Differential Non-linearity, Integral Non-linearity, and Signal to Noise Ratio of an Analog to Digital Converter

Y. C. Jenq and Qiong Li
Department of Electrical & Computer Engineering
Portland State University
P. O. Box 751, Portland, OR, 97207 USA
(503)725-5414, jenq@ece.pdx.edu

Summary

The Analog to digital converter (ADC) has been widely used in all kinds of modern electronic instruments, so it is desirable to find out the relationship between various errors and the performance of ADC and improve the performance with a low cost operation. This paper studies and tries to quantify the relationship between the differential non-linearity (DNL) error, the integral non-linearity (INL) error and the signal to noise ratio (SNR) performance of an ADC, and investigates various methods to reduce the effect of these errors to increase the SNR. Major results are obtained by computer simulation, while a simple analytic model is also investigated. It is found that the loss of SNR due to DNL roughly follows the "6 dB per LSB" rule, and the major cause of the significant SNR loss is primarily due to harmonic distortion introduced by the INL. The investigation of the analytic model of the INL shows that it can predict the SNR loss very similar to the result obtained by computer simulation. We also study various methods to improve the SNR performance of the ADC. It is found that the Midpoint method, although simple, can almost completely eliminate the effect of the INL. It is also shown that if the ratio of the sampling frequency to the input signal frequency is large, the DNL can be further reduced to improve the SNR by Grouping and Sorting method.

1. Introduction [1,2]

DNL and INL are two important non-linearity errors of an ADC, which always leads to relatively severe loss of SNR. In this paper we will investigate how DNL and INL errors affect the SNR performance of an ADC, and study various methods to reduce their effects. Some basic concepts of an ADC are introduced first. Then the relationship between DNL, INL and the loss of SNR of an ADC are studied respectively. An analytical model for the INL is also proposed. Next, various methods to improve the SNR performances of ADCs are investigated.

2. Basic Concepts

An ADC represents analog inputs within a certain range by a limited number of digital output codes. The transfer function of an ideal ADC is a uniform staircase characteristic.

But, for a real-life ADC, its transfer characteristic is generally not uniform due to various kinds of errors, such as offset error, gain error, DNL error and INL error. Each error can be expressed in LSB (Least Significant Bit). The DNL error is the difference between an actual step width and the ideal value of 1 LSB. There is a possibility that there can be missing codes, i.e., one or more of the possible 2^n binary codes are never outputted. If the midpoints of steps of an ideal ADC transfer function are connected, we will get a straight line passing through the origin with a 45-degree angle. The INL error is the maximum deviation from the straight line after gain and offset errors have been nullified [4].

The real world analog input to an ADC is a continuous-time signal with an infinite number of possible states, whereas the digital output is by its nature a discrete-time function with a finite number of different states determined by the resolution of the ADC. Some information has been lost and error (noise) has been introduced into the signal. This is called the quantization noise [5]. For the ideal staircase transfer function of an ADC, the quantization error is uniformly distributed in the range $\pm 1/2$ LSB. If the input is $f(x)=A\sin(x)$ and the Full Scale Range (FSR) of an B-bit ADC is 2A, the signal noise ratio is

$$SNR = 6.02\,B + 1.76\,dB \qquad (1)$$

3. Simulation Study of DNL, INL and SNR

In this section we use computer simulation to study the relationship between DNL, INL and SNR. The input signal under test is a sine wave (S1) with half of FSR amplitude and a frequency of 268.55 KHz. The input signal is sampled at 10 mega (10^7) samples per second and the record length is 4096. The choice of frequencies are rather arbitrarily except to meet the condition that the sampling frequency is not an integer multiple of the signal frequency. A saw-tooth wave (S2) with the same signal frequency, the same sampling frequency is used to perform histogram test to ensure that the ADC model works properly. For simplification, we make FSR = 2 for all ADCs.

