SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM

TMLAI TRANSACTIONS ON
MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE

VOLUME 2, ISSUE 5
ISSN: 2054 - 7390

Behavioral Modeling of Typical Non-Ideal Analog to Digital Converter Using MATLAB

[1]Sayed Mostafa Taheri and [2]Bahareh Mohammadi
[1] Signal Processing and Communications Group, Electronics Engineering Department
Barthawa Institute of Technology, Mashhad, Iran;
[2] Communication and Information Systems Group, Electronics Engineering Department
Barthawa Institute of Technology, Mashhad, Iran;
[1]s.m.taheri@ieee.org; [2]b.mohammadi@agb.prof.ir

ABSTRACT

Analog-to-digital converters (ADC) are one the most fundamental and crucial parts of an electronic device which needs to convert analog inputs into the digital format. Nowadays, you can find some type of ADCs in almost every communication device. Consequently, simulating an ADC plays a challenging and very fundamental role while aiming to simulate real-world instruments and designed systems. Despite the fundamental significance of this task, a general simulation model for typical non-ideal ADCs have not been presented yet. In this paper, we consider the most important non-ideality parameters of a typical ADC and initiate simulation models in the Simulink environment of MATLAB, as the most versatile widely-used simulation software for engineering tasks. For this purpose, we first reconsider the non-ideality parameters of a typical ADC from general perspective, aiming to bring a unified view in hand. Afterwards, the potential models for the non-ideality parameters are initiated separately, and then, are combined together to make the whole model. By creating a MATLAB Block of the whole general-topology model at the end, and defining the non-ideality parameters as variable inputs, we simulate the behavior of a typical ADC using practical non-ideality parameters data, and compare with an ideal ADC. This additionally enables a user interface for quick input of non-ideality parameters of any practical ADC for future clients. Despite the pure generality of the model, simulation results demonstrate acceptable outcomes proving the applicability of this model in wide range of engineering simulation tasks, and an appreciable step towards higher precision general simulations of typical ADCs.

Keywords: Analog to digital converter, non-ideal ADC, behavioral modeling and simulation of ADC, MATLAB Simulink, typical modeling of non-ideal ADC.

1 Introduction

Analog to digital conversion has been a tremendously active field in digital signal processing and communications realm. Due to its wide vast range of applications, as well as high demand for up-to-date and capable analog-to-digital converters (ADC or A/D) to be used in digital communication and electronic systems satisfying today's requirements, it has always been an attractive and ongoing field of study and research. Efforts have been put to design and implement high precision, fast, and reliable ADCs over the past decades. There have been variety of different types of ADCs being proposed for

DOI: 10.14738/tmlai.25.567
Publication Date: 3th November, 2014
URL: http://dx.doi.org/10.14738/tmlai.25.567

different applications having various characteristics and features. As a general concept, ADCs are known to be non-linear non-ideal instruments as a result of being comprised of non-linear circuits. Therefore, the non-ideality parameters of ADCs usually imply some constraints to be looked after while employing them in real-world system designs. From this perspective, it is quit crucial to be able to predict and handle these non-idealities while designing electronic or communications systems. To do this, many researches have been done to study, improve, and overcome the non-ideality parameters of different types of ADCs; [1]–[15] are some examples. These studies could be generally divided into two main categories. The first category consist of researches which are focused on infrastructure design and improvements of ADCs, i.e., circuit-level studies. The second class contains the works aiming to predict or simulate the performance of different types of ADCs based on their infrastructure characteristics and features, usually in MATLAB Simulink or Spice. [2], [3], [6]–[10], [12]–[14] are some examples for this category. Generally speaking, the first category has attracted more interest than the other over the past decade. However, with the development of capable computer systems, the need for more works in the second category has been raised. Given these facts, simulation, as a powerful, cost-effective, and time-saving tool plays an unsurpassed role to address these issues. Nevertheless, despite the fundamental importance of simulation tasks they has not been truly employed to predict and evaluate the performance of ADCs, typically. Performing high precision, fast, and cost-efficient simulations is therefore a real need in the present circumstances. Nonetheless, due to the generally complicated circuitry characteristics of these converters, modeling their non-ideality parameters is a challenging task. As a result, previous works are mainly focused on modeling and simulation of *some* of the particular non-ideality parameters of *specific types* of ADCs. However, the employability of these works are presumably restricted as a consequence of being proposed for only specific types of ADCs, and can not be reused for others. Moreover, in real world situations and practical applications, we are sometimes confronted with circumstances wherein the type and infrastructure properties of the ADC is fully unknown. Presumably, these sort of methods will not be employable in such circumstances. In this paper, we propose considering an ADC from a general point of view with its main characteristics and features, and try to initiate models which are appropriate for the most influential nonideality parameters of a typical ADC. As a result, the generality of this work is quite vaster than previous researches, being capable to be applied on and employed for generally any analog to digital converter without having to look at its infrastructure properties. Additionally, it makes it competent to be employed in practical situations where the type and circuitry features of the ADC is not known. The initiated models are described with appropriate details in Simulink environment, and eventually implemented for a typical ADC using practical non-ideality parameters data. The simulation results are presented to show the effect of the non-ideality parameters being modelled. Simulation results support the potential of the initiated models for the future higher precision general simulations of typical ADCs. The proposed Simulink model is designed in such a way to be reproducible and employable for fast and cost-effective handy simulation of a typical non-ideal ADC repeatedly.

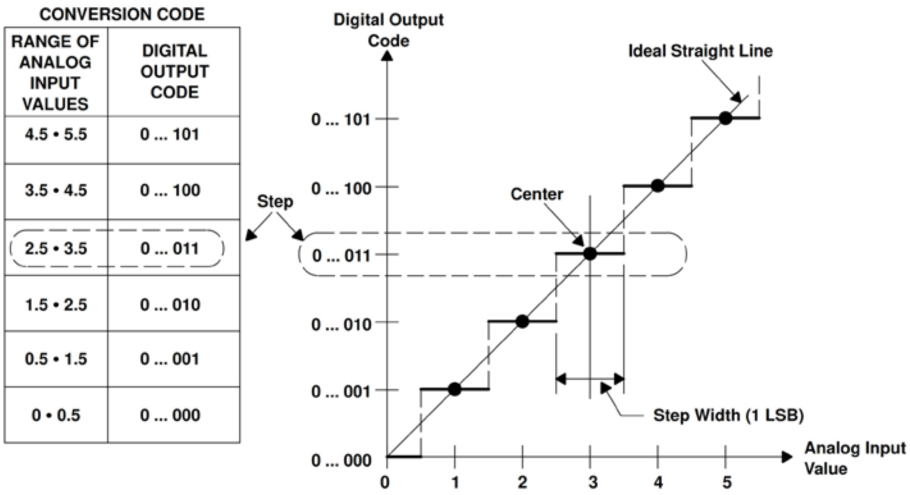
The rest of the paper is organised as follows. In Section 2 we reconsider the required preliminaries and bring a general view of the most important non-ideality parameters of a typical ADC into hand. In Section 3 we commence initiating appropriate models for the non-ideality parameters of a typical ADC discussed in Section 2 in Simulink environment. In Section 4, by creating a Simulink Block called typical non-ideal digital converter (TNDC), we perform the simulation task using real-world values and compare some results with an ideal ADC. Section 5 presents the limitations and describes some directions for future research, and eventually in Section 6, the conclusion of the work is presented.

2 Preliminaries and Unified View

Although there are various types of ADCs having different structures, nevertheless, the non-ideality parameters of a typical ADC can be generally classified into two main categories: static specifications and dynamics. The transfer function of an ideal ADC has been shown in Figure. 1 [16].

Static specifications of an ADC causing inaccuracy in conversion can be completely described by the following errors [16]:

- Finite resolution
- Offset error
- Differential nonlinearity (DNL)
- Integral nonlinearity (INL)



CONVERSION CODE

RANGE OF ANALOG INPUT VALUES	DIGITAL OUTPUT CODE
4.5 • 5.5	0 ... 101
3.5 • 4.5	0 ... 100
2.5 • 3.5	0 ... 011
1.5 • 2.5	0 ... 010
0.5 • 1.5	0 ... 001
0 • 0.5	0 ... 000

Figure 1: General transfer function of an ideal ADC [16].

Followings can be categorised as the dynamic specifications of a non-ideal ADC:

- Signal-to-noise ratio (SNR)
- Total harmonic distortion (THD)
- Signal-to-noise-and-distortion-ratio (SNDR)
- Effective number of bits (ENOB)
- Spurious-free dynamic range (SFDR)

While we do not aim to demonstrate extensive explanations of the ADC parameters in this article, however, reminding the fundamental concepts from a unified point of view can be handful for the next section. We assume these classified expressions along with the included figures can remind the reader about more in-depth details.

2.1 Static Non-Ideality

The static non-ideality parameters of an ADC can be stated in terms of both the least significant bit (LSB) and percentage of the full scale range $\%FSR$, which will be defined as follows. These could easily be converted to each other, as will be presented in Equation (39).

2.1.1 Finite Resolution

The resolution of an ADC is usually the number of bits which shows the maximum number of distinct levels available in conversion process [17]. In other words, the number of bits generally determines the resolution of the data acquisition system. A/D resolution can be mathematically expressed as follows:

$$V_{LSB} = V_{FSR}/(2^n - 1) \tag{1}$$

where V is the maximum input voltage and n is the number of conversion bits. LSB is the least significant bit at low output, and FSR is the full scale range, i.e. maximum input voltage at 2^n output, which is unreachable. In other words, there are 2^n possible digital output codes for an n-bit converter which each value is an equal fraction of the total input voltage range. Equation (1) is for a binary converter. However, it can be rewritten for decimal ADCs as follows:

$$V_{LSB} = V_{FSR}/(10^D) \tag{2}$$

wherein D represents the number of decimal digits. In this paper, we work with binary A/Ds due to their vast popularity.

