can be used to model the filter. The function is available at DFD→Fixed point tools. To use this vi, one needs to specify word lengths and integer word lengths for the input signal and word lengths for the output signal as well as the rounding mode for the output signal.

vi. **Simulating a Fixed-Point Filter Model:** Simulation is the last level of filter design verification. If the sample data is available, it can be used at this point with DFD FXP simulation function vi. It is possible to filter the signal using FP reference and FXP filters. It is expected that both will look similar in their performance. The simulation report gives statistical information about the quantizers except coefficients a and b. Generally, it reports max value, min value, overflows, underflows, and operations. For optimum performance, overflows and underflows should both be zero or acceptable values depending on the filter performance and resolution required. To remove or reduce non-zero values, integer word lengths are modified in the modelling vi.

Although this design process is arranged in a nice order, it cannot work with all filter design specifications. For example, Equi-ripple and Kaiser window for a signal sampled at 19.44 MHz and cut off frequency of 5 kHz is not practicable (National Instruments, 2014). In addition, some filter design algorithms (such as sinC filter) that are popular in this application because of their ability to separate signals of different frequencies, are not supported by LabVIEW (National Instruments, 2009). Fortunately, there are other ways to achieve the same or superior result without using the express vi's. In fact, express vi's are not recommended for application design because they are very heavy and thus occupy a lot of space and cause unacceptable delays during execution (Kehtarnavaz & Kim, 2011).

Other ways to design the filter include filter design by coefficients (using transfer functions) and using non-express vi's (as demonstrated in figure 3). These are the methods utilized in this design. In designing the filter by coefficients, the filter design algorithm is chosen first (such as DFD Remez order estimator). This sub vi is later used to design the FXP filter using transfer function. The other approach is using DFD classical design vi and following the steps similar to the those stated earlier. With this approach it is possible to design Kaiser window and Dolph-chebyshev window FIR filters. A block diagram for designing Kaiser Window or Dolph-Chebyshev window is presented in figure 5. The vi inside the rectangle is for code generation and its detail discussion is presented in the next section.

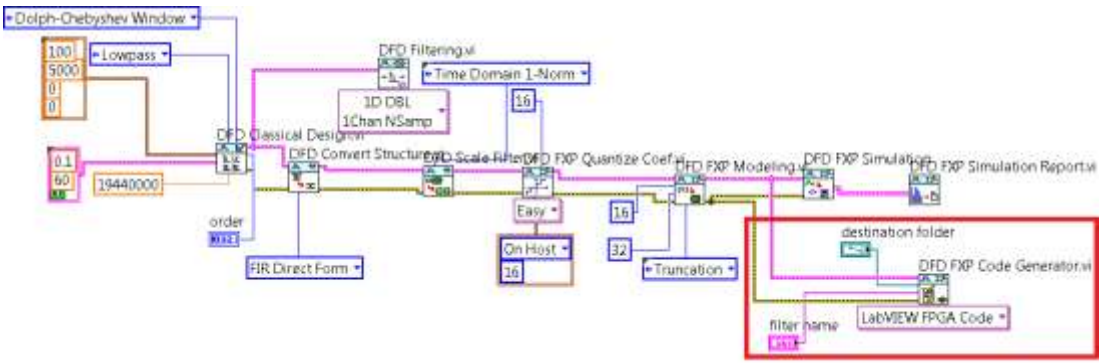

Figure 5: *Block diagram showing the stages followed to design Dolph-Chebyshev filter. The vi inside the rectangular boarder represents the FPGA code generation level.*

Based on the nature of the D-S encoded data (which is in time domain), the best filter should be able to differentiate signals of different frequencies. This is because, after quantisation, the quantisation noise has been shifted to higher frequencies outside the band of interest. Sample data acquired from a signal generator processed by the D-S system is presented in figure 6.

