Characterization and Digital Correction of Multi-Stage Analog-to-Digital Converters

by

Dragos Cartina

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the the degree of Masters
of Engineering

Ottawa-Carleton Institute for Electrical Engineering
Department of Electronics
Faculty of Engineering
Carleton University
Ottawa, Canada

The undersigned recommend to the Faculty of Graduate Studies

and Research acceptance of the thesis

"Characterization and Digital Correction of Multi-Stage

Analog-to-Digital Converters"

submitted by Dragos Cartina (B.Eng.) in partial fulfillment of the

requirements for the degree of Master of Engineering

Chairman, Department of Electronics
Professor Jim Wight

Thesis Supervisor
Professor Martin Snelgrove

Carleton University

December 10, 1997

Abstract

Multi-stage analog-to-digital converters are the dominant choice in applications that require both high speed and high accuracy, such as video and wideband radio.

Existing characterization methods only provide information about the performance of the ADCs at the system level, and therefore offer little insight into the causes of nonlinearity. In this thesis we propose a method geared specifically toward multi-stage architectures that can characterize the internal blocks and estimate the differential nonlinearity (DNL) and the integral nonlinearity (INL) of the A/D and D/A subconverters, and the gain of the inter-stage amplifiers. This method has been applied to a two-stage ADC and a pipeline ADC.

An adaptive on-line digital correction algorithm derived from the characterization method is also presented. A pipeline ADC was designed to test this algorithm, and digital correction resulted in an improvement of the Spurious-Free Dynamic Range by 6dB over three quarters of the bandwidth.

Acknowledgments

The two years I spent working on my M.Eng. thesis have been a great experience and changed my life in many (positive) ways.

All this has been possible thanks to Martin Snelgrove who supported and guided me all along, and provided me with excellent ideas and solutions. I am heavily indebted to him.

Many thanks to Phil Lauzon for designing my test boards, to Alana Wirta for helping me with the paperwork, and to the other people in the lab for the many times they answered my questions and gave me good advice.

I am very grateful to Timo Rahkonen and Antti Mantyniemi from the University of Oulu for helping me design my first chip. Also thanks to Analog Devices for providing us with several of their chips.

Finally, I would like to express my gratitude to my parents and sister for their love, support, and understanding.

Table of Contents

List of Figures

Glossary of Terms

A, A_k	actual stage/inter-stage gain
$\hat{A}$, $\hat{A}_k$	estimated stage/inter-stage gain
$\tilde{A}$, $\tilde{A}_k$	ideal stage/inter-stage gain
A/D	Analog-to-Digital
ADC	Analog-to-Digital Converter/Subconverter
d	digital output of overall A/D converter
Δ	length of reference region for bound estimation
δ	adjustment step for digital correction algorithm
d_k, d_{k-j}	digital output of A/D subconverter
D_k	digital output of conversion stage of pipeline ADC
D_k^+, D_k^-	comparator outputs of conversion stage of pipeline ADC
D_{k_j}	digital output of substage (magamp) of two-stage ADC
D/A	Digital-to-Analog
DAC	Digital-to-Analog Converter
Δl_k	D/A subconverter error: $l_k - \tilde{l}_k$
DNL	Differential Nonlinearity (def. page 24)
$\Delta t_{1,k}$	coarse/first A/D subconverter threshold error: $t_{1,k} - \tilde{t}_{1,k}$
$\Delta t_{2,k}$	fine/second A/D subconverter threshold error: $t_{2,k} - \tilde{t}_{2,k}$
DVM	digital voltmeter
$E[\ ]$	expected value
ENOB	Effective Number of Bits (def. page 25)
FFT	Fast Fourier Transform
FS	Full Scale
h	bin height
$\bar{h}$	average bin height
h_j	height of bin j of unidimensional histogram

$h_{j,k}$	height of bin (j, k) of bidimensional histogram
H_j	height of bin j of reference histogram
$\overline{\hat{h}_k^+}$, $\overline{\hat{h}_k^-}$	average bin height in the vicinity of the transition region
INL	Integral Nonlinearity (def. page 25)
l_k	actual D/A subconverter output levels
$\hat{l}_k$	estimated D/A subconverter output levels
$\tilde{l}_k$	ideal D/A subconverter output levels
LMS	Least Mean Square
LSB	Least Significant Bit
LSB_k	voltage corresponding to LSB of A/D suconverter k
m_k	actual lower bound of section k of residue
$\hat{m}_k$	estimated lower bound of section k of residue
M_k	actual upper bound of section k of residue
$\hat{M}_k$	estimated upper bound of section k of residue
MSB	Most Significant Bit
pdf	probablity density function
R, R_1, R_2	nominal signal range at the input of converter / subconverter
R_{in}, R_{in1}, R_{in2}	input range of ADC converter / subconverter
r_k^+	reference for the calculation of upper bound
r_k^-	reference for the calculation of lower bound
RSD	Redundant Signed Digit
SFDR	Spurious-Free Dynamic Range (def. page 25)
SINAD	Signal-to-Noise-and-Distortion Ratio (def. page 25)
SNR	Signal-to-Noise Ratio (def. page 25)
Σ	summer
σ	variance
t_k	overall ADC thresholds
$t_{1,k}$	actual thresholds of coarse/first A/D subconverter

$\hat{t}_{1,k}$	estimated thresholds of coarse/first A/D subconverter
$\tilde{t}_{1,k}$	ideal thresholds of coarse/first A/D subconverter
$t_{2,k}$	actual thresholds of fine/second A/D subconverter
$\hat{t}_{2,k}$	estimated thresholds of fine/second A/D subconverter
$\tilde{t}_{2,k}$	ideal thresholds of fine/second A/D subconverter
TH	Track-and-Hold
$\hat{V}_{in}$	reconstructed input voltage
V^{+}_{comp}, V^{-}_{comp}	actual comparator thresholds in conversion stage
$\hat{V}^{+}_{comp}$, $\hat{V}^{-}_{comp}$	estimated comparator thresholds in conversion stage
V_{ref}, V^{+}_{ref}, V^{-}_{ref}	actual reference voltages in conversion stage
$\hat{V}_{ref}$, $\hat{V}^{+}_{ref}$, $\hat{V}^{-}_{ref}$	estimated reference voltages in conversion stage
$V_{res,k}$	residue of stage k
$V_{res,k-j}$	residue of suconverter containing stages k to j

CHAPTER 1 Introduction

This thesis proposes a new method of characterization of multi-stage A/D converters, capable of estimating the errors introduced by each of the internal blocks. It also proposes an adaptive digital correction mechanism derived from the aforementioned method, which was verified to effectively reduce the harmonic distortion of a real converter.

1.1 Motivation

Multi-stage analog-to-digital converters predominate in applications that require high speed and numbers of bits in excess of ten, such as video and wideband radio applications. Flash converters, though faster, are not common beyond eight bits because their complexity increases exponentially, while folding and interpolation architectures, though showing good performance, are rarely found in commercial parts for the time being.

The (numerous) techniques currently used to test and characterize this type of A/D converters are identical to those used for any other type of A/D converters: the beat frequency and envelope tests, the servo-loop code transition measurement, the Fast Fourier Transform (fft) test, the sine-fitting test, the code density test, and many others. A common shortcoming of these methods is that they only provide information about the performance of the ADC at the system level, such as the overall differential nonlinearity (DNL) and integral nonlinearity (INL).

We propose a method geared specifically toward multi-stage architectures that can characterize their internal blocks and estimate the DNL and INL of the A/D and D/A subconverters, and the gain of the interstage amplifiers. The characterization can even go one step further and identify the sources of problems at the component level, for instance by revealing V_{be}, resistor, capacitor, and current mismatches, low gain in amplifiers, or hysteresis. One might ask, who needs this kind of information? Probably not the users of the ADCs, since they only need to know whether the chip meets the requirements imposed by their applications. On the other hand, the design engineers might find this inside information very handy: after all, if they know which blocks degrade the performance of the ADC, they also know what to do in order to fix it. The component-level information could also be used to characterize a process, especially in terms of matching, with a better accuracy (more than ten bits) than it would otherwise be possible.

The method proposed here is to some extent related to the standard code density test, inasmuch as both use histograms. However, they do not use the same kind of histograms: the latter uses unidimensional histograms while the former uses bidimensional histograms. There is also a significant difference of complexity. In one of the variants of the new method, the code density test is but one of the steps of a long procedure involving system solving, polynomial root finding, and least-squares approximations. There are however simpler variants, applicable to particular architectures, some of them simple enough to be turned into on-line digital correction logic. The extra complexity certainly pays off by the wealth of insightful information it offers.

The method can be applied to any multi-stage analog-to-digital converter, from two-stage to pipeline architectures, and was actually tested on two different chips: a two-stage A/D converter, namely the AD9042 from Analog Devices [AD95], [Mur95], and a 14-stage pipeline ADC which we redesigned and fabricated from a previous version [Man96a], [Man96b], designed by Timo Rahkonen, Antti Manyniemi, and Antti Ruha, from the Department of Electrical Engineering of the University of Oulu, Finland.

Since the application of the proposed method requires an in-depth knowledge of the converters, a large part of this thesis is devoted to their analysis. Mathematical models are provided for both two-stage and pipeline ADCs, allowing for many possible nonidealities.

1.2 Thesis Outline

The thesis could have been organized in more than one way. We could have presented first the general theory behind the characterization method and then the applications, but this would have made the presentation rather dry, so that we decided to explain the theory along with the applications.

The thesis comprises five chapters, apart from this introduction.

Chapter 2 deals with the architecture of a typical two-stage ADC, namely the AD9042, focusing on possible sources of nonlinearity and developing a mathematical model to be used in the characterization process.

Chapter 3 introduces the bidimensional histogram, which is the basis of the new method of characterization. After a short overview of current testing and characterization methods, the new method of characterization is presented step by step in its most elaborate form (to date). The advantages and limitations of the method are also discussed at length.

Chapter 4 describes the architecture of the pipeline ADC and provides a pseudo-two-stage model for it, in order to emphasize the similarities with the two-stage architecture presented before.

Chapter 5 introduces a simplified version (same idea, different implementation) of the new characterization method and an interesting spin-off of this method: an adaptive digital correction mechanism which, when tested on the output data stream of the pipeline ADC, improved the Spurious-Free Dynamic Range by 6dB over three quarters of the bandwidth.

Chapter 6 discusses the contributions of this thesis and possible future enhancements.

CHAPTER 2 Two-Stage A/D Converter Architecture

Classic characterization methods can be applied in the same way to any A/D converter, regardless of its architecture. On the contrary, the method that we propose, while offering much more internal information than classic methods, requires an exact knowledge of the converter architecture. Therefore, it seems only natural that we should devote a chapter to the description and mathematical modeling of the ADC to be characterized.

The converter discussed here is the AD9042 from Analog Devices [AD95], [Mur95]. Although its implementation may be different from those of other two-stage converters, the mathematical model can be applied with minor modifications to any of them.

2.1 Introduction

The AD9042 is a high-speed 12-bit A/D converter fabricated in a fast complementary bipolar technology. Its block diagram is presented in Figure 2.1.

The analog signal is applied at the input of the ADC, amplified by A1, sampled and held by TH1, and then quantized by the coarse ADC. The DAC turns the 6-bit digital output of this first conversion (d_1), into an analog signal to be subtracted from the output of TH2, which is simply a delayed version of the output of TH1. In the subsequent clock cycle, the difference is held by TH3, amplified by A2, and quantized by the 7-bit fine ADC.

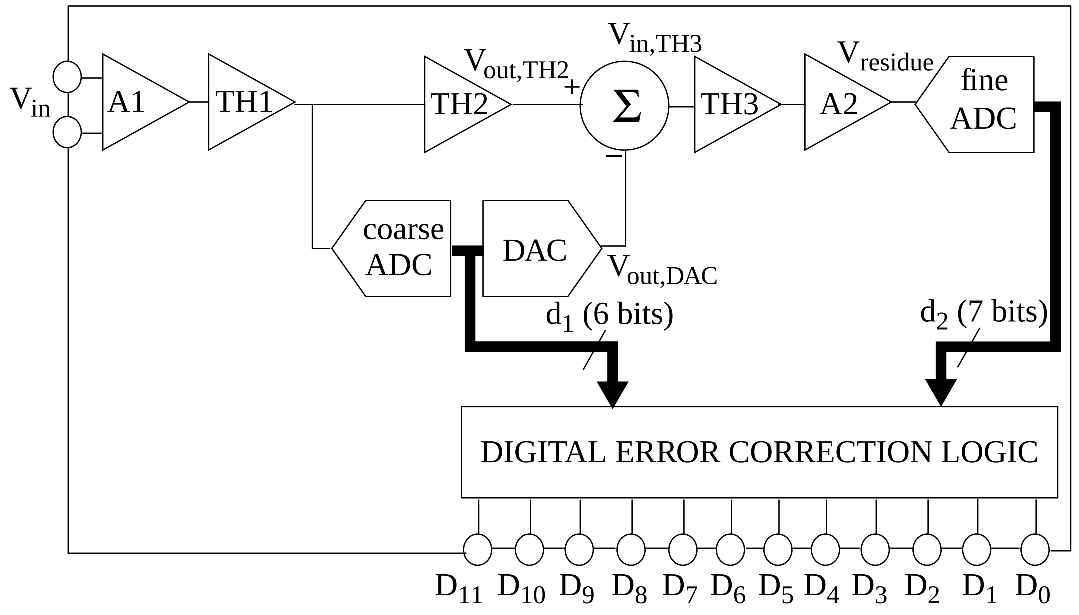


FIGURE 2.1 Two-Stage ADC Architecture (AD9042)

The operation of the two-stage ADC is illustrated in Figure 2.2. Assuming ideal components, the transfer characteristic of the branch containing the coarse ADC and the DAC will be a perfect staircase, and the residue will have a sawtooth shape, with identical "teeth".

The outputs of the two A/D subconverters are combined in the digital error correction block by overlapping the MSB of d_2 (D_{2_6}) and the LSB of d_1 (D_{1_0}), so that the overall digital output is calculated as

$$\begin{array}{ll} d_1(6\text{ bits}) \rightarrow & D_{1_5}D_{1_4}D_{1_3}D_{1_2}D_{1_1}D_{1_0} \qquad\qquad\quad + \\ d_2(7\text{ bits}) \rightarrow & \underline{\qquad\qquad\qquad D_{2_6}D_{2_5}D_{2_4}D_{2_3}D_{2_2}D_{2_1}D_{2_0}} \\ d(12\text{ bits}) \rightarrow & D_{11}D_{10}\,D_9\;D_8\;D_7\;D_6\;D_5\;D_4\;D_3\;D_2\;D_1\;D_0 \end{array} \qquad \text{(EQ 2.1)}$$

or simply

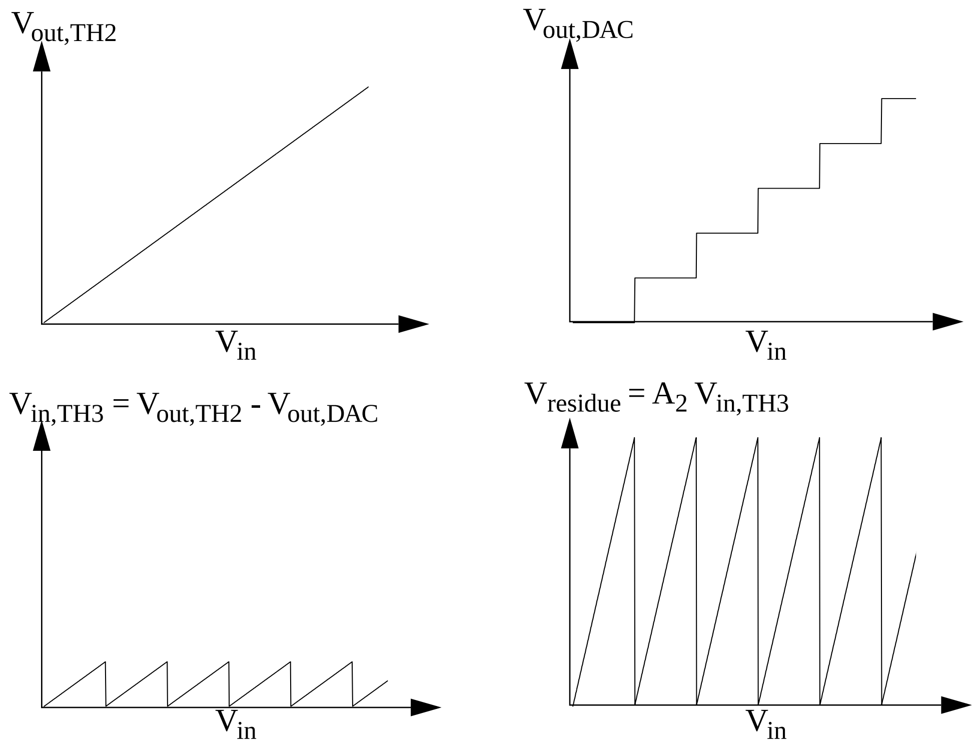


FIGURE 2.2 **Residue as the Difference Between Input and Quantized Input, Amplified by A2**

$$d = 2^6 \cdot d_1 + d_2 \qquad \text{(EQ 2.2)}$$

Without the one-bit overlap the input-output characteristic might have deadbands because of component nonidealities. This technique is called range overlap and is explained in the following.

2.2 Range Overlap and Digital Correction

The input range of an A/D subconverter is defined by the number of bits and the voltage corresponding to one bit (i.e. the ideal value of the difference between two consecutive thresholds), as follows:

$$R_{in} = 2^N LSB \tag{EQ 2.3}$$

Let R_{in2} be the input range of the fine ADC, R_1 the nominal signal range at the input of the coarse ADC, and R_2 the nominal signal range at the input of the fine ADC. From the block diagram in Figure 2.1, it is apparent that

$$R_2 = \frac{A_2}{2^6} \cdot R_1 \tag{EQ 2.4}$$

When designing a two-stage ADC, we could choose the gain A_2[1] so that the signal at the input of the fine subconverter covers its whole input range. In other words, the nominal signal range would be equal to the input range (R_2 equal to R_{in2}). Then,

$$A_2 = 2^6 \cdot \frac{R_{in2}}{R_1} \tag{EQ 2.5}$$

In reality, the coarse ADC will suffer from errors, and the residue will have an irregular form, as depicted in Figure 2.3 (right). Other components, such as the DAC, the summer, the track-and-hold circuits, and the amplifier A2, will also suffer from nonidealities, but the errors in the coarse ADC are by far the largest input-referred errors.

As a result of these errors, if A_2 from (EQ 2.5) is used, parts of the residue will exceed the input range of the fine ADC and will be clipped: any voltage above the input range will be quantized as 2^7-1, and any voltage below the input range will be quantized as 0. Consequently, the overall input-output characteristic will have deadbands. The solution used in the AD9042 to avoid clipping and deadbands is to make the gain A_2 half of what it could be in the ideal case:

1. the notation A2 is used for the amplifier as such and A_2 for its gain

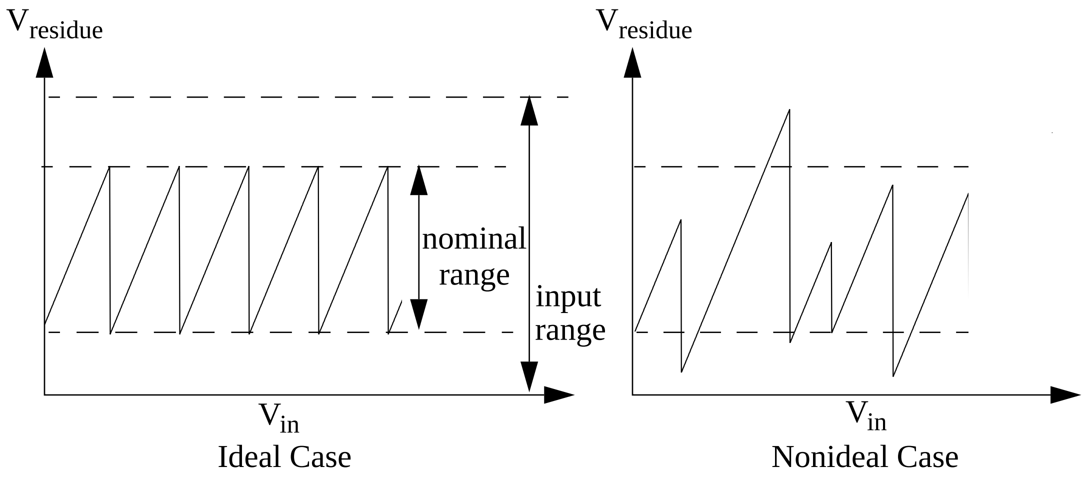


FIGURE 2.3 **Residue for Ideal and Nonideal Case**

$$A_2 = \frac{1}{2} \cdot 2^6 \cdot \frac{R_{in2}}{R_1} \qquad \text{(EQ 2.6)}$$

Thus, for each value of d_1, only about half of the range of the fine ADC is actually used. This redundancy is normally meant to enable the overall converter to tolerate coarse ADC errors (see Section 2.4 for a mathematical analysis and proof). In this thesis we will make further use of this redundancy to characterize each of the components of the ADC (Chapter 3).