2.1.2 Offset Error

The offset error is usually defined as the difference between the nominal and actual offset points on the analog output value vs. digital output code diagram of the ADC (Figure. 2)[16]. The nominal offset point for an ADC is the midstep value for which the digital output code is zero. This error has an equal influence on all codes by the same amount [16].

2.1.3 Gain Error

The gain error is usually defined as the difference between the nominal and actual gain points on the analog input value vs. digital output code transfer function of the ADC (Fig. 2) after the offset error is rectified to zero [16]. For an ADC, the nominal gain point is the midstep value for which the digital output code is at full range. In essence, this non-ideality error points out a change in the slope of the actual and ideal transfer functions, and thus matches the same percentage error at each step [16].

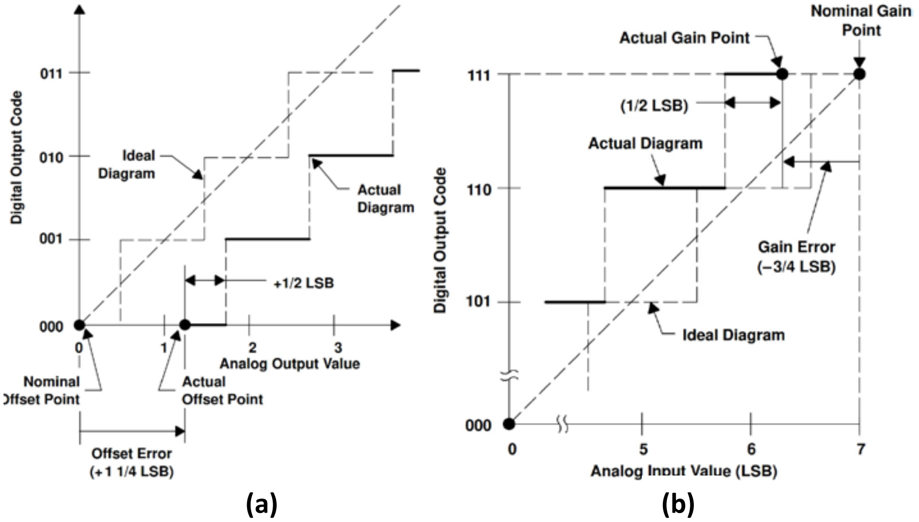


Figure 2. 3-bit converter: (a) Offset Error; (b) Gain Error.

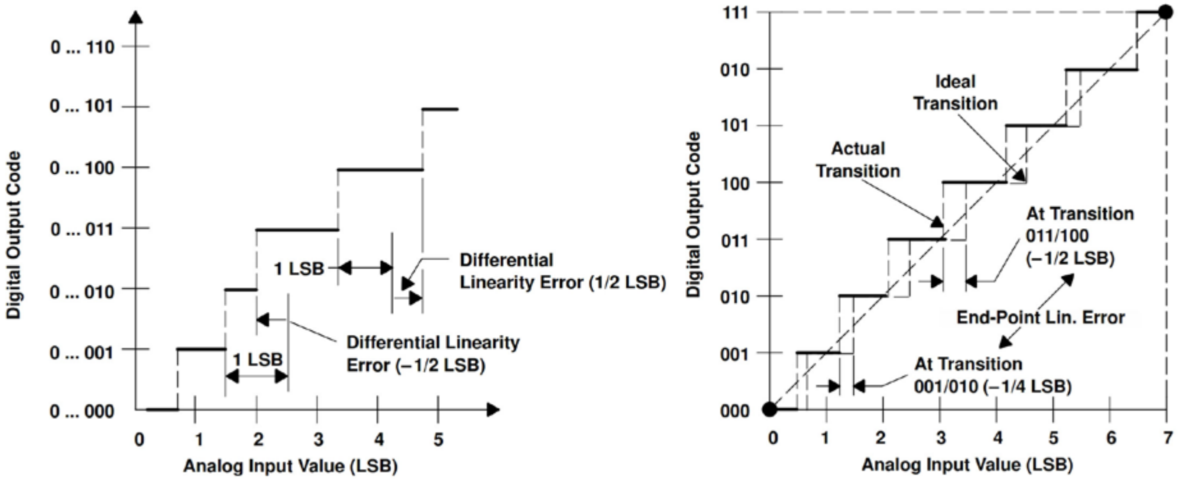
2.1.4 Differential Nonlinearity Error

The differential nonlinearity error for an ADC, as shown in Figure. 3, is the dissimilarity between each actual steps widths and the ideal value of 1 LSB widths [16]. Hence, if the steps widths are exactly 1 LSB, in that case the differential nonlinearity error is exactly zero. Nonetheless, in cases where the DNL exceeds the ideal 1 LSB width, there are possibility of encountering variety of non-linear behaviours, including the A/D turns to be nonmonotonic. This signifies that the magnitude of the output could become smaller in spite of increment in the magnitude of the input. Some other possibilities of non-linear behaviour could include having missing codes i.e., one or more of the possible 2^n binary values are never seen output. The following equation could therefore be used to describe or evaluate the DNL error for each digital (D) code mathematically:

$$DNLm = \frac{V_m^Q - V_{m-1}^Q - V_{LSB}}{V_{LSB}}. \tag{3}$$

The overall DNL is usually referred to as the maximum of individual DNLs:

$$DNL = \max(DNL_m), \qquad m \in [0, 2^n - 1]. \tag{4}$$



(a)

(b)

Figure 3. 3-bit converter: (a) Differential nonlinearity error; (b) Integral nonlinearity error [16]. Offset and gain errors are set to zero.

2.1.5 Integral Nonlinearity Error

The integral nonlinearity error is referred to the deviation in values of the actual transfer function from an ideal straight line. Figure. 3 shows this concept [16]. This straight line can be drawn so as to minimize the deviations, or it can also be a line fitted between the end points of the transfer function while the gain and offset errors have been corrected to zero e.g. by trimming. For a typical ADC these deviations are measured at the transition points from one step to the other. In fact, the summation of the individual differential nonlinearities (DNLs) from the bottom up to a particular step i.e.,k, determines the value of the integral nonlinearity at that step. The following equation describes this fact:

$$INL_k = \sum_{i=1}^{i=k} DNL_i, \tag{5}$$

or

$$INL_m = \sum_{m=0}^{2^n-1} DNL_m, \tag{6}$$

and the overall INL is defined as follows:

$$INL = \max(INL_m), \qquad m \in [0, 2^n - 1]. \tag{7}$$

2.2 Dynamic Non-Ideality

2.2.1 SNR

SNR refers to the power of signal in comparison to the power of *permanent* noise, simply written as:

$$SNR = \frac{P_s}{P_n}. \tag{8}$$

This, while converted to *dB*, follows the subsequent equation:

$$SNR_{dB} = 10\log\left(\frac{P_s}{P_n}\right) = 6.02N + 1.76 \quad dB \tag{9}$$

wherein N denotes the converter's total number of bits (maximum resolution). The noise mentioned in above equations are usually counted as quantization noise (error) which is inevitable even in ideal ADCs. This obeys the following formula:

$$V_{Q,rms} = \frac{V_{LSB}}{\sqrt{12}} = \frac{V_{FSR}}{2^N\sqrt{12}} \tag{10}$$

in which Q stands for quantization and rms is the root mean square; and equation (9)can be concluded from (10) [18], [19]. As a result, the SNR defined above is the ideal signal to noise ratio in any N-Bit A/D converter.

2.2.2 THD

Due to the non-linear characteristics of converter circuits' elements, other harmonics of the input signal also appear at the output. In other words, the multiplies of the main signal's frequency also appear at the output and is present along with the main frequency (i.e., nf_{sig} or nf_{in}). Assume having a single-tone signal at the input, this property is usually described by the following equations:

$$x_i = a_i\sin(2\pi f_i t + \varphi) \tag{11}$$

$$y_o = \sum_{k=1}^{m} \zeta_k x_i^k = \zeta_1 x_i + \zeta_2 x_i^2 + \cdots$$

$$= \zeta_1 a_i \sin(2\pi f_i t + \varphi) + \zeta_2 a_i \sin^2(2\pi f_i t + \varphi) +$$

$$= \xi_1 \sin(2\pi f_i t + \varphi) + \frac{\xi_2}{2}(1 - \cos(4\pi f_i t + \varphi)) + \ldots, \tag{12}$$

in which a_i, ζ_k and ξ_k are the amplitude coefficients, and x_i and y_o represent the input and output signal to the non-linear system, respectively. As can be seen, second, third and so forth multiplies of the f_i appears at the output unwontedly. Similarly, equations (11) and (12) could be modified to the following forms in cases of working with multi-tone signals:

$$x'_i = \sum_{i=1}^{n} a_i \sin(2\pi f_i t + \varphi) \tag{13}$$

and hence,

$$y_o = \sum_{k=1}^{m} \zeta_k x_i^k = \sum_{k=1}^{m} \sum_{i=1}^{n} \zeta_k a_i^k \sin^k(2\pi f_i t + \varphi)$$
$$= \sum_{k=1}^{m} \sum_{i=1}^{n} \zeta_k a_i^k \left(\sum_{j=1}^{p} \sin(2\pi(kf_i \pm k' f_i)t + \psi)\right) \tag{14}$$