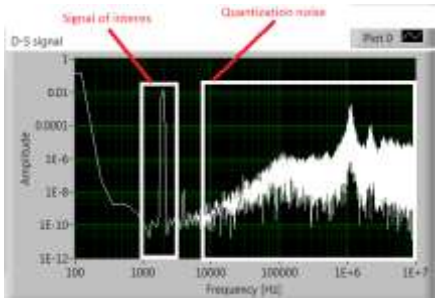

Figure 6: *FFT of signal recorded from a typical signal generator and passed through D-S signal system. The effect of D-S quantisation can be observed with interest band and noise band been clearly separated*

The filter design by transfer function was in two stages: first, coefficients quantization and generation (figure 7); second, design of Remez filter by transfer function (figure 8). In principle Remez filter is very similar to Equi-ripple filter (National Instruments, 2014). Other approaches of filter design were also considered. But because of the nature of the D-S code and the restrictions imposed by PXIe (which is controlled by FPGA code) most of these designs were not adopted. Many LabVIEW digital filter design vi's are based on double precision data, which is not acceptable by FPGA. Although IIR filter design involves fewer operations, it can be very complex to incorporate into FPGA because of its many mathematical operations.

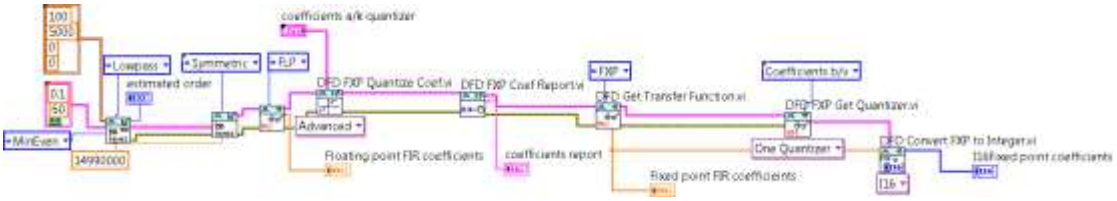

Figure 7: *Block diagram for coefficients quantization display for Remez filter*

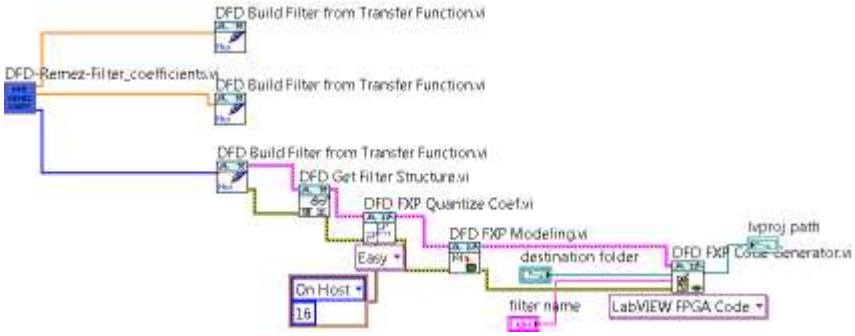

Figure 8: *Block diagram for Remez filter design from transfer function*

LabVIEW-FPGA CODE GENERATION

At the point of implementation of the filter designed, there is need to modify its design to integer based or FPGA code. This can be achieved simply by adding DFD FXP code generation vi in the block diagram as depicted in figure 4. The vi is included in the DFD tool kit module. Code generation vi can also generate C code as well as integer code. In the current application, integer or FPGA code are configured into the PXIe via first input first output (FIFO). Figures 9 and 10 show the LabVIEW integer and FPGA code respectively for Dolph-chebyshev window filter.

