

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

Technical Note	LIGO-T11XXXXX-vX	2015/05/27
<b>Noise Shaping in ADCs to reduce Quantization Noise in Advanced LIGO Digital Control Systems</b>		
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](mailto: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](mailto: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](mailto: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](mailto:info@ligo.caltech.edu)

## 1 Basic Theory and Introduction

The LIGO setup measures strain in the range  $10^{-20}$  to  $10^{-25}$  metres as a consequence of which the frequency at which gravitation waves are detected in the setup currently in use is in very low frequency range ( $\leq 1\text{KHz}$ ). 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.

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 other arises quite frequently and that is where ADCs and DACs come into picture.

## 2 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 [1]. 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" [1]. 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

$$f_s > 2f_{max} \quad (1)$$

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 [1] 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 [2]. Hence, delta-sigma ADCs do not require much focus on the anti-aliasing filter, which further simplifies the design.

### 3 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 [3], 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:

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

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.

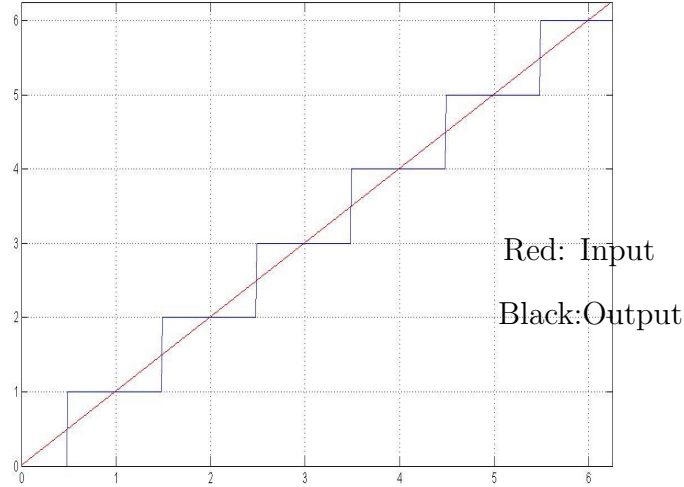


Figure 1: The Output given by the Mid Tread Quantizer, the converted digital form

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 [3] as mentioned earlier is the mid-riser quantizer which is given by:

$$Q(x) = k.sgn(x).q. \left( \left\lceil \left\lceil \frac{x}{q} \right\rceil \right\rceil + \frac{1}{2} \right) \quad (3)$$

The output of this quantizer is shown in figure ??:

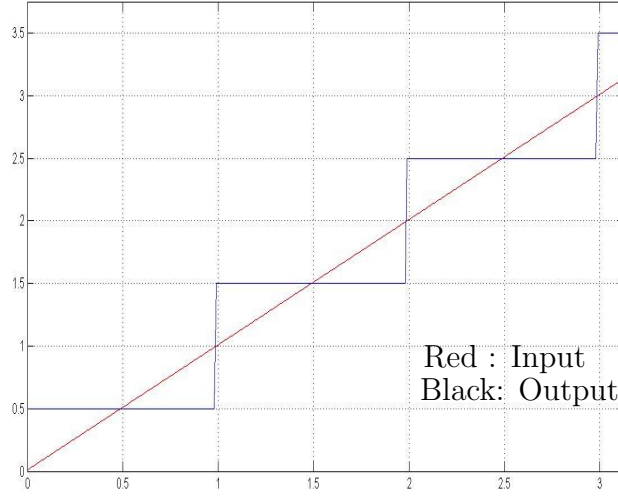


Figure 2: The Output given by the Mid Riser Quantizer, the converted digital form

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 ??