In analog to digital conversion realm, similar to the second form mentioned above is usually called intermodulation distortion (IMD), which is the presence of distortions with additive frequencies at the output in addition to the correct multiples of the main frequency. In this paper we work with a tone sinusoid. After simplifications, equations (12) can be rewritten as:

$$y_o = HD_1 \sin(2\pi f_i t + \varphi_1) + HD_2 \sin(4\pi f_i t + \varphi_2) +$$

$$= \sum_{i=1}^{n} HD_i \sin(2\pi n f_i t + \varphi_i). \tag{15}$$

Generally, the amplitude of the distortion harmonics above the fourth (i.e. HD_4) is very small and negligible, i.e. $HD_i \approx 0$ for $i > 4$. Consequently, the total harmonic distortion (THD) can be calculated by calculating the distortion power as follows:

$$P_{dist} = \sum_{i=2}^{\infty} (\frac{HD_i}{\sqrt{2}})^2 = \sum_{i=2}^{\infty} \frac{HD_i^2}{2}, \tag{16}$$

and eventually:

$$THD = \frac{\sqrt{\sum_{i=2}^{\infty} HD_i^2}}{V(f_{sig})}. \tag{17}$$

Given the fact explained above, equation (17) reduces to the following:

$$THD = \frac{\sqrt{\sum_{i=2}^{4} HD_i^2}}{V(f_{sig})}, \tag{18}$$

where V represents the voltage or amplitude of the signal. THD is usually specified in units of dBc (dB to carrier, i.e. f_i).

2.2.3 SNDR

Signal to noise and distortion ratio (SNDR) is usually referred to the fact that, the SNR contains the inevitable quantization noise only, as explained before. However, there are other types of noises involved, i.e. thermal noise. As a result, the SNDR parameter takes into account the quantization noise, thermal noise, and the total harmonic distortion. The following equations demonstrate this fact:

$$SNDR_{dB} = 10\log\left(\frac{P_{signal}}{P_{noise+dist.}}\right) \tag{19}$$

$$SNDR = 10\log\left(\frac{V_{sig}^2}{V_Q^2 + V_{n,th}^2 + V_{dist}^2}\right). \tag{20}$$

Therefore, we can easily conclude that the following inequality is always true:

$$SNDR_{dB} \leq SNR_{dB}. \tag{21}$$

It is worth mentioning that SNDR is sometimes recalled as signal to noise and distortion ratio (SINAD) as well.

2.2.4 SFDR

The spurious-free dynamic range (SFDR) refers to the strongest harmonic's (f_{spur}) power to the main signal's power. It is therefore usually given in units of dBFS (dB full scale) or dBc (dB to carrier). The following equation describes this:

$$SFDR_{dB} = 10\log\left(\frac{V_{f_{sig}}^2}{V_{f_{spur}}^2}\right). \tag{22}$$

Commonly, the ADC's manufacturer specifies the harmonic which has the highest power e.g. HD_i. Figure. 4 shows this concept.

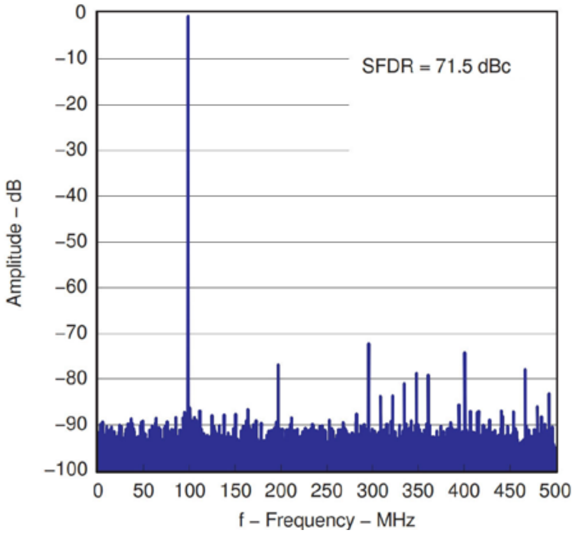


Figure 4: Spurious-free dynamic range (SFDR) [20].

For a pure sine wave input, SFDR can be seen as the ratio of the amplitude of the averaged Fourier transform (FT) value at f_{sig} or f_i, to the averaged FT amplitude of the largest magnitude harmonic or spurious signal component being detected in the full Nyquist band i.e., $\max\{V_{avg}[f_h] \cup V_{avg}[f_{spur}]\}$. The following equation describes this [1]:

$$SFDR_{dB} = 20\log\left(\frac{|V_{avg}(f_{sig})|}{\max\limits_{f_{spur} \cap f_h}\left(|V_{avg}(f_{spur})| \cup |V_{avg}(f_h)|\right)}\right) \tag{23}$$

in which avg denotes the averaged value and h stands for harmonic.

2.2.5 ENOB

Effective number of bits (ENOB) shows the concept associated with the SNDR explained in Section 2.2.3. Recall equation (9) where N represents the ADC's total number of bits. However, based on inequality (21)the *practical* signal to noise ratio is always less than the *nominal* one in equation (9). Thus, if we replace the total number of bits in (9) with the effective number of bits, we reach the value specified in (20), hence:

$$ENOB = \frac{(SNDR - 1.76)_{dB}}{6.02}.$$

3 Non-Ideality Parameters and Modeling

In this section, we initiate considering the most important and influential nonideality parameters of a typical ADC having general point of view in mind. This will assure that the research is not identical to previous efforts taking the infrastructure of ADCs into account, which thus restricted the generality of the proposed models for other types of ADCs. As a result, in this paper the initiated models are not produced by discussing circuit-level characteristics of a specific type of ADC. To achieve this main objective, we ought to consider the nonideality parameters of a practical ADC, looking from a widespread perspective. To do this, we propose the initiated models for these non-ideality parameters recalling the unified view discussed in Section 2. It is worth noting that the original analog input signal could be generally defined to be the sum of ℓ signals, $x_{in,\ell}(t)$, having the following format:

$$x_{in,\ell}(t) = a_{\ell,0} + \sum_{k=1}^{+\infty} a_{\ell,k}\sin(2\pi f_{\ell} k t + \psi_{\ell,k}), \tag{24}$$

in which the coefficients of the respective Fourier series are used. Consequently, the total input signal can be written as follows [14]:

$$x_{in}(t) = \sum_{\ell=0}^{\ell-1} x_{\ell}(t). \tag{25}$$

In this paper we aim at working with a tone conventionally. Nevertheless, the proposed model has the flexibility to be employed for the above definitions as well.

3.1 Total Harmonic Distortion

There are two ways of modeling the THD. In ideal situations, we assume that the input signal is known, or specified by the client manually. Therefore, the parameters of the input signal such as frequency, amplitude and phase are fully known. Having this, we can easily model the THD by defining the input frequency (and other parameters if applicable) as a variable, converting the HD_i data from datasheet to pure amplitudes, define a new wave with the frequency equal to if_{sig} and add them up together. Figure. 5 shows the model diagram for the second, third, and fourth distortion harmonics.

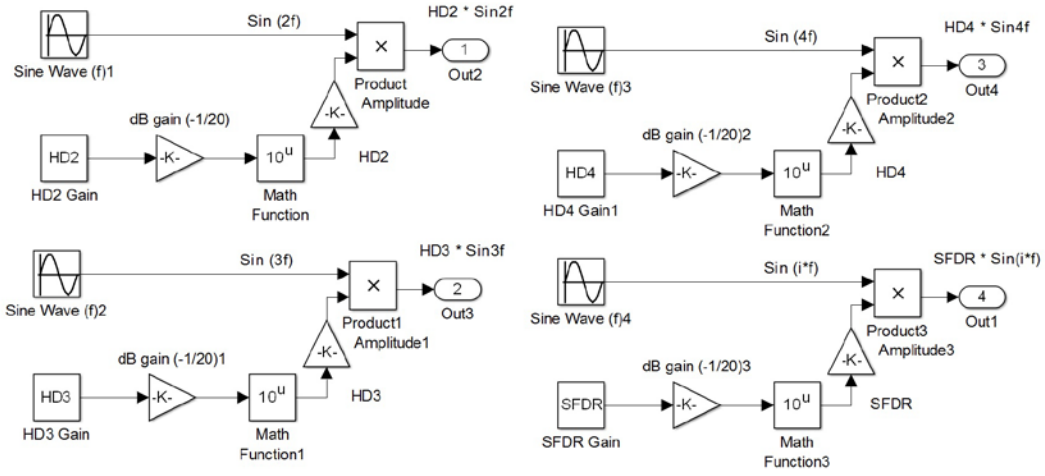


Figure 5: model (for $HD_i \approx 0$ for $i \pm 5$), and SFDR model ($f_{spur} = if_{sig}$). More harmonics can be added if needed.

We can obviously add more distortion harmonics to the above model as needed, and create a subsystem for ease of use. Nevertheless, as the second case, we can assume that the full information about the input signal might not be available. In this situation, we must use the input signal itself to generate the distortion harmonics. This is possible using sine and cosine expansions:

$$\sin^2\Theta = \frac{1-\cos 2\Theta}{2} \tag{26}$$

$$\cos^2\Theta = \frac{1+\cos 2\Theta}{2} \tag{27}$$

Other powers of the sine and cosine can be generated using the above main formulas. The cosine in (26) can be generated from the input signal by applying the Hilbert transform. We should note that for simulation purpose, x samples delay equal to the half of the Hilbert transform order should be made. It is also worth noting that shifting the input phase by $\pi/2$ will make some problems for the simulation and should not be always used. Another method could be usage of input derivatives, however, this can also make some problems, specially if the input signal is sampled or discrete in time. Hence, the output after this step is as follows:

$$x_{out,THD}(t) = x_{in}(t)|_{f=f_{sig}} + \sum_{i=2}^{p} HD_i x_{in}(t)|_{f_i=if_{sig}} \tag{28}$$

in which $x_{out,THD}$ denotes the output after the THD modeling step.

