

LASER INTERFEROMETER GRAVITATIONAL WAVE OBSERVATORY
- LIGO -
CALIFORNIA INSTITUTE OF TECHNOLOGY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Technical Note	LIGO-T11XXXXX-vX	2015/06/05
Report I — LIGO — SURF — 2015 : Quantization Noise in Digital Control Systems		
Ayush Pandey		

California Institute of Technology
LIGO Project, MS 18-34
Pasadena, CA 91125
Phone (626) 395-2129
Fax (626) 304-9834
E-mail: info@ligo.caltech.edu

Massachusetts Institute of Technology
LIGO Project, Room NW22-295
Cambridge, MA 02139
Phone (617) 253-4824
Fax (617) 253-7014
E-mail: info@ligo.mit.edu

LIGO Hanford Observatory
Route 10, Mile Marker 2
Richland, WA 99352
Phone (509) 372-8106
Fax (509) 372-8137
E-mail: info@ligo.caltech.edu

LIGO Livingston Observatory
19100 LIGO Lane
Livingston, LA 70754
Phone (225) 686-3100
Fax (225) 686-7189
E-mail: info@ligo.caltech.edu

1 Introduction

The Project Proposal[1] gave a brief outline about the timeline and the basic background information related to the project. The past three weeks have almost followed the path what was planned in [1]. The project is basically divided into the following three parts:

1. Quantization Noise Measurement using a Software Tool, and enhancements of the same.
2. Analysis of Filter Structures and developing ways to modify the structures (using techniques mentioned later) such that the quantization noise is minimized.
3. Quantization Noise Analysis and Ways to reduce quantization noise in the desired frequency band (Noise Shaping) with respect to the noise in Analog to Digital Converters (ADCs) and Digital to Analog Converters that are present in the Digital Control loop.

In this report, we look into each in detail. We also focus on the current progress and the progress expected in the coming months for each part of the project. Hence, this report presents the all the work that has been completed and also gives us an idea as to what kind of work we would be doing later.

2 Software Tool to Measure Quantization Noise Level Automatically

Software Tool to Automatically Check All Digital Filters for quantization noise in various filter structures in the Advanced LIGO Digital Controllers has been developed.

The tool has been developed on MATLAB and is based on an existing code to check any particular digital filter. The existing code by Dennis as in [2] checks any digital filter, on providing the name of the filter, the channel and the filter bank name as arguments, for quantization noise in the Direct Form II and the "Biquad Form" as described in [12]. The plot that results on running the above code shows the input signal, the output and the quantization noise level for both the filters. Hence, using this code, one could check for noise level in any filter and determine whether the noise is an important source or not and whether one needs to think about mitigating it or could do with it in its current level itself (if very low with respect to the input and output). [2] calculates noise for single precision by taking the difference between the output of single and double precision. Implicitly assuming that the double precision is almost as good as being without any noise. To make sure the assumption holds, the code multiplies the noise level by a factor of 2^{-m} where m is the difference in number of bits in single and double precision. The factor has been obtained empirically and hence the validity of it is only established through results and not theoretically accurate.

In this project, we aim to achieve the following two things with respect to this software tool:

1. As mentioned above, a new tool has been developed which has automated the above procedure of checking the digital filter for noise. The scripts are available at [3]. The

new tool parses the foton software file which has names of all the filters in the LIGO digital control systems.

2. The project also aims to solve the current problem, mentioned previously, that the code actually calculates single precision noise and then empirically extrapolates it to double precision. But, realizing the fact that this is not such a robust way of achieving quantization noise analysis, we aim to modify the code such that it calculates double precision noise itself by making use of a long double format (that is a precision level better than double precision). This is necessary because at all LIGO sites, already double precision is in use for all digital controllers (or most of them) and hence there is no point analyzing single precision noise. The update version of the code is available at [3]. We are currently working on this aspect and hope to complete it in the coming weeks.

3 Digital Filter: Structures and Noise Analysis

Digital Filters are represented mathematically using Transfer function, say $H(z)$, which is unique for given position of poles and zeros in the z -plane. The implementation of a given transfer function can be done in infinite different ways. Various different filter structures are described in detail in [15]. The quantization noise, our primary concern in this project, also depends on the filter form realization. By different realization of filter form, we mean different order of additions and/or multiplications in the filter structures. Some of the most commonly used filter structures for representation of Second Order Sections (SoS), along with their important features is described below:

1. Direct Form I: The direct form I is implementation of the difference equation, as is. The roundoff noise for this realization has been found to be much more compared to other structures which has been proved in [15]. The advantage of having direct form I structure is in the applications where least hardware complexity is required. DF I leads to lowest chip area required for implementation on a Digital Signal Processor. [9]
2. Direct Form II : The number of multiplications and delay registers are least for this form. [15] shows the quantization noise analysis for fixed point implementation. DF II form minimizes the coefficient sensitivities of the filter with respect to the quantization error which is proved in [9].
3. State-Space Representation of Digital Filters is used widely to realize low noise form structures as described later in this report. The major disadvantage of using state space form filters is that it requires a very high computational time, i.e. a complexity of $(N + 1)^2$ for an Nth order filter.[7]

For the Advanced LIGO digital controllers, previously the digital filters were implemented in the direct form II structures pertaining to its lower number of additions and multiplications. But, as described above, state-space representations leads to low noise forms increasing the computational time in the process. [12] went on this line and suggested the use of a kind of biquad filter derived from state-space representation but only using one addition extra in the process. The noise analysis shown in [12] proves that this indeed is a better choice for the digital filter structure as it provides for great increase in the SNR compared to the computation time penalty suffered.

The following (intuitive) assumptions have been made:

1. Higher the precision, the lower the quantization error. So, the double precision would perform much better with respect to the quantization noise.
2. The more the computational time taken, the lesser would be the quantization error.

To measure single precision noise [2] suggested a way, by taking the difference between the single precision output and the double precision signal output. The double precision output noise is so less compared to the single precision noise that the difference output is equal to single precision quantization error. Hence, in essence, considering double precision to be a perfect representation (i.e. with no quantization error). All digital controllers in the Advanced LIGO setup specifically use the floating point representation of numbers. A good analysis has been done by [12] for floating-point representation wherein, calculations for various digital filter forms have been provided. However, in this project we aimed to provide a better literature review and background to the floating point quantization noise analysis.