3.1 Validation of Simulation

Before we compute the SNR of a simulation ADC system, we should check if the simulation model works properly. Since we need to construct many ADC simulation models to represent many DNL and/or INL conditions, a quick check of the accuracy of the model is important. Saw-tooth wave is a

good choice to validate the model because it is uniformly distributed in time. When an ADC under test digitizes an analog input, the relative number of occurrences of the distinct digital output codes is termed code density [1]. The data are viewed in the form of a normalized histogram showing the frequency of each code from zero to full-scale. If one uses a saw-tooth wave as input, the code density should be proportional to the size of corresponding step size. Histogram test is a useful tool to quickly check if a particular ADC simulation model reflects the intended transfer characteristic.

3.2 Calculation of SNR

Once a simulation model is validated by the histogram method, it is used to digitize an input test signal and the SNR is calculated. SNR is the ratio of the average signal power to the noise power and it is one of the most important performance criterions of an ADC. In this test, calculations of SNR of an ideal digitized sine wave and real-life digitized sine are required. For the ideal digitized sine wave, Equation 1 is used. For the real-life digitized sine wave, we use Fast Fourier Transform (FFT) method to compute the SNR. Generally speaking, signal is the two maximum points of all FFT values and the rest are noises. So SNR can be computed readily by the ratio of the sum of the square of the signal components to those of the noise components. The input frequency is properly chosen to avoid the energy leakage due to windowing effect.

3.3 Simulation Model for DNL

To obtain an ADC with a specific DNL, we first set up an ideal ADC with uniform staircase transfer function, then the thresholds are adjusted by random quantities which are uniformly distributed between a specific range, such as ±0.2 LSB or ±0.3 LSB. But we keep the first and last threshold of the transfer function fixed to ensure the offset error and gain error are zero. To avoid the possibility that two adjacent thresholds could cross over, we restrict the threshold adjustment to be less than ±0.5 LSB. That is to say, the maximum DNL is 1 LSB. By changing the random number range from 0 to ±0.5 LSB, ADCs with different DNL values can be obtained. The ADC with DNL larger than 1 LSB will be generated by different technique described in the next section.

Following the procedure described above, the relationship between DNL and loss of SNR (as compared to the ideal ADC) are obtained and plotted in Figure 1. It is a set of 8-bit ADCs with different values of DNL. From Figure 1, it is seen that the relationship between DNL and SNR approximately follows Equation (1). Take the last point as an example, when the average DNL is 0.5 LSB, and the loss of SNR should be about 3 dB according to Equation, and our experiment data shows about 2.86 dB.
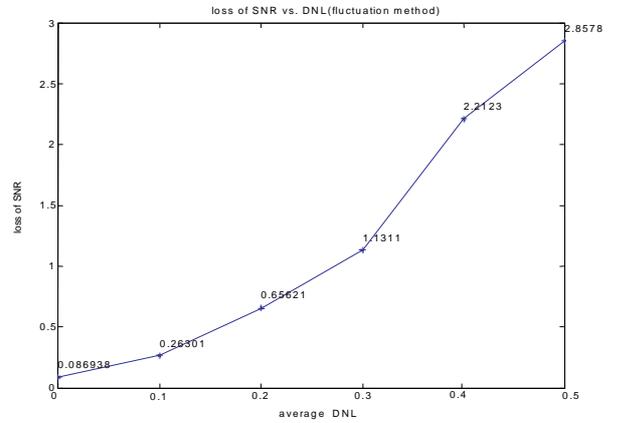