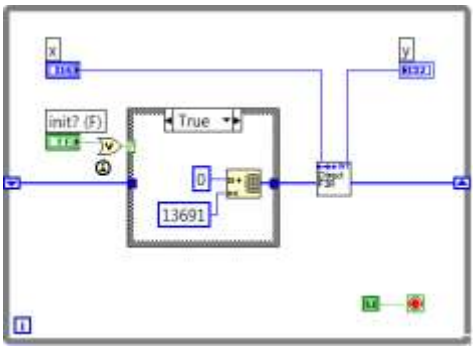

Figure 9: *LabVIEW integer code of Dolph-Chebyshev window filter*

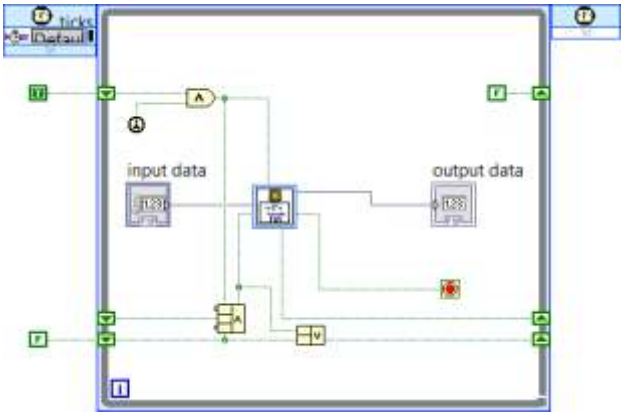

Figure 10: *LabVIEW FPGA code of Dolph-Chebyshev window filter*

DIGITAL FILTER PERFORMANCE EVALUATION

Evaluation of digital filter performance involves studying its main response parameters such as step response, impulse response and frequency response. Apart from these parameters, a simulated or sample signal is passed into the filter to examine its performance. Step response of a filter consists of passband, transition band and stop band as depicted in figure 11. Since the filter may be time or frequency domain, frequency response is used to evaluate frequency domain filter, while the step response is suitable for time domain filter.

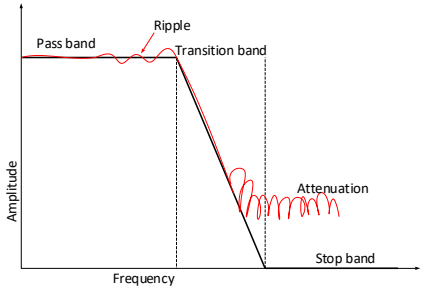

Figure 11: *Frequency response of a low pass filter and its components. The path drawn using red line indicates some abnormalities in practical situations*

A fast roll-off in the transition band is required for a filter intended to be used in the separation of closely spaced frequencies (narrowband). Ripple in the pass band alters the passband frequencies. In order to have the passband frequencies unaffected, there should be no ripple. In the stopband regime, excellent attenuation is desired. Since the choice of filter design in this application is frequency domain, the filters designed were evaluated in this format. In addition, real signal filtering was also assessed using the D-S encoded signal presented in figure 5. Phase

response has not been given particular attention here because; all FIR filters have linear phase response. Even if IIR filter is chosen as an option, there are ways of modifying the digital filter to have zero phase shifts. Frequency responses of some of the filters designed in this work are presented in figures 12 and 13 for Dolph-Chebyshev and Kaiser Window filters respectively.

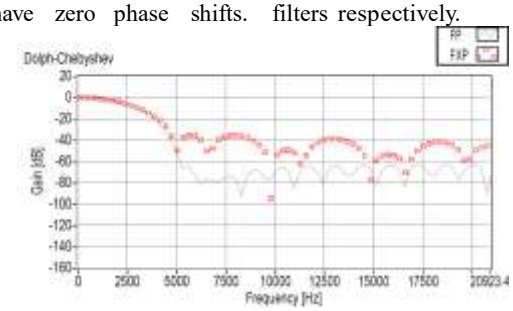

Figure 12: *Frequency response of Dolph-chebyshev low pass filter. The red doted points represent fixed point design and the thick black line is for floating point design*

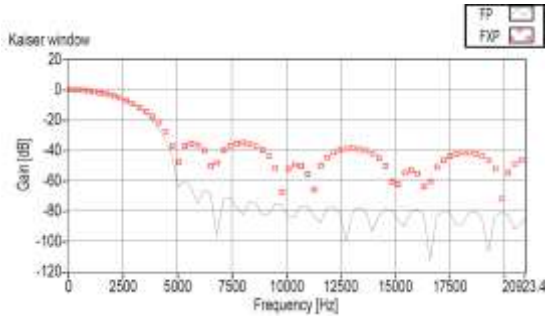

Figure 13: *Frequency response of Kaiser Window low pass filter. The red doted points represent fixed point design and the thick black line is for floating point design*

Apart from this form of evaluation, direct comparison was made between unfiltered D-S signal and the filtered one. Although, there was a warning on using windowing in filter design, it can be concluded that the encoded D-S signal is not affected.

RESULTS AND DISCUSSION

In order to test the effect of the filters designed on the Quantum voltage signal, the original, filtered and decimated signals were plotted on the same graph as shown in figures 14 – 17. It is interesting that, each filter and windowing combination give different treatment to the original signal. In some cases, the amplitude of the original signal is reduced after being passed through the filtering system (see figures 15 and 17).