3.2 Spurious-Free Dynamic Range

The SFDR can be easily modeled by considering the information in Section 2.2.4. To do this, while applying the THD method discussed in Section 3.1 the SFDR of the ADC can also be constructed taking the information in ADC manufacturer's datasheet into account. For instance, the SFDR for the Texas Instruments ADS5400 A/D at f_{in} = $1200MHz$ is $66dBc$, and this occurs at the second harmonic (i.e., the spur's amplitude is HD_2). Hence, by generating a harmonic at the spur's frequency and applying the relevant amplitude given, the SFDR parameter is modelled (Figure. 5). Needless to say, that the relevant building blocks must be chosen according to the SFDR's given unit, to convert it back to pure amplitude usable. The output of this stage can be written as follows:

$$x_{out,SFDR}(t) = x_{out,THD}(t) + SFDRx_{in}(t)|_{f=if_{sig}}. \tag{29}$$

3.3 Offset Error

The offset value, δ, of the ADC which is entered by the user can be added after the THD subsystem. This will affect the output according to the offset error's definition. The output after this stage can be presented as follows:

$$x_{out,Off}(t) = x_{out,SFDR}(t) + \delta. \tag{30}$$

It is again notable that the relevant units across the different building blocks must be unified first, and dissimilar units to be converted to the appropriate ones before performing the simulation task.

3.4 Gain Error

The output of the last part discussed in Section 3.3 could be multiplied by the gain error (converted from %FS (%full-scale) to pure amplitude), η_G, and be added up with the previous signal to model the gain error effect. The output after this stage is as follows:

$$x_{out,G}(t) = x_{out,Off}(t) + \eta_G x_{out,Off}(t). \tag{31}$$

3.5 Aperture Delay

Aperture delay, which is the delay time between the rising edge of the input sampling clock and the actual time at which the sampling starts [20], can be modelled by placing a delay line right after the sampling clock input of the sample-and-hold (S/H) block. This will delay the actual starting point of the sampling task while the simulation time has already been started. Fig. 6 describes this method.

It is worth noting that, to model the aperture delay non-ideality of ADC we ought to employ the S/H block rather than using the simple zero-order-hold (ZOH).

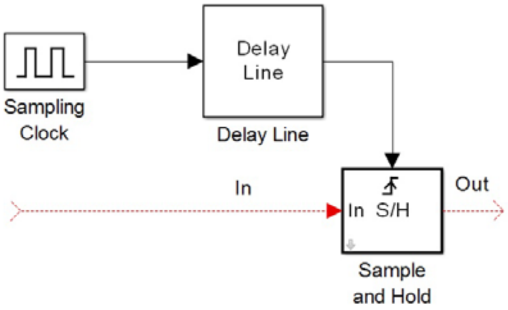


Figure 6: General aperture delay model. Maximum allowable aperture jitter for a given frequency f_i and finite resolution N is $t_j = 1/\pi f_{in} \times 2^{N+1}$.

3.6 Differential and Integral Non-Linearity

To model the DNL and INL, we initiate using a completely uniform random source to be added to the output node of the gain error, described in Section 3.4. Using this will assure having the maximum randomness for the aim of moving the ideal transfer function and changing the transition code length for the aim of modeling INL and DNL, respectively. The range of this uniformly random source equals to DNL, and the mean is equal to the INL value. We should also define these parameters as variables to be entered by the user. We do this using the mask property in Simulink. Therefore, the general random source to be added is:

$$Uniform\left\{\left(\frac{INL-DNL}{(2^n)-1} \times V_{FS}\right) : \left(\frac{INL+DNL}{(2^n)-1} \times V_{FS}\right)\right\}. \tag{32}$$

Another method can also be presented if we know the internal structure of the A/D. A third order power series is used to model the INL and DNL. Although this method's nature is far from the intuitive understandable ideas for modeling the INL and DNL like the previous method, it usually provides good accuracy, given the fact that it is similar to the conventional circuit non-ideality modeling. The elementary idea of this method was first proposed in a project at the University of California Berkeley under supervision of Professor Bernhard Boser [21]. Figure. 7 shows this method.

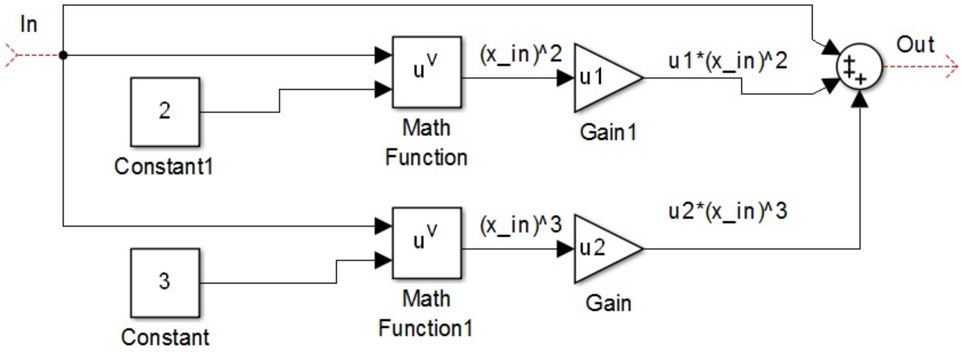


Figure 7: Second method for modeling DNL&INL.

$$\Gamma_{out} = \Gamma_{in} + \mu_1\Gamma_{in}^2 + \mu_2\Gamma_{in}^3. \quad (33)$$

It should be noted that we are restricted to choose either of the two positive or negative values for both the DNL and INL, and cannot afford producing a method which can entirely model the range from minimum negative to the maximum positive, as we are looking at the ADC from general perspective and do not intend to involve circuit or infrastructure level properties. The output after this stage will hence be:

$$x_{out,(D\&I)NL}(t) = x_{out,G}(t) + \rho|_{Uniform\{\alpha:\beta\}} \quad (34)$$

if we use the first method, and:

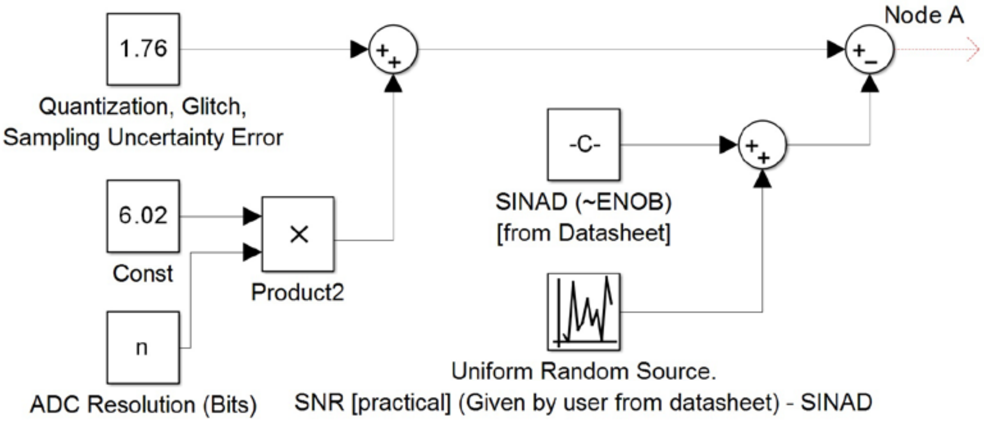
$$x_{out,(D\&I)NL}(t) = x_{out,G}(t) + \Gamma_{out}|_{\Gamma_{in}=x_{out,G}(t)} \quad (35)$$

if the second method is employed. ρ has the uniform distribution described in (32).

3.7 Other Non-Idealities Modeling

Recall the equations (19), (20), (21), and (9), and the discussions in Sections (2.2.3) and (2.2.1). Paying in-depth attention to those, we can propose an intuitive working way to achieve modeling other non-idealities of the ADC, helping to skip and overcome the theoretical difficulties. We explained that the difference between the theoretical (unreachable) signal to noise ration (known as SNR in ADC realm, or more accurately SNR_{ideal}) is caused by other types of noises except the unavoidable quantization noise. Those include thermal noise, distortions and many other things. Although some of these are partly predictable knowing the circuit level information of the specific type of ADC, nevertheless, it is hidden to us from systematic point of view. So far, we have modeled some non-idealities including the total distortion caused by the ADC's non-linear nature. We can now assume that any other non-ideality causing the SNDR to be lower than the theoretical SNR, (21), can be modeled by adding a white Gaussian noise having the power coming from an external port. As the mean of the Gaussian noise is zero, the power evaluation block will determine the variance. To do this, we need to go through the following steps:

1) We calculate the ideal SNR using (9). To do this, we only need the ADC's total number of bits, given by the user.
2) Sometimes in practical and more commonly in industrial ADCs, there is very small and negligible difference between the SINAD value and the 'practical' SNR, which expectedly should be the same. To take this effect into account as well, we add a uniform random source δ ranging between zero and the $\max(SNR_{ptactical} - SINAD)$ to the SINAD building block. This presumably owns a very small value and will have negligible effect on simulation results. The SINAD ($\sim$ ENOB) and the practical SNR comes from the ADC's datasheet and is entered by the user. Likewise before, by defining variable parameters at each building block of the simulation model using the mask property of the Simulink, we can use them in other blocks. Figure. 8 demonstrates the first two steps.