There are some basic trade-offs in the design of filter which are mentioned as follows:

1. The Hardware Complexity in the Design of the filter (the chip area required of a DSP), though this doesn't play any role if specialized DSP chips are not in use and implementation is being done on a computer.
2. The software complexity, i.e. the computational time the filter takes in calculations and the complexity of the structure algorithm
3. Quantization noise level in the filter and its effects
4. Sensitivity to disturbances and perturbations of coefficients

[10] showed through his analysis for direct form filters that if poles and zeros are tightly clustered on the z-plane, then even small coefficient quantization errors may cause large shifts in the position of poles and zeros and hence changing the response of the filter and even even

tending to become unstable.

Since the filter structure changes so many factors, one possible goal of this project for the next month would be to develop some way to optimize the quantization noise level by either reordering of poles and zeros [5] or implementing some other optimization technique keeping in mind the trade-offs mentioned above.

3.1 Floating-Point Precision Noise analysis

For floating point representations a key point to keep in mind is that the quantization noise can no more be assumed to be independent of the input signal. In fact, the noise is directly dependent on what input is being given to the filter. Also, the noise cannot be assumed to be white in this representation hence the noise analysis in floating point is a difficult task. An advantage due to the use of an exponent in the floating point representation is that the Overflow condition is eliminated and hence the complexity of the structure reduces as we don't need to care about the overflow during calculations.

The book by Widrow in Chapter 12, [14] looks at floating point quantization in depth and provides detailed floating-point noise analysis and calculations. Some direct formula to calculate quantization noise level in different filter structures have been given in [12] along with a focus to the signal in LIGO. The presentation [12] shows that noise is lower in a state space representation of a filter realized analogous to the analog biquad filter compared to the Direct Form II (which is widely considered as the best filter structure). This filter structure having low noise has been referred to as a "biquad filter" in [12]. For floating point precision some very important results are available in the literature already existing. [8] mentions two theorems proving that the floating point quantization noise level is independent both of the ordering of sections in a cascade form and even independent of ordering of poles and zeros between sections. The paper assumes that ordering of calculations and poles and zeros within a section remain the same throughout. Theoretically, the quantization noise could achieve super low noise [7] form when error feedback is used in a state-space representation of digital filter. It has already been shown in various journal articles that for any filter the lowest quantization noise can be achieved in a state space representation [7]. Any filter can be represented in the general State variable representation as:

$$x_{k+1} = Ax_k + Bu_k \quad (1)$$

$$y_k = Cx_k + Du_k \quad (2)$$

On a given transformation T , to achieve low noise structure, we have

$$T_{k+1} = A'T_k + B'u_k \quad (3)$$

$$y_k = C'T_k + Du_k \quad (4)$$

where $A' = T^{-1}AT$; $B' = T^{-1}B$; $C' = C^tT$

To implement a filter as in the above form, a complexity of $(N + 1)^2$ is required and hence these filters are not widely used. Moreover, as shown in [11], error feedback could be used to achieve super low noise forms. This is hard to implement practically as in practical system

we don't have the error accessible as a signal which could be fed back. [13] shows how the error feedback helps in minimizing the error. Another important result that comes from the literature is that when extra bits are used in floating point representation for accumulation register, the optimization methods developed for fixed point analysis can be applied 'as is' for floating point representation[4]. This is one of the reasons why [5] was cited above for optimization using reordering of poles and zeros even though the analysis given in [5] is for fixed point.

This project, hence, would look to focus upon various issues such as coming up with a proper trade-off between computational complexity and noise reduction so that better filter structures could be devised.

4 Noise Shaping, Dither, Noise Analysis and Simulations

Lastly, as previously planned in [1] we would be looking into Noise shaping techniques for DAC and ADC in the digital controller setup in this project. ADCs and DACs are present in a Digital Control System since the control loops are mixed signal setups where both analog and digital signal exist. Hence, the need to convert from one form to another arises quite frequently and this is where ADCs and DACs come into picture.

Since, the LIGO setup measures strain in the range 10^{-20} to 10^{-25} metres as a consequence of which the frequency at which gravitational waves are detected in the setup currently in use is in very low frequency range ($< 1KHz$). Realizing the fact that this is quite a low frequency and also since we are heavily concerned about any and all kinds of noises that exist in the control loop or elsewhere, Noise Shaping technique would be very useful in decreasing the noise power contributed to by the systematic noise called Quantization noise. Quantization Noise occurs both in DACs as well as ADCs and if noise shaping is implemented in both of the architectures the sensitivity of measurement would increase leading to better detection of gravitational waves.

We would be looking into the DAC noise shaping as the last part of this project. The DAC noise shaping has been analyzed in part and is available in [1].

But the noise shaping analysis for ADCs has been completed and is shown below. The actual implementation on hardware depends on various other factors and hence currently we are not looking forward to it, but if time permits we would be going into that as well.

4.1 Sampling

A simple ADC takes in analog signals at a given frequency and samples it. Sampling is a technique of taking input values at discrete time intervals. It is a linear process and the most common form of achieving this is by using the Sample and Hold amplifier arrangement [15]. The sampling frequency plays a very important role here because it is the sampling frequency which decides how precisely the signal is given to the quantizer for it to be digitized. For low frequencies, as one would expect the signal won't be represented accurately at the output of the sample and hold. The dilemma of choosing the correct frequency has been addressed

in the famous "Shannon-Nyquist Sampling Theorem" [15]. It says that sampling should be done at-least at the double of the maximum frequency of the signal being sampled. Hence, various ADC architectures choose the sampling frequency

$$f_s > 2f_{max} \quad (5)$$

where f_s is the sampling frequency and f_{max} is the maximum frequency coming in at the input called the Nyquist Frequency. Undesirable effects such as aliasing might occur if the sampling frequency is an integer multiple of the signal frequency which is explained in detail in [15] under the topic Aliasing. Anti-aliasing filters which are basically Low Pass filters have to be used to prevent it.