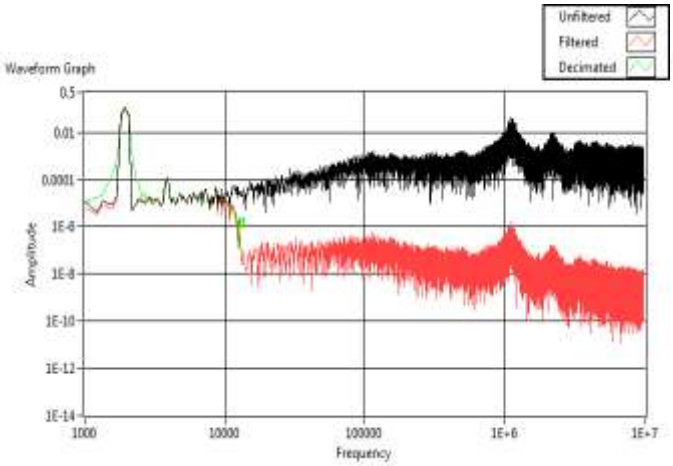

Figure 14 Dolph-chebyshev low pass filter

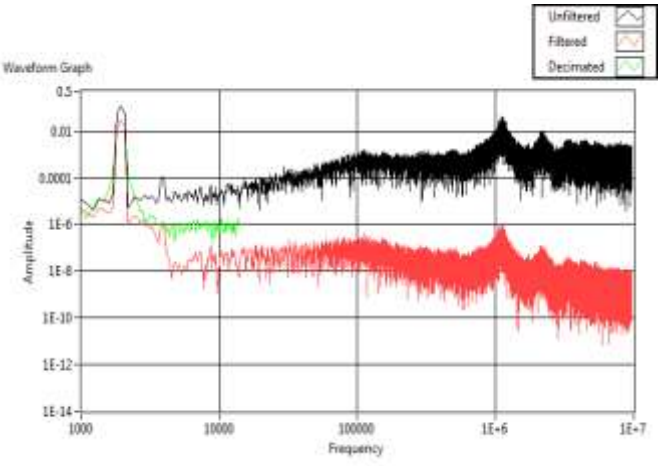

Figure 15 Band pass Hamming

For other designs, the original signal is not reduced, and the noise is reduced to minimum. This implies that, the choice of filtering combination is a tread-off between noise removal completely with some reduction in original signal and reduction of noise while maintaining the size and shape of the original signal.

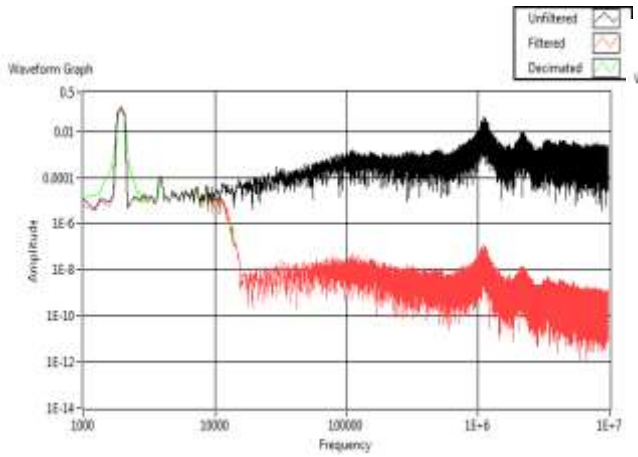

Figure 16 Kaiser window

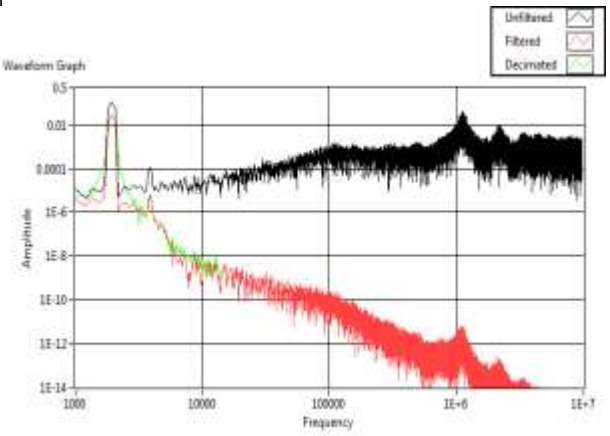

Figure 17 Blackman

Based on the assessment of the performance of each filtering combination, it is clear that Kaiser window is a good option for the application.