Figure 1. Loss of SNR vs. DNL

3.4 Simulation Model for Larger DNL and INL

As mentioned before, the preceding method which creates DNL by simply adjusting the thresholds by a random quantity from an ideal ADC with uniform step size has the maximum DNL limitation of one LSB and the INL obtained in the experiment is also less than one LSB. We are interested in constructing model to simulate much bigger DNL and also to be able to simulate cases with the same average DNL but different INL. In this way, we can determine the effects on SNR from DNL and INL, separately. In the following test, 2^B segments of random length are generated to represent steps of the transfer function of an ADC. The sum of all segments constitutes the FSR of the ADC and the average length of the segments is the 1 LSB. If the 2^B segments are connected in sequence, the transfer function is constructed. If we simply generate these step sizes from a specific distribution, we can control the average DNL, but not the INL. In fact, if we start with a fixed set of step sizes, and therefore a specific DNL, we can have many different INL if we simply rearrange the order of the steps. In other words, we need to have some control over the order of the steps with different step sizes. To begin with a simple case, let us consider the case with two different average step sizes. We first divide the FSR of the ADC into two parts each with a different average step size. To determine the number of steps associated with each of the two average sizes, let us assume that the first average step size is b_1 LSB ($b_1 < 1$) and the second average step size is b_2 LSB ($b_2 > 1$). Let X be the total length of the first part (which is equal to the sum of all steps with average step size b_1 LSB), then the length of the second part is R-X. Therefore, we have $X/(R*b_1) + (R-X)/(R*b_2) = 1$, and hence $X = Rb_1(b_2-1)/(b_2-b_1)$. Let N be 2^B, then it is clear that there are $NX/(b_1R)$ steps in part 1 and $N(R-X)/(b_2R)$ steps in part 2. That is, there are $NX/(b_1R)$ segments in the first bunch of steps, and all those segments add up to X and their average value is b_1 LSB, and there are $N(R-X)/(b_2R)$ segments in the second bunch of steps, and all those segments add up to R-X and their average value is b_2 LSB. With this method, one can control DNL and INL of

an ADC by using different divisions of the full-scale and different average values of the step sizes in each part.

In order to study the relationship of INL and SNR, a series of ADCs with the same DNL and different INL can be obtained by the method described above. Similar to the test on DNL, saw-tooth wave and histogram are used to validate every ADC model before the simulation is carried out.

According to the definition of INL, it is easy to calculate the INL from the transfer function of an ADC. Digitized sine wave S1 is used as the input test signal to calculate the SNR. Compared to an ideal ADC, the relationship between the loss of SNR and INL of a series of 8-bit ADCs with the same DNL (the average DNL is 0.7LSB) and different INL is shown in Figure 2 as 'Δ' curve. It is clear that INL is the dominant error in these cases. In Figure 3, we plot the FFT of a digitized sine wave through an ADC with INL error of 34.99 LSB. It is seen that the harmonic distortions caused by the INL contributes to the major loss of SNR.
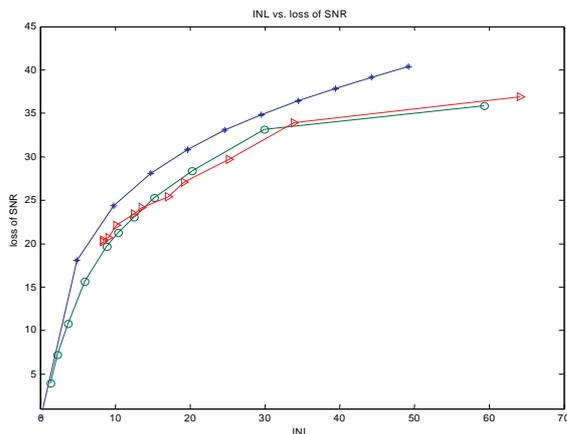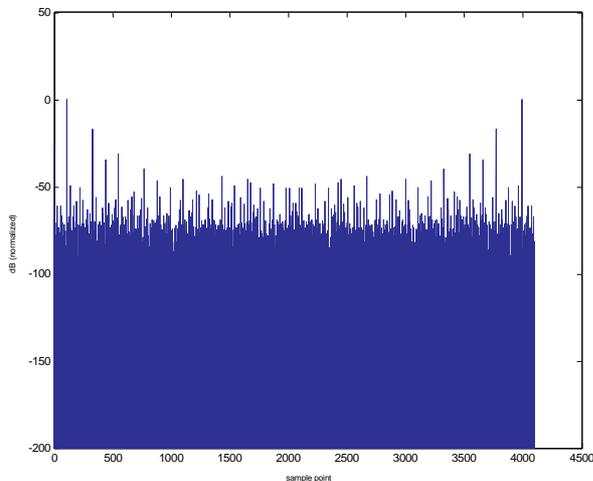