Among various types of ADCs that are commercially available, the delta-sigma ADCs provide major advantages in our case, which is explained in detail in this report later. One of the important parts of delta-sigma ADCs is the use of a technique known as oversampling. In oversampling, the signal is sampled at much higher frequency rates, say upto 10 times (or more) the nyquist frequency. Oversampling helps in reduction of Quantization noise in the desired band as well as it helps in making the aliasing effect negligible [16]. Hence, delta-sigma ADCs do not require much focus on the anti-aliasing filter, which further simplifies the design.

4.2 Quantization Noise and its Spectrum

After Sampling, the signal is discretized in its amplitude, which is called the Quantization of signal. It is here that the signal is actually converted into discrete values which is then called the converted digital form of the signal. With respect to the output produced, Quantizers can be classified into two types, the mid-tread and the mid-riser Quantizer.

For an example [17], let us take a ramp input signal. A ramp is an input signal which is a straight line when drawn on a 2D plane, i.e., $y=x$. For this ramp input signal the quantization is achieved as shown in the equation and in the simulations for the same:

The input signal is a ramp input given which needs to be converted into digital domain.

The quantizer is given by:

$$Q(x) = k.q.sgn(x). \left\lfloor \left\lceil \frac{x}{q} \right\rceil + \frac{1}{2} \right\rfloor \quad (6)$$

where,

k is any arbitrary scaling factor

q is equal to 1 Least Significant Bit (LSB), shows the resolution of ADC and,

x is the input to which Q(x) is the output.

The output is shown in the figure 1.

As is clear from the figure, the output looks like a staircase and around zero value the it is flat (representing a tread of stair) hence this type of quantization is called the mid-tread quantizer.

Another type of quantizer [17] as mentioned earlier is the mid-riser quantizer which is given by:

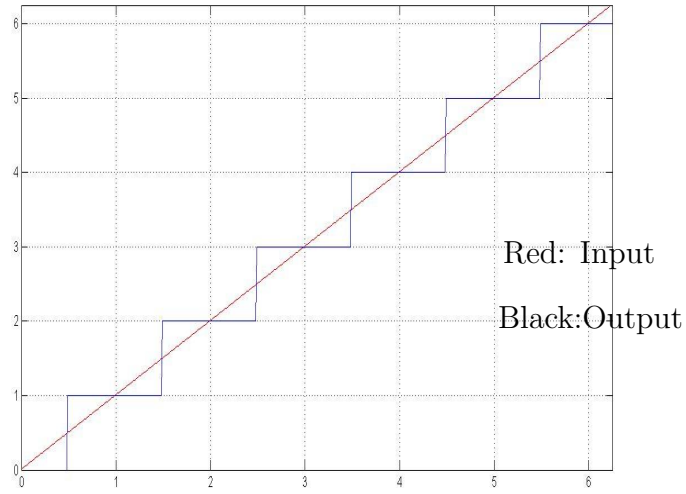


Figure 1: The Output given by the Mid Tread Quantizer, the converted digital form

$$Q(x) = k.sgn(x).q. \left(\left\lfloor \left| \frac{x}{q} \right| \right\rfloor + \frac{1}{2} \right) \quad (7)$$

The output of this quantizer is shown in figure 2.

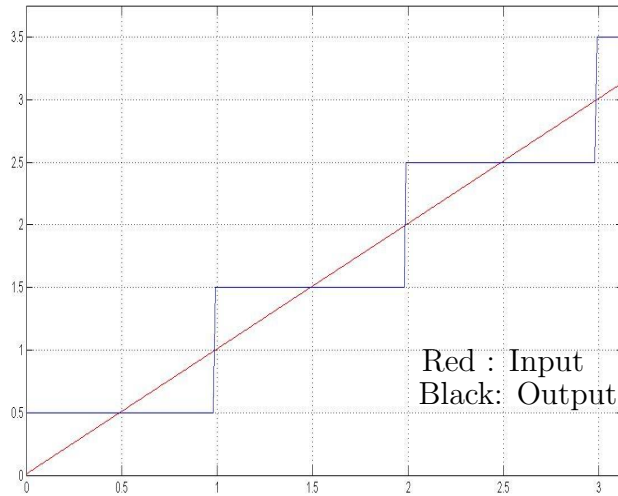


Figure 2: The Output given by the Mid Riser Quantizer, the converted digital form

The quantization error that occurs in the above system is given by the difference between the input and the output. The Root Mean Square (RMS) level of quantization noise can be derived for easier reference later. The error between input and output can be approximated as (the difference between input and output) shown in Figure 3.

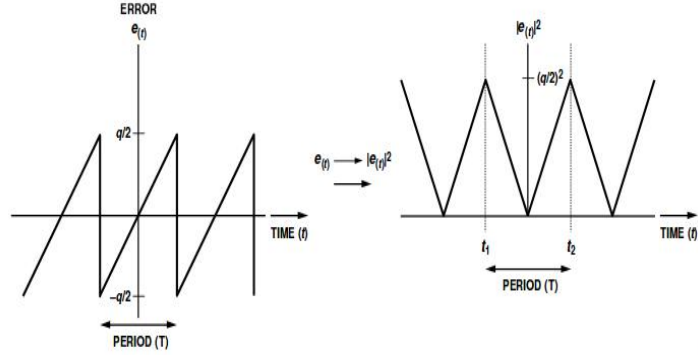


Figure 3: The Approximated Wave Form for the Quantization Error taken from Analog.com

To calculate the RMS we take a time interval $T = t_1 - t_2$ and write the equation of straight line in $y=mx+c$ form as:

$$e(t) - \frac{q}{2} = \left(\frac{\frac{q}{2} - \frac{-q}{2}}{t_2 - t_1} \right) \cdot (t - t_2) \quad (8)$$

On simplifying, we get,

$$e(t) = \left(\frac{q}{T} \right) \cdot t + q \cdot \left(\frac{1}{2} - \frac{t_2}{T} \right) \quad (9)$$

For the quantization error the RMS is given by:

$$e_{rms}^2 = |e^2(t)| = \frac{1}{T} \cdot \int_{t_1}^{t_2} |e(t)|^2 dt \quad (10)$$

Simplifying,

$$e_{rms}^2 = \frac{1}{T} \cdot \int_{t_1}^{t_2} \left(\frac{q^2}{T^2} \right) \cdot t^2 dt - \frac{1}{T} \cdot \int_{t_1}^{t_2} q^2 \cdot \left(\frac{(t_1 + t_2)^2}{4T^2} \right) dt \quad (11)$$

On solving we get (Skipping the simple algebraic manipulations) that the RMS of the error is given by:

$$e_{rms} = \frac{q}{\sqrt{12}} \quad (12)$$

where q is the LSB of the quantizer representing the resolution of the taken ADC.