Node A
1.76
Quantization, Glitch, Sampling Uncertainty Error
6.02
Const
n
ADC Resolution (Bits)
Product2
-C-
SINAD (~ENOB) [from Datasheet]
Uniform Random Source. SNR [practical] (Given by user from datasheet) - SINAD

Figure 8: First steps of modeling other non-idealities of ADC

$$NodeA = (1.76 + 6.02N) - (SINAD + \vartheta)$$

$$\vartheta|_{Uniform(0:SNR_{practical} - SINAD)} \tag{36}$$

3) At this step, we need to compute two power quantities: the total input power (i.e., $P(x_{in}) = P_{in}$) and the power after the INL and DNL modeling block (i.e., $P(x_{out,(D\&I)NL}(t))$, point P2 in Figure. 9). The reason is, from the total difference between the SNR_{ideal} and the SINAD values, we have already modeled some non-idealities till point P2. Therefore, before adding a Gaussian noise to the model to compensate the rest, we should calculate the effect of the previous modelled parts on the noise level (i.e., how much noise has virtually been added from the beginning till point P2), subtract it from the $SNR_{ideal} - SINAD$, and then add the remaining unconsidered noise to the model to modify and construct the correct difference between the

SNR_{ideal} and SINAD in our simulation. To do these, we place a power evaluation block at the beginning of the model to determine the input power i.e., P_{in}. It is notable that we cannot use the mathematical formulations to calculate the total input power, as we have considered the input to be unknown in general. To use the evaluated data, the output is connected to a "Data Store Write" block accompanied by a "Data Store Memory". Similarly, the total power at point P2 can be evaluated. The difference between the total input signal power and the power at point P2 would therefore be the power of added noise, i.e. n'. Hence:

$$P_{n'} = P_{in,signal} - P2 \qquad (W), \tag{37}$$

$P_{n'}$ must be converted to dB.

4) The output of the above (which is in dB) must be converted back to amplitude to be usable by data store and memory blocks, and then be entered as the input variance of the white Gaussian noise needed. Figure. 9 presents the above procedure.

After doing the above procedures, we can now place an additive white Gaussian noise (AWGN) to the model which its variance comes from external port (i.e., the Data Store Read block) (Figure. 8).

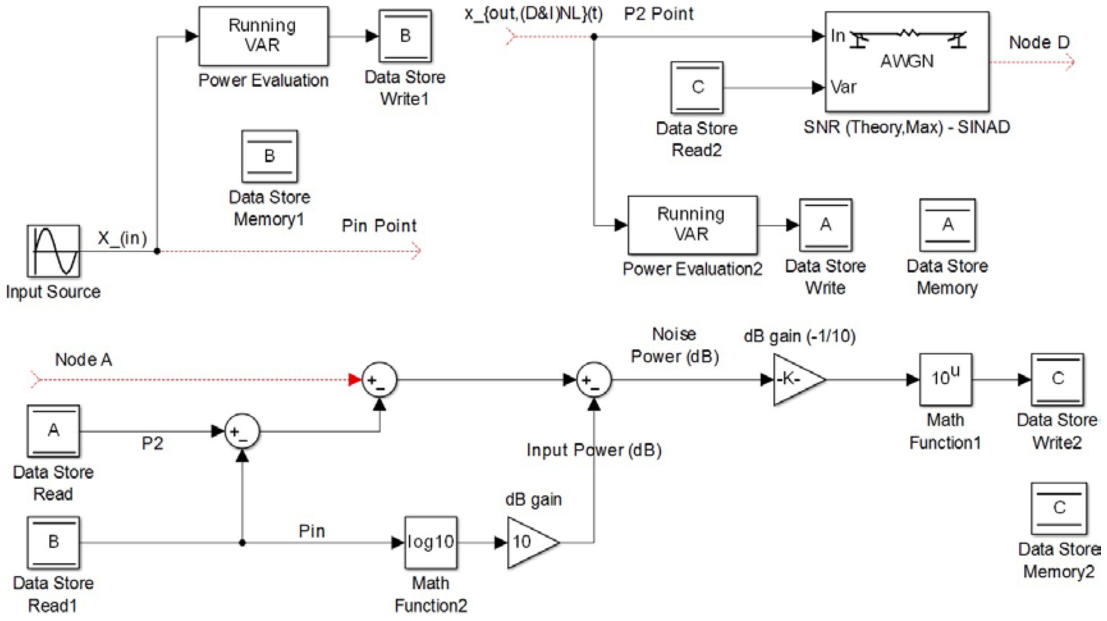


Figure 8: First steps of modeling other non-idealities of ADC.

4 Simulation Results

In this section, by placing all the aforementioned building blocks in a serial fashion, we have completed the whole simulation model having the non-ideal ADC in hand. By creating a Simulink subsystem we can make a handy block to be used each time we aim to simulate a new ADC having different non-ideality parameters. For this purpose, we define the input values of the subsystem (TNDC Block) to be the same as the values needed for each of the previously considered blocks. In other words, by doing this, the non-ideal ADC parameters will be entered by the user each time. To do this, we can define the non-ideality parameters' values at each individual block discussed previously as a double integer. In this case, at each individual block, we will enter the respective defined parameters names instead of a constant integer. This values are specified by the ADC's manufacturer and is findable in their respective datasheet. In addition, some points should be taken into consideration while putting the individual blocks in serial to each other. The output of each building block which is the input to the next block must have same characteristics to be mathematically acceptable. For instance, the output of one block could not be in LSB units while entering a block which works with pure amplitudes. The following formula is useful for conversion:

$$n\ \ LSB = x\ \ \%FSR$$

$$n \times \frac{V_{FSR}}{2^{N}-1} \times 100 = x\%\ \ V_{FSR}$$

$$n_{LSB} \times \frac{100}{2^{N}-1} = \%x_{FSR}. \qquad (38)$$

Table 1: The overall initiated model for simulating a typical non-ideal analog to digital converter in MATLAB Simulink

NON-IDEALITY PARAMETERS (MOSTLY FOR TEXAS INSTRUMENTS ADS5400)	VALUE
Input amplitude ($\boldsymbol{a}$)	1
Input frequency (rad/sec) ($\boldsymbol{f_i}$ or $\boldsymbol{f_{sig}}$)	$2\pi125e6$
Input phase (rad) ($\boldsymbol{\varphi_i}$)	0
ADC Resolution (Bits) ($\boldsymbol{N}$)	12
SFDR (dBFS or dBc)	66
SINAD (~ENOB) (dB)	57.5
SNR (Practical, from datasheet)(dB)	57.6
Offset Error (V)	$2.5e-3$
Gain Error (%FS)	0.05
Differential Non-Linearity (DNL) (LSB)	0.7
Integral Non-Linearity (INL) (LSB)	2
ADC Sampling Time (~$\boldsymbol{f_s}$)	$1/1200e6$
i-th Harmonic Distortion ($\boldsymbol{HD_i}$)(dBc)	$HD_2 = SFDR = 66$ $HD_3 = 70$
Min input voltage at logical low output (V)	-1
Max input voltage at $\mathbf{2^N}$ output (V)	$+1$

By entering the output of the above serial to S/H and quantizer blocks (or simply ZOH and quantizer if aperture delay or clock duty cycle is not aimed to be modeled) we can have the simulation task started. In this paper, we will have the following parameters to be entered by the client before the simulation process starts: Input amplitude, Input frequency (rad/sec), Input phase (rad), Input (sine) sample time (if applicable; i.e., sinusoids sampled with ultra high frequency), ADC resolution (Bits), SFDR (dBFS or dBc), SINAD (~ENOB)(dB), SNR (dB), Offset (V), Gain Error (%FS), DNL (LSB), INL (LSB), ADC sampling time (plus sampling clock duty cycle, if applicable), HD2, HD3, HD4, HD5 (in case that a particular HD_i has considerable amplitude, the user can simply change the mask variable in THD modeling section to have the frequency equal to if_{sig} and enter the relevant amplitude to model that specific harmonic too), Min input voltage at logical low output (V), and Max input voltage at 2^n output (V). Practical data listed in Table. 1 is used for the simulation (i.e., values will be entered by the user based on his own data) tasks.

A strong advantage of the proposed system is, that the employed data in Table. 1 can be easily changed by the user according to different practical ADCs, owing to the structure of the model. The overall employed model which is used for simulation task is depicted in Figure. 9.

Needless to mention that, the basic parameters of the ADC apart from the non-ideality parameters can be modelled using internal parameters of individual building blocks being employed in this model. For instance, the ADC number of bits or minimum and maximum voltage levels can be entered in the idealized quantizer block. In case of using ordinary quantizer block in Simulink, we should enter the quantization interval instead. The simulation is performed for a 12-bit non-ideal ADC having practical parameters' values listed in Table 1, and for 1024 data samples. The following figures demonstrate the simulation results and output waveforms. The output spectrums can be obtained by applying appropriate fast fourier transforms.