Figure 2. Loss of SNR vs. INL

Figure 3. FFT of A Digitized Sine Wave

In previous test, the step sizes of an ADC are generated by a random number generator in the range between 0 and 2×average step size. To obtain a tighter control of the INL we can use the same step size in part 1 and part 2 (for example, every step size in part 1 is 0.7 LSB and every step size in part 2 is 2 LSB). We then perform the same simulation and the results are shown in Figure 2 as 'o' curve.

Because the transfer function of ADC with two constant step sizes can be adjusted nearly symmetrical, smaller INL can be obtained. For example, when part 2 is divided into 60 parts presented by the first point of 'o' curve in Figure 2, INL is 1.4 LSB and the loss of SNR is mainly produced by DNL under this condition. It is noted from 'Δ' and 'o' curves in Figures 2 that the loss of SNR for both cases are very similar in the high INL area.

4. A simple Analytic Model on INL and SNR

In this section, an analytical model for the INL is proposed. A group of cubic curves are considered to represent the transfer functions of ADC. The function of the curve is given by $y = ax^3 + bx^2 + cx + d$. Using the property that the transfer function of an ADC has a full scale range of 2, three points contained in the curve are: *(-1, -1), (0, 0) and (1, 1)*. Hence, the function is $y = ax^3 + (1-a)x$. By changing the value of the coefficient '*a*', we can obtain a group of curves with different features.

According to the definition of INL, it is the maximum deviation of the curve from the line, *y=x*. The deviation between the curve and the line, *y=x*, can be represented by $D(x) = x - (ax^3 + (1-a)x) = ax - ax^3$, and a local maximum happens when $D'(x) = 0$, i.e., $x = 3^{-1/2}$. Therefore, at that point we have INL $= a (3^{-1/2} - 3^{-3/2})$.

Suppose a sine wave, $x = sin(\omega t)$, is digitized by an ADC with the transfer function $y = ax^3 + (1-a)x$, then the output is

y $= a\,sin^3(\omega t) + (1-a)sin(\omega t)$

$= (1-0.25a)\,sin(\omega t) - 0.25a\,sin(3\omega t)$　　　(2)

From Equation 2, the SNR of output is

$SNR = 20\,log_{10}\,[(4-a)\,/a]$

In order to compare with experiment data of the past experiments, INL of this set of ADCs are normalized by a uniform step size of an ideal 8-bit ADC and the loss of SNR is also calculated by comparing with the SNR of an ideal 8-bit ADC. When '*a*' is assigned from 0 to 1, we can obtain the relationship between INL and loss of SNR, shown as the '***' curve in Figure 2. From Figure 2, the relationships between INL and loss of SNR obtained from analytic model and different simulation methods are consistent to some extent.

5. Methods to Increase SNR of ADC with INL and DNL

For a specific real-life ADC, the transfer function can be obtained by histogram method. Several methods to increase SNR of the ADC are considered in this section.

5.1 Midpoint Method

From the previous simulations, it is clear that the loss of SNR due to DNL (without significant INL) roughly follows the "6 dB loss per LSB DNL" rule, and the major cause of the significant SNR loss is primarily due to harmonic distortion introduce by the INL. Therefore, the major improvement can be obtained by minimizing the INL. We first propose a simple technique where each output code is simply replaced by the midpoint value of the corresponding input step. Let us assume a part of thresholds of an ADC transfer function are listed in Table 1. The top line indicates the output codes, and the bottom line shows the corresponding threshold values.