On simulation of the above in MATLAB and calculating the RMS, we observe that the RMS of the error is indeed equal to as given in equation (8). The script files for the same are available for reference on the GitHub Repository at the URL mentioned in [18].

The signal to noise ratio taking quantization noise into account could be calculated as follows: Assuming the input to be sinusoidal, we have,

$$V_t = \frac{q \cdot 2^N}{2} \cdot \sin(2\pi \cdot ft) \quad (13)$$

The RMS for sinusoidal signal is known to be equal to the signal divided by a $\sqrt{2}$. Hence, calculating the SNR in decibels using the above result for RMS of the error. We have,

$$SNR_{dB} = 20 \cdot \log \left(\frac{RMS_{Input}}{RMS_{noise}} \right) \quad (14)$$

which can be written as,

$$SNR_{dB} = 20 \cdot \log \left(\frac{\frac{q \cdot 2^N}{2 \cdot \sqrt{2}}}{\frac{q}{2 \cdot \sqrt{3}}} \right) \quad (15)$$

On simplification, we arrive at the following well known result,

$$SNR = (6.02N + 1.76) \text{ dB} \quad (16)$$

4.3 Dithering

Throughout the above calculations, we have assumed that the quantization noise is independent of the input signal. This is a valid assumption since in practical systems, the use of a dithering additive signal results in independence between quantization noise and the input. Dithering is basically a process of adding random noise to the input before feeding it into the Quantizer [6]. Though it increases the noise power to a certain extent, but the advantage is that the quantization noise after adding a dither is very close to white noise (hence easy to model and analyze). A half LSB broadband noise is mostly used as a dither signal for most purposes. A block diagram representation of adding a dither signal is shown in figure 4.

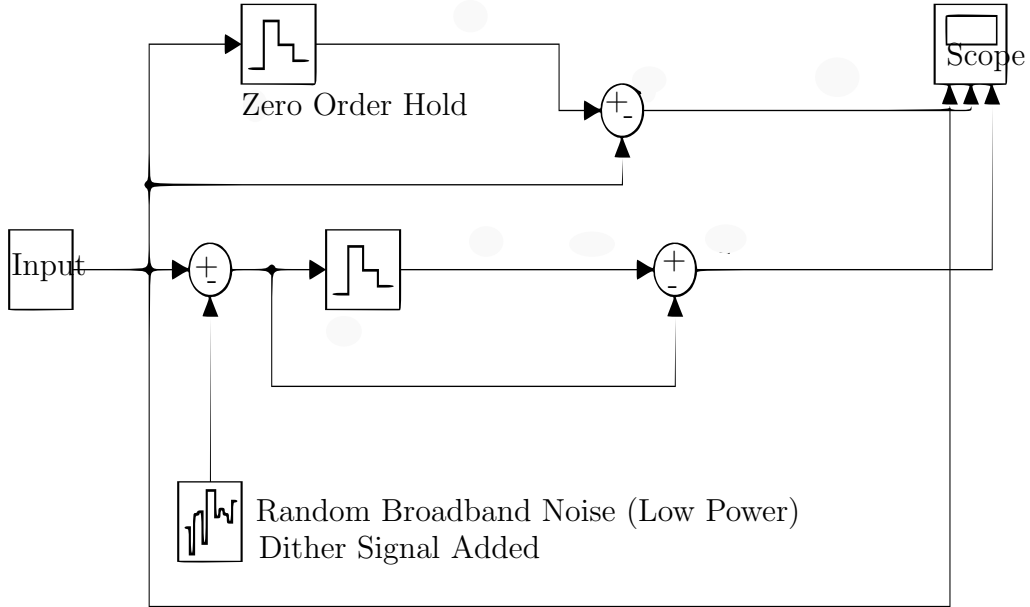


Figure 4: The Block Diagram Representation of an ADC with Dither Added to Input

In the above figure, the Zero Order Hold block represents the ADC, rest of the blocks are self explanatory. We are observing the effects of dither and comparing it with the signal

when no dither is being used. A MATLAB script was generated to simulate the same and the results are shown in figures 5 and 6.

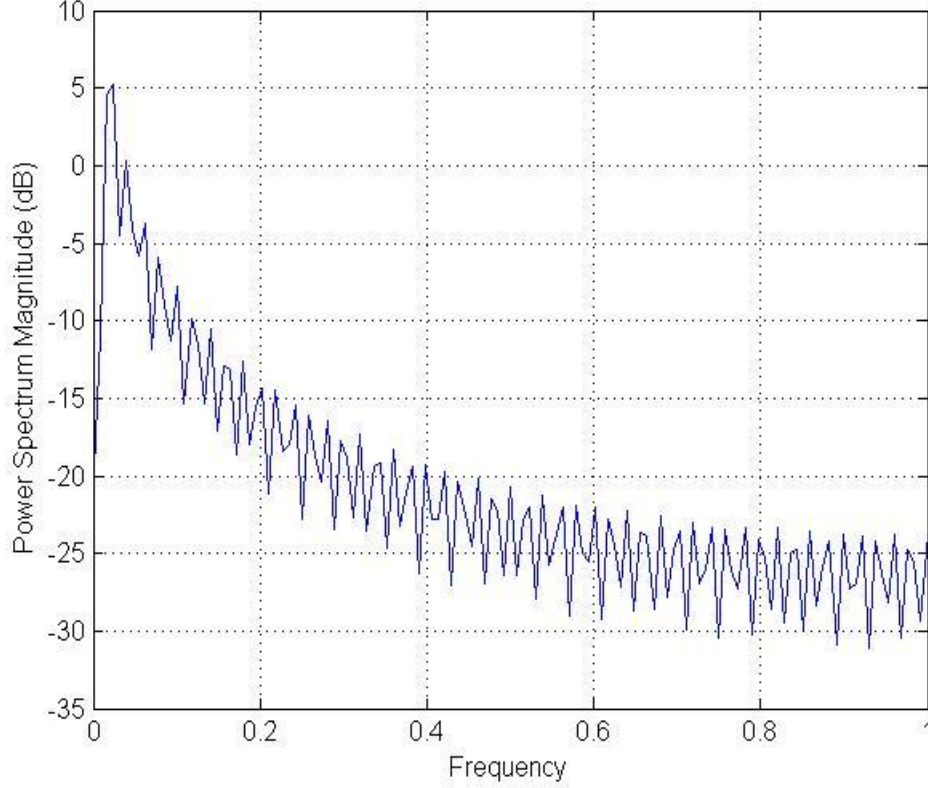


Figure 5: The Power Spectrum Density of the Quantization Noise when No Dither is added to the Input