Note that the reduction of $\frac{1}{2}$ in the analog domain must be accompanied by a similar reduction in the digital domain, otherwise the overall input-output characteristic would not be linear. This means that the coefficient of d_1 in (EQ 2.2) is 2^6 and not 2^7, although the fine subconverter has 7 bits.

2.3 Sources of Nonlinearity in a Two-Stage ADC

The purpose of this thesis is to provide a method capable of characterizing and if possible fixing the nonidealities that affect (negatively) the performance of the ADC. These nonidealities are briefly surveyed here and will be analyzed in more detail in subsequent chapters.

A complete block diagram of the AD9042 has been presented in Figure 2.1. We will verify (see Section 2.4) that the blocks relevant for linearity are the D/A subconverter, the amplifier A2, and the fine A/D subconverter.

We will also prove (also in Section 2.4) that for this particular architecture the errors in the coarse A/D subconverter do not affect the overall ADC linearity. As to the other blocks, in a well-designed ADC the track-and-hold circuits (TH1, TH2, TH3), the amplifier (A1) and the summer (Σ) can be assumed not to cause any significant increase in nonlinearity. Note that in terms of nonlinearity, a gain error in A1 is irrelevant, while a gain error in A2 can be a serious problem.

Figure 2.4 presents a simplified block diagram, including all the blocks capable of causing significant nonlinearity. Some of the other blocks are not shown. All the calculations will be based on this diagram and the notation will be explained at the beginning of the next section.

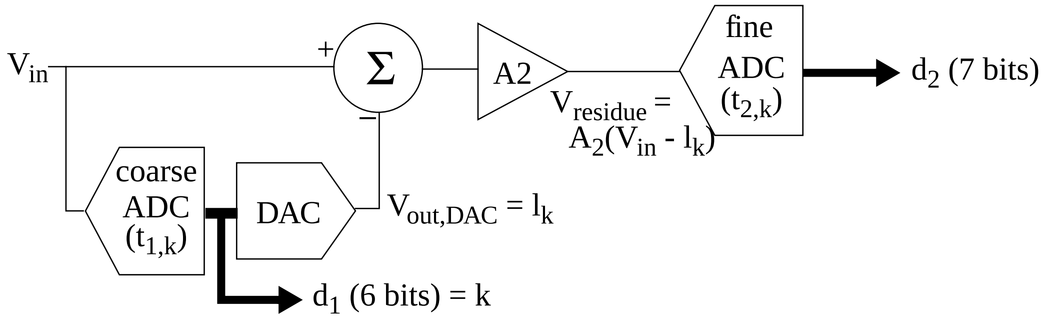


FIGURE 2.4 Simplified Block Diagram of the AD9042

2.4 Residue Modeling

Most of the calculations required by the characterization process revolve around the residue. Therefore it is necessary to provide a formal mathematical model for it.

Let V_{in} be the analog input signal, $t_{1,k}$, k=0...64, the thresholds of the coarse A/D subconverter, $t_{2,k}$, k=1...127 the thresholds of the fine A/D subconverter, l_k, k=0...63, the output levels of the DAC, and LSB_1 and LSB_2 the voltages corresponding to one bit of the coarse and the fine subconverter respectively. The actual thresholds are from $t_{1,1}$ to $t_{1,63}$ and from $t_{2,1}$ to $t_{2,127}$. We also introduced $t_{1,0} = 0$, $t_{1,64} = 64\ LSB_1$, $t_{2,0} = 0$, and $t_{2,128} = 128\ LSB_2$ to simplify the equations and we assumed the input to be between $t_{1,0}$ and $t_{1,64}$, otherwise the signal is clipped.

This choice of values for $t_{1,0}$, $t_{1,64}$, $t_{2,0}$, and $t_{2,128}$ implies that we are ignoring the overall offsets and gains of the subconverters. In practice these parameters are of less interest than nonlinearity, and therefore they are not covered in this thesis. The characterization method that we propose (Chapter 3) could be modified to take them into account, but some extremely accurate signal generators and measurement equipment would be required.

Ideal values will be marked with a tilde sign (e.g. $\tilde{t}_{1,k}$) to distinguish them from real values (e.g. $t_{1,k}$), and the difference between them will be marked with Δ (e.g. $\Delta t_{1,k} = t_{1,k} - \tilde{t}_{1,k}$). Estimated values will have a circumflex accent (e.g. $\hat{t}_{1,k}$).

The output of the coarse A/D subconverter may be defined as

$$d_1(V_{in}) = k\ , \qquad t_{1,k} < V_{in} \le t_{1,k+1} \qquad k = 0\ldots63 \tag{EQ 2.7}$$

This digital signal is fed to the D/A subconverter which turns it into an analog signal:

$$V_{out,DAC}(d_k) = l_k\ , \qquad k = 0\ldots63 \tag{EQ 2.8}$$

Replacing (EQ 2.8) in (EQ 2.7) we obtain:

$$V_{out,DAC}(V_{in}) = l_k , \qquad t_{1,k} < V_{in} \le t_{1,k+1} \qquad k = 0 \dots 63 \tag{EQ 2.9}$$

This signal is then subtracted from V_{in} and amplified by A_2 so that the residue is

$$V_{residue}(V_{in}) = A_2(V_{in} - l_k) , \qquad t_{1,k} < V_{in} \le t_{1,k+1} \qquad k = 0 \dots 63 \tag{EQ 2.10}$$

This equation enables us to visualize the effect on residue of coarse ADC errors ($\Delta t_{1,k}$), DAC errors (Δl_k), and gain errors, as shown in Figure 2.5. Note that only four sections of the residue are represented (out of 64).

It can be seen that each of these errors changes the residue in a distinct way: each coarse ADC error moves horizontally the transition between sections of residue corresponding to

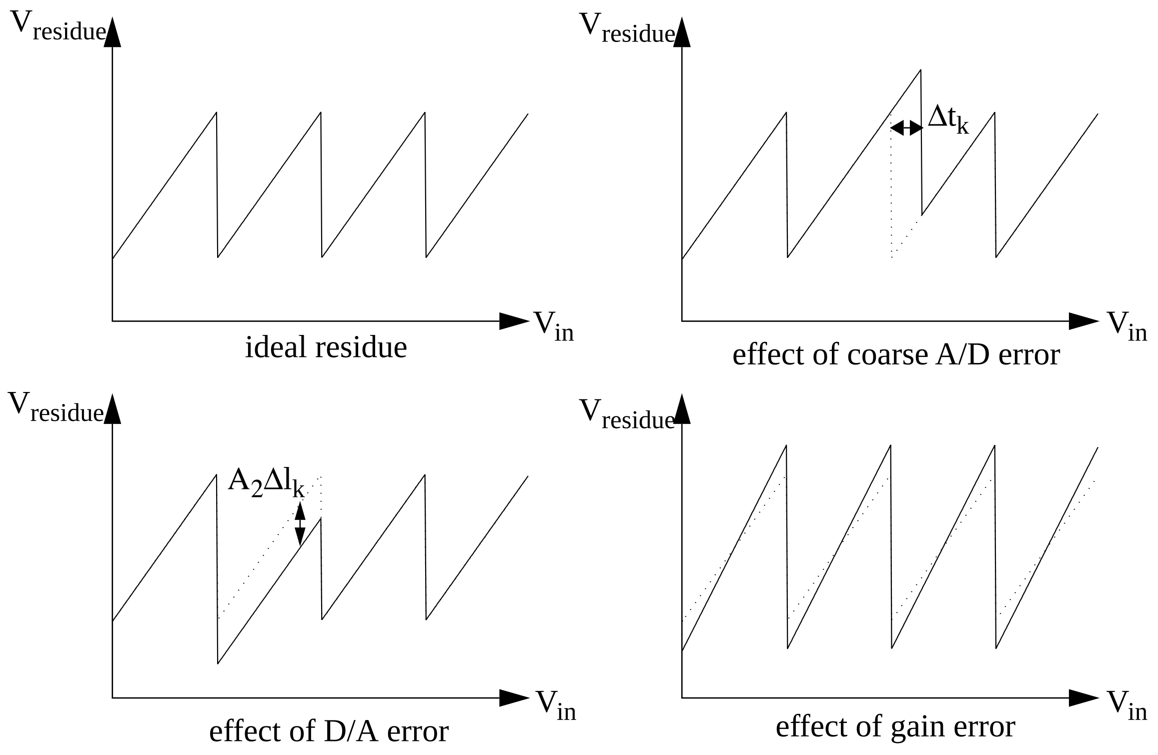


FIGURE 2.5 **Effect of Nonidealities on Residue**

consecutive d_1 codes, each DAC error moves vertically the section of the corresponding d_1 code, and a gain error will either lengthen or shorten all the sections of the residue.

Assuming an ideal fine A/D subconverter with infinite precision (i.e. assuming $V_{residue} = A_2 \cdot (V_{in} - l_k) = d_2 \cdot LSB_2$), for a given digital output (d_1 and d_2) the reconstructed input is equal to the sum of the contributions of the first stage ($d_1 \cdot LSB_1$) and second stage ($\frac{d_2 \cdot LSB_2}{\tilde{A}_2}$):

$$\hat{V}_{in} = d_1 \cdot LSB_1 + \frac{d_2 \cdot LSB_2}{\tilde{A}_2} \qquad \text{(EQ 2.11)}$$

$$= \tilde{l}_k + \frac{A_2 \cdot (V_{in} - l_k)}{\tilde{A}_2}$$

$$= \frac{A_2}{\tilde{A}_2} \cdot V_{in} + \left(\tilde{l}_k - l_k \cdot \frac{A_2}{\tilde{A}_2}\right) \qquad \text{(EQ 2.12)}$$

From (EQ 2.12) we can draw a number of conclusions:

- threshold errors in the coarse ADC ($\Delta t_{1,k} = t_{1,k} - \tilde{t}_{1,k}$) do not appear in the formula and therefore have no effect whatsoever on the linearity (assuming they do not push the residue out of the range of the fine ADC)
- DAC errors ($\Delta l_k = l_k - \tilde{l}_k$) cause nonlinearity in the overall ADC input-output characteristic
- a gain error in A2 causes both nonlinearity and gain error in the overall ADC input-output characteristic. However, in some cases, a small gain error (first term in (EQ 2.12)) is acceptable but discontinuities are not (second term). In order to eliminate the second term, we only need to know the ($A_2 l_k$) products; we do not have to estimate A_2 and l_k, k=0...63, separately.

How these errors affect the overall ADC characteristic is shown in Figure 2.6 (again, only part of the characteristic is shown).

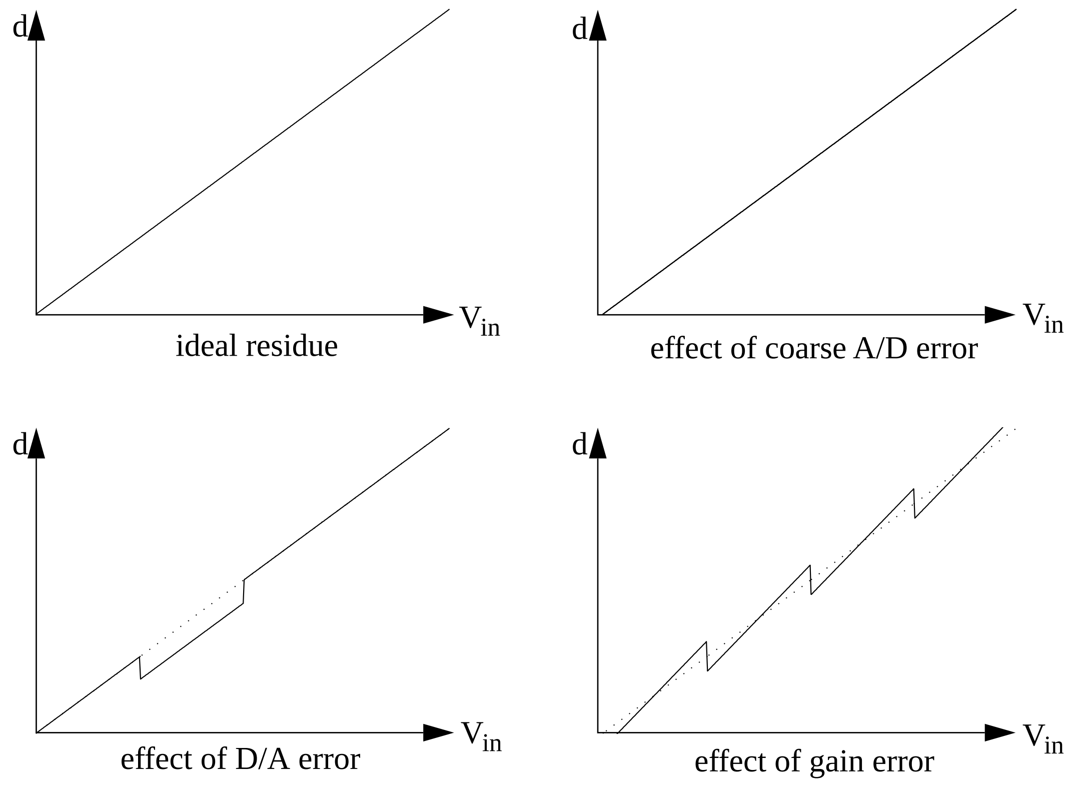


FIGURE 2.6 **Effect of Nonidealities on ADC Characteristic**

Errors in the fine ADC must also be taken into account as they affect the overall linearity by making $V_{residue}$ different from $d_2\ LSB_2$.

2.5 Residue Bounds

Of particular importance to the characterization process are the lower and upper bounds of each of the 64 sections of the residue (Figure 2.7), where a section is considered to be the part of the residue corresponding to a single d_1 code. We will denote these bounds by m_k and M_k, respectively, with k=0...63.

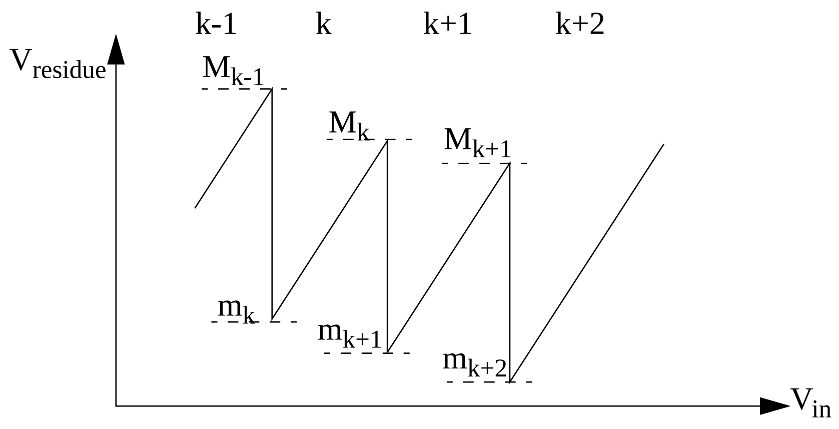


FIGURE 2.7 Lower and Upper Bounds of the Residue

We can calculate m_k and M_k from (EQ 2.10) :

$$\begin{aligned} M_k &= A_2 \cdot (t_{1,\mathrm{k+1}} - l_k) \\ m_k &= A_2 \cdot (t_{1,\mathrm{k}} - l_k) \end{aligned} \tag{EQ 2.13}$$

Let us assume for the moment that all M_k and m_k are known (we will see later how they can be estimated). Then it is possible to form a system of equations using (EQ 2.13) for k = 0...63. This system has 128 unknown variables (A_2, $t_{1,k}$ (k=1...63), and l_k (k=0...63)) and 128 equations, and is in fact quite easy to solve. First we arrive at

$$M_k - m_k = A_2 \cdot (t_{1,\mathrm{k+1}} - t_{1,\mathrm{k}}) \text{ ,} \qquad k = 0 \ldots 63 \tag{EQ 2.14}$$

which by summing for all k yields:

$$\sum_{k=0}^{63} (M_k - m_k) = A_2 \cdot (t_{1,64} - t_{1,0}) \tag{EQ 2.15}$$

Since $t_{1,64}$ and $t_{1,0}$ are known, we can calculate A_2:

$$A_2 = \frac{\sum_{k=0}^{63} (M_k - m_k)}{64 \cdot LSB_1} \tag{EQ 2.16}$$

Then the coarse A/D thresholds and the D/A output levels can be determined progressively from (EQ 2.13) :

$$t_{1,\mathrm{k+1}} = \frac{M_k - m_k}{A_2} + t_{1,\mathrm{k}} \quad , \qquad k = 0 \ldots 62 \tag{EQ 2.17}$$

$$l_{k+1} = \frac{M_k - m_{\mathrm{k+1}}}{A_2} + l_k \quad , \qquad k = 0 \ldots 62 \tag{EQ 2.18}$$

where l_0=0 and $t_{1,0}$=0.

The following chapters will show that the bounds (M_k and m_k) can be estimated, and (EQ 2.16) , (EQ 2.17) , and (EQ 2.18) will be used to calculate the desired parameters.

Alternatively, if all we need are the ($A_2 l_k$) products, we can use

$$(A_2 l_{\mathrm{k+1}}) = (A_2 l_k) + M_k - m_{k+1} \, , \qquad k = 0 \ldots 62 \tag{EQ 2.19}$$

Now if we go back to (EQ 2.2) we can see that the first term is

$$2^6 \cdot d_1 = \tilde{A}_2 \cdot d_1 = \tilde{A}_2 \cdot \frac{\tilde{l}_k}{LSB_1} \tag{EQ 2.20}$$

A better formula for the digital output (EQ 2.2) can be found by replacing this term with the actual value, $\frac{A_2 l_k}{LSB_1}$:

$$d = \frac{(A_2 l_k)}{LSB_1} + d_2 \qquad \text{(EQ 2.21)}$$

since the reconstructed input $\hat{V}_{in}$ in this case would be

$$\begin{aligned}\hat{V}_{in} &= \frac{(A_2 l_k)}{\tilde{A}_2} + \frac{d_2 \cdot LSB_2}{\tilde{A}_2} \qquad \text{(EQ 2.22)}\\ &= \frac{(A_2 l_k)}{\tilde{A}_2} + \frac{A_2 \cdot (V_{in} - l_k)}{\tilde{A}_2}\end{aligned}$$

$$= \frac{A_2}{\tilde{A}_2} \cdot V_{in} \qquad \text{(EQ 2.23)}$$

Please note the differences between (EQ 2.2) and (EQ 2.21) , and between (EQ 2.12) and (EQ 2.23) . Although in (EQ 2.23) we still have a gain error, the nonlinear term from (EQ 2.12) has disappeared.

It is apparent that although (EQ 2.16) , (EQ 2.17) , and (EQ 2.18) are more suitable for characterization, (EQ 2.19) may be a better option for correction, since it is easier to implement (it does not require division).