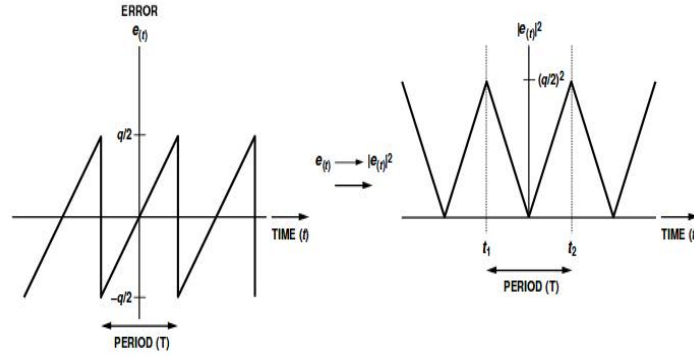


Figure 3: The Approximated Wave Form for the Quantization Error taken from Analog.com

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:

$$e(t) - \frac{q}{2} = \left( \frac{\frac{q}{2} - \frac{-q}{2}}{t_2 - t_1} \right) \cdot (t - t_2) \quad (4)$$

On simplifying, we get,

$$e(t) = \left( \frac{q}{T} \right) \cdot t + q \cdot \left( \frac{1}{2} - \frac{t_2}{T} \right) \quad (5)$$

For the quantization error the RMS is given by:

$$e_{rms}^2 = |e^2(t)| = \frac{1}{T} \cdot \int_{t_1}^{t_2} |e(t)|^2 dt \quad (6)$$

Simplifying,

$$e_{rms}^2 = \frac{1}{T} \cdot \int_{t_1}^{t_2} \left( \frac{q^2}{T^2} \right) \cdot t^2 dt - \frac{1}{T} \cdot \int_{t_1}^{t_2} q^2 \cdot \left( \frac{(t_1 + t_2)^2}{4T^2} \right) dt \quad (7)$$

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

$$e_{rms} = \frac{q}{\sqrt{12}} \quad (8)$$

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 [4].

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

$$V_t = \frac{q \cdot 2^N}{2} \cdot \sin(2 \cdot \pi \cdot f t) \quad (9)$$

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,

$$SNR_{dB} = 20 \cdot \log \left( \frac{RMS_{Input}}{RMS_{noise}} \right) \quad (10)$$

which can be written as,

$$SNR_{dB} = 20 \cdot \log \left( \frac{\frac{q \cdot 2^N}{2 \cdot \sqrt{2}}}{\frac{q}{2 \cdot \sqrt{3}}} \right) \quad (11)$$

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

$$SNR = (6.02N + 1.76) dB \quad (12)$$

## 4 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

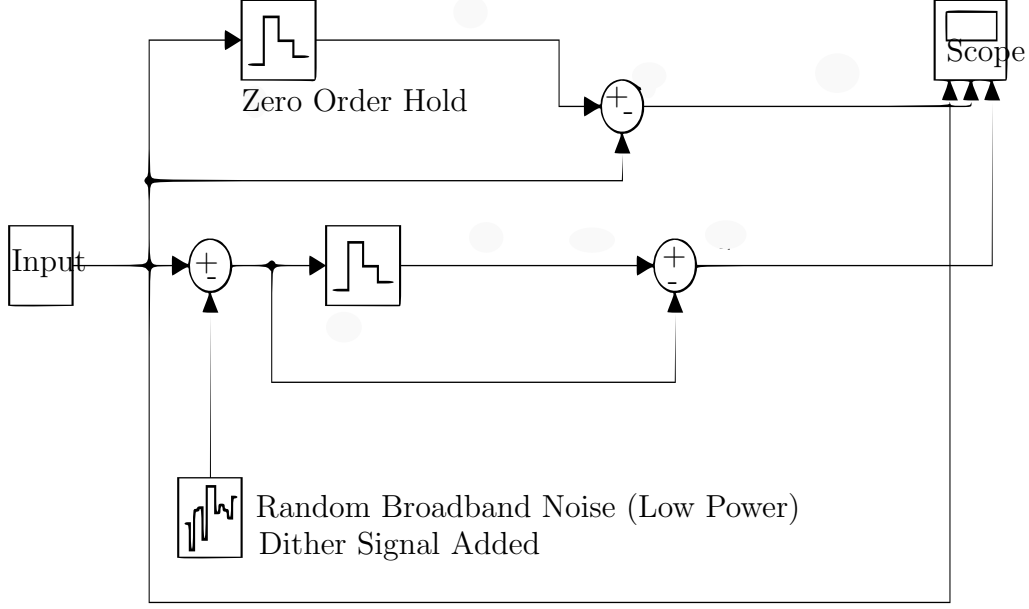


Figure 4: The Block Diagram Representation of an ADC with Dither Added to Input

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. 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.

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.

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 [2].