As we can see, the Power Spectrum Density (PSD) of the error tends towards being 'white' more in the case when dither is added to the input signal. Hence, low power dither is usually preferred while performing ADC. Now, this shape of the noise (flat, white noise) can be shaped such that the desired frequency band has low power of the noise. This technique is called Noise Shaping. Noise shaping is, in essence, pushing the noise into a bandwidth which is undesirable to us. As previously stated, the bandwidth of interest while detecting gravitational waves in the LIGO is in the lower ranges and the higher frequency band is undesirable. Hence, using noise shaping technique we can push the noise spectrum to be more in the higher frequency band with respect to that in the low frequency band. This increases the SNR greatly in the desired band of frequencies. Post Noise Shaping, a Low Pass filter could be used to cut out all the undesirable frequencies (the higher band of frequencies) and hence removing the quantization noise [16].

4.4 Quantization Noise Shaping in ADCs

Regular ADCs such as the Successive Approximation Register (SAR) ADC or the Flash ADC face various problems with respect to their complexity and noise power spectrum. The

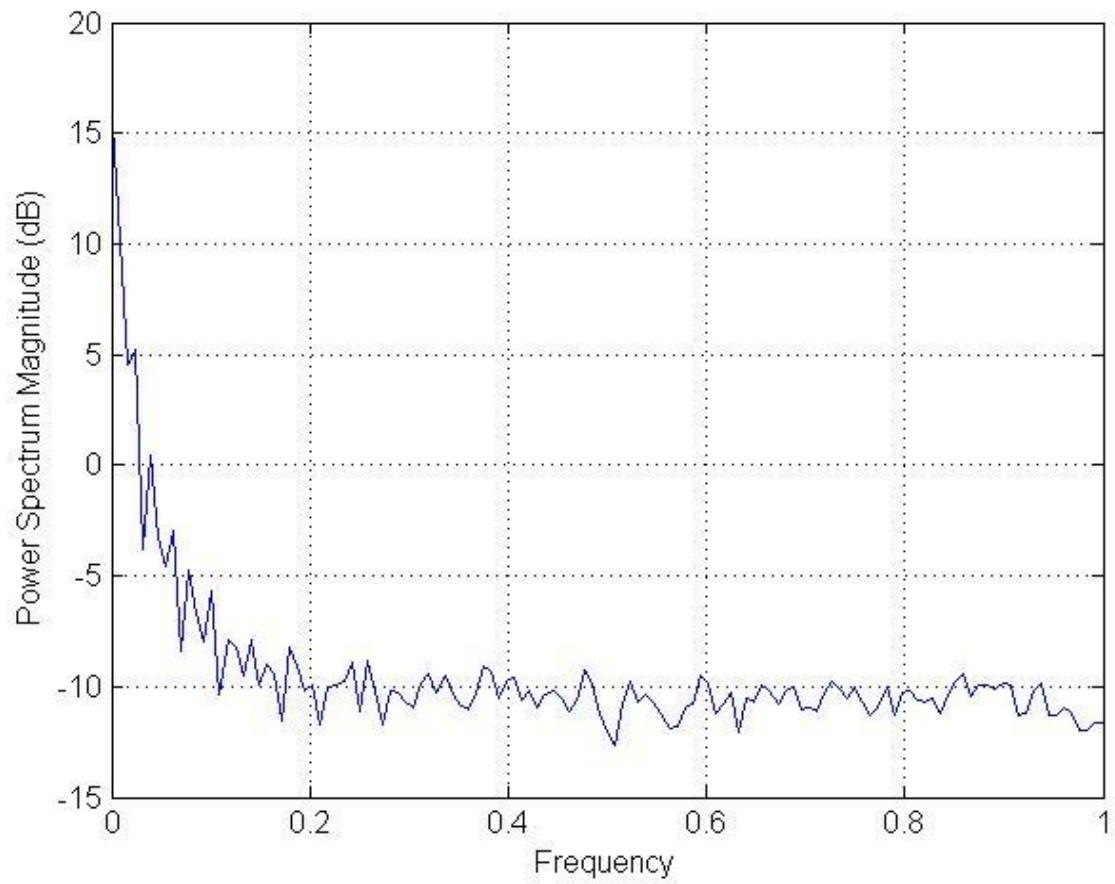


Figure 6: The Power Spectrum Density of the Quantization Error when Dither is added to the Input

details of these and other types of ADCs are explained in [15].

Delta-Sigma ADCs aim to solve this problem by reducing the complexity of the design and along with it, it introduces noise shaping, which leads to a higher value of SNR in the desired frequency range. It basically consists of the following stages, Oversampling, Quantization using Density of Pulses called the Sigma-Delta Modulation and finally a Decimator which reduces the output rate to frequency which can be processed further by the Digital Signal Processor (DSP) where the digital filter or the digital controller is implemented.

Noise shaping is achieved in Delta-Sigma ADC using feedback of the output via 1-bit DAC, which is basically a switch. This feedback brings about appropriate noise shaping in the signal which shall be explained using the figure 7.