2.6 A/D Subconverter Architecture

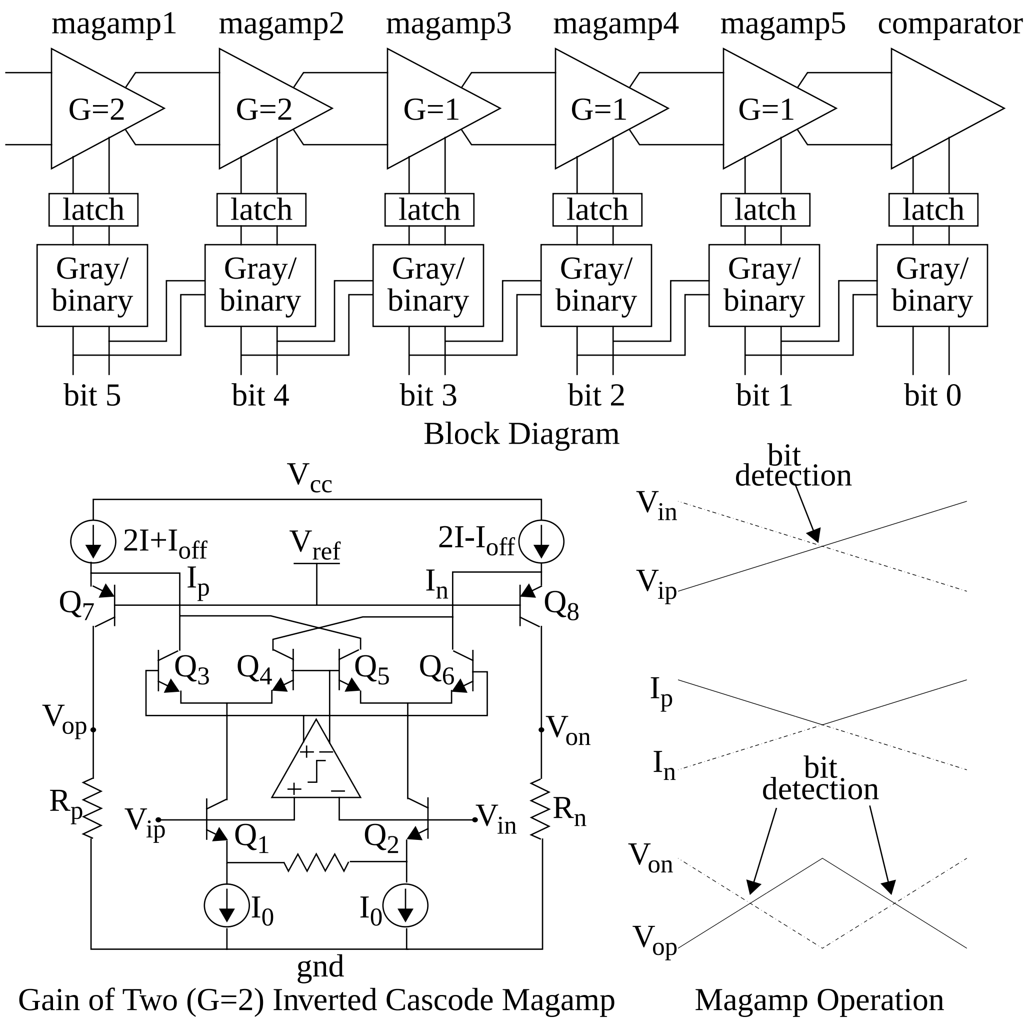
In Chapter 3 we will estimate the errors in the coarse and fine A/D subconverters. Some knowledge of the architecture of these components is a prerequisite to interpreting the estimated errors. Therefore, a brief description of the structure and operation of the subconverters of the AD9042 is included in this chapter. Since their architecture is quite unusual, the typical flash (subconverter) architecture is also presented.

The two coarse and fine subconverters of the AD9042 are similar, except for the number of bits, so that only the coarse ADC is explained here (Figure 2.8, top). This is based on a cascaded architecture with five magnitude amplifiers (magamp) and a comparator. The design of the first two magamps (those with gain of two, G=2) is shown in Figure 2.8, middle-left, and their operation is illustrated in Figure 2.8, middle-right. The other magamps (those with gain of one, G=1) have a different design, but their behavior is almost the same, and at any rate their effects, referred to the input of the subconverter, are smaller, so that they are not analyzed here.

The magamps are fully differential which means that as we (dc) sweep the positive input voltage V_{ip}, the negative input voltage V_{in} will be swept in the opposite direction (Figure 2.8, middle-right; note that the horizontal axis is always V_{ip}). The current through Q_1 will increase and the one through Q_2 will decrease. At first V_{ip} is less than V_{in} so that the comparator keeps Q_4 and Q_5 on and Q_3 and Q_6 off, thus steering the collector currents of Q_1 and Q_2 through Q_4 and Q_5, respectively, and subtracting them from the currents of the sources, $2I\text{-}I_{off}$ and $2I\text{+}I_{off}$, respectively. Since I_n increases, the current through R_n and the voltage across it, V_{on}, decrease. Similarly, I_p decreases, therefore the current through R_p and V_{op} increase.

When V_{ip} becomes greater than V_{in}, the comparator turns Q_4 and Q_5 off and Q_3 and Q_6 on, forcing the collector currents of Q_1 and Q_2 to go through Q_3 and Q_6, respectively. These currents are subtracted from those of the sources. Since I_p now increases and I_n decreases, the currents through and the voltages across the resistors R_p and R_n will reverse their direction of change. The characteristic is shown in Figure 2.8, middle-right.

It is easy to see that with each magamp, the number of bit detection points doubles (Figure 2.8, bottom). Due to the shape of the transfer characteristic (V_{op} vs. V_{ip}), with a positive slope at first and a negative slope afterward, the outputs are inherently in Gray code and



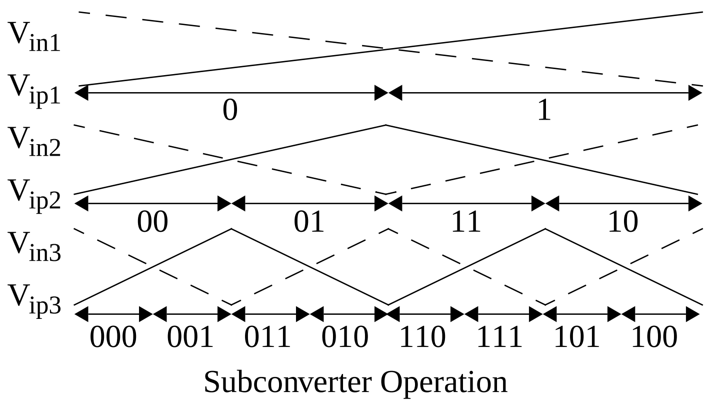


FIGURE 2.8 Coarse ADC Architecture

must be subsequently converted to binary. The latching is done after all the magamps have settled.

In practice many things can go wrong with this circuit. The worst offenders are, in all likelihood, V_{be}, resistor, and current mismatches. The effect of each of these mismatches occurring in the MSB stage (magamp 1) on the thresholds of the following stages can be seen in Figure 2.9. Both the threshold errors (Δt_k) and the difference between consecutive threshold errors (Δt_k-Δt_{k-1}) have been plotted, and we will refer later to these plots to understand the estimated INL and DNL (defined on page 25 and page 24, respectively) of the subconverters. Since the $\Delta t_k - \Delta t_{k-1}$ plots have been derived here from the Δt_k plots, the reader is urged to focus on the latter for the following explanation of the shape of these curves.

In the case of V_{be} mismatch either in Q_1-Q_2 or in the input transistors of the comparator, the currents through Q_1 and Q_2 are switched (Figure 2.9-1.b) not at the point where they are equal but at a different point, which causes discontinuities in the currents I_p and I_n. These discontinuities also appear in V_{op} and V_{on} in the central region (Figure 2.9-1.c) but leave the rest of the characteristic unaffected, and as a result, the thresholds of the following stages are not changed ($\Delta t_k = 0$). The only threshold changed is that of magamp 1 (the one in the middle in Figure 2.9-1.e).

A current mismatch in $2I + I_{off}$ for instance will not affect I_p (Figure 2.9-2.b), but will shift V_{op} (Figure 2.9-2.c) up or down. Let us assume that the curve is shifted up. Then the thresholds (of the following stages) on the left side will decrease (Figure 2.9-2.e, watch for instance the left bit-detection point which is one of the thresholds due to magamp 2, considered to be ideal), while the thresholds on the right side will increase (watch the right bit-detection point, which is the other threshold due to magamp 2).

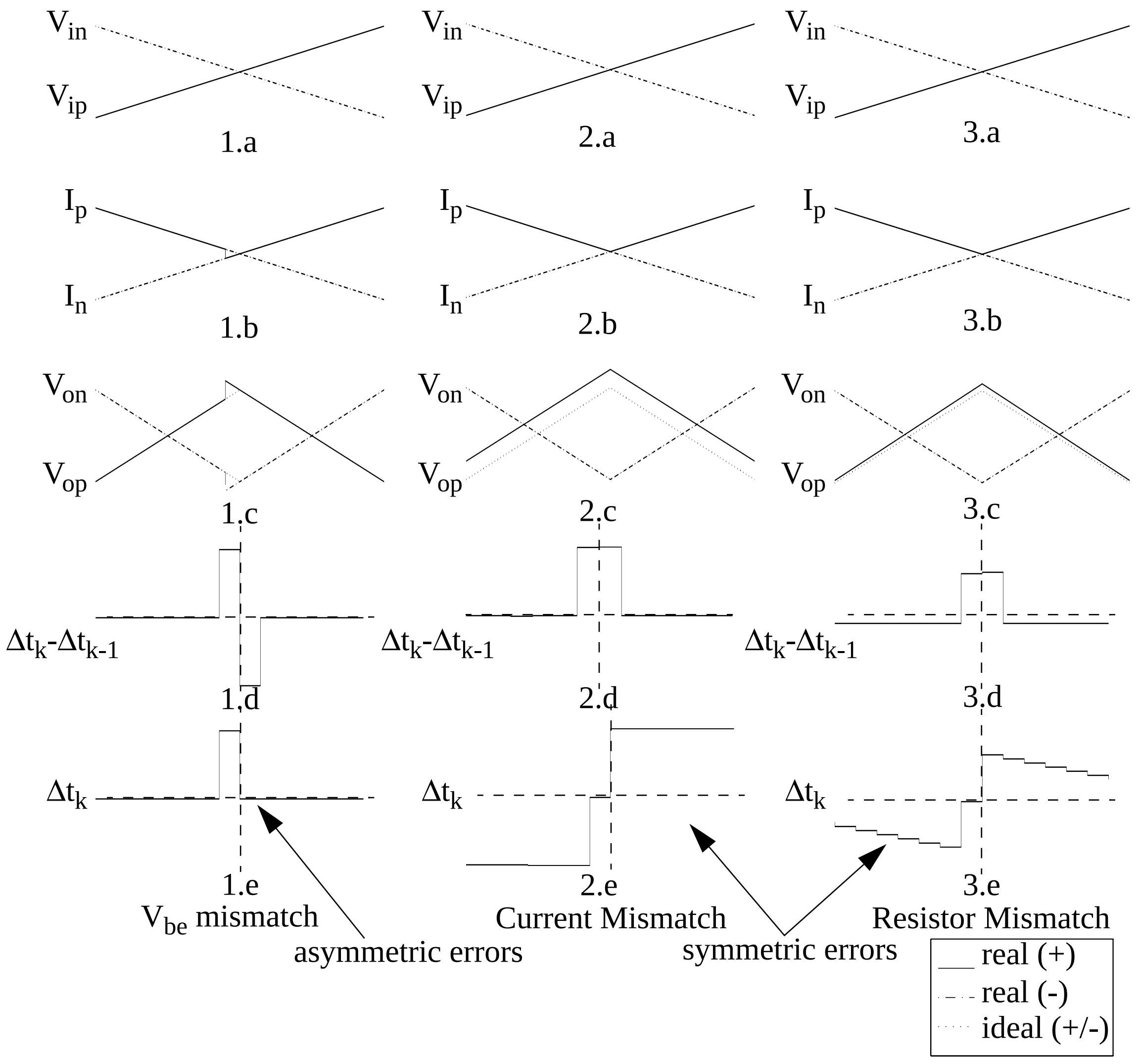


FIGURE 2.9 Effect of Mismatches

A resistor mismatch in R_p for example will only affect V_{op} (Figure 2.9-3.c), and will cause the thresholds on the left side to decrease and those on the right side to increase. As opposed to the previous case, the change is not the same for all thresholds, because the change in V_{op} is not constant.

A common error in data converters is hysteresis. Although the AD9042 suffers from this problem, we believe it does not originate from the A/D subconverters, so it is not discussed here (see Section 3.9).

Other types of errors are also possible, but, in our opinion, all of them fall into two categories:

- errors that cause symmetrical changes in all the thresholds of the following stages except those next to the bit transition, e.g. resistor and current mismatches, and
- errors that cause only an asymmetric change in the thresholds next to the bit transition, e.g. V_{be} mismatch.

Later we will derive plots of the threshold errors in both subconverters, and we will see that both categories of errors are readily distinguishable.

The cascaded architecture presented so far is far from being a common choice. Most two-stage ADCs use flash subconverters, like the circuit presented in Figure 2.10. For n bits, there are $2^n - 1$ comparators which compare the input signal to a number of $2^n - 1$ equally spaced reference voltages, in this case generated from a single reference by a voltage divider with 2^n equal-size resistors. Suppose that the input is between the references V_j and V_{j+1}. The outputs of the comparators from A_1 to A_j will be 1, and the outputs of the comparators from A_{j+1} to A_{2^n-1} will be 0. It is apparent that these outputs constitute a thermometer code, which can be converted to a binary representation by the decoder.

2.7 Summary

In this chapter we have seen the basic principles of the two-stage ADC architecture, including range overlap and digital correction. A model has been developed for this

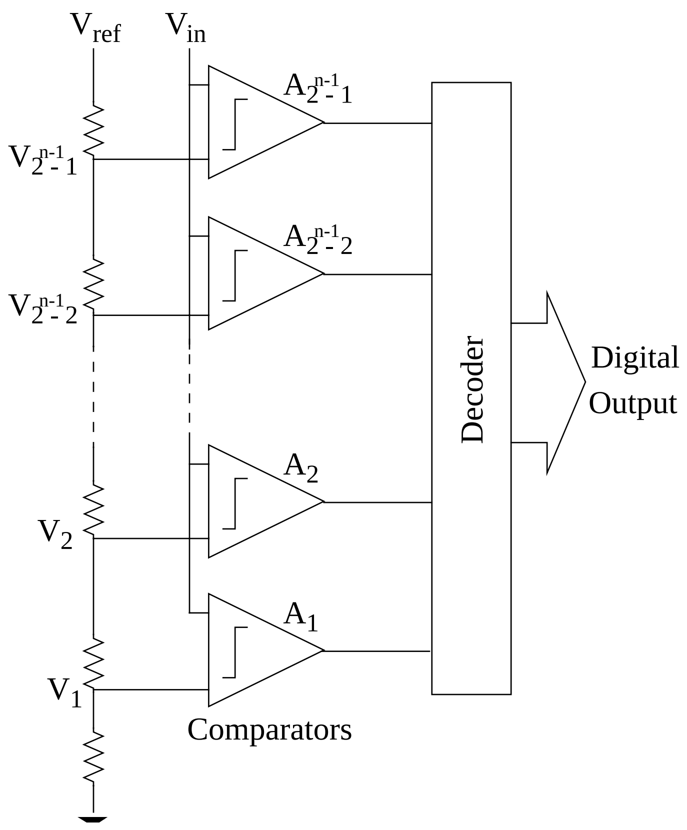


FIGURE 2.10 **Typical Flash Architecture**

architecture, and the components relevant for linearity have been identified. Particular attention has been given to the residue and equations relating internal parameters to residue bounds have been found. The architecture of the A/D subconverters has also been explained briefly.

CHAPTER 3 Characterization of Two-Stage A/D Converters

Many testing and characterization methods are available for A/D converters. This chapter, after overviewing the existing methods, all of which are based on a black-box approach, presents a new method which actually diagnoses the chip by offering more information about what is happening inside the chip and how different blocks degrade the overall performance. The application of this method is described in detail, as are its advantages and limitations.

3.1 Definition of ADC Specifications

A number of performance metrics are used for the characterization of nonlinearity in multi-stage A/D converters. The ones that are used throughout this thesis are defined in the following [Raz95].

Differential Nonlinearity (DNL) The deviation of the actual voltage difference between two consecutive thresholds from its ideal value, *LSB*:

$$DNL_k = (t_{k+1} - t_k) - LSB \qquad \text{(EQ 3.1)}$$

where t_k is the threshold that separates code k-1 from code k. Sometimes only the maximum value is indicated:

$$DNL = max(|DNL_k|) \tag{EQ 3.2}$$

Integral Nonlinearity (INL) The deviation of the actual value of a threshold from its ideal value:

$$\begin{aligned} INL_k &= t_{k+1} - k \cdot LSB \\ INL &= max(|INL_k|) \end{aligned} \tag{EQ 3.3}$$

Signal-to-Noise Ratio (SNR) The ratio of the signal power to the noise power (excluding harmonics and dc) at the output of the ADC, usually for an input 1dB below full scale, expressed in dB:

$$SNR = 10 \cdot \log_{10}\left(\frac{P_s}{P_n}\right) \tag{EQ 3.4}$$

Signal-to-Noise-and-Distortion Ratio (SINAD) The ratio of the signal power to the power of all other spectral components (including harmonics but excluding dc)

Spurious-Free Dynamic Range (SFDR) The ratio of the power of the signal to the power of the peak spurious spectral component, in the worst case over the frequency range and the best case over the level range.

Effective Number of Bits (ENOB) Defined by the following equation:

$$ENOB = \frac{SINAD - 1.76}{6.02} \tag{EQ 3.5}$$

3.2 Prior Art of ADC Characterization

Almost any book on data converters has a chapter devoted to testing and characterization methods. These methods range from quick, qualitative tests (e.g. beat frequency and envelope tests) to thorough tests that require extensive mathematical processing on a computer (e.g. sine-fitting and fft tests). A few of the more precise tests are briefly presented here.

Servo-Loop Code Transition Measurement. A negative feedback loop like the one in Figure 3.1 allows us to measure the threshold between any two codes [Bla94], [Cap95], [Dem91], [Raz95], [She86]. Suppose that the output of the integrator is zero and we apply a code *d* at the input B of the digital comparator. Then, A<B, so that $+V_{ref}$ will be coupled to the integrator and its output will increase until A≥B, when $-V_{ref}$ will be coupled to the integrator and its output will decrease. The integrator output will then oscillate around the threshold which separates codes *d*-1 and *d*, and this threshold can be measured with a digital voltmeter (DVM). Once all the thresholds are known, DNL and INL plots can be derived. Since the noise is integrated out and the digital comparator introduces an infinite gain, the accuracy is very good even with a low-quality integrator. Furthermore, the method can be easily automated. On the negative side, the method is rather slow, and its performance is limited by the correctness of the model used. For instance it does not take into account hysteresis.

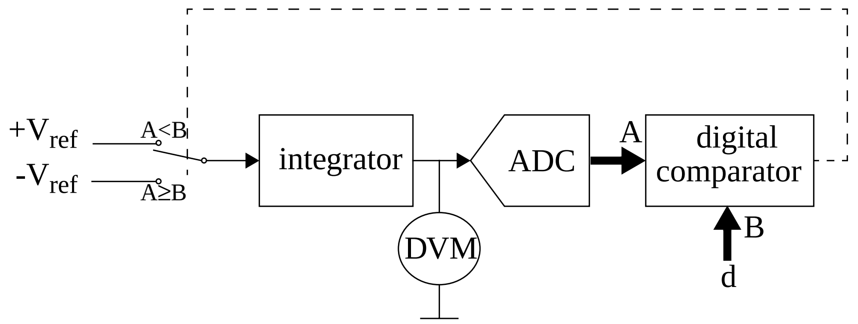


FIGURE 3.1 **Servo Loop**

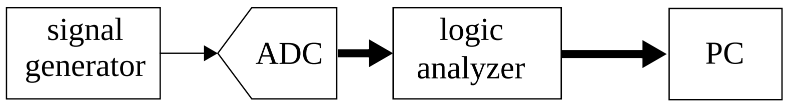


FIGURE 3.2 **Set-Up for the FFT, Sine-Fitting, and Code Density Tests**

A number of other tests (FFT, sinewave fitting, code density) can be performed using the circuit shown in Figure 3.2, which contains a signal generator, a logic analyzer, and a computer. The digital output of the ADC is sampled by the logic analyzer and transferred to a computer.

FFT Test. The discrete Fourier transform of the digital output signal for a sinusoidal input (see Figure 3.3) provides information on noise and harmonic distortion [Dem91], [Raz95]. The power of the signal, harmonics, and noise, SNR, SINAD, and SFDR can be determined from the spectrum. The method is very reliable and efficient, since it requires relatively few samples compared to the code density test (which is explained later in this chapter). The signal used for testing must be a very pure sinewave, and some LC filtering may be required to meet this condition.