000		001	010
-1.0000	-0.8552	-0.6948	-0.6336

Table 1. Thresholds of an ADC Transfer Function

For example, if an output code is 010, which indicates that the input is in the input step interval (-0.6948,-0.6336), it will be replaced by a new value (-0.6336-0.6948)/2 = -0.6642.

5.2 Uniform Random Number Method

In this case, a simple variation of the midpoint method is considered. Each output code is replaced by a value uniformly randomly selected in the interval of the corresponding step. For the transfer function in Table 1, if an output code is 010, a random number uniformly distributed between -0.6336 and -0.6948 is generated to replace it.

5.3 Gaussion Random Number Method

In this case, each output code is replaced by a number generated by a Gaussian distribution with mean m = midpoint value of the corresponding input step and standard variance σ = one sixth of the corresponding step width. For the transfer function in Table 1, if an output code is 010, a Gaussian random number with m =- 0.6642 and σ = (-0.6336+0.6948)/6 = 0.0102 will replace it.

5.4 Gaussion Random Number Grouping Method

In this method, the total output codes are grouped in sequence. When some consecutive points have the same output code, they are in a group. Next, every group is replaced by an array of Gaussian numbers with mean m = midpoint and standard variance σ = one sixth of the corresponding step width.

5.5 Gaussion Random Number Grouping and Sorting Method

This method is developed based on the technique described in section 5.4. After adjustment as described in section 5.4, the new values in one group are sorted in either increasing or decreasing order of three adjacent groups. For example, we have a group of output codes: 000, 000, 010, 010, 010, and 100. They are grouped as (000, 000), (010, 010, 010) and (100). Take the second group as an example, an array of Gaussian numbers (0.6598, 0.6472, 0.6655) with m = -0.6642 and σ = 0.0102 are generated. Because the output code sequence is in an increasing order, the Gaussian number are sorted as 0.6472   0.6598   0.6655 and replace the original 3 points respectively. The technique described here is trying to reduce the DNL in order to improve SNR further.

5.6 Comparison of Different Methods

The results of the five methods described above indicate that when the ratio of the sampling frequency to the input signal frequency is large, the method 5.5 performs the best. But, in general, method 5.1 seems to outperform the rest of methods. It is interesting to see that a simple method such as the Midpoint method almost completely eliminates the effect of INL.

6. Conclusion

In this paper, we have studied the relationship between differential non-linearity (DNL), integral non-linearity (INL), and signal to noise ratio (SNR) of an ADC, and the method to reduce the effect of these errors to increase the SNR. Major results are obtained by computer simulation, while a simple analytic model is also investigated. It is found that the loss of SNR due to DNL roughly follows the "6 dB per LSB" rule, and the major cause of the significant SNR loss is primarily due to harmonic distortion introduce by the INL The analytic model of INL predicts the SNR loss very similar to the result obtained by computer simulation. We also proposed and studied various methods to improve the SNR performance of the ADC. It is found that the Midpoint method, although simple, can almost completely eliminate the effect of INL. It is also shown that if the ratio of the sampling frequency to the input signal frequency is large, the DNL can be further reduced to improve the SNR by grouping and sorting method.

References

[1]	Y.C.Jenq, "Asynchronous Dynamic Testing of A/D Converters", Handshake, VOL.13 NO.2 summer 1988, pp.4-7.

[2]	B.E. Peetz, A.S. Muto, and J.M.Neil, "Measuring Waveform Recorder Performance," Hewlett-Packard Journal, Nov. 1982, pp.21-29.

[3]	"A/D Converter Definition Of Terms", National Semiconductor, January 2000, pp.1-2.

[4]	 "Understanding Data Converters", Texas Instrument Report, July 1995, pp.4-8.

[5]	B.P.Lathi, Modern Digital and Analog Communication Systems, Saunders College Publishing, 1989, pp.136-141.