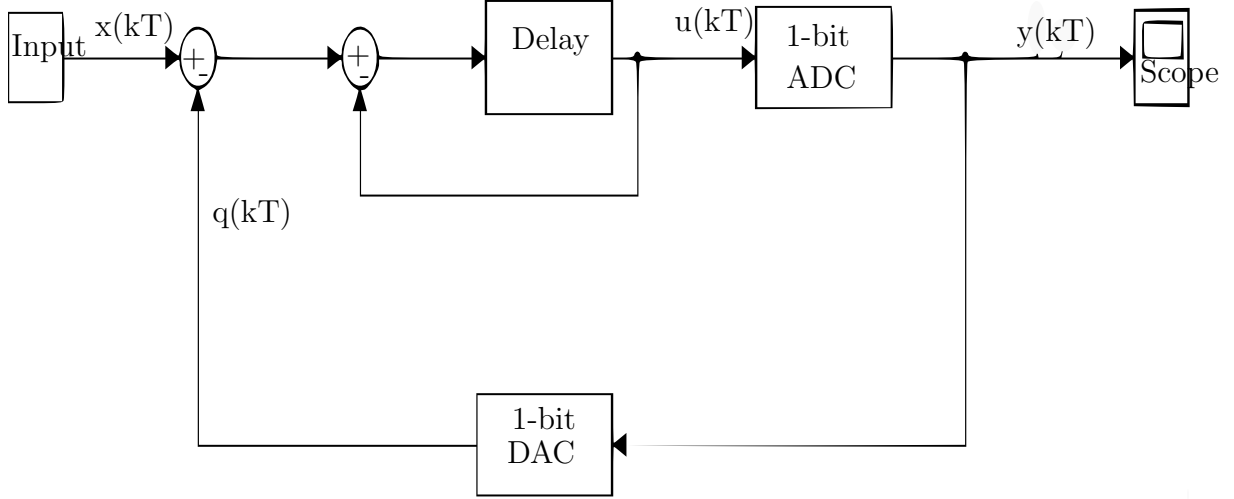


Figure 7: The Block Diagram of a Delta-Sigma ADC

$x(kT)$ is the input in analog which is to be converted into digital domain and $y(kT)$ is the output. $q(kT)$ is the feedback signal and $u(kT)$ is the signal which is quantized and converted to digital form using the 1-bit ADC shown.

As is clear from the diagram itself that the ADC design architecture when implemented this way is very simple since it is using one bit ADC and DAC. The Delta-Sigma ADC basically trades off for much higher frequency rate (oversampling) over high resolution of a single converter. The 1-bit ADC may be implemented using a comparator and a 1 bit DAC is basically a switch which changes between $+V_{ref}$ and $-V_{ref}$ according to whether the output $y(kT)$ is high or not. The reason why we have used such an architecture would be clear once we derive the effects on quantization noise it has.

$$u(kT) = x(kT - T) + u(kT - T) - q(kT - T) \quad (17)$$

$$q(kT) = y(kT) \quad (18)$$

The quantization error for the 1 bit ADC is given by:

$$Q_e(kT) = y(kT) - u(kT) \quad (19)$$

Using 17 and 19 we have,

$$y(kT) = x(kT - T) + u(kT - T) - q(kT - T) + Q_e(kT) \quad (20)$$

Using 18 and 20 we have finally,

$$y(kT) = x(kT - T) + Q_e(kT) - Q_e(kT - T) \quad (21)$$

The output of the ADC is equal to the previous input added to the difference between previous and current quantization errors. Hence, the error cancels itself which leads to better signal to noise ratio in the desired frequency range, which shall be shown by using the block diagram as shown in figure 8 (another simplified version of the previous diagram, this time signals in voltages). From feedback control theory we have the following relation:

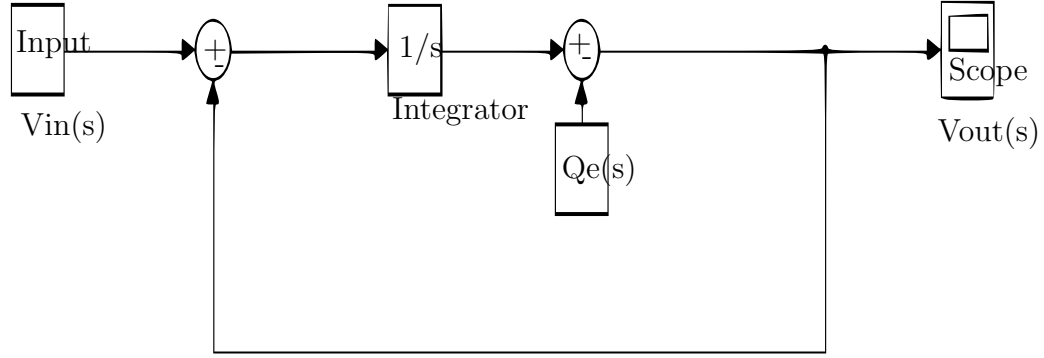


Figure 8: A simplified version of the Block Diagram of a Delta-Sigma ADC

$$V_{out}(s) = Q_e(s) + \frac{V_{in}(s) - V_{out}(s)}{s} \quad (22)$$

From the above equation we have:

$$V_{out}(s) = \frac{Q_e(s) \cdot s}{s + 1} + \frac{V_{in}(s)}{s + 1} \quad (23)$$

The quantization noise is multiplied with a transfer function which represents that of a High Pass filter which essentially proves how noise shaping takes place. The input signal is a low pass filter which is decimated (down sampled) to feed into the digital signal processor further in the control loop.

4.5 Simulations in MATLAB

The delta-sigma ADC was simulated using the Simulink tool in MATLAB. Figure 9 is the detailed block diagram with all scopes and power spectrum measurement models shown as well (obtained using the dspdadcd tool in MATLAB toolkit):

The results are as follows: For given input, a square wave of 80 Hz frequency, the fig-