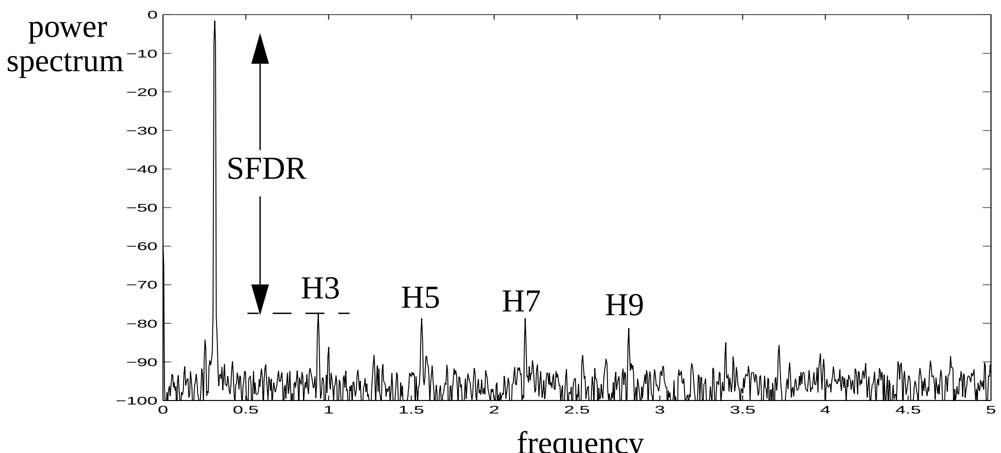


FIGURE 3.3 **Power Spectrum of ADC Output for a Sinusoidal Input**

Sinewave Fitting. Another approach is to collect the samples of a sinewave, estimate its parameters (offset, amplitude, frequency, phase) using mathematical software, and calculate the noise and harmonic distortion by subtracting the estimated sinewave from the actual sinewave [Dem91], [Raz95].

Code Density Test. Suppose we apply a triangular waveform at the input of the ADC under test and collect a large number of samples at the output. Assuming that the signal frequency is not commensurate to the clock frequency (i.e. not related to the clock frequency by a ratio of integers), the value of the signal can be considered to be a random variable with uniform pdf. The probability of occurrence of a code will then be indicative of its DNL. If the two thresholds that define the code are closer to each other than they should be (DNL<0), the probability will be lower than average, since a smaller interval is less likely to be hit. Conversely, if the thresholds are farther from each other (DNL>0), the probability will be higher.

By plotting the number of occurrences of each code (this number is referred to as code density) a so-called histogram is obtained (Figure 3.4, right). The number of samples should be large so as to make the experimental code density plot an accurate replica of the

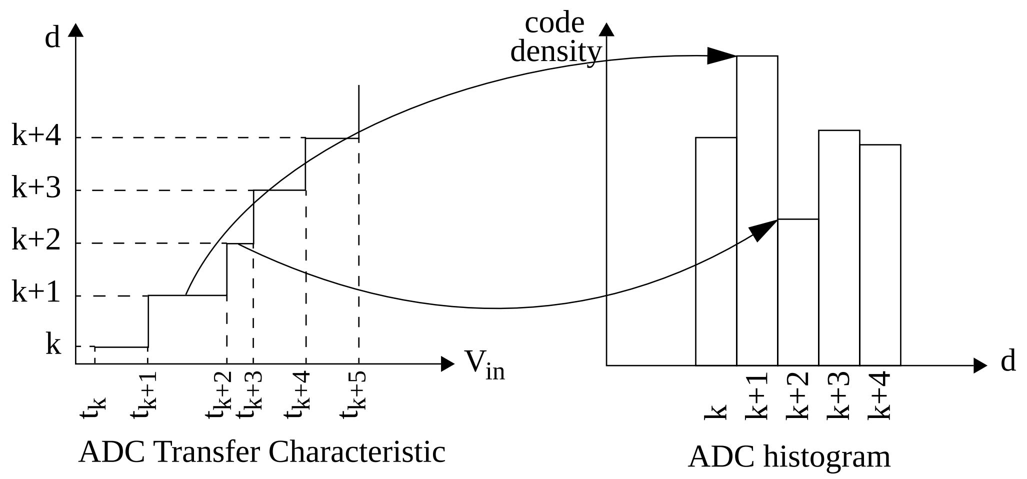


FIGURE 3.4 **Code Density Test**

theoretical code probability plot. DNL and INL can be calculated from these histograms using a procedure which will be described in Section 3.5 [Cap95], [Chi96], [Dem91], [Mor94], [Raz95], [Wag91].

In practice it is difficult to ensure that the triangular signal has the linearity required, so sinewaves are used instead. Although they do not have a constant pdf, they are well-defined (ultralinear signal generators are employed), and therefore their pdf can be calculated with high accuracy. The code density is then normalized with respect to the pdf, and the procedure is identical to the one used for triangular waveforms.

What all of the above tests have in common is that they only characterize the ADC at the system level, and therefore offer little or no information about the internal blocks of the converter. The explanation is that they do not use the knowledge about the architecture of the converter. In the following chapters we will show how this knowledge can be used to devise a new method, that can characterize each of the blocks relevant for the overall linearity.

3.3 Bidimensional Code Density Tables

In Section 2.1 we showed that the digital output d of a two-stage A/D converter results from combining the outputs of two A/D subconverters, d_1 and d_2 (refer to Figure 2.4). The whole characterization process proposed here is based on the idea that we should build code density tables not as a function of the overall output d, but as a function of d_1 and d_2.

A problem with this approach that comes to mind immediately is how to actually get d_1 and d_2. The AD9042, like all commercial parts, only provides d at the output and it is not possible to calculate d_1 and d_2 from d, since some information is lost in the digital error correction process. Analog Devices provided us with two modified parts for testing, in which an internal signal, the MSB of d_2, could be accessed via two of the normally unused pins. Once this signal is available, d_2 can recovered from its MSB and the last six bits of d, and d_1 from the first six bits of d less the MSB of d_2.

Unfortunately, one of the parts suffered from missing codes, so that it could not be used for testing. Therefore, all the plots and computations in this chapter refer to a single AD9042 part, and we will not make any attempt to generalize.

The new code density table will be a 2^6 by 2^7 matrix instead of a vector with 2^{12} components as in a classic code density test. An example is shown in Figure 3.5, using a plane representation with the degree of darkness proportional to the frequency of (d_2, d_1) pairs. Dark areas indicate pairs that occur frequently, while light areas indicate pairs that occur rarely. We will call this representation a bidimensional histogram.

As we will show in Section 3.8, errors in the amplifier A2, the coarse A/D subconverter, and the D/A subconverter can be estimated from these bidimensional histograms. Based on these estimations we can reconstruct to a good approximation the residue, as in Figure 3.6. A quick comparison between Figure 3.5 and Figure 3.6 reveals the connection between the bidimensional histogram and the residue: the residue dictates the shape of the histogram and, as a result, the histogram contains information about the residue. It is this information that will enable us to characterize the components of the ADC.

We are also interested in individual columns of the matrix, like the one plotted in Figure 3.7 using a stairstep representation. We mentioned before that the section of the residue corresponding to a given d_1 code only occupies about half of the input range of the fine subconverter. The same happens to the columns of the bidimensional histogram: for a given value of d_1, only about half of the possible values of d_2 occur. A more in-depth discussion of the shape of these histogram columns is provided in Section 3.4.

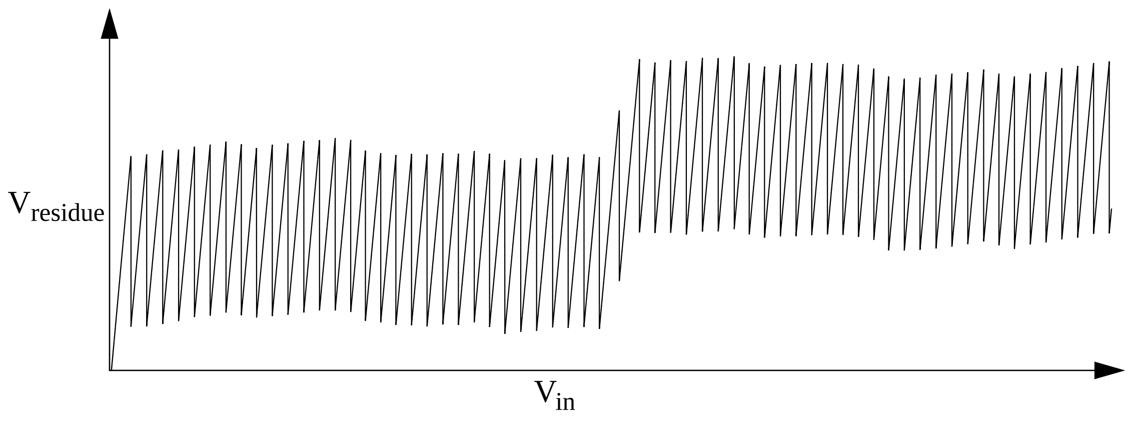


FIGURE 3.6 **Residue Plot**

The last issue that will be approached here is the relationship between unidimensional and bidimensional histograms, illustrated in Figure 3.8. The columns of the bidimensional histogram, offset in steps of 64 LSB_2 and added together result in the unidimensional histogram. On the other hand, it is not possible to reconstruct the bidimensional histogram from the unidimensional histogram.

The relationship can be expressed as:

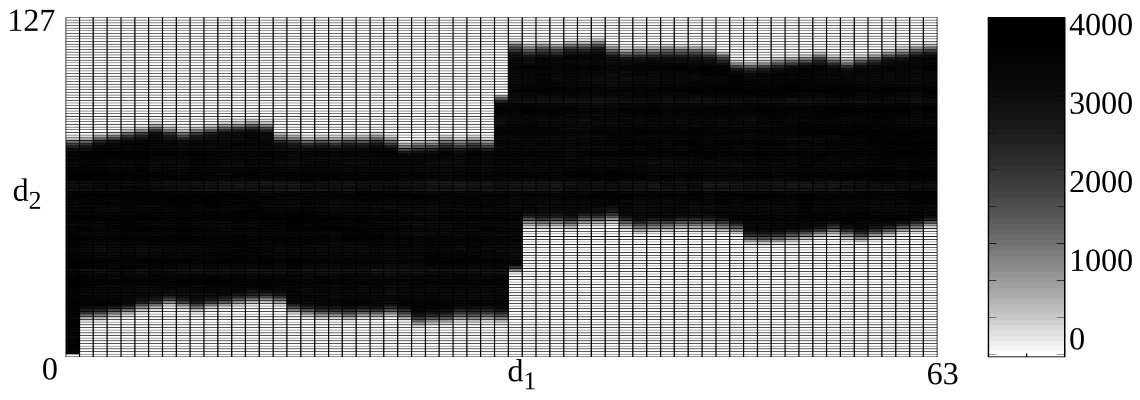


FIGURE 3.5 **Bidimensional Histogram**

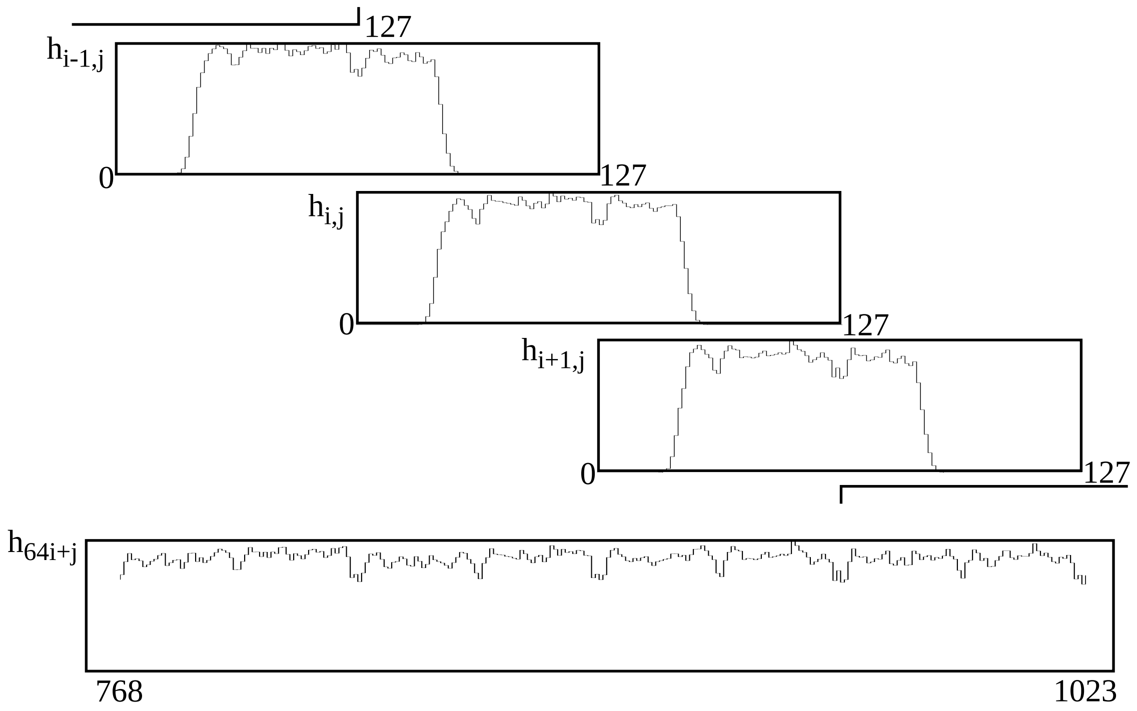


FIGURE 3.8 **Relationship between Uni- and Bidimensional Histograms**

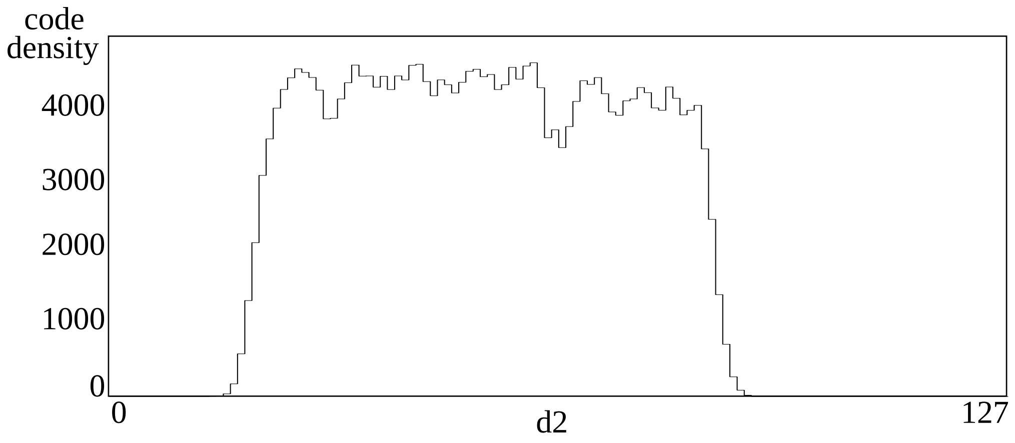


FIGURE 3.7 **Code density for d_1 = 3 and d_2 from 0 to 127**

$$h_j = h_{0,j} \qquad j = 0\ldots63 \text{ (first half of first column)}$$
$$h_{64i+j} = h_{i,j} + h_{i+1,j-64} \qquad i = 0\ldots62,\ j = 64\ldots127 \qquad \text{(EQ 3.6)}$$
$$h_{64\cdot62+j} = h_{63,j} \qquad j = 64\ldots127 \text{ (second half of last column)}$$

where *h* with an index refers to the unidimensional histogram, and *h* with two indices refers to the bidimensional histogram.

3.4 Columns of the Bidimensional Histogram

In order to extract all the information available in the columns of the bidimensional histogram, we have to understand the factors that affect them. These factors are analyzed in the following.

We have mentioned before that the residue dictates the shape of the histogram. The mechanism is illustrated in Figure 3.9. Since the residue signal for code d_1=k ranges from m_k to M_k, the output codes (d_2) will range from m_k/LSB_2 to M_k/LSB_2. Hence it is possible

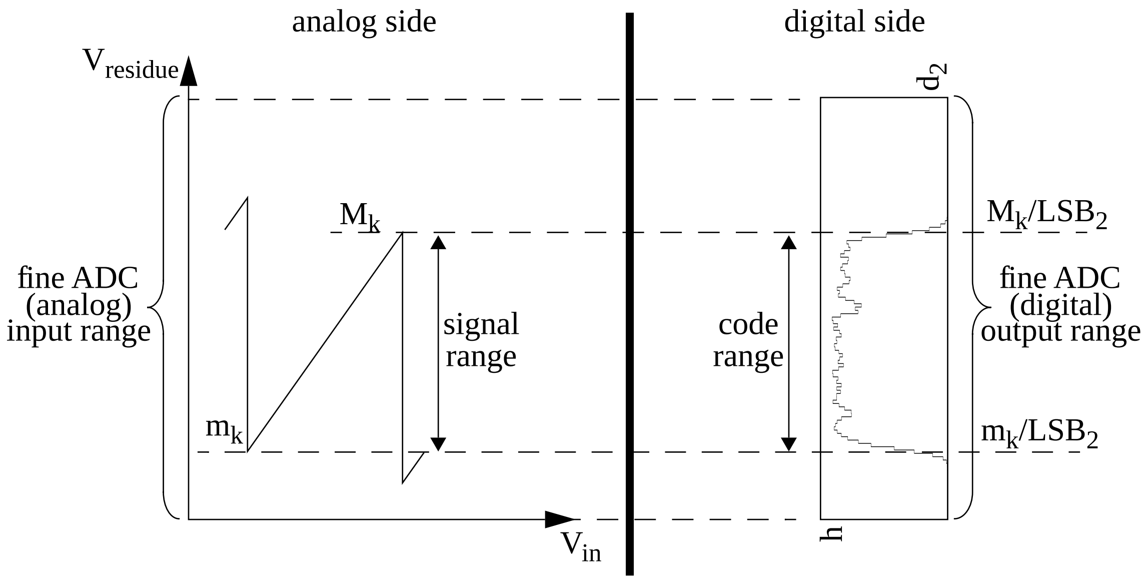


FIGURE 3.9 Relationship between Residue Sections and Histogram Columns

to estimate the bounds of each residue section from the bounds of the corresponding histogram column. For convenience, from now on all the voltages will be expressed in terms of LSB_1 or LSB_2 so that we can use the same notation for analog values and their digital equivalent. For instance, M_k will be both the analog upper bound of the residue and the projection thereof on the digital axis.

A closer look at the histogram column reveals that its shape is not what we might expect to see for an ideal ADC in which the residue has the same bounds (see Figure 3.10). In the

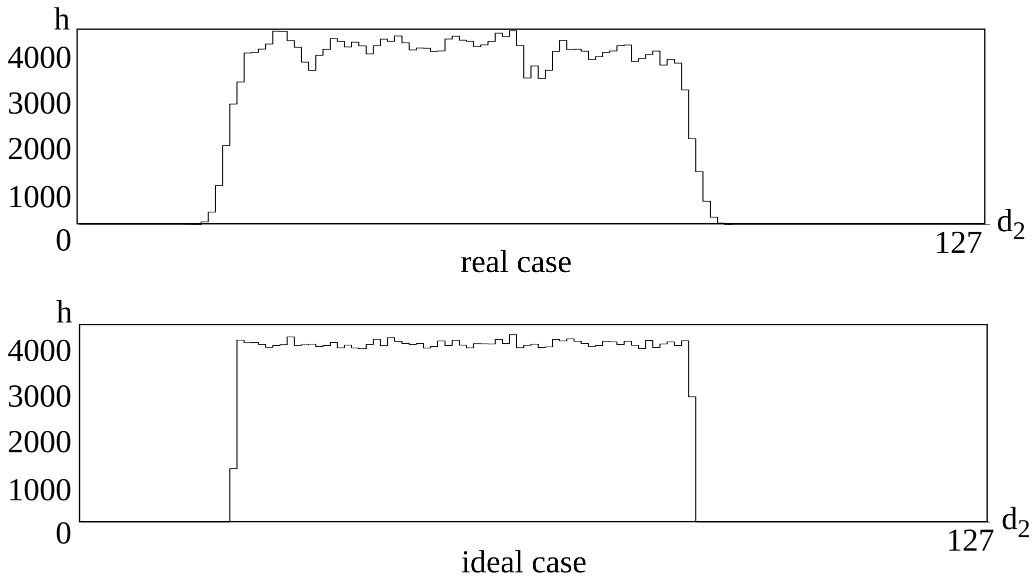


FIGURE 3.10 Histogram Columns in Real and Ideal ADC

ideal case (simulated using a random uniform distribution with the same number of samples as in the real case), the heights of the non-zero bins are about equal, except for two incomplete bins at the left and right end. The presence of incomplete bins is explained by the relative position of m_k and M_k with respect to the closest thresholds of the fine ADC. For instance if m_k coincides with the threshold $t_{2,j}$ then bin j-1 will be empty while bin j will be of average height, and if m_k is exactly in the middle between $t_{2,j}$ and $t_{2,j+1}$, bin j will only be of half the average height.