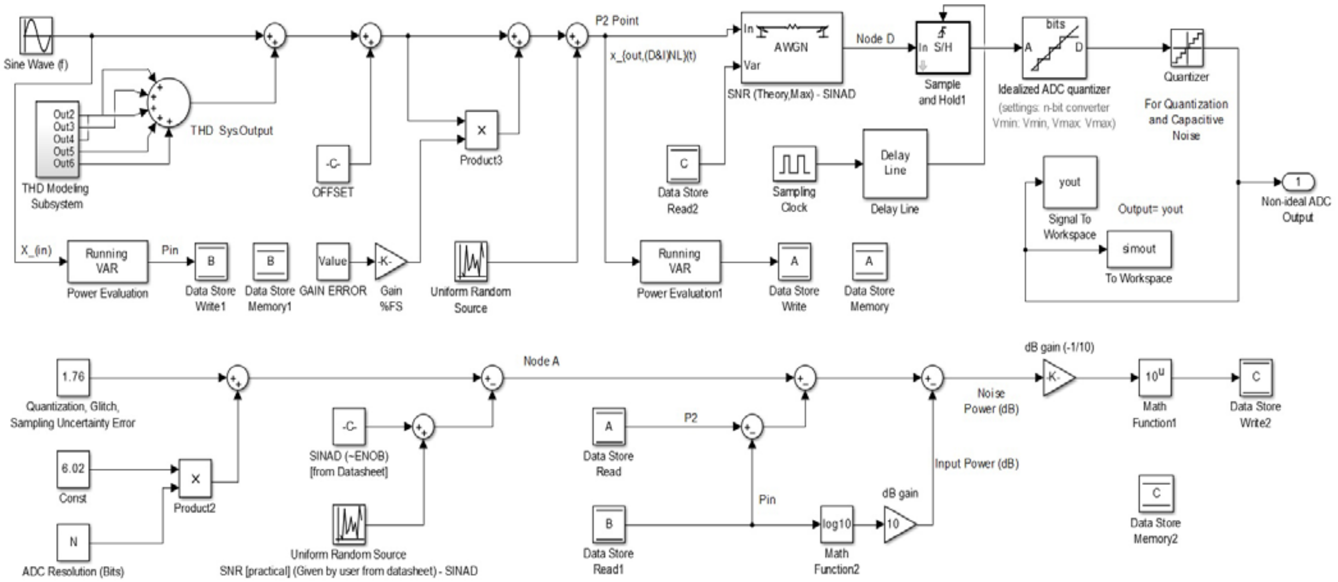


Figure 9: The overall initiated model for simulating a typical non-ideal analog to digital converter in MATLAB Simulink.

Since the distorting harmonics have very low amplitude comparing to the amplitude of the main frequency, their effect on the signal cannot be easily seen in time domain. However, Figure. 11 shows that the proposed model in Section 3.1 perfectly carries out the THD modeling. The simulation results can be exported to the MATLAB Workspace to be used for mathematical manipulations, as well as being accessible for MATLAB functions and/or M-Files. You can see the ideal ADC's results being compared to the non-ideal ADC modelled in this paper.

At this stage, it is important to remember that although the proposed TNDC simulation Block was a fundamental step forward towards creating typical ADC models and paving the way for handy and inexpensive simulation tasks, it cannot be compared to manufacturers' experimental results, due to the following facts and obstacles, which some of them could to be addressed by future works:

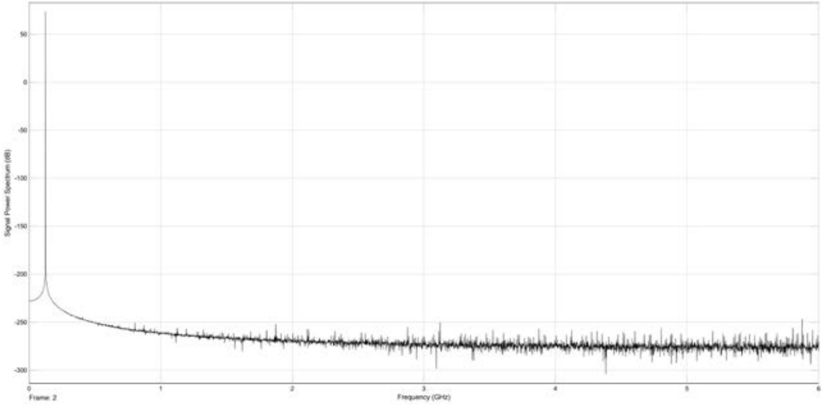


Figure 10: Power spectral after adding offset and gain errors.

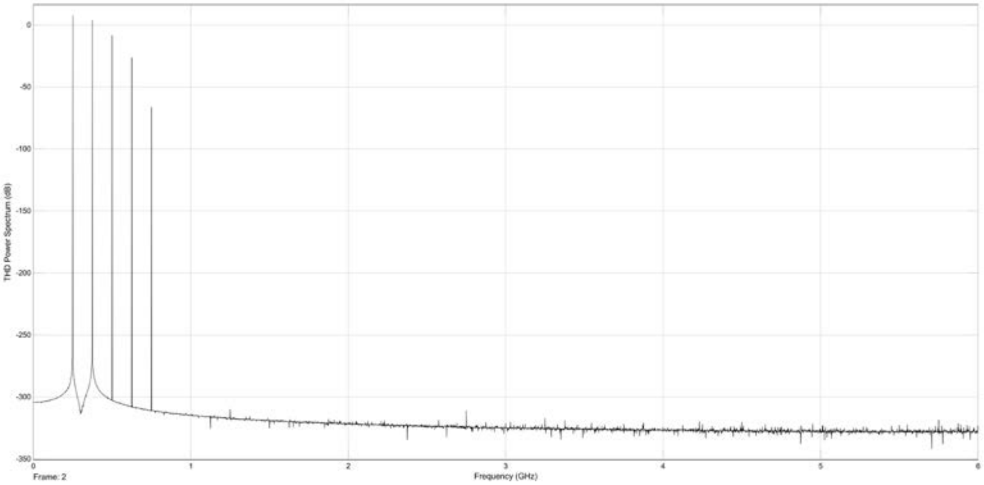


Figure 11: THD power spectral.

1. Most manufacturers use their own set of tools for extracting characteristics and experimental plots. Those are deeply linked with the infrastructure of each particular ADC.

2. The results in practical ADCs' datasheets are obtained by practical test experiments as mentioned above, and not performing simulation tasks. Consequently, you can find different results based on different local temperatures and so forth. Evidently, there are many affective issues (including physical circumstances) that cannot be taken into account in simulation tasks, especially when we do not emphasize on infrastructure and circuitry features of each particular converter.

3. Practical testing diagrams of ADCs always contain many other linear and non-linear elements which does affect the obtained experimental results, however, cannot be modelled here.

4. There are other non-ideality parameters apart from those modelled in this paper, which although are not much important, nevertheless, they do affect the simulation results somehow.

5. Above all the aforementioned points, there are some other non-ideality parameters that can be neither correctly measured nor being involved in simulation procedure, e.g. packaging non-idealities. Once again, we should emphasize that having a handy TNDC Block which can acceptably simulate typical ADCs according to their fundamental non-ideality parameters, can be notably helpful while simulating complicated projects. Employing the model and running the simulation repeatedly does not increase any overhead cost while offering a valuable vision about what's going on.

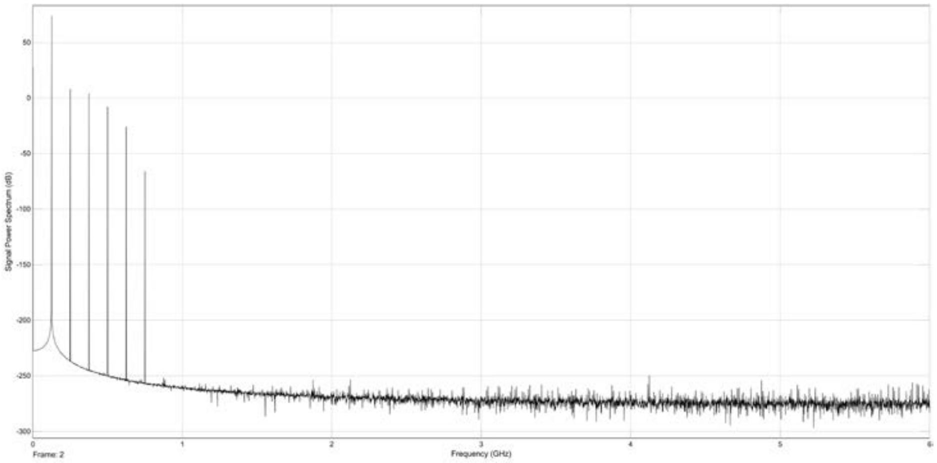
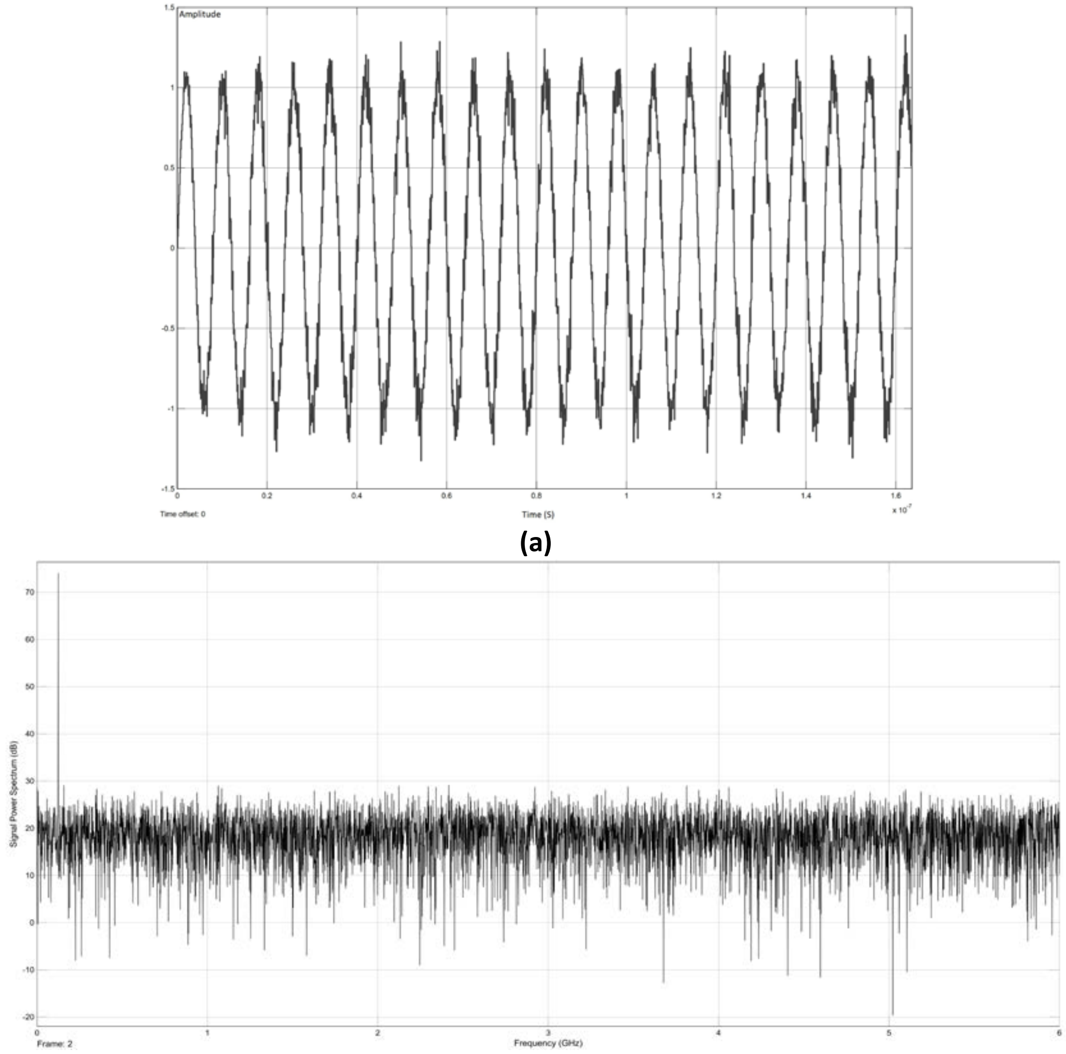