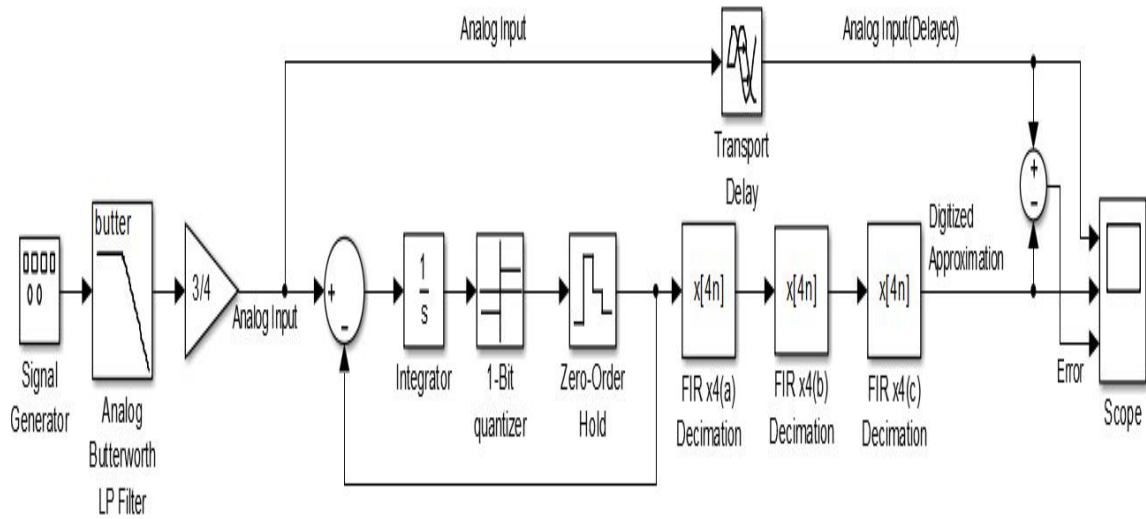


Figure 9: The Waveform of Input, Output and the Quantization Error

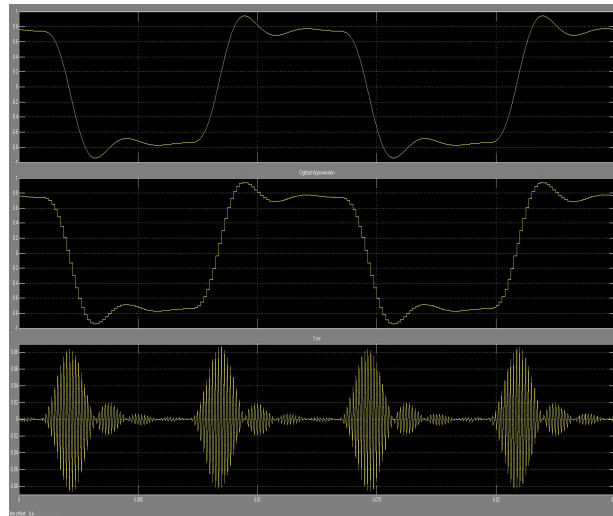


Figure 10: The output waveforms of ADC Conversion along with Quantization Noise

ure 10 shows the time domain output of the ADC. As we can see in the third segment of the figure 10 which represents the error, noise shaping is taking place as we described earlier. This would be further clearer from the frequency plots. Figure 11 shows the frequency spectrum of the output in the low frequency range (i.e. the range where signal is most prominent) and it is clear from the plot that the signal exhibits maximum magnitude in this band and falls off steeply after that. Figure 12 shows the frequency spectrum of the output for the full

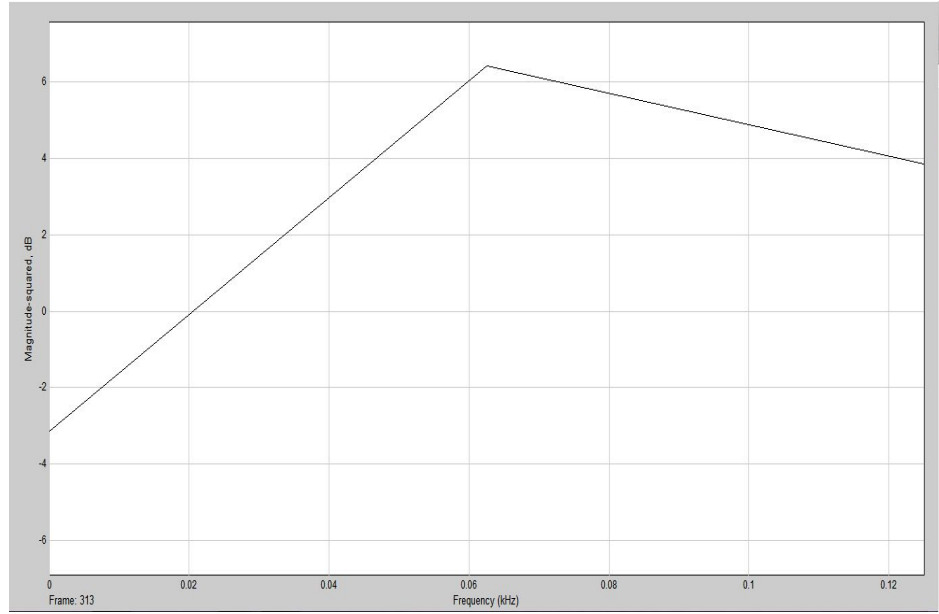


Figure 11: The output spectrum in the lower range of frequency

range of frequencies, which further acts as evidence for the fact that the power spectrum of the output is maximum in the desired range (low frequency range) only.

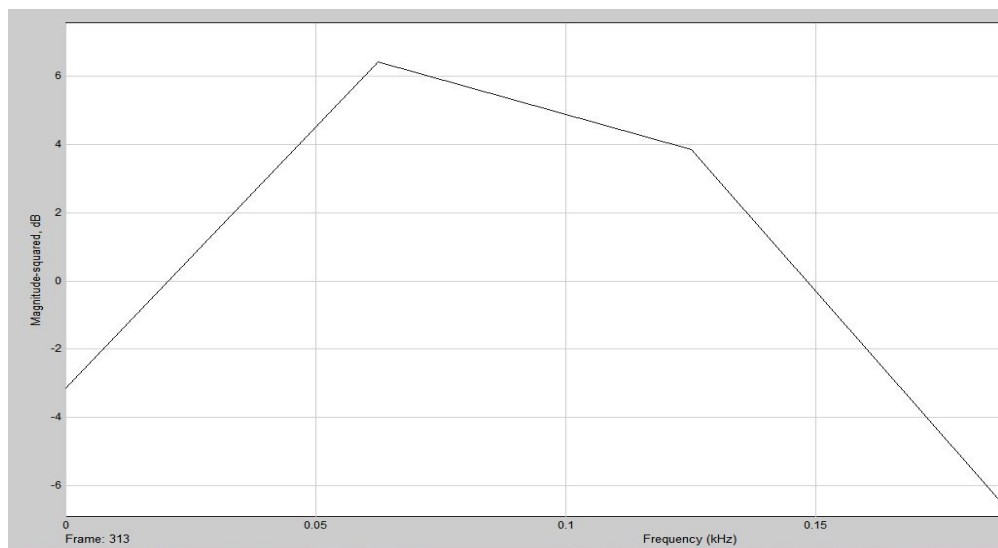


Figure 12: The output spectrum showing complete range of frequencies

Figure 13 is the key figure as it shows how nicely the quantization noise is shaped, taking its

lowest magnitude in the lower frequency band and rising after that and in figure 14 the full range of frequencies is showing how the quantization noise has the highest power spectrum in the undesirable (higher bandwidth) range.