Note that in both cases (real and ideal) there is some randomness in the middle region of the histogram. The signal applied at the input when building the histograms is in fact deterministic (triangular or sinusoidal waveforms), but we may consider it random because the number of samples used is very large (16 megasamples) and also the frequencies of the signal and of the clock were chosen to be incommensurate, to ensure that all the codes occur. Note that we are not interested in the time when a particular code occurs (this point of time cannot be considered random), but only in how often it occurs in a relatively long period of time.

However, two major differences can be noticed between the ideal and the real case. First, in the real case the lower and the upper bounds are not abrupt but rounded, which makes their accurate estimation a bit more difficult. This can be attributed to phenomena such as noise, hysteresis, drifts, and so on, which occasionally combine to push the residue outside of the nominal interval $[m_k, M_k]$.

Second, in the real case the middle region is not as flat as in the ideal case. The explanation lies in the nonidealities of the fine A/D subconverter: if the thresholds of this subconverter are different from the ideal values, we should not expect the bins to have the same height (since the bin height is proportional to the distance between the two thresholds that define the code of the bin). This suggests that the agreement between the residue and the bidimensional histogram is qualified by the linearity of the fine subconverter and it is this linearity that we should study first.

3.5 Fine A/D Subconverter Characterization

Residue bounds contain information about the errors in the amplifier A2, coarse A/D subconverter, and D/A subconverter, and will be used for the characterization of these components. On the other hand, the bounds do not contain any readily interpretable information about the errors in the fine ADC, so a distinct method must be devised to evaluate them.

Code density tests provide part of the answer. If we had a histogram covering the whole range of the fine ADC, we would be able to estimate its errors, but such a histogram is not available. All we have are the columns of the bidimensional histogram, and there are two problems associated with them: most of them occupy only about half of the range (Figure 3.10, top), and there is a large amount of randomness in them, even for large data sets of 16 megasamples.

A solution to the latter problem is to increase the number of samples until the randomness is small enough compared to the expected bin heights. How many samples would be necessary? It can be easily demonstrated that if p_0 is the probability of hitting a bin (e.g. for a uniform distribution and a 12-bit ADC, p_0 is 2^{-12}) and n the number of samples, the expected value of the bin height is

$$\bar{h} = E[h] = np_0 \qquad \text{(EQ 3.7)}$$

and its variance is

$$\sigma_h = \sqrt{E[h^2 - E[h]^2]} = \sqrt{np_0(1-p_0)} \qquad \text{(EQ 3.8)}$$

It follows that the coefficient of dispersion is

$$\frac{\sigma_h}{\bar{h}} = \sqrt{\frac{1-p_0}{np_0}} \qquad \text{(EQ 3.9)}$$

(EQ 3.9) indicates that the accuracy is inversely proportional to square root of the number of samples. For instance, for a coefficient of dispersion of 1% we need 40 megasamples.

Getting so many samples can be a long and tedious task in the absence of automated test equipment. An alternative to increasing the number of samples is to use information not just from a single column, but from all the 64 columns. In Figure 3.11, two columns (31

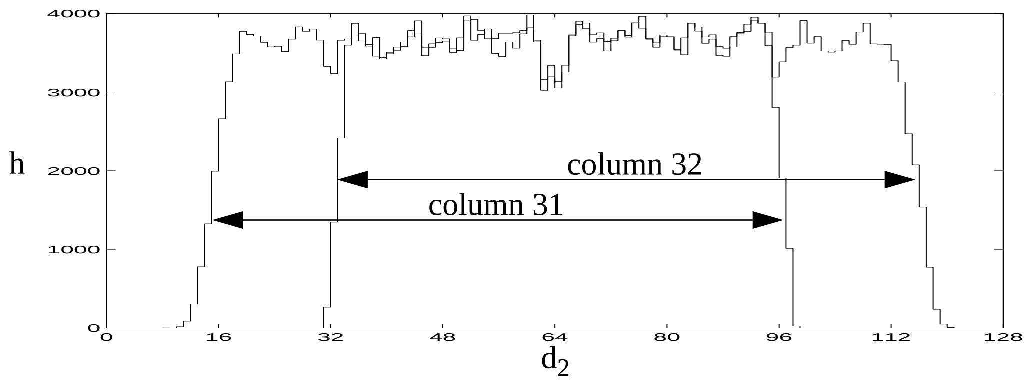


FIGURE 3.11 **Overlapped Columns of Bidimensional Histogram (Uniform pdf)**

and 32) have been overlapped. It is easy to see that in the central region, the histograms follow the same pattern. In the case of a uniform pdf (for a triangular waveform), this is valid for all the columns, so we could simply average (the full bins in) the rows of the bidimensional histogram to get a more accurate estimate. Incidentally, this would also solve in part the other problem. Individual columns cover only half of the range, but some of them are shifted to the right, and some to the left, and combined together they cover more than half of the range.

However, if the input pdf is not uniform (e.g. for a sinusoidal waveform), we have the situation depicted in Figure 3.12. Although the columns seem to follow a similar pattern, there is a substantial difference between them, stemming from the fact that these columns correspond to different sections of the pdf of the input signal.

This dependence on the pdf must be removed from the histogram, before any averaging can take place. Just like in code density tests, this can be accomplished by normalizing the histogram with respect to the estimated pdf. A general solution (that works for any input signal pdf) is to find a polynomial *p*(*i*) that fits the unidimensional histogram, in a least-squares sense. To normalize the unidimensional histogram, we would use

$$h_i^{norm} = \frac{h_i}{p(i)} \qquad i = 0\ldots4095 \qquad \text{(EQ 3.10)}$$

Taking into account the relationship between the unidimensional and the bidimensional histograms (see Section 3.3), we can normalize the latter using:

$$h_{i,j}^{norm} = \frac{h_{i,j}}{p(64i+j)} \qquad i = 0\ldots63,\ j = 0\ldots127 \qquad \text{(EQ 3.11)}$$

In the following we will assume that the bidimensional histogram has been normalized. In our calculations (in MATLAB), second-order polynomials have been used for *p*(*i*), since no improvement has been noticed for higher-order polynomials. The result of normalization and averaging is the reference histogram presented in Figure 3.13.

We explained in Section 2.6 that the errors in the A/D subconverters of the AD9042 fall into two categories: those which result in symmetrical changes on both sides of the vertical axis corresponding to the threshold, and those which result in asymmetrical changes in the codes next to the axis. The effect of these two categories of errors can be

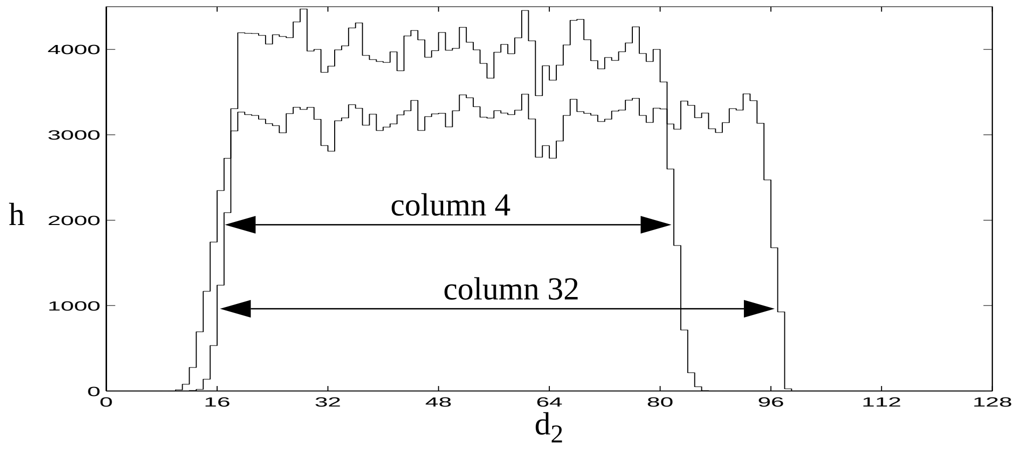


FIGURE 3.12 **Overlapped Columns of Bidimensional Histogram (Nonuniform pdf)**

clearly seen in Figure 3.13. The left half (codes 0 to 63) is mirrored in the right half (codes 64 to 127), with respect to the MSB axis of symmetry d_2=63.5. Also the first quarter is mirrored in the second quarter, with respect to the secondary axis of symmetry d_2=31.5. The second quarter is mirrored into the third (d_2=63.5) which is mirrored into the fourth (d_2=95.5). In other words, bins x, 63-x, 64+x, and 127-x should be equal, where x ranges from 2 to 30. Only the codes next to the axes of mirroring are not symmetrical, which means that the relationship between the bins mentioned above is not valid for bins 62-65 (next to the MSB axis), bins 0-1 and 127-128 (which would have to be symmetric to those next to the MSB axis), and bins 31-32 and 95-96 (next to secondary axes of symmetry). In the case of the AD9042, we can use these symmetries to recreate the bins that were left empty because no column of the bidimensional histogram covered them. The final reference histogram for the AD9042 is shown in Figure 3.14.

However, most two-stage A/D converters use flash subconverters, in which case we have to make do with the histogram in Figure 3.13, since the other bins cannot be reconstructed.

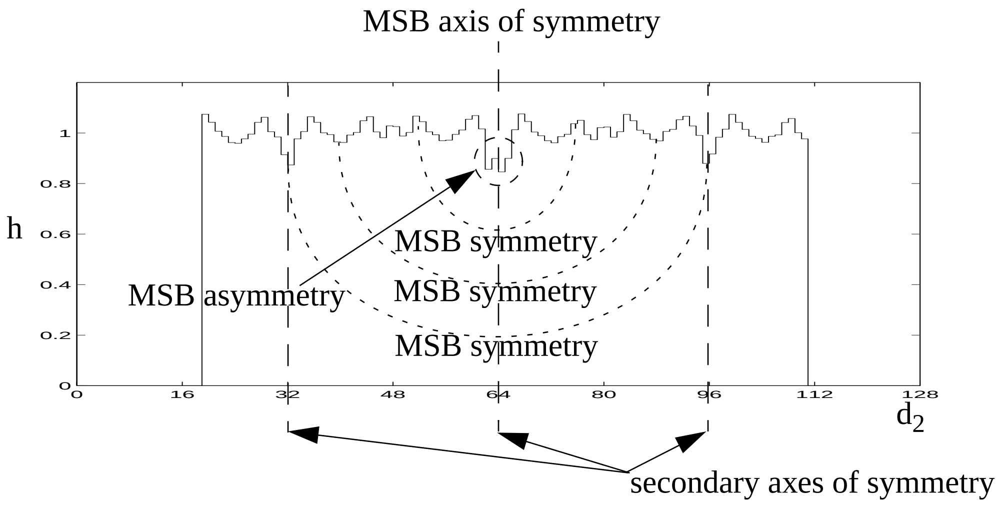


FIGURE 3.13 Symmetries and Asymmetries in the Reference Histogram

Note that the new histogram (Figure 3.14) covers the whole range of the fine ADC, and has significantly less randomness than the individual histogram columns from which it was created, because it contains information from all of them.

The next step is to apply the usual procedure from code density tests to the reference histogram, denoted here by *H*. We know that the bin height H_k is proportional to the difference between the thresholds that define the bin:

$$H_k = q \cdot (t_{2,k+1} - t_{2,k}) \qquad \text{(EQ 3.12)}$$

where *q* is a constant. Adding (EQ 3.12) for *k* from 0 to 127 we obtain:

$$\sum_{k=0}^{127} H_k = q \cdot (t_{2,128} - t_{2,0}) \qquad \text{(EQ 3.13)}$$

$t_{2,128}$ and $t_{2,0}$ are by definition $128LSB_2$ and 0, respectively (see Section 2.4). *q* can be then calculated from (EQ 3.13) as:

$$q = \frac{\sum_{k=0}^{127} H_k}{t_{2,128} - t_{2,0}} \qquad \text{(EQ 3.14)}$$

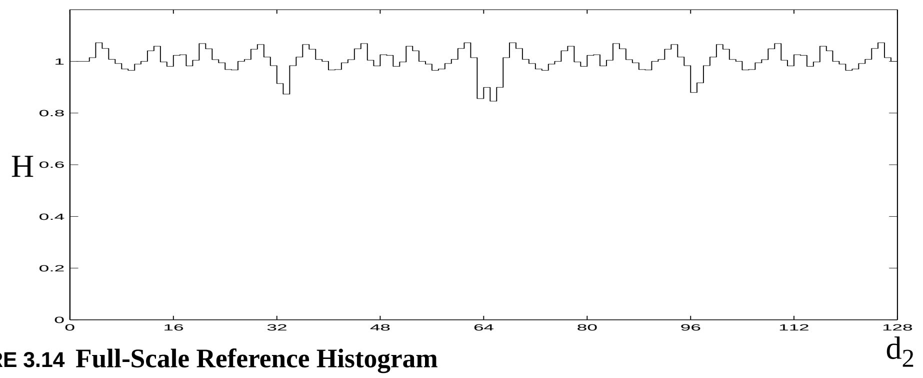


FIGURE 3.14 Full-Scale Reference Histogram

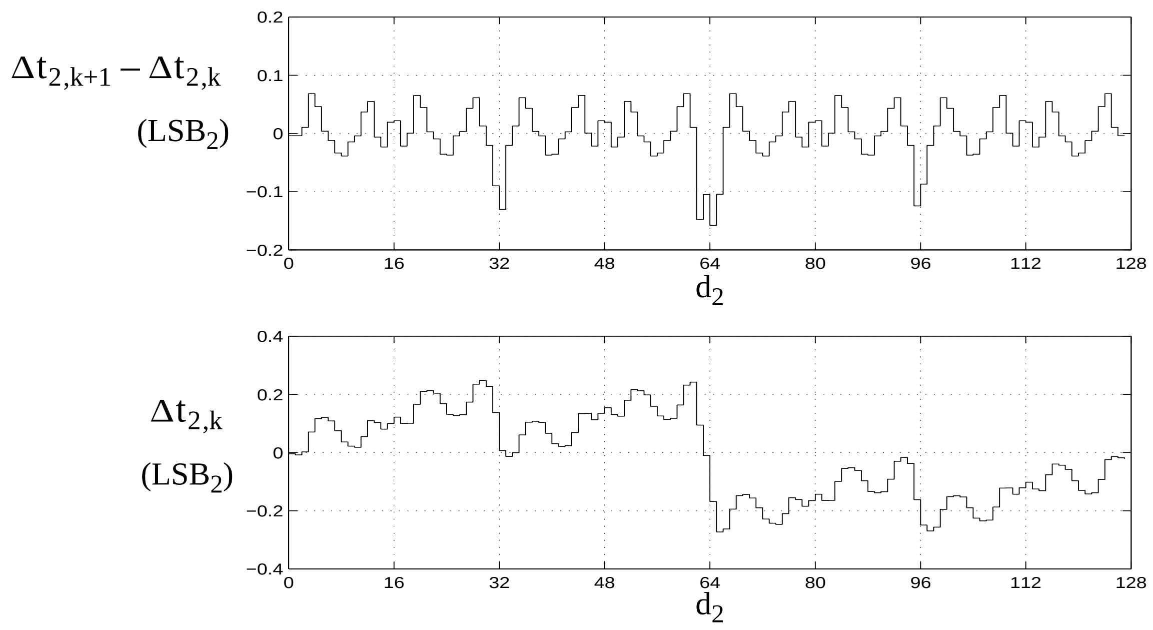


FIGURE 3.15 **Threshold Errors in the Fine A/D Subconverter**

By replacing *q* in (EQ 3.12) and rearranging the terms, a formula for the thresholds of the fine subconverter can be derived:

$$\hat{t}_{2,k+1} = \hat{t}_{2,k} + H_k \cdot \frac{t_{2,128} - t_{2,0}}{\sum_{k=0}^{127} H_k} \qquad \text{(EQ 3.15)}$$

The results have been plotted in Figure 3.15. Only the estimated errors are presented, i.e. the deviations ($\Delta\hat{t}_{2,k}$) of the estimated values ($\hat{t}_{2,k}$) from the ideal values ($\tilde{t}_{2,k} = k$). The top plot represents the difference between two consecutive errors, which is indicative of differential nonlinearity. One can verify that this plot is simply a scaled and offset version of Figure 3.14. The bottom plot is the cumulative sum of the differences shown in the top plot and is indicative of integral nonlinearity.

Note that the bottom plot has a different kind of symmetry from the top plot: the mirroring happens with respect not to a vertical axis but to a point. For instance, the left half is a mirrored version of the right half with respect to the point {63.5,0}. These symmetries are in good agreement with the expected behavior of the subconverter, as explained in Section 2.6.

The change corresponding to bit 6 (MSB) of the fine subconverter can be clearly seen at d_2=64. Also the changes corresponding to bit 5 are quite obvious at d_2=32 and d_2=96. Overall, these errors do not exceed +/-0.25 LSB_2, and they must be taken into account when estimating the bounds of the histograms.

3.6 Histogram Smoothing

Now that the fine ADC has been characterized, out next goal is to characterize the coarse ADC, the DAC, and A2, based on residue bounds which can be estimated from histograms. Before proceeding to estimate them, it is necessary to remove from the histograms any distortions caused by the fine ADC errors. We call this procedure histogram smoothing.

We have already talked briefly in Section 3.4 about the distortion of the histograms caused by the fine ADC. For an ADC with no threshold errors, the envelope of the histogram is proportional to the pdf, as depicted in Figure 3.16 (MATLAB simulations with a 100,000 samples from a signal with a triangular pdf). Note that we have represented the histograms using a stem representation, instead of the customary stairstep, in order to make the envelope obvious. It is easy to notice that the bins in the code density plot are proportional to the area delimited by two consecutive thresholds, marked on the pdf plot with vertical dashed lines.

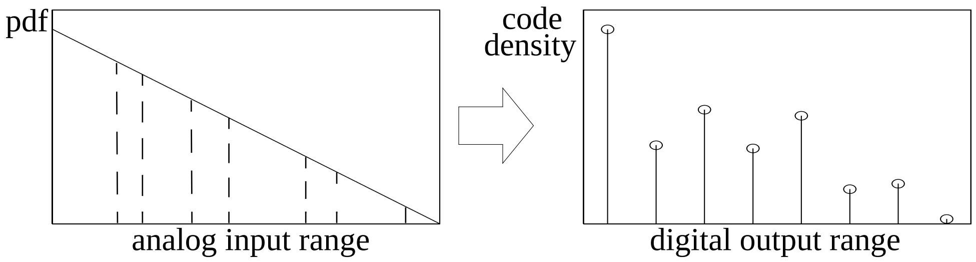


FIGURE 3.17 **Relationship between pdf and Histogram for ADC with Threshold Errors**

In reality, the fine ADC suffers from threshold errors, as we have seen in the previous chapter. As a result, the histogram might not reflect accurately the pdf. Figure 3.17 illustrates this situation, with random threshold errors between -0.3 and 0.3 *LSB*. It is hard to believe that the histogram at the right actually originates from the pdf at the left, but this is indeed the case.

What we would need is a procedure to make the histogram agree better with the pdf, and this should not be too difficult since we have already calculated the errors that cause the distortion.

Let $p(x)$ be the pdf, and t_k the thresholds of an ADC. If the number of samples is very large, we can estimate the bin height by:

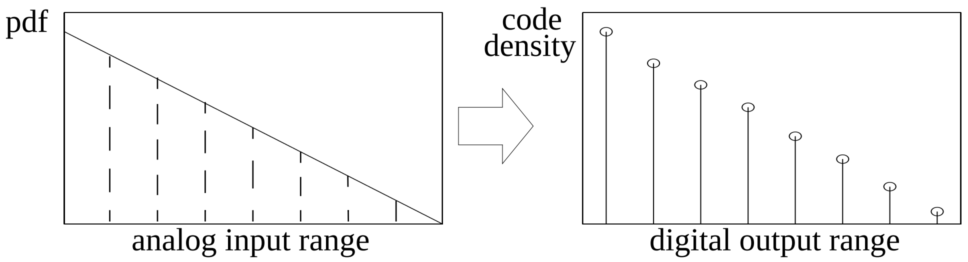


FIGURE 3.16 **Relationship between pdf and Histogram for Ideal ADC**

$$h_k \cong const. \int_{t_k}^{t_{k+1}} p(x)dx \quad \text{(EQ 3.16)}$$

This formula can be approximated as

$$h_k \cong const.(t_{k+1} - t_k) \cdot p\left(\frac{t_k + t_{k+1}}{2}\right) \quad \text{(EQ 3.17)}$$

Therefore the bin height is set by the pdf at the midpoint between the two thresholds and the difference between the two thresholds.