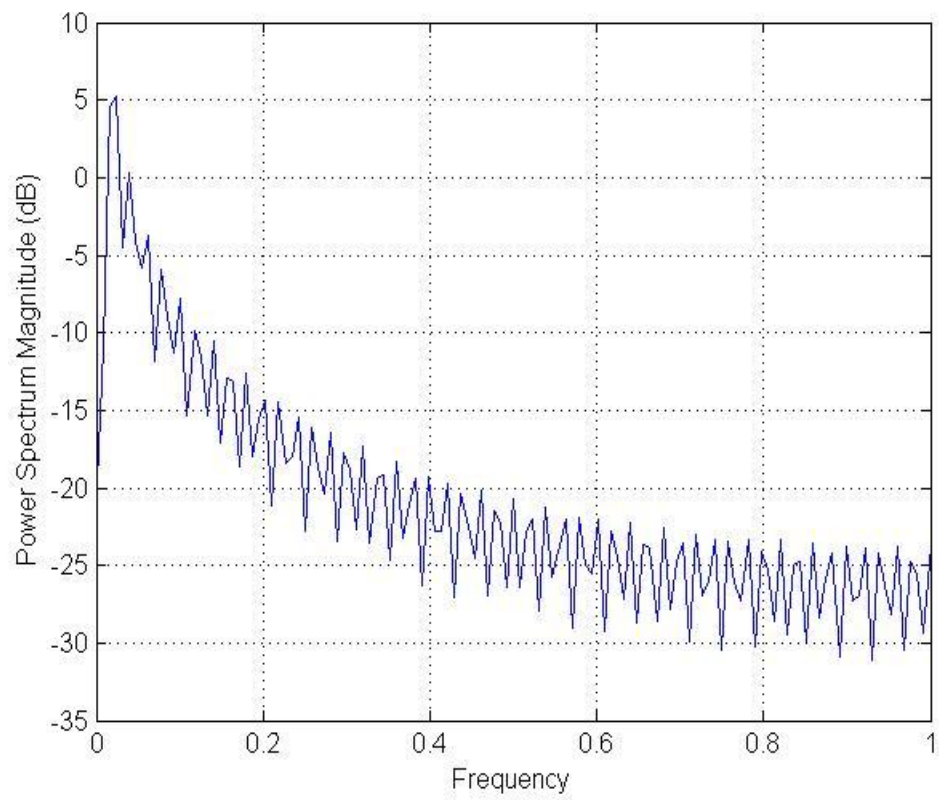


Figure 5: The Power Spectrum Density of the Quantization Noise when No Dither is added to the Input