Figure 12: Power spectral after modeling offset and gain errors with THD.



(a)

(b)

Figure 13: Signal after AWGN block: (a) time-domain; (b) Power spectral.

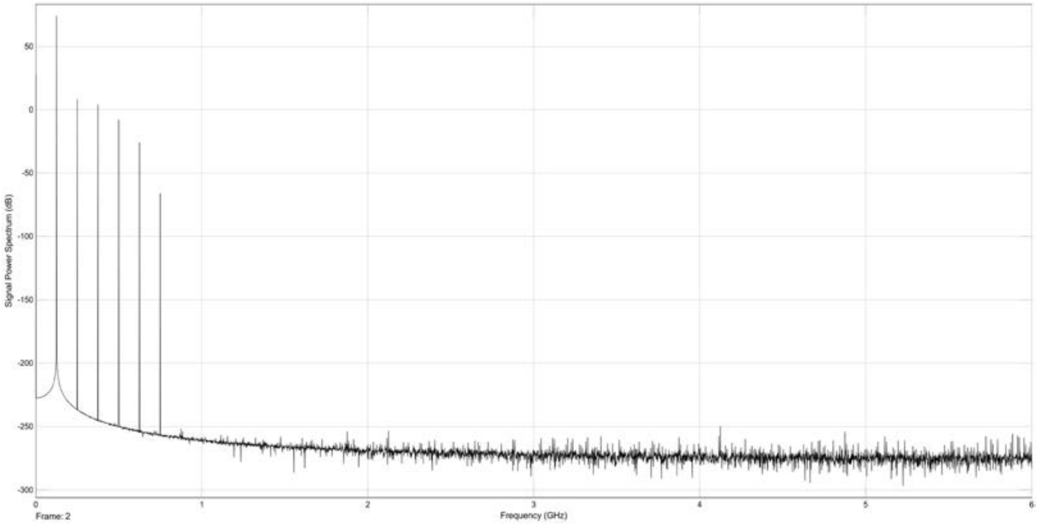
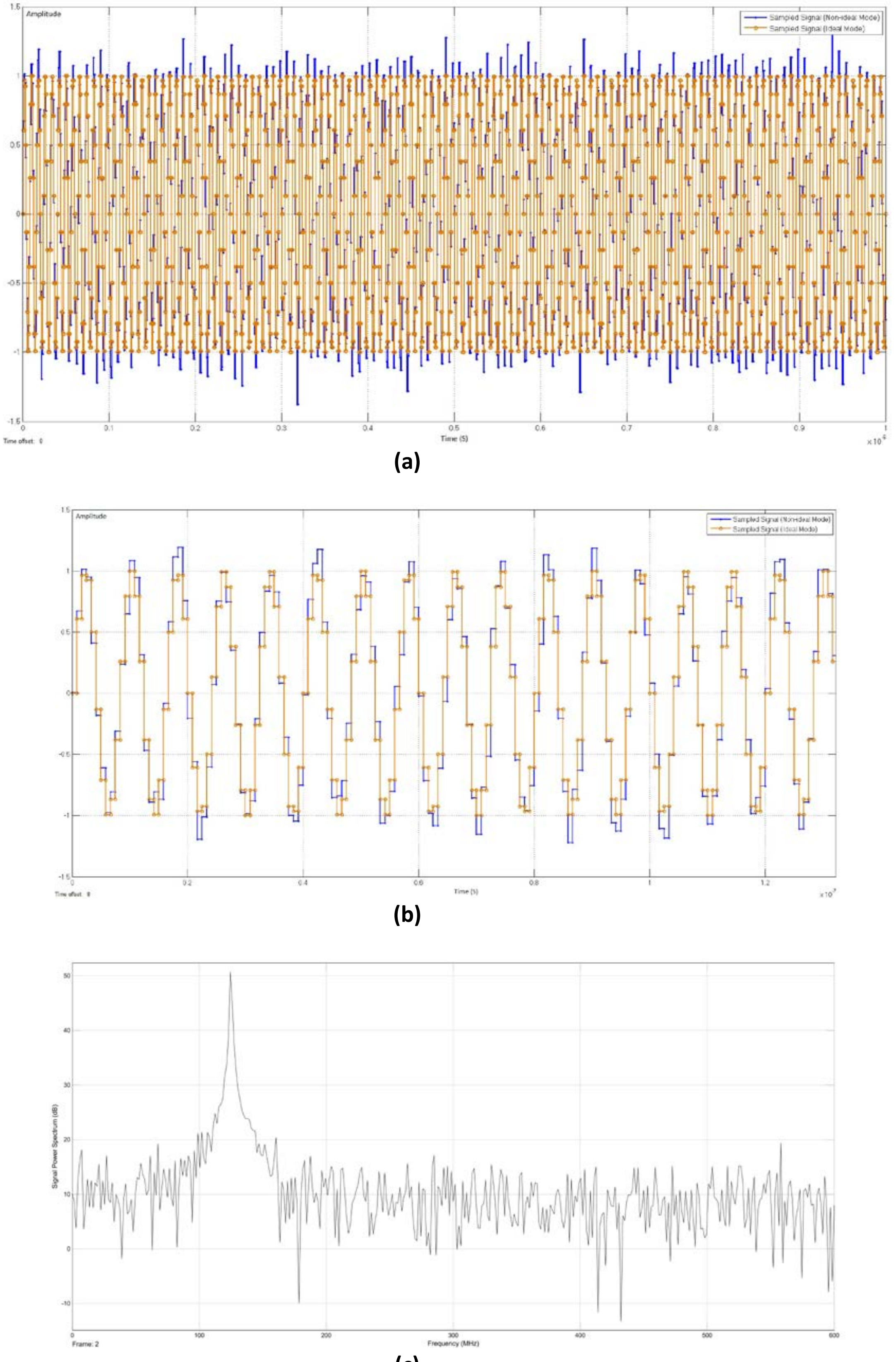


Figure 14: Power spectral after modeling offset and gain errors with THD.