(EQ 3.17) indicates that the histogram is in fact a plot of pairs $\left\{\frac{\tilde{t}_k + \tilde{t}_{k+1}}{2}, const.(t_{k+1} - t_k) \cdot p\left(\frac{t_k + t_{k+1}}{2}\right)\right\}$. On the other hand, the plot of the pdf is made up of pairs $\{x,p(x)\}$. A comparison between the two forms suggests what we have to do to correct the distortion in the histogram. First we have to adjust the bin height

$$h_k \leftarrow h_k \cdot \frac{LSB}{\hat{t}_{k+1} - \hat{t}_k} \quad \text{(EQ 3.18)}$$

and then we have to adjust its position too. Instead of $\frac{\tilde{t}_k + \tilde{t}_{k+1}}{2}$ we should put it at $\frac{\hat{t}_k + \hat{t}_{k+1}}{2}$ (note that we are using estimated values, because this is all we have). The result can be seen in Figure 3.18. Note that the bins are now at irregular intervals, but their envelope reflects the pdf with high accuracy.

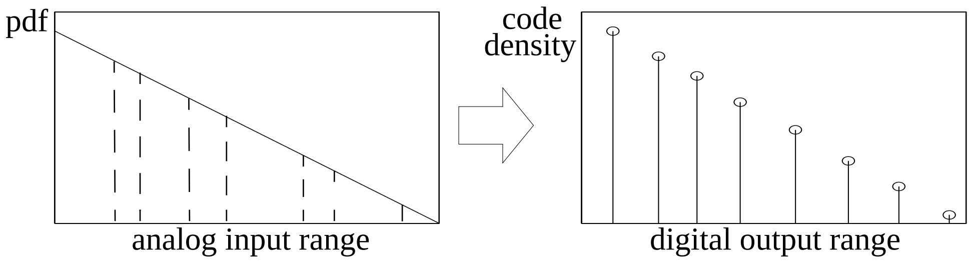


FIGURE 3.18 Relationship between pdf and Smoothed Histogram for Real ADC

We can now apply this procedure to the columns of the bidimensional histogram, using the thresholds estimated in Section 3.5, ($\hat{t}_{2,k}$). An example of a smoothed histogram is shown in Figure 3.19.

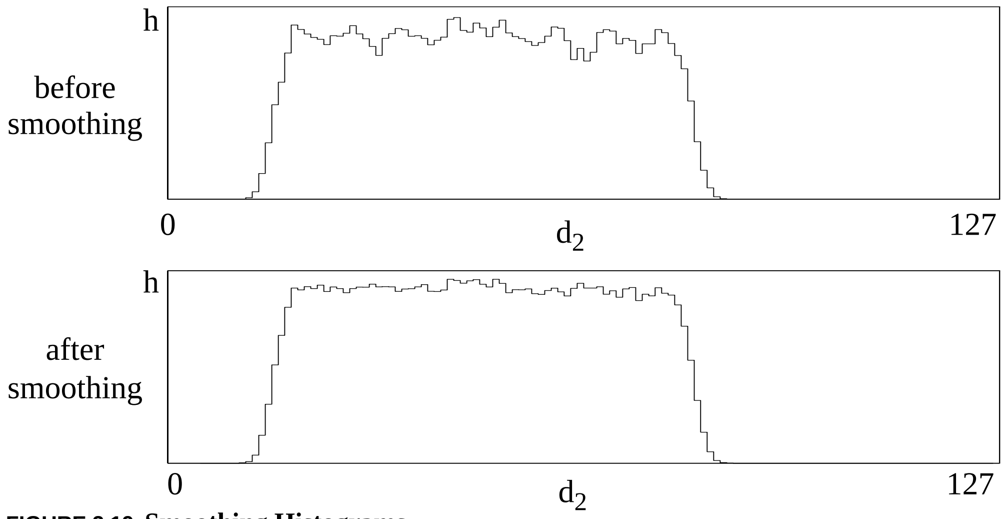


FIGURE 3.19 Smoothing Histograms

There is still randomness in the histogram because we used a finite number of samples, but systematic errors have been eliminated, so that we can now go on to the next step.

3.7 Bound Estimation

After a histogram has been normalized and smoothed, the histogram bounds should coincide with the residue bounds. The last difficulty in estimating these bounds is the fact that they are rounded by noise.

Histogram bounds without and with noise are depicted in Figure 3.20. In the noiseless case the estimation of the bounds is straightforward. If *j* is the rightmost full bin and *j*+1 is the empty bin next to it (Figure 3.20, left), we would estimate M_k as:

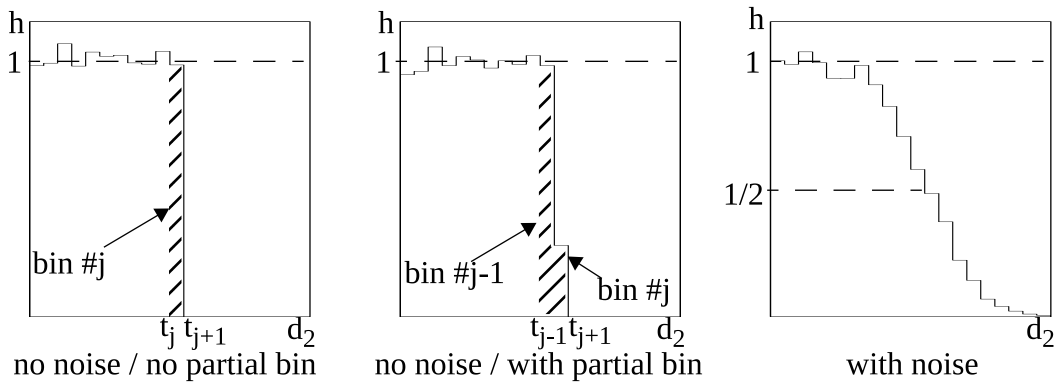


FIGURE 3.20 Histogram Bound without and with Noise

$$\hat{M}_k = \hat{t}_{j+1} \tag{EQ 3.19}$$

If there is a partial bin after the last full bin, the estimate would be

$$\hat{M}_k = \hat{t}_j + \frac{h_j}{\bar{h}}(\hat{t}_{j+1} - \hat{t}_j) \tag{EQ 3.20}$$

where h_j is the height of bin j and $\bar{h}$ is the average bin height. It is easy to see that when h_j goes to zero (partial bin almost empty) the estimate $\hat{M}_k$ goes to $\hat{t}_j$, and when h_j goes to $\bar{h}$ (partial bin almost full) the estimate $\hat{M}_k$ goes to $\hat{t}_{j+1}$, which agrees with (EQ 3.19) .

This formula does not work any more for the case with noise. M_k is probably located somewhere between the last full bin (i.e. bin not affected by rounding) and the first empty bin.

There are a number of solutions to this problem, all of which assume that the noise is additive and has symmetric pdf. The solution presented here is based on the observation that in the presence of noise, if there were a bin centered exactly at the bound, its height would be about 1/2 (see Figure 3.20, right), i.e. half of the average bin height which is 1 because the histogram has been normalized.

Let us approximate the shape of the transition region with a polynomial f using the least-squares method, so as to minimize

$$\sum_{j_1}^{j_2}\left(f\left(\frac{\hat{t}_j+\hat{t}_{j+1}}{2}\right)-h_j\right)^2$$

where j_1 and j_2 are the limits of the transition region, and j_2 - j_1 is typically 7-8 LSB_2. A low order polynomial (second- or third-order) has to be used to ensure that the randomness is averaged out. Regardless of the noise level, the value of the polynomial calculated at the actual bound M_k should be equal to $1/2$, so that

$$\hat{M}_k = \left\{x,\ f(x) = \frac{1}{2}\right\} \qquad \text{(EQ 3.21)}$$

A similar approach can be used to estimate the lower bounds, m_k. The estimates of the upper and lower bounds thus obtained are shown in the figure below. The reader is urged to compare Figure 3.21 to Figure 3.5. The former is in fact the estimated contour of the latter, and they are in good agreement.

We also need to know the variance of these estimates, preferably expressed in LSB_2. Finding a mathematical formula is quite a difficult task, so we preferred an empirical approach. We used N=16 sets of 1 megasample derived from the same input signal, calculated the bound estimates for each of them, and then estimated the variance of the population of bound estimates:

$$\sigma_M(1Meg) = \sqrt{\frac{\sum_N (\hat{M}_k - \overline{\hat{M}_k})^2}{N-1}} \qquad \text{(EQ 3.22)}$$

This is valid for 1 megasample and must be divided to $\sqrt{N}$ for N megasamples. The resulting variance for 16 megasamples is less than 0.1 LSB_2.

The same approach was used for the estimates of some of the component errors in the next section (coarse A/D errors and D/A errors) and the resulting variance was less than 0.1 LSB_2 for simple estimates and less than 0.35 LSB_2 for cumulative sum estimates.

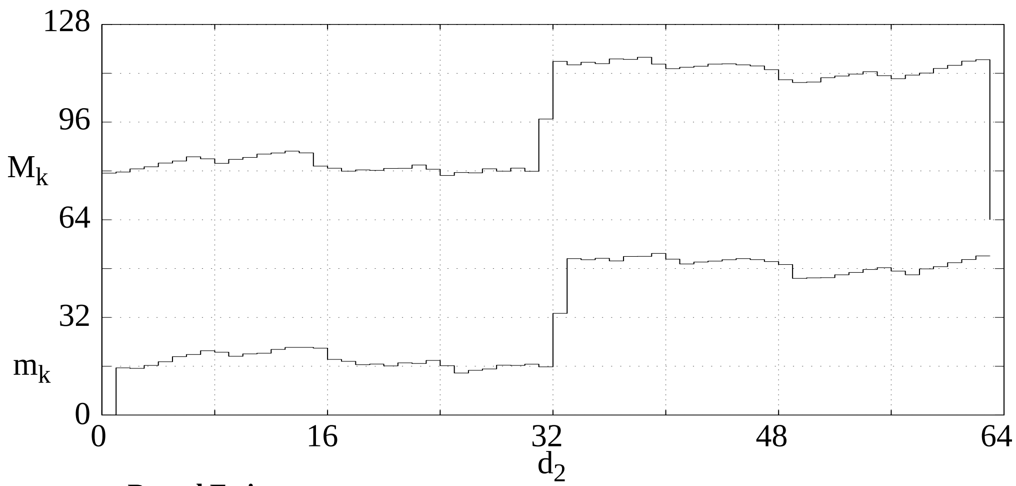


FIGURE 3.21 **Bound Estimates**

3.8 Component Characterization

We have seen previously that the parameters of the components can be calculated from residue bounds (Section 2.5), and we estimated the residue bounds (Section 3.7). We can now proceed to calculate the parameters from the estimated residue bounds. The components covered here are the amplifier A2, the coarse ADC, and the DAC.

The first parameter we should calculate is the gain, since we need it for the other parameters. From (EQ 2.16) we can obtain:

$$\hat{A}_2 = \frac{\sum_{k=0}^{63} (\hat{M}_k - \hat{m}_k)}{64} \tilde{A}_2 = 1.0005\tilde{A}_2 \qquad \text{(EQ 3.23)}$$

In the following chapters we will use the value 1.0005·64 for the gain, although its effect is small enough not to have any impact.

Next we calculate the threshold errors in the coarse A/D subconverter. Using (EQ 2.17) , we arrive at:

$$\Delta\hat{t}_{1,k+1} = \Delta\hat{t}_{1,k} + \frac{\hat{M}_k - \hat{m}_k}{\hat{A}_2} - LSB_1 \ , \qquad k = 0 \dots 62 \qquad \text{(EQ 3.24)}$$

Since $\Delta t_{1,0}$ is by definition always zero, (EQ 3.24) enables us to calculate all the $\Delta\hat{t}_{1,k}$ errors one by one. The results may be seen in Figure 3.22. Both the differences between two consecutive threshold errors and the actual threshold errors have been plotted, since the former are indicative of differential nonlinearity and the latter of integral nonlinearity.

Although this plot seems to be very different from the plot of threshold errors in the fine ADC, Figure 3.15, in fact it obeys the same rules of symmetry, because the two subconverters have similar architectures. For instance, in the top plot, the left half is mirrored in the right half with respect to the middle vertical axis. The transition corresponding to bit 5 (MSB) can be seen clearly at d_1=32, and the transitions corresponding to bit 4 can be seen at d_1=16 and d_1=48.

The results shown in Figure 3.22 can be further used to obtain information about errors at the component level. Consider for instance the middle of the top plot. The two errors corresponding to the MSB stage (at 31 and 32) stand out: their asymmetric component is due to V_{be} mismatch, and their symmetric component is due either to resistor or current mismatch. A comparison with Figure 2.9 indicates that we are probably dealing with a current mismatch. We can also study the following stages, but the errors become smaller and smaller and are more difficult to distinguish.

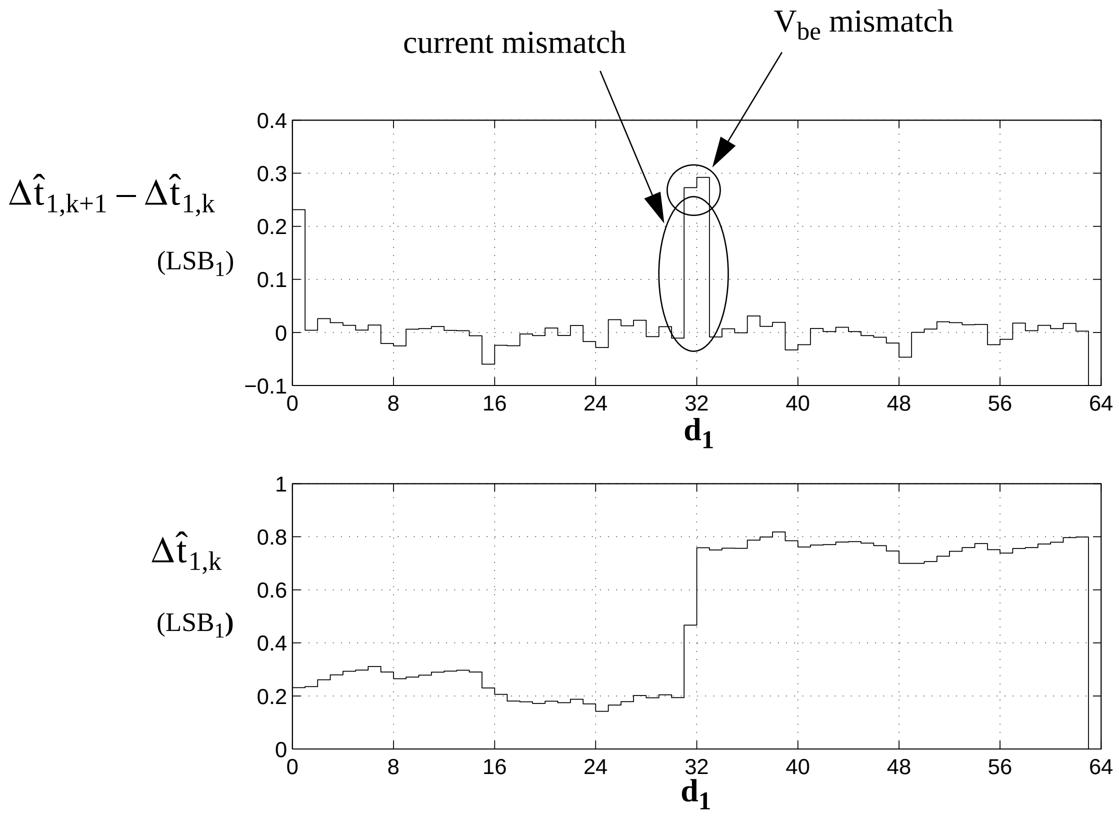


FIGURE 3.22 Threshold Errors in the Coarse A/D Subconverter

In the same fashion we can calculate the output level errors in the D/A subconverter. (EQ 2.18) can be rewritten as:

$$\Delta\hat{l}_{k+1} = \Delta\hat{l}_k + \frac{\hat{M}_k - \hat{m}_{k+1}}{\hat{A}_2} - LSB_1 \ , \qquad k = 0\ldots62 \qquad \text{(EQ 3.25)}$$

where Δl_0 is zero.

The results are shown in Figure 3.23. Note that the units used to plot the coarse A/D errors and the D/A errors are different: the former are in LSB_1, the latter in LSB_1/A_2. The errors in the coarse ADC are much larger, but do not affect the linearity of the overall ADC since

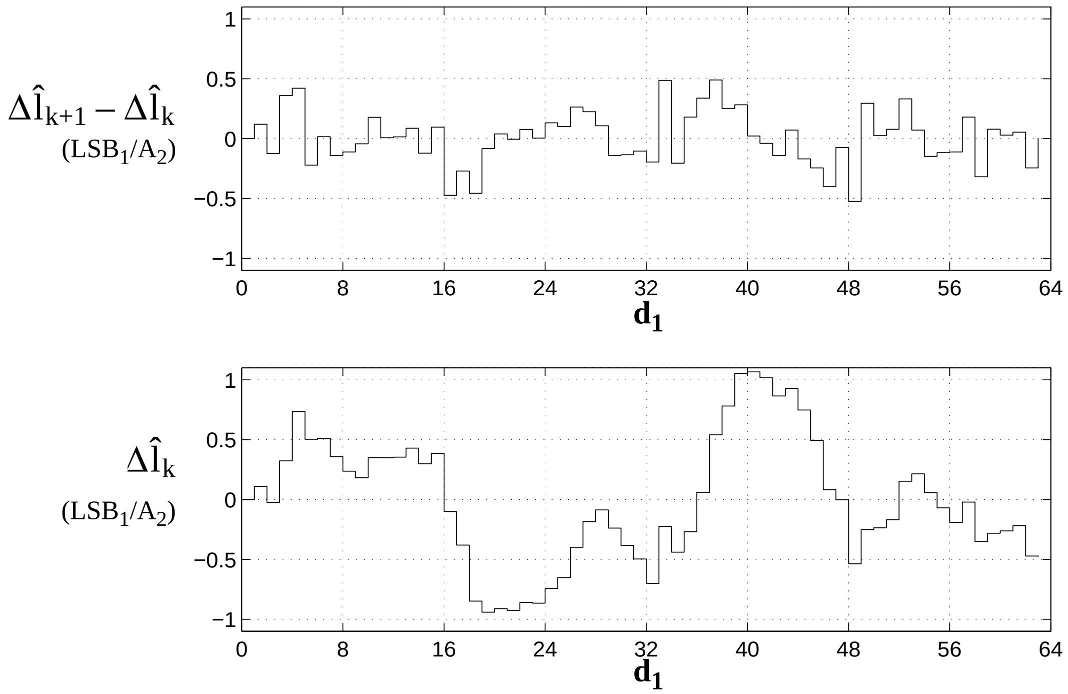


FIGURE 3.23 **Output Level Errors in the D/A Subconverter**

they are removed by the mechanism of range overlap and digital correction (see Section 2.4). On the contrary, the errors in the DAC, albeit an order of magnitude smaller, result in significant nonlinearity. In fact, of all the sources of nonlinearity discussed in Section 2.4

and estimated in the current chapter (DAC errors, inter-stage gain error, fine ADC errors) the DAC errors are the main cause of ADC nonlinearity.

3.9 Discussion of Characterization Method

Before concluding Chapter 3, we would like to review the steps involved in the histogram-based ADC characterization method that we are proposing. Some fine points, the limitations of the method, and other possible uses of the bidimensional histogram are also covered here.

The steps of the method, described in detail from Section 3.5 to Section 3.8, are outlined in the following.

1. Data sampling

Two issues must be considered when sampling data for characterization. First, the larger the number of samples, the more accurate the estimates are. The largest sample sets we used were of 16 megasamples, which led to a 0.35 *LSB* accuracy. Automated test equipment, like the one used for standard code density tests would be most helpful.