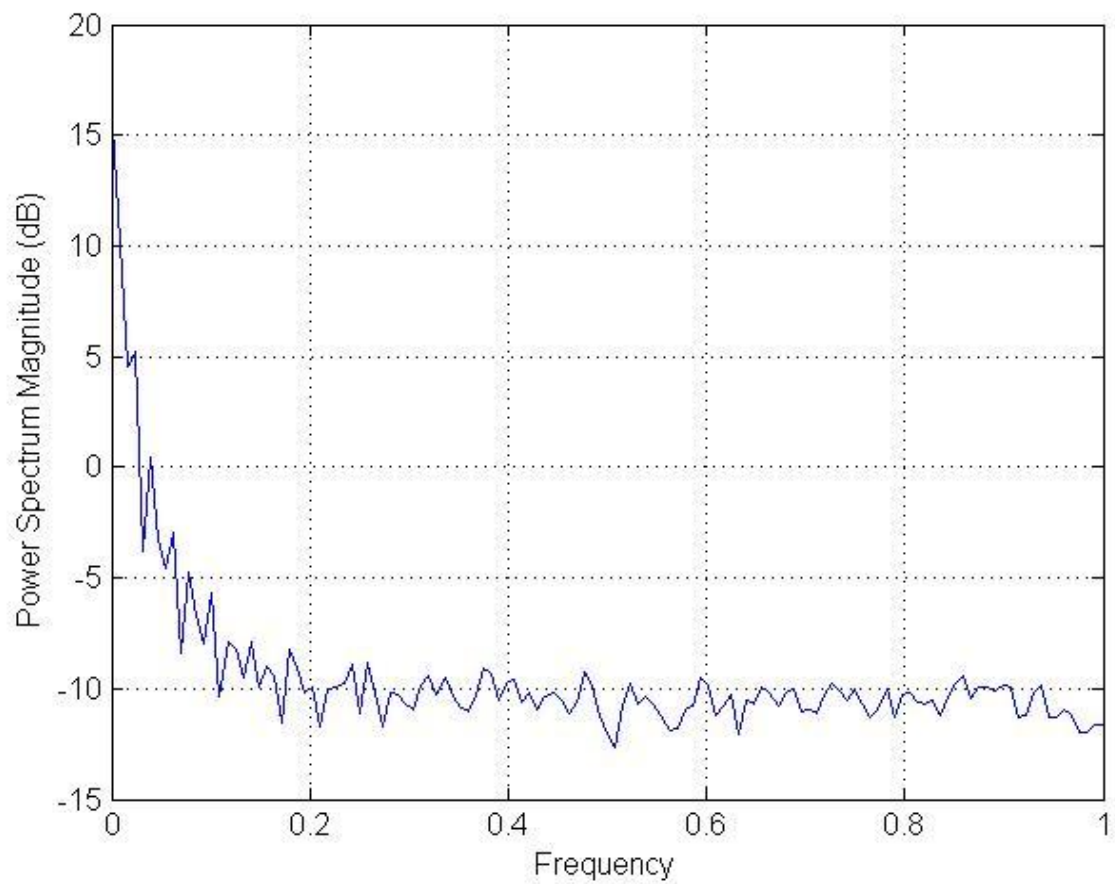


Figure 6: The Power Spectrum Density of the Quantization Error when Dither is added to the Input



## 5 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 details of these and other types of ADCs are explained in [1].

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.

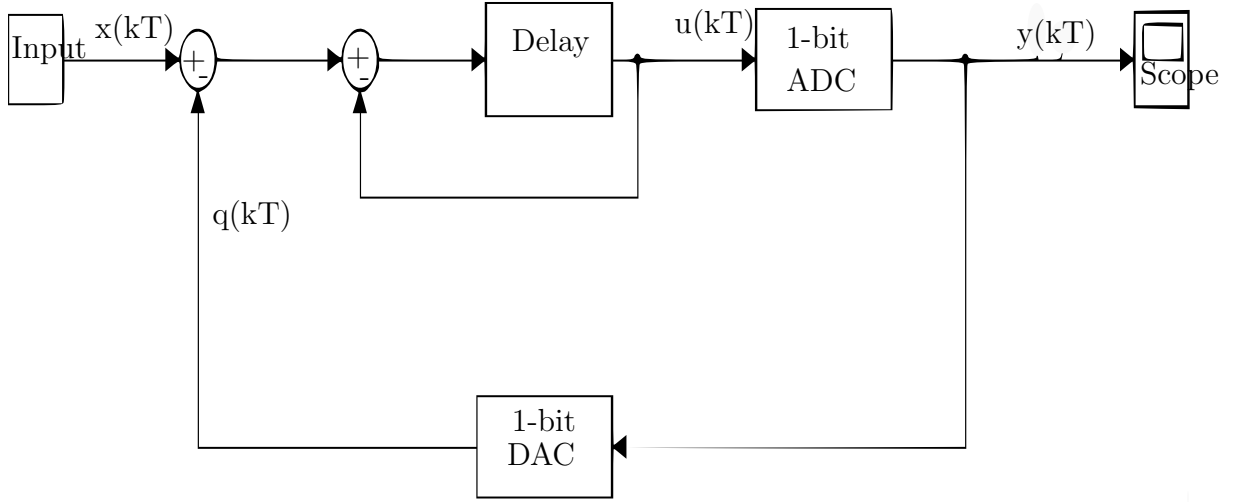


Figure 7: The Block Diagram of a Delta-Sigma ADC

$x(kT)$  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.

$$u(kT) = x(kT - T) + u(kT - T) - q(kT - T) \quad (13)$$

$$q(kT) = y(kT) \quad (14)$$

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

$$Q_e(kT) = y(kT) - u(kT) \quad (15)$$

Using 13 and 15 we have,

$$y(kT) = x(KT - T) + u(kT - T) - q(kT - T) + Q_e(kT) \quad (16)$$

Using 14 and 16 we have finally,

$$y(kT) = x(kT - T) + Q_e(kT) - Q_e(KT - T) \quad (17)$$

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:

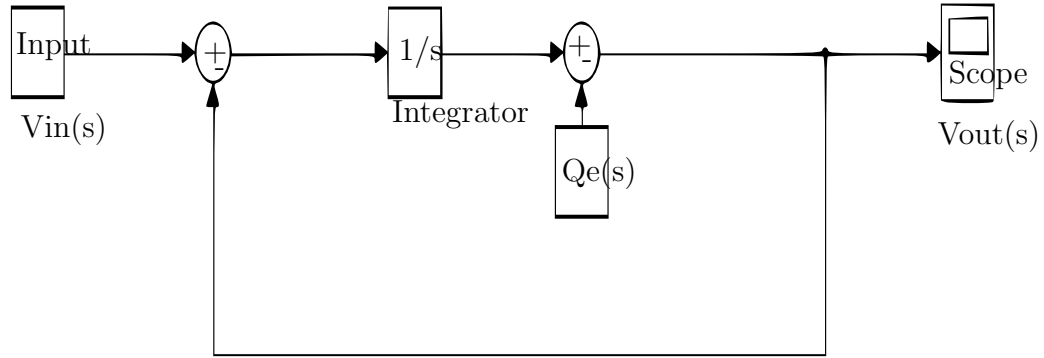


Figure 8: A simplified version of the Block Diagram of a Delta-Sigma ADC

$$V_{out}(s) = Q_e(s) + \frac{V_{in}(s) - V_{out}(s)}{s} \quad (18)$$

From the above equation we have:

$$V_{out}(s) = \frac{Q_e(s) \cdot s}{s + 1} + \frac{V_{in}(s)}{s + 1} \quad (19)$$

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.

## 6 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

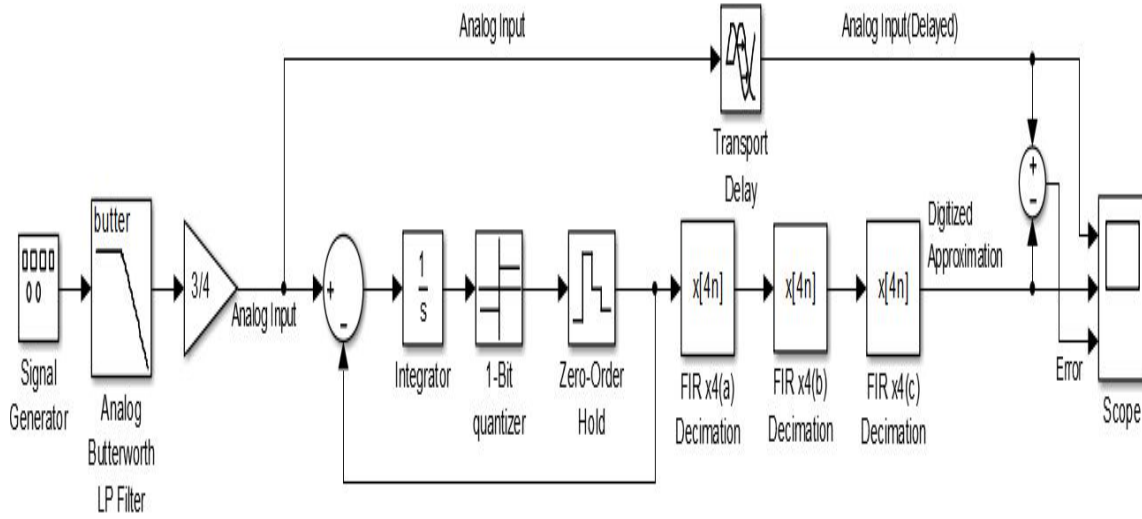


Figure 9: The Waveform of Input, Output and the Quantization Error

well (obtained using the dspstdc tool in MATLAB toolkit):