CONCLUSION AND RECOMMENDATIONS

So far, a number of digital filters design have been considered for the D-S application. Amongst all the programs designed and tested, Kaiser window appears to have better performance without reduction of quality of the original signal.

One of the best frequency domain filters (sinC filter) is not currently supported by National Instruments (NI) and some filters tested cannot be easily converted to integer or LabVIEW-FPGA code. Other limitations have to do with the LabVIEW software licence. LabVIEW 2013 64 bit does not support FPGA, but 32 bit can support it. Also, programs developed using LabVIEW 2015 and higher versions cannot be saved as 2013 versions due to changes in vi structure during upgrade. Although this design has shown some level of acceptable performance, it is important to note that its performance may vary with the real-time signal in practice and complete loop (i.e. together with JJA's) as depicted in figure 2.

Based on the current D-S system and the digital filters design, optimum performance can be achieved by considering the following recommendations:

- Parameters (band specifications) for Remez filter have to be selected with care because, all the 4 values of frequency specifications are required.
- There is a little discrepancy between FXP and FP filters for Dolph-chebyshev and Kaiser window. These can be improved by varying word lengths (both for output and input) in the modelling vi.
- Simple filter parameters are not enough to evaluate a filter, the raw data is important

as well. This is because filter performance does not depend on its intrinsic properties but also, on the nature and complexities of the data.

REFERENCES

Agilent Technologies. (2008). Designing a Digital Filter. Retrieved 6 August 2020, from http://literature.cdn.keysight.com/litweb/pdf/ads2008/dfilter/ads2008/Designing_a_Digital_Filter.html

Analog Devices. (1995). A Beginner's Guide to Digital Signal Processing (DSP) | Design Center | Analog Devices. Retrieved 6 August 2020, from https://www.analog.com/en/design-center/landing-pages/001/beginners-guide-to-dsp.html#

Baker, B. (2011). How delta-sigma ADCs work, Part 1. *Analog Applications, 3Q*. Retrieved from http://www.ti.com/lit/an/slyt423/slyt423.pdf

Ireland, J., Cryer, A., Williams, M, J., Houtzager, E., Hornecker, R., & Van den Brown, E, H. (2016). *Design of delta-sigma feedback loop for quantum voltage digitizer*. Internal communication.

Isen, F. W. (2009). DSP for MATLABTM and LabVIEWTM I: Fundamentals of Discrete Signal Processing - Morgan & Claypool books. In *DSP for MATLABTM and LabVIEWTM I: Fundamentals of Discrete Signal Processing , Morgan & Claypool*, (p.213). Retrieved from https://ieeexplore.ieee.org/document/6813571?arnumber=6813571

Jackson, L. B. (1996). FIR Filter Design Techniques. In L. B. Jackson (Ed.), *Digital Filters and Signal Processing: With MATLAB® Exercises* (pp. 289–321). Boston, MA: Springer US. https://doi.org/10.1007/978-1-4757-2458-5_9

Jarman, D. (1995). No. AN9504 May 1995 Harris Data Acquisition.

Kehtarnavaz, N., & Kim, N. (2011). *Digital signal processing system-level design using LabVIEW*. Newnes.

National Instruments. (2009). Designing a Fixed-Point Filter - LabVIEW 2009 Digital Filter Design Toolkit Help - National Instruments. Retrieved 10 April 2016, from http://zone.ni.com/reference/en-XX/help/371988C-01/TOC23.htm

National Instruments. (2014). Advanced FIR Filter Design VIs - LabVIEW 2014 Digital Filter Design Toolkit Help - National Instruments. Retrieved 14 April 2016, from http://zone.ni.com/reference/en-XX/help/371988G-01/lvdigfiltdestk/dfd_adv_fir_vis/

Smith, S. W. (2002). *The scientist and engineer's guide to digital signal processing*. California Technical Pub.

Smith, S. W. (2003). *Digital signal processing: a practical guide for engineers and scientists*. Newnes. Retrieved from http://books.google.com/books?hl=en&lr=&id=PCrcintuzAgC&oi=fnd&pg=PA1&dq=%22available.