Second, the input signal used for characterization cannot be completely arbitrary. It must cover the whole input range of the A/D converter, without leaving empty spots, and it should have a pdf easy to describe mathematically so as to be able to normalize the histogram (see next step). This requirement excludes some classes of signals, such as those which can take only a finite number of values (e.g. an ideal squarewave can only take two values), or periodic signals synchronized to the clock (e.g. a sinewave whose frequency is $\frac{m}{n} f_{clock}$ can only take a maximum of n values). On the other hand, one can use sinewaves or combinations of them that are not (all) synchronized to the clock as long as they are overdriven, so as to cover the whole range.

2. Histogram Normalization

This step removes the dependence of bin heights on pdf. If the input signal is not known or hard to describe, polynomial functions can be used to fit the unidimensional histograms. These polynomials are then used to normalize the bidimensional histograms. For sinusoidal waveforms, one can use instead

$$p(i) = \frac{1}{\pi\sqrt{\hat{A}^2 - (i - \hat{V}_{offset})^2}} \qquad \text{(EQ 3.26)}$$

where the amplitude A and the offset V_{offset} must be estimated [Raz95]. This formula accurately describes the pdf of such waveforms.

3. Fine ADC Characterization

The threshold errors of the fine ADC can be estimated from the columns of the bidimensional histogram. Based on the architecture of the subconverter, we can predict the form of its threshold errors and use these predictions as a reality check for the estimates.

4. Histogram Smoothing

The correspondence between histogram columns and residue sections is distorted by the errors of the fine ADC. Since estimates of these errors are readily available (from the previous step), their effect on histograms can be removed through a simple procedure.

5. Bound Estimation

In the case of rounded histogram edges, the bounds can be estimated as the points where the histograms are half the average full-bin height, which in the case of normalized histograms is equal to one. Other approaches are also possible (see Section 5.2).

6. Coarse ADC, DAC, and A2 Characterization

A simple system of equations relates the residue bounds to the component parameters, which can therefore be calculated, providing information about the blocks of the ADC most likely to degrade its performance.

The reader should keep in mind that these results are only as good as the model used for the ADC. The residue was assumed to be made up of linear sections (Section 2.4), all having the same slope. This may or may not be true; for instance, TH2, TH3, A2, or Σ may have a nonlinear characteristic and this may result in distorted results. However, in a well-designed converter, the linearity of these components is much better than that of the subconverters. It is interesting to note that nonlinearity in TH1 or A1 is not necessarily a problem for this method of characterization (although this would obviously degrade the performance of the ADC) since it can be removed, to some extent, by histogram normalization.

So far we have derived bidimensional histograms from all the samples of the input signal. Nothing prevents us from constructing separate histograms for special conditions. An example of such a condition is the sign of the slope of the input signal. In Figure 3.24 we have superimposed a column from a positive-slope histogram and the same column from a negative-slope histogram (both histograms coming from the same input signal). One can see that their bounds do not coincide, revealing the existence of a hysteresis of about $2LSB_2$, which makes the transitions regions slightly wider than they would be in the presence of noise only. We believe that this hysteresis is not due to the A/D subconverters, because if it were, it should have different signs for different thresholds (because of the folding characteristic), and different magnitudes (for instance the threshold corresponding to the MSB substage should be the largest). In reality the hysteresis turns out to be constant, and it may be caused by the S/H circuits.

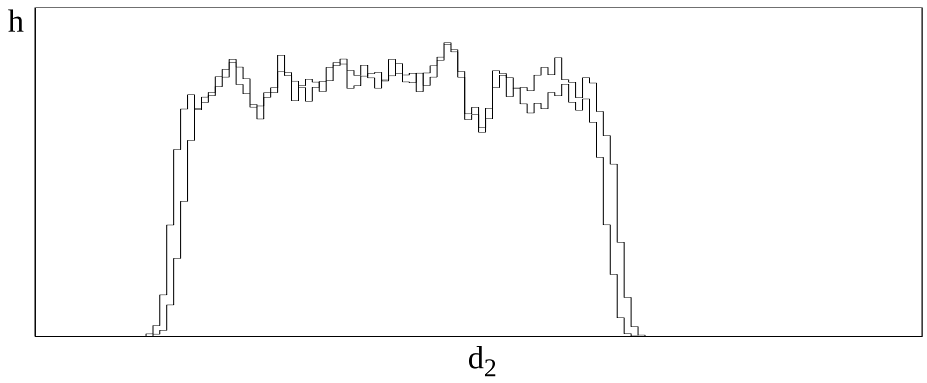


FIGURE 3.24 **Histogram Columns for Positive and Negative Slope Input**

One may also envision histograms for different slopes, temperatures, power supply voltages, and so on, in order to study how the parameters of the components of the ADC vary with these conditions.

Another interesting application is fault detection. A/D converters occasionally suffer from missing codes. Classic code density tests can point out this kind of problem, and with a bit of detective work, the test engineer may be able to determine which block is out of order. This task is much easier with a bidimensional histogram. Consider the bidimensional histogram shown in Figure 3.25, obtained from a real chip.

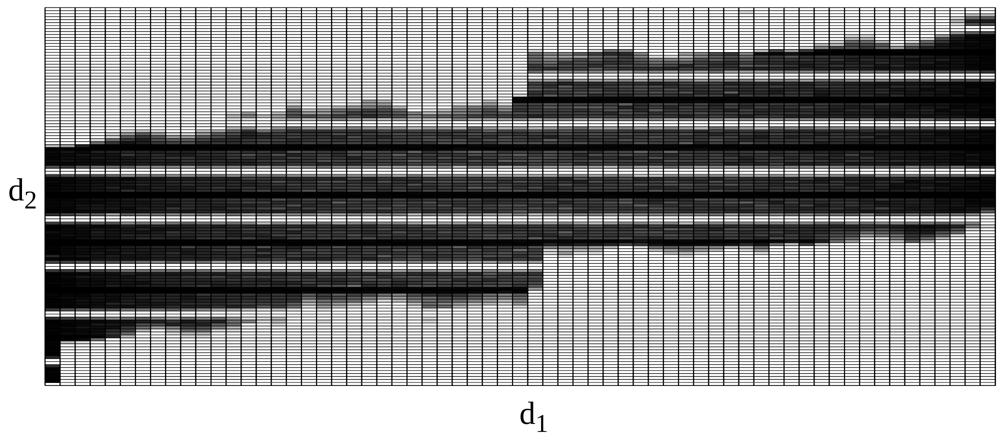


FIGURE 3.25 **Bidimensional Histogram of Faulty ADC**

One can see horizontal light-colored lines through the histogram, evidencing missing codes. We can tell immediately that the problem is located in the fine A/D subconverter, because the lines are horizontal. If they were vertical, the problem would have been in the coarse A/D subconverter. The missing d_2 codes are 7, 8, 23, 24, 39, 40, and so on, all of which, written in Gray code as they are internally coded, end in ...100. On the other hand, the Gray codes ending in ...000 are of double height, which means that all the ...100 codes are turned into ...000 codes. A standard code density test would also reveal the missing codes, but it is much easier to identify the location of the fault on a bidimensional histogram than on a unidimensional histogram.

As with any other ADC characterization method, the cost is an important issue. The hardware requirements are very simple: one internal signal (the MSB of d_2 or the LSB of d_1) must be accessed so as to reconstruct d_1 and d_2, and build the bidimensional histograms. An extra buffer and output pad can be easily included in the design to this purpose, especially if it is a test run. The software requirements are also easy to meet since any mathematical software package with root-finding, system-solving, and least-squares optimization capabilities can be used to write the programs necessary to analyze the bidimensional histograms.

In Chapter 5 we will show that the errors estimated through this procedure can be used for digital correction in the case of a pipeline ADC, leading to an increase of 6dB in the SFDR over most of its bandwidth. When we applied the same approach to the AD9042, no improvement has been noticed. A probable explanation is the presence of dynamic errors. Using another digital correction technique, namely phase-plane compensation [Hum92a], [Hum92b], [Hum93], [Hum96a], [Hum96b], [Iro91], and [Iro96] (also see Section 5.5), we have been able to prove the existence of such errors which are not taken into account by our method. The use of the histogram-based characterization method for digital correction is presumably limited to multi-stage ADCs whose performance is limited by static errors.

3.10 Summary

In this chapter bidimensional histograms have been introduced as a means of analyzing two-stage A/D converters. An experimental part was characterized and diagnosed as to its key sources of nonlinearity, which were subconverter errors and interstage gain errors. There are a number of difficulties with this approach, resulting from pdf nonuniformity, fine ADC nonlinearity, noise, and other phenomena, and methods to go around these difficulties have been presented. The nonidealities of the ADC can thus be estimated with reasonable accuracy, providing valuable information to the design engineers about the limitations of the blocks that make up the converter.

CHAPTER 4 Pipeline ADC Architecture

In Chapter 5 we will show that the theory we have developed to estimate the errors in a two-stage ADC can be applied with minor modifications to a pipeline ADC as well. The present chapter is devoted to explaining the architecture of the particular pipeline converter that was used to test the method and finding a suitable model for it.

4.1 Introduction

We have seen in Chapter 2 that in order to build the bidimensional histograms and characterize a two-stage ADC, we need access to an internal signal. It will be shown in Section 4.3 that the same is true when applying characterization to pipeline converters: we need some internal signals in addition to or instead of the normal output. The problem is that in commercial pipeline ADCs the required signals are not available out of the chip for testing.

Consequently, we decided to design a chip that meets the access requirement. Fortunately, we did not have to start from scratch, since we had access to an earlier design by Timo Rahkonen, Antti Mantyniemi, and Antti Ruha from the Department of Electrical Engineering, University of Oulu, Finland. In its first version [Man96a], [Man96b](Figure 4.1), the ADC had 10 conversion stages, each of them having three possible outputs (-1, 0,

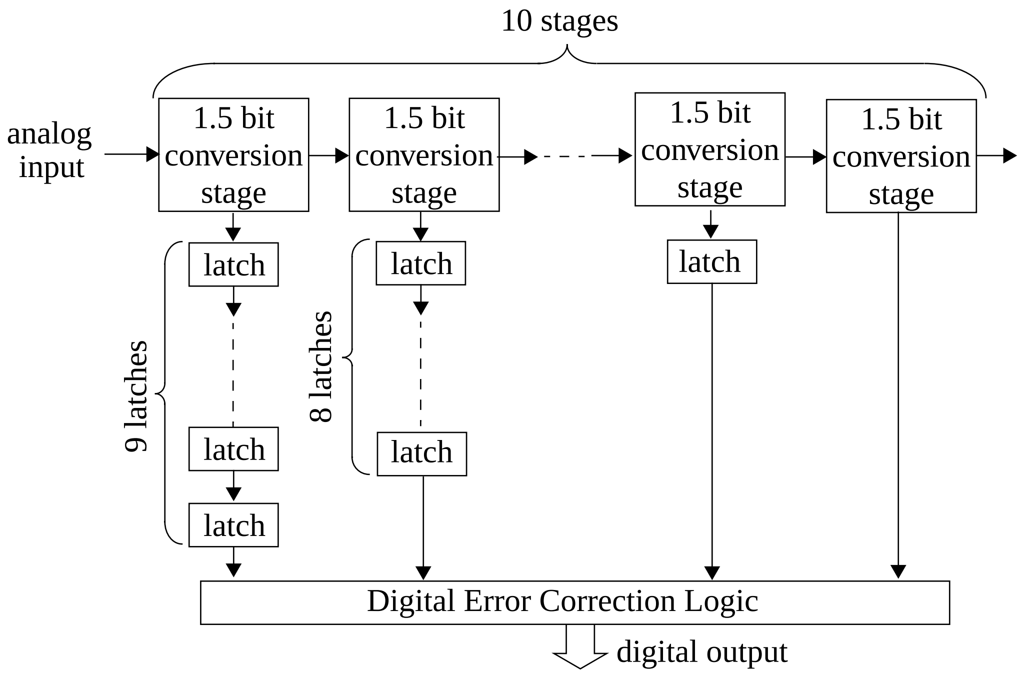


FIGURE 4.1 **Block Diagram of 10-Stage Pipeline ADC**

or +1). This type of stage is called a 1.5-bit conversion stage, and when more of them are combined into a pipeline ADC, this is called a Redundant-Signed-Digit architecture (RSD).

The operation of the stages is interleaved. The clock has two phases and the odd stages clock on phase 1 (ϕ_1), while the even stages clock on phase 2 (ϕ_2). Each stage generates (on ϕ_2 if it is odd, and on ϕ_1 if it is even) a digital output which is passed through latches to the digital error correction logic, and also an analog signal (the residue) which is applied at the input of the next stage. The residue produced by each stage is delayed from the preceding stage by half a clock cycle, so that 10 phases (i.e. 5 clock cycles) are necessary for the signal to propagate up to the output of the last stage. The digital outputs of the stages are delayed by latches a number of phases to make up for the latency on the analog path so that all the bits applied at the input of the digital correction logic during a clock cycle correspond to the same analog input sample.

The chip was fabricated in a double-metal double-poly 0.8µm BiCMOS technology (Austria Mikro Systeme) and had the following performance metrics:

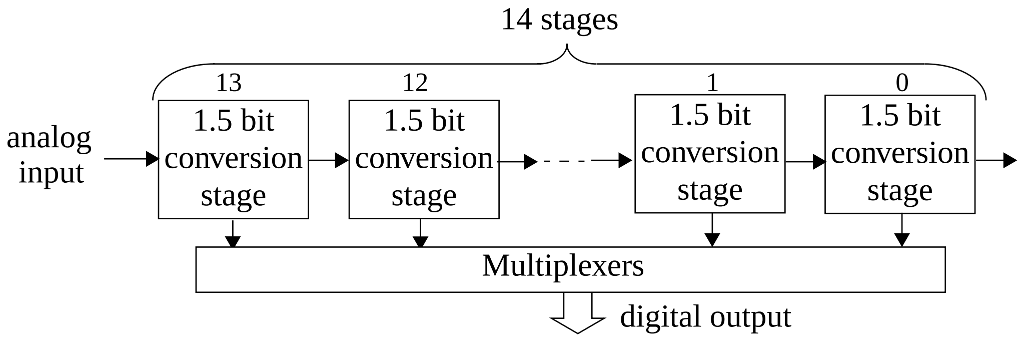
- power consumption 5.4mW at 2.7V (the chip was specifically designed as a low power circuit)
- sampling frequency 300kSamples/sec
- SNR=58.5dB
- ENOB=9.4 (effective number of bits)
- SFDR=64dB
- DNL=0.5LSB
- INL=1.5LSB

Our version (Figure 4.2) has 14 1.5-bit stages (extra stages are provided for characterization/correction) and its performance before (histogram-based) correction is very similar to that of its predecessor. The 28 digital outputs of the 14 conversion stages are not combined on-chip to build the overall digital output, but are made available out of the chip through multiplexing. The odd stages supply their bits at the output on ϕ_1, and the even stages on ϕ_2. The output is sampled on both phases, and the samples are collected by a logic analyzer and transferred to a computer for analysis. Since no latches are included, for reasons of simplicity, the delaying is done by software. This implementation, although inconvenient for normal ADC operation, is ideal for testing the characterization method because it offers access to all the internal signals.

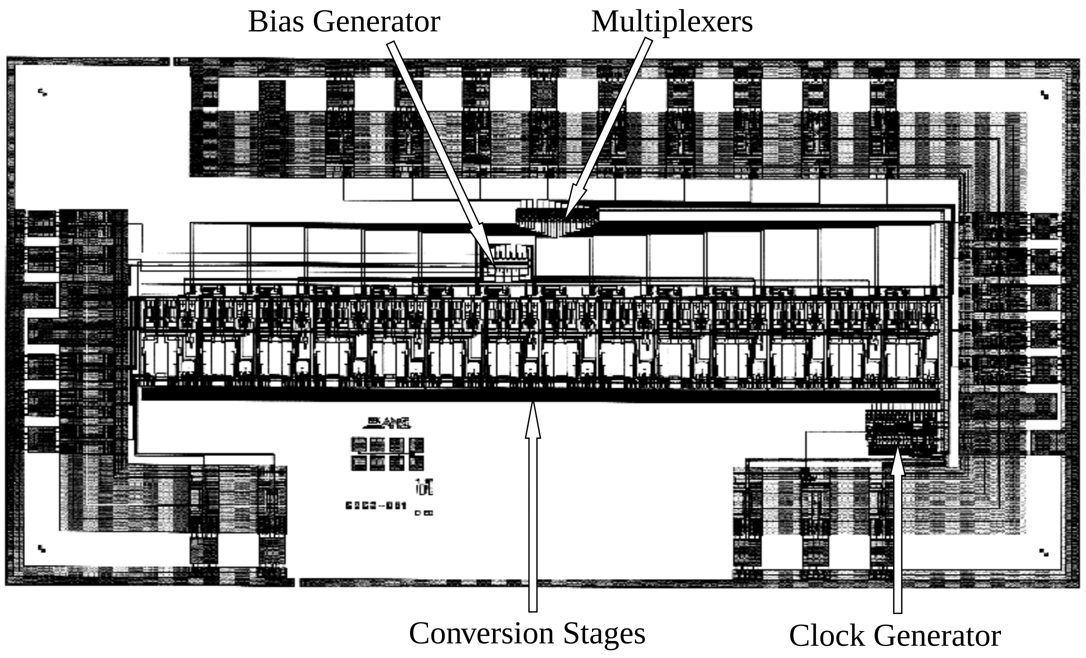
Each 1.5-bit conversion stage provides at the output a pair of bits (D_k^-, D_k^+), which can take the values (0,1), (0,0), and (1,0) (more on this topic in the following chapter). These two bits are then subtracted from each other, so that the actual output of the stage is defined as a ternary value

$$D_k = D_k^+ - D_k^- \qquad \textbf{(EQ 4.1)}$$

which can take the values +1, 0, and -1, and the overall ADC output is calculated as



Block Diagram



Layout

FIGURE 4.2 14-Stage Pipeline ADC with Access to Internal Bits

$$d = \underbrace{\begin{Bmatrix} +1 \\ 0 \\ -1 \end{Bmatrix}}_{D_{13}} 2^{13} + \underbrace{\begin{Bmatrix} +1 \\ 0 \\ -1 \end{Bmatrix}}_{D_{12}} 2^{12} + \dots + \underbrace{\begin{Bmatrix} +1 \\ 0 \\ -1 \end{Bmatrix}}_{D_{1}} 2^{1} + \underbrace{\begin{Bmatrix} +1 \\ 0 \\ -1 \end{Bmatrix}}_{D_{0}} 2^{0} \qquad \text{(EQ 4.2)}$$

It is apparent that there is a lot of redundancy in the definition of d: there are 14 stages each with 3 possible outputs, for a total of 3^{14} combinations, but d can only take $2^{15} - 1$ values (between $-2^{13} - 2^{12} - \ldots - 2^1 - 2^0$ and $2^{13} + 2^{12} + \ldots + 2^1 + 2^0$), so that the average number of combinations for each possible output is 145.969. This number is different for different outputs: for instance $d = 16383$ can only be written as $2^{13} + 2^{12} + \ldots + 2^1 + 2^0$, while $d = 16381$ can be written either as $2^{13} + 2^{12} + \ldots + 2^1 - 2^0$ or as $2^{13} + 2^{12} + \ldots + 0 + 2^0$, and other outputs can be written in hundreds of ways.

The purpose of this redundancy is to make the ADC insensitive to threshold errors, and consequently eliminate the deadbands from the overall input-output characteristic. The idea is that if a conversion stage makes an error in calculating the digital output (because of threshold errors), the error will be corrected in the following stages and the output will be right, though resulting from a different combination. For instance, suppose that the analog input is such that d should be 2^{12}. Then, we would expect stage 13 to supply a 0, stage 12 to supply a +1, and all the following stages to supply a 0, according to (EQ 4.2) . If stage 13 makes an error and supplies a +1 instead of a 0, its residue will be negative, and stage 12 will supply in all likelihood a -1, and the next stages will supply 0. The output will be $2^{13} - 2^{12} + 0 + 0 + \ldots = 2^{12}$, which is the right value.

Note that for this circuit we started the numbering of the conversion stages from the left (stage 0 is the LSB and stage 13 is the MSB). For the AD9042 we started the numbering from the right (stage 1 was the coarse subconverter and stage 2 is the fine subconverter). We used these (conflicting) notations because they are customary for pipeline and two-stage converters, respectively.

The basic unit of this design is the conversion stage, which is explained in the following chapter.

4.2 1.5-Bit Conversion Stage

The structure and operation of each stage is presented in Figure 4.3. To minimize area and power consumption, each stage contains only one offset-compensated operational amplifier, in addition to capacitors and two latched comparators. As it was mentioned in the previous chapter, the operation of consecutive stages is interleaved and consecutive stages clock on alternate phases, ϕ_1 or ϕ_2.