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

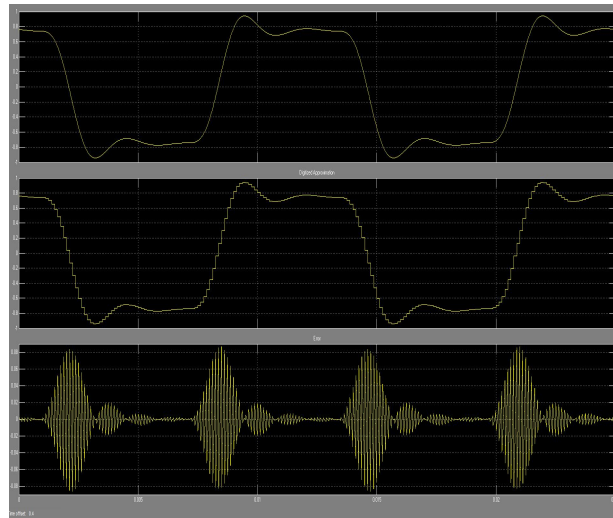


Figure 10: The output waveforms of ADC Conversion along with Quantization Noise

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

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

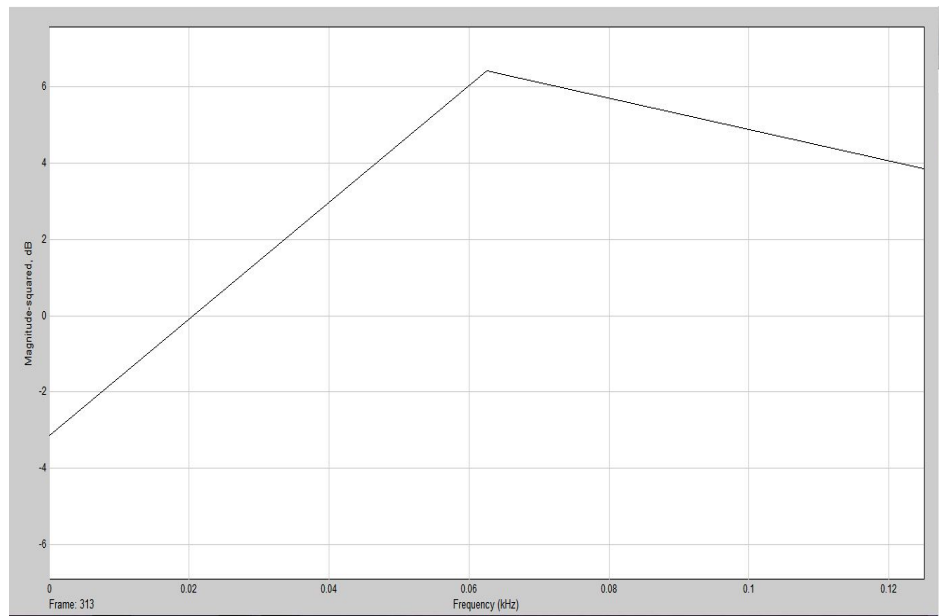


Figure 11: The output spectrum in the lower range of frequency

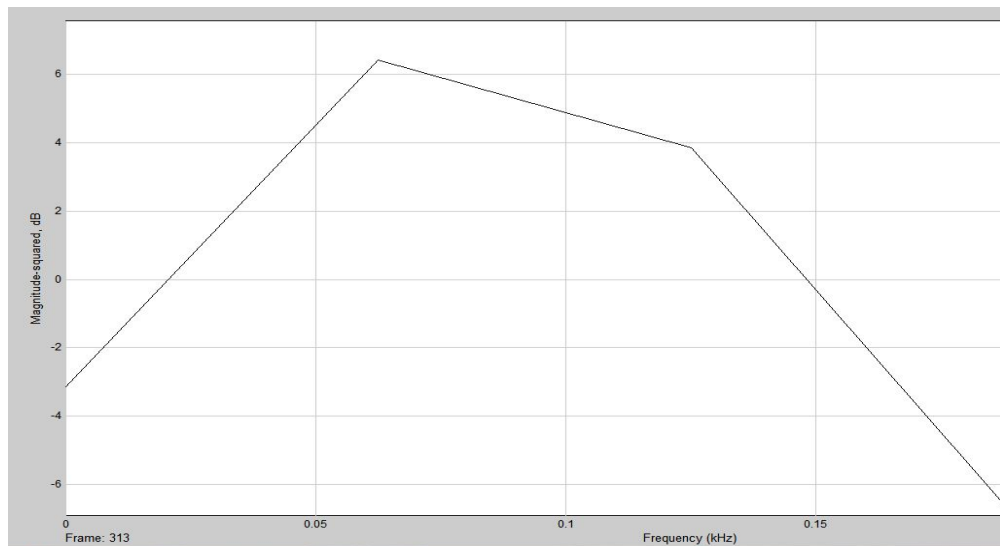


Figure 12: The output spectrum showing complete range of frequencies

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

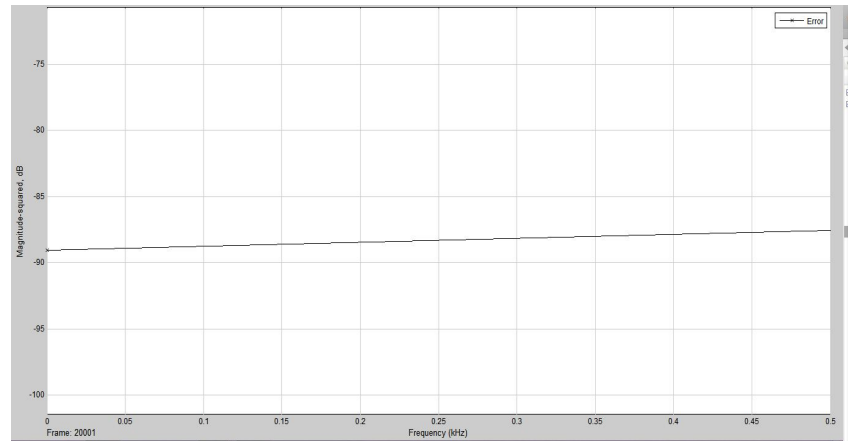


Figure 13: The Quantization Error spectrum showing only the lower range of frequency