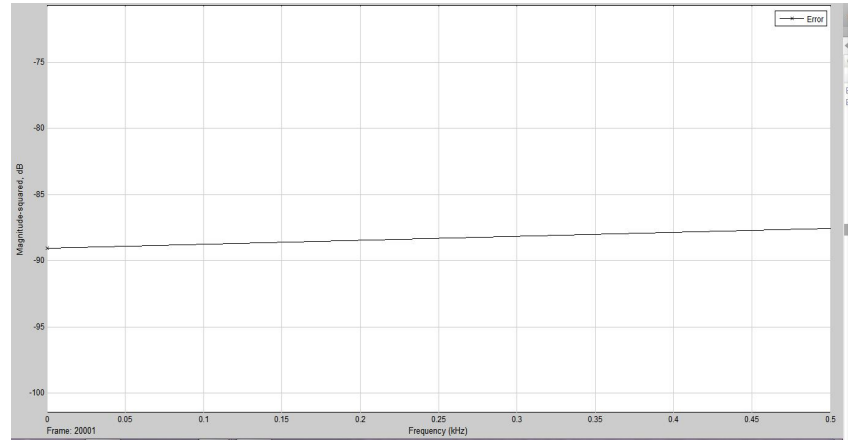


Figure 13: The Quantization Error spectrum showing only the lower range of frequency

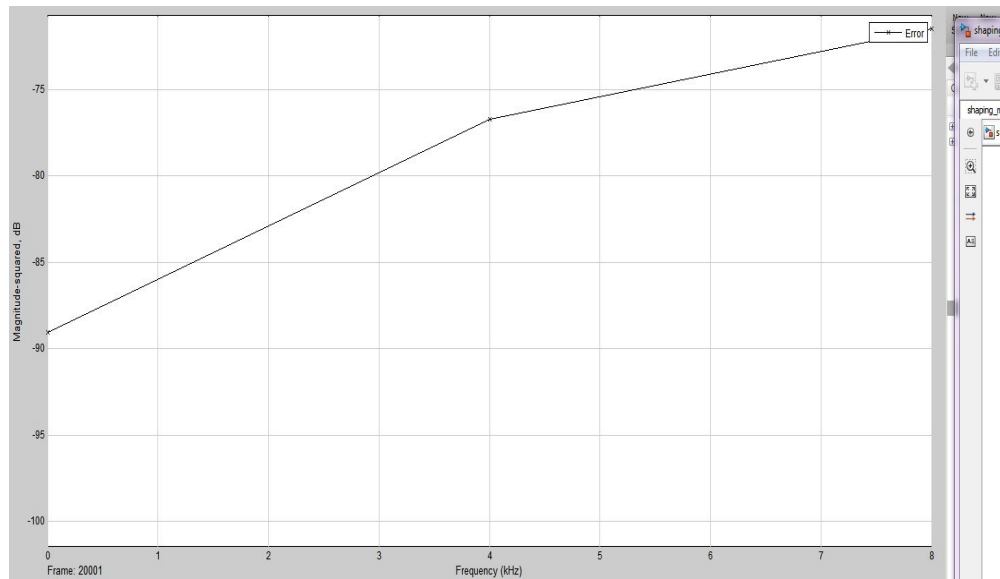


Figure 14: The Quantization Error spectrum showing complete range of frequency

The simulink block diagram was built and compiled to get the code to implement the same. The code along with other supporting files is available at the url mentioned in [18].

References

- [1] Ayush Pandey, *Quantization Noise in Digital Control Systems* <https://dcc.ligo.org/LIGO-T1500243>

- [2] Denis Martynov, *Checking Digital System* <http://github.com/denismartynov/quantization> <http://nodus.ligo.caltech.edu:8080/40m/?mode=full&attach=1&reverse=0&npp=510&Author=den&subtext=digital>
- [3] Software Tool Code, <http://github.com/ayush9pandey/quantization>
- [4] Bomar, *Roundoff Noise Analysis of State-Space Digital Filters implemented on Floating-Point Digital Signal Processors*
- [5] Dehner, *Noise Optimized IIR digital filter design-tutorial and some new aspects*
- [6] Widrow, *Chapter 19: Dither, Quantization Noise Book*
- [7] T. L. Chang, *A Unified Analysis of Roundoff Noise Reduction in Digital Filters*
- [8] Bing Zeng, *Analysis of Floating-Point Roundoff errors using dummy multiplier coefficient sensitivities*
- [9] S. Rahmanian, *An Optimal Structure for Implementation of Digital Filters*
- [10] Kaiser, *An approach to implement Digital Filters*
- [11] Chang, *Comparison of Roundoff Noise variances in several low round off noise digital filter structures*
- [12] Matthew Evans, *Digital Filter Noise*
- [13] *Roundoff Noise in digital filters: Frequency transformations and invariants*
- [14] Widrow, *Chapter 12, Basics of Floating-Point Quantization, Quantization Noise Book*
- [15] Oppenheim and Schaffer, *Discrete-Time Signal Processing*
- [16] Basics of Sigma Delta ADCs, a Tutorial, <http://www.analog.com/media/en/training-seminars/tutorials/MT-022.pdf>
- [17] Quantization of Signals, [http://en.wikipedia.org/wiki/Quantization_\(signal_processing\)](http://en.wikipedia.org/wiki/Quantization_(signal_processing))
- [18] DAC Denoising <http://github.com/ayush9pandey/DACdenoising/>