(a)

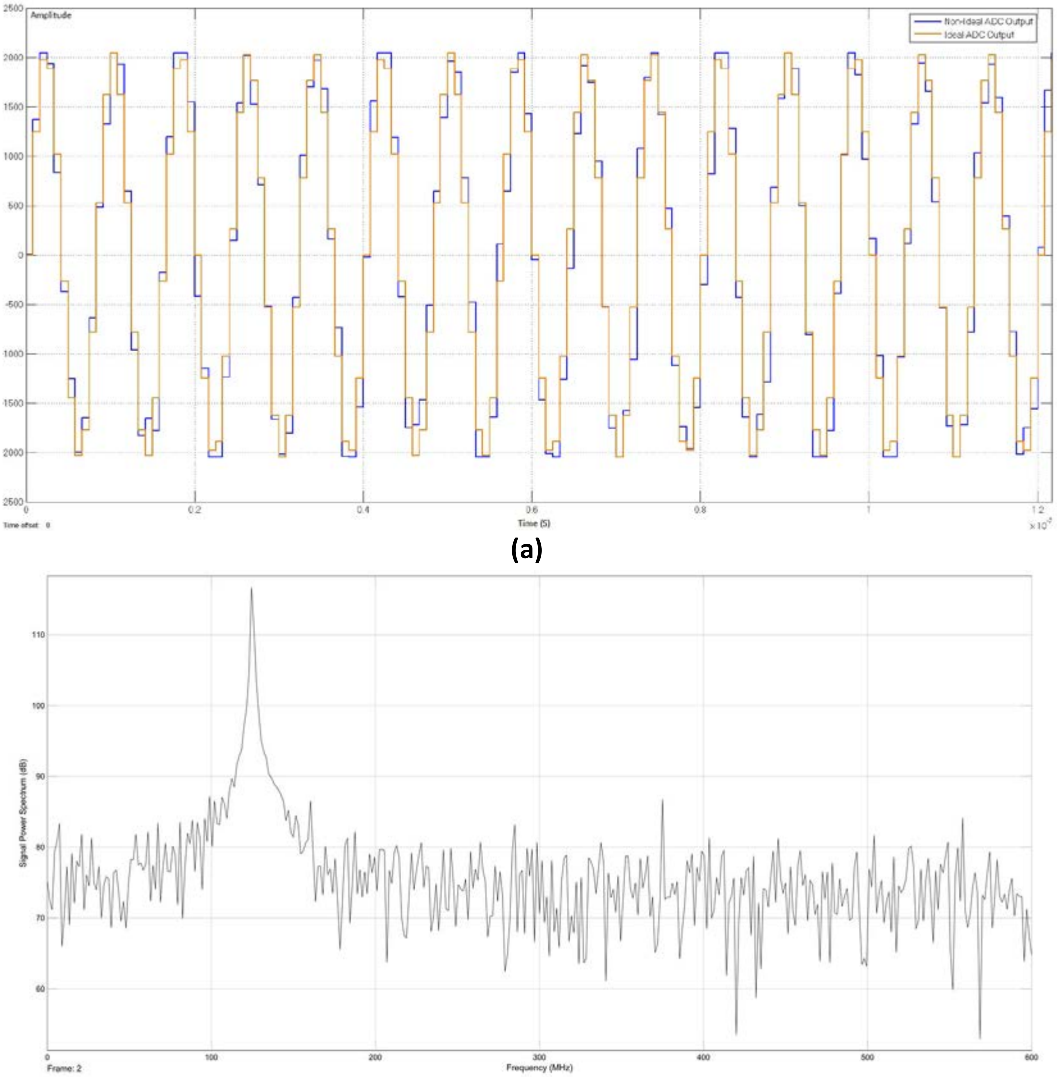
(b)

(c)

Figure 15: Sampled Signal: (a),(b) time-domain; (c) Power spectral.

5 Limitations and Future Research

As mentioned earlier, although this paper presented some first steps towards simulating practical analog to digital converters, it is essential to be aware of the need to considerable more in-depth works to achieve more realistic assumptions and implementations. Considering a non-ideal ADC to be a typical black box has always been a challenge in simulation and implementation of real-world projects, and was the main challenge of this paper too. However, this obstacle has to be somehow overcome in order to pave the way for more handy and inexpensive simulations of bigger projects. In addition to the points mentioned at the end of Section 4 to be addressed, considering an ADC as a typical converting block having some non-ideality parameters as its inputs has some drawbacks inevitably, as a result of not contemplating the actual infrastructure of each ADC itself. Therefore, some more advanced research should be carried out to consider the relations of non-ideality parameters with the infrastructural features of ADCs.



(a)

(b)

Figure 16: Signal after quantization (digital signal): (a) time-domain; (b) Power spectral.

Moreover, a unified version of these parameters and models based on different types of ADCs can notably direct to simulate typical ADCs more realistically. The above suggestions could be pointed as some directions, needless to say, there are a lot more to do than what described.

6 Conclusion

In this paper, we initiated a complete MATLAB Simulink model for a typical ADC without restricting it to a specific type. Considering an ADC as a black box while being competent for simulating non-ideality parameters was a notable challenge. By presenting an initiative model for most important non-ideality parameters of an ADC, in general, we proceeded an appreciable step towards more in-depth simulations of typical ADCs in future. This work can potentially lead to quick, handy and inexpensive rough simulation of ADCs in bigger projects, where splitting the whole project's diagram into parts can play an effective role in simplifying the overall simulation tasks, and considerably decreasing complicatedness. By placing individual building blocks in serial fashion, and defining the non-ideality parameters as parametric variables using Simulink mask feature, we created the whole Simulink model, namely, typical non-ideal digital converter (TNDC) Block, which takes the non-ideality parameters' values from the client before the simulation process starts. By showing the output of each of important TNDC blocks' nodes, we demonstrated the performance of the proposed method. At the end, we employed the TNDC Block to simulate a typical ADC using practical values for its non-ideality parameters. The results prove the validity of the proposed models according to the non-ideality parameters' definitions.

ACKNOWLEDGEMENT

This work was supported by the Research Committee of Barthawa Institute of Technology. The authors would also like to thank respected reviewers for their comments.

REFERENCES

[1]. IEEE-SA_Standards_Board, “IEEE standard for terminology and test methods for analog-to-digital converters,” 2000.

[2]. H. Zare-Hoseini, I. Kale, and O. Shoaei, “Modeling of switched-capacitor delta-sigma modulators in Simulink,” *IEEE Transactions on Instrumentation and Measurements*, vol. 54, no. 4, pp. 1646–1654, 2005.

[3]. P. Malcovati, S. Brigati, F. Francesconi, F. Maloberti, P. Cusinato, and A. Baschirotto, “Behavioral modeling of switched-capacitor sigma-delta modulators,” *IEEE Transactions on Circuits & Systems I Fundam. Theory Appl.*, vol. 50, no. 3, pp. 352–364, 2003.

[4]. G. Li, Y. M. Tousi, A. Hassibi, and E. Afshari, “Delay-line-based analog-to-digital converters,” *IEEE Transactions on Circuits & Systems II Express Briefs*, vol. 56, pp. 464–468, 2009.

[5]. H. S. Lee and C. G. Sodini, “Analog-to-digital converters: Digitizing the analog world,” *Proceeding of IEEE*, vol. 96, pp. 323–334, 2008.

[6]. S. Brigati, F. Francesconi, P. Malcovati, D. Tonietto, A. Baschirotto, and F. Maloberti, “Modeling sigma-delta modulator non-idealities in Simulink,” *1999 IEEE Int. Symp. Circuits Syst. VLSI*, vol. 2, 1999.

[7]. A. Fornasari, P. Malcovati, and F. Maloberti, “Improved modeling of sigma-delta modulator non-idealities in Simulink,” in *Proceedings - IEEE International Symposium on Circuits and Systems*, 2005, pp. 5982–5985.

[8]. S. Jaykar, P. Palsodkar, and P. Dakhole, “Modeling of sigma-delta modulator non-idealities with two step quantization in MATLAB/Simulink,” in *Proceedings - 2011 International Conference on Computational Intelligence and Communication Systems, CICN 2011*, 2011, pp. 532–536.

[9]. V. Kledrowetz and J. Haze, “Analysis of non-ideal effects of pipelined ADC by using MATLAB - Simulink,” *Adv. Sensors, Signals Mater. Anal.*, pp. 85–88.

[10]. D. P. Jayker, Shashant, Palsodkar Prachi, “Modeling of sigma-delta modulator non-idealities in MATLAB/Simulink,” in *Proceedings - 2011 International Conference on Communication Systems and Network Technologies, CSNT 2011*, 2011, pp. 525–530.

[11]. M. Koe and J. Z. J. Zhang, “Understanding the effect of circuit non-idealities on sigma-delta modulator,” *Proc. 2002 IEEE Int. Work. Behav. Model. Simulation, 2002. BMAS 2002.*, 2002.

[12]. J. M. Lei, X. W. Dai, X. C. Zou, and Z. G. Zou, “Modeling non-idealities of sigma delta ADC in Simulink,” in *2008 International Conference on Communications, Circuits and Systems Proceedings, ICCCAS 2008*, 2008, pp. 1040–1043.

[13]. G. Qingbo, J. Xinzhang, and T. Hualian, “Co-simulation of pipeline ADC using simulink and PSpice,” in *Proceedings - 4th International Conference on Intelligent Computation Technology and Automation, ICICTA 2011*, 2011, vol. 2, pp. 487–490.

[14]. C. Vogel and H. Koeppl, “Behavioral modeling of time-interleaved ADCs using MATLAB,” in *Proceedings of the AUSTROCHIP 2003*, 2003, no. October, pp. 45–48.

[15]. M. Webb and H. U. A. Tang, “System-level simulation for continuous-time delta-sigma modulator in MATLAB Simulink,” in *Proceedings of the 5th WSEAS Int. Conf. on Circuits, Systems, Electronics, Control & Signal Processing, Dallas, USA.*, 2006, pp. 236–241.

[16]. T. I. Inc, “Understanding data converters, SLAA013,” *Mix. Prod. Appl. Reports*, pp. 1–22, 1999.

[17]. B.-B. Texas Instruments (Inc., “Principles of data acquisition and conversion,” *BURR-BROWN Appl. Bull.*, no. 602, pp. 1–6, 1994.

[18]. M. J. M. Pelgrom, *Analog-to-digital conversion*. Springer New York Press, 2012.

[19]. I. Glover and P. Grant, *Digital Communications*. McGrew Hill Publications, 2009.

[20]. T. I. Inc, “ADS 5400, 12-Bit , 1-GSPS analog-to-digital converter,” *Texas Instruments Inc.*, no. March, 2010.

[21].] E. Ng and M. Bohsali, “Multifrequency cell impedence measurement,” University of California, Berkeley, USA., 2010.