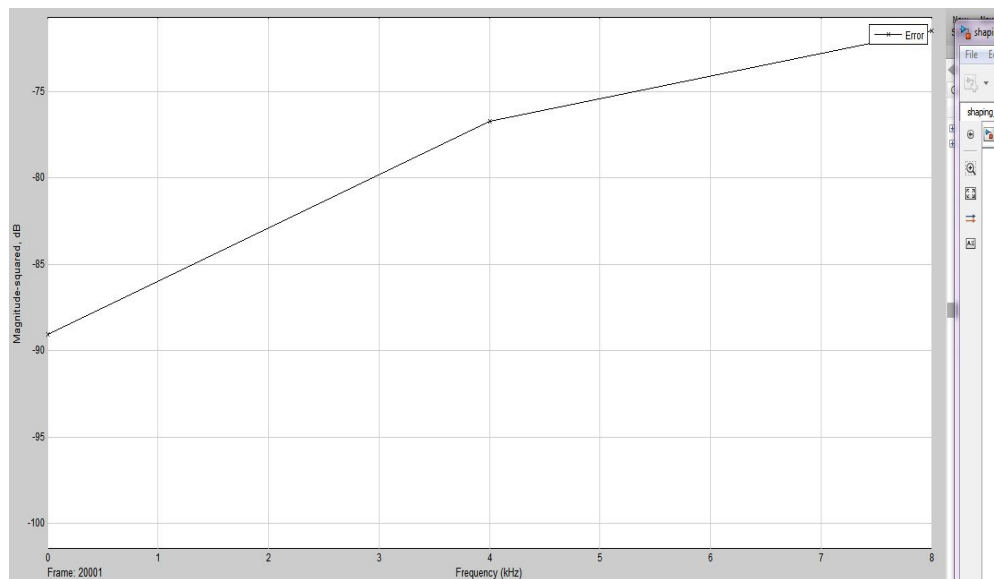


Figure 14: The Quantization Error spectrum showing complete range of frequency

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 [4].

## References

- [1] Oppenheim and Schaffer, *Discrete-Time Signal Processing*
- [2] Basics of Sigma Delta ADCs, a Tutorial, <http://www.analog.com/media/en/training-seminars/tutorials/MT-022.pdf>

- [3] Quantization of Signals, [http://en.wikipedia.org/wiki/Quantization\\_\(signal\\_processing\)](http://en.wikipedia.org/wiki/Quantization_(signal_processing))
- [4] DAC Denoising <http://github.com/ayush9pandey/DACdenoising/>