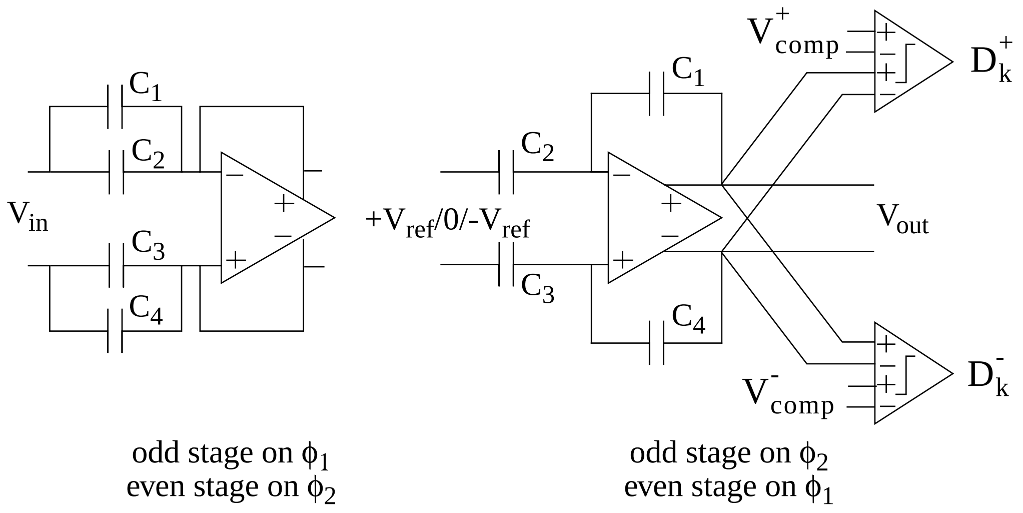


FIGURE 4.3 **Two-Phase Operation of 1.5-Bit Conversion Stage**

Suppose we analyze an odd stage. On ϕ_1 the signal V_{in} is applied at the input and the (equal-sized) capacitors are charged to

$$Q_1 = \frac{CV_{in}}{2} \tag{EQ 4.3}$$

On ϕ_2, the configuration of the switches is changed, and $+V_{ref}$, 0, or $-V_{ref}$ is applied at the input, depending on the result of the conversion in the previous stage. Say we apply $+V_{ref}$. Then, the two capacitors at the input are charged to

$$Q_{21} = \frac{CV_{ref}}{2} \quad \text{(EQ 4.4)}$$

while the two capacitors in the feedback loop are charged to

$$Q_{22} = \frac{CV_{out}}{2} \quad \text{(EQ 4.5)}$$

Charge conservation requires that

$$2Q_1 = Q_{21} + Q_{22} \quad \text{(EQ 4.6)}$$

so that

$$V_{out} = 2V_{in} - V_{ref} \quad \text{(EQ 4.7)}$$

We can derive in a similar fashion formulae for the other cases, when 0 or -V_{ref} is applied at the input, and we can write the stage output as

$$V_{out} = \begin{cases} 2V_{in} - V_{ref} \\ 2V_{in} \\ 2V_{in} + V_{ref} \end{cases} \quad \text{(EQ 4.8)}$$

To allow for possible nonidealities, we will use the following form:

$$V_{out} = \begin{cases} AV_{in} - V_{ref}^{+} \\ AV_{in} \\ AV_{in} + V_{ref}^{-} \end{cases} \quad \text{(EQ 4.9)}$$

where A is nominally 2, and V_{ref}^{+} is nominally equal to V_{ref}^{-}.

Also on ϕ_2, V_{out} is compared by the two comparators to a pair of reference voltages, V^{+}_{comp} and $-V^{-}_{comp}$ (where $V^{+}_{comp} > 0 > -V^{-}_{comp}$) to determine the digital output and implicitly the reference that will be applied at the input of the next stage:

$$D_k^{+} = \begin{cases} 1 & V_{in} \geq V^{+}_{comp} \\ 0 & V_{in} < V^{+}_{comp} \end{cases}$$

$$D_k^{-} = \begin{cases} 0 & V_{in} \geq -V^{-}_{comp} \\ 1 & V_{in} < -V^{-}_{comp} \end{cases} \qquad \text{(EQ 4.10)}$$

$$D_k = D_k^{+} - D_k^{-} = \begin{cases} 1 & V_{in} \geq V^{+}_{comp} \\ 0 & -V^{-}_{comp} \leq V_{in} < V^{+}_{comp} \\ -1 & V_{in} < -V^{-}_{comp} \end{cases}$$

This model will be used in the next chapters to define the residue of the first and second stages.

4.3 Residue Modeling for Partition 1/13

The pipeline architecture presented here seems to be quite different from the two-stage architecture we analyzed in Chapters 2 and 3. However, it can be described using a very similar model and therefore it can be characterized using the same method.

First we divide the pipeline ADC conceptually into two subconverters, and examine the residue produced by the first one and fed to the second one. We can choose the two subconverters to include 1 and 13 conversion stages, respectively, or 2 and 12, or 3 and 11, and so on. We will see later that, as we increase the number of stages in the first subconverter, the estimation accuracy for its parameters decreases due to error accumulation and to the smaller number of bits in the second subconverter (which is used to characterize the first one). Only the first two cases, 1/13 and 2/12, are studied here.

We will begin by analyzing the partition 1/13. (EQ 4.9) can be applied to stage 13:

$$V_{\text{res},13}(V_{in}) = \begin{cases} A_{13}V_{in} - V^{+}_{\text{ref},13} & V_{in} \geq V^{+}_{\text{comp},13} \\ A_{13}V_{in} & -V^{-}_{\text{comp},13} \leq V_{in} < V^{+}_{\text{comp},13} \\ A_{13}V_{in} + V^{-}_{\text{ref},13} & V_{in} < -V^{-}_{\text{comp},13} \end{cases} \quad \text{(EQ 4.11)}$$

which can be rewritten as

$$\mathrm{V}_{\mathrm{res},13}(\mathrm{V}_{\mathrm{in}}) = A_{13}(V_{in} - l_{13,k}) , \quad \mathrm{t}_{13,\mathrm{k}} < \mathrm{V}_{\mathrm{in}} \leq \mathrm{t}_{13,\mathrm{k+1}} \quad k = 0\ldots2 \quad \text{(EQ 4.12)}$$

where

$$l_{13,k} = \left\{-\frac{V^{-}_{ref}}{A_{13}}, 0, \frac{V^{+}_{ref}}{A_{13}}\right\}$$

$$t_{13,\,k} = \left\{-\frac{V_{FS}}{2}, -V^{-}_{comp}, +V^{+}_{comp}, +\frac{V_{FS}}{2}\right\}$$

Note that the equation above is essentially the same as (EQ 4.9) , except for some changes in notation, which might seem unusual for the pipeline architecture. The reason for these changes is that we want to emphasize the similarities between the two-stage architecture from Chapter 2 and the conceptually two-stage (but really 14-stage) architecture introduced in the current chapter.

We also need to define the digital outputs of the first and second subconverters. The former is obviously

$$d_{13} = D_{13} \quad \text{(EQ 4.13)}$$

and the latter is

$$d_{12\text{-}0} = D_{12}2^{12} + D_{11}2^{11} + \ldots + D_1 2^1 + D_0 2^0 \qquad \text{(EQ 4.14)}$$

The overall output is then calculated as:

$$d = d_{13}2^{13} + d_{12\text{-}0} \qquad \text{(EQ 4.15)}$$

The digital output of the first stage can be further written as

$$d_{13}(\mathrm{V_{in}}) = k - 1 , \qquad \mathrm{t_{13,\,k}} < \mathrm{V_{in}} \le \mathrm{t_{13,\,k+1}}, \quad \mathrm{k{=}0\ldots 2} \qquad \text{(EQ 4.16)}$$

At this point we should note that d_{13} and $d_{12\text{-}0}$ cannot be reconstructed from d, because some information is lost when the first two are combined into the last one. For example, if $d = 2^{13} - 1$, we cannot know for certain if $d_{13} = 1$ and $d_{12\text{-}0} = -1$ or $d_{13} = 0$ and $d_{12\text{-}0} = 2^{13} - 1$. Both combinations are theoretically possible, and we cannot know which one actually happened unless we have at least access to D_{13}. Many similar examples can be found.

Based on the equations above we can draw the transfer characteristic of the stage as in Figure 4.4. Just as in Section 2.4, it can be (easily) proved that errors in the reference levels ($l_{13,k}$), gain (A_{13}), and the second subconverter (stages 12 to 0) are relevant for linearity, while errors in the thresholds ($t_{13,k}$) are not.

We introduce now the upper and the lower bounds of the residue, (M_k and m_k, also indicated in Figure 4.4), as the maximum and minimum values that the residue can take on section k (i.e. the part of the residue corresponding to code $d_{13} = k$ - 1), in the absence of noise. These bounds can be expressed in terms of the parameters of the first subconverter using (EQ 4.12) :

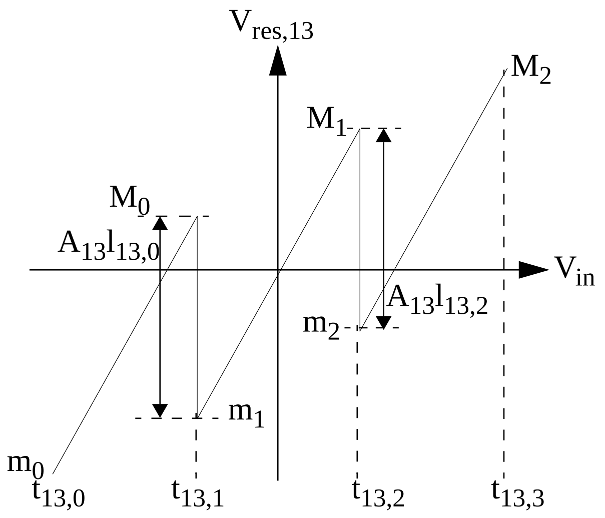


FIGURE 4.4 **Transfer Characteristic of First Stage**

$$M_k = A_{13} \cdot (t_{13,\mathrm{k+1}} - l_{13,k})$$
$$m_k = A_{13} \cdot (t_{13,\mathrm{k}} - l_{13,k}) \qquad \text{(EQ 4.17)}$$

This is a system of 6 equations and 8 unknown variables (A_{13}, $t_{13,k}$, for k = 0...3, and $l_{13,k}$, for k = 0...2). Since we are mostly interested in linearity (and not in an overall offset of the characteristic), we can choose $l_{13,1}$ to be 0, and we also know that $t_{13,0} = -t_{13,3}$, so that the system can be solved (see page 15 for solution details) leading to the following equations:

$$A_{13} = \frac{\sum_{k=0}^{2}(M_k - m_k)}{2LSB_{13}} \qquad \text{(EQ 4.18)}$$

$$t_{13,\mathrm{k+1}} = \frac{M_k - m_k}{A_{13}} + t_{13,\mathrm{k}} \ , \qquad k = 0\ldots2 \qquad \text{(EQ 4.19)}$$

$$l_{13,\mathrm{k+1}} = \frac{M_k - m_{\mathrm{k+1}}}{A_{13}} + l_{13,\mathrm{k}} \ , \qquad k = 0\ldots1 \qquad \text{(EQ 4.20)}$$

(please compare (EQ 4.18) to (EQ 2.16) , (EQ 4.19) to (EQ 2.17) , and (EQ 4.20) to (EQ 2.18)). In conclusion, all the parameters can be calculated if the bounds are known.

If a small gain error is acceptable and only the nonlinearity must be removed, we only need to calculate the $A_2 l_{13,k}$ products, using

$$(A_{13} l_{13,\mathrm{k+1}}) = (A_{13} l_{13,\mathrm{k}}) + M_k - m_{\mathrm{k+1}} \;, \qquad k = 0\ldots1 \tag{EQ 4.21}$$

A similar model for partition 2/12 is also required and will be calculated in the next chapter.

4.4 Residue Modeling for Partition 2/12

We begin this section by defining the digital outputs of the first and second subconverter for partition 2/12, $d_{13\text{-}12}$ and $d_{11\text{-}0}$, respectively:

$$d_{13\text{-}12} = D_{13}2 + D_{12} \tag{EQ 4.22}$$

$$d_{11\text{-}0} = D_{11}2^{11} + D_{10}2^{10} + \ldots + D_1 2^1 + D_0 2^0 \tag{EQ 4.23}$$

and the overall output is

$$d = d_{13\text{-}12}2^{12} + d_{11\text{-}0} \tag{EQ 4.24}$$

Since both D_{13} and D_{12} can take three values, theoretically there are nine possible $\{D_{13}, D_{12}\}$ combinations: $\{\text{-}1,\text{-}1\}$, $\{\text{-}1,0\}$, $\{\text{-}1,+1\}$, $\{0,\text{-}1\}$, $\{0, 0\}$, $\{0,+1\}$, $\{+1,\text{-}1\}$, $\{+1,0\}$, and $\{+1,+1\}$. The corresponding values of d_{13-12} are respectively: -3, -2, -1, -1, 0, +1, +1, +2, +3. Note that some values appear twice because they can result from two different combinations. Thus, according to (EQ 4.22) {-1,+1} is equivalent to {0,-1}. In the real chip, depending on the actual comparator thresholds of the conversion stages, we might have {-1,+1} or {0,-1} or both. It turns out (as it will be shown in Section 5.1) that for this circuit we only have {0,-1}. The combination {-1,+1} never occurs, and the same applies to {+1,-1} which is equivalent to {0,+1}.

As a result, the residue of the first subconverter for this partition will only have seven branches, instead of nine. The parameters of this residue, which is in fact the residue of stage 12, are calculated in the following.

The thresholds of the first subconverter are easy to determine: we already know four of them, namely the two ends of the scale, $-\frac{V_{FS}}{2}$ and $\frac{V_{FS}}{2}$, and the two thresholds of stage 13, namely $-V^{-}_{\text{comp},13}$ and $V^{+}_{\text{comp},13}$. We also have to add the thresholds of stage 12, referred to the input. Since the residue of stage 13 is calculated by subtracting $-V^{-}_{\text{ref},13}$, 0, or $V^{+}_{\text{ref},13}$, from the input and amplifying the result by A_{13}, referring the thresholds of stage 12 to the input means dividing them by A_{13}, and adding $-V^{-}_{\text{ref},13}$, 0, or $V^{+}_{\text{ref},13}$.

The resulting input-referred thresholds are $\frac{-V^{-}_{\text{ref},13}-V^{-}_{\text{comp},12}}{A_{13}}$, $\frac{-V^{-}_{\text{ref},13}+V^{+}_{\text{comp},12}}{A_{13}}$, $\frac{-V^{-}_{\text{comp},12}}{A_{13}}$, $\frac{V^{+}_{\text{comp},12}}{A_{13}}$, $\frac{V^{+}_{\text{ref},13}-V^{-}_{\text{comp},12}}{A_{13}}$, and $\frac{V^{+}_{\text{ref},13}+V^{+}_{\text{comp},12}}{A_{13}}$. However, we have to eliminate from this list two thresholds, $\frac{-V^{-}_{\text{ref},13}+V^{+}_{\text{comp},12}}{A_{13}}$ and $\frac{V^{+}_{\text{ref},13}-V^{-}_{\text{comp},12}}{A_{13}}$, since they correspond to the two combinations that do not occur, $\{-1,+1\}$ and $\{+1,-1\}$.

The remaining thresholds are $\left\{-\frac{V_{FS}}{2}, \frac{-V^{-}_{\text{ref},13}-V^{-}_{\text{comp},12}}{A_{13}}, -V^{-}_{\text{comp},13}, \frac{-V^{-}_{\text{comp},12}}{A_{13}}, \frac{V^{+}_{\text{comp},12}}{A_{13}}, V^{+}_{\text{comp},13}, \frac{V^{+}_{\text{ref},13}+V^{+}_{\text{comp},12}}{A_{13}}, \frac{V_{FS}}{2}\right\}$.

(EQ 4.9) can be applied to stage 12 as follows:

$$V_{\text{res,13-12}} = \begin{cases} A_{12}V_{\text{res,13}} - V^{+}_{\text{ref,12}} & V_{\text{res,13}} \geq V^{+}_{\text{comp,12}} \\ A_{12}V_{\text{res,13}} & -V^{-}_{\text{comp,12}} \leq V_{\text{res,13}} < V^{+}_{\text{comp,12}} \\ A_{12}V_{\text{res,13}} + V^{-}_{\text{ref,12}} & V_{\text{res,13}} < -V^{-}_{\text{comp,12}} \end{cases} \quad \text{(EQ 4.25)}$$

Using (EQ 4.11) and (EQ 4.25) we can calculate the parameters $l_{\text{13-12,k}}$ as the sum of the input-referred contributions of stage 13, $-\frac{V^{-}_{\text{ref,13}}}{A_{13}}$, 0, or $\frac{V^{+}_{\text{ref,13}}}{A_{13}}$, and the input-referred contributions of stage 12, $-\frac{V^{-}_{\text{ref,12}}}{A_{12}A_{13}}$, 0, or $\frac{V^{+}_{\text{ref,12}}}{A_{12}A_{13}}$.

Then, the residue of the first subconverter can be written as

$$\begin{aligned} \mathrm{V}_{\text{res,13-12}}(\mathrm{V}_{\text{in}}) &= A_{\text{13-12}}V_{in} - l_{\text{13-12},k}\,, \\ & t_{\text{13-12,k}} < V_{in} \leq t_{\text{13-12,k+1}},\ k = 0\ldots6 \end{aligned} \quad \text{(EQ 4.26)}$$

where

$$A_{\text{13-12}} = A_{13}A_{12}$$

$$l_{\text{13-12,k}} = \left\{ -\frac{V^{-}_{\text{ref,13}}}{A_{13}} - \frac{V^{-}_{\text{ref,12}}}{A_{12}A_{13}}, -\frac{V^{-}_{\text{ref,13}}}{A_{13}}, -\frac{V^{-}_{\text{ref,12}}}{A_{12}A_{13}}, 0, \frac{V^{+}_{\text{ref,12}}}{A_{12}A_{13}}, \frac{V^{+}_{\text{ref,13}}}{A_{13}}, \frac{V^{+}_{\text{ref,13}}}{A_{13}} + \frac{V^{+}_{\text{ref,12}}}{A_{12}A_{13}} \right\}$$

$$t_{\text{13-12,k}} = \left\{ -\frac{V_{FS}}{2}, \frac{-V^{-}_{\text{ref,13}} - V^{-}_{\text{comp,12}}}{A_{13}}, -V^{-}_{\text{comp,13}}, \frac{-V^{-}_{\text{comp,12}}}{A_{13}}, \frac{V^{+}_{\text{comp,12}}}{A_{13}}, V^{+}_{\text{comp,13}}, \frac{V^{+}_{\text{ref,13}} + V^{+}_{\text{comp,12}}}{A_{13}}, \frac{V_{FS}}{2} \right\}$$

The plot of the residue is shown in Figure 4.5. Incidentally, both residue plots have been made using parameters estimated through histogram analysis from a real chip, so we believe they are accurate representation of the real transfer characteristics. Since the architecture is insensitive to threshold errors, when the chip was designed little attention has been given to setting the thresholds to a specific value. As a result, the characteristic looks very uneven, which is absolutely normal (it does not lead to errors).

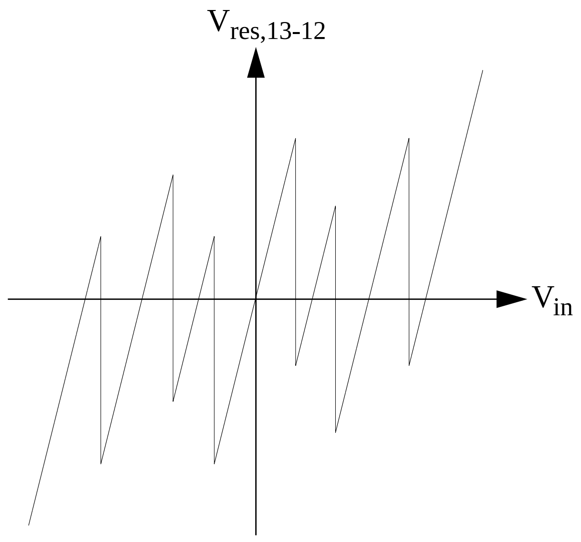


FIGURE 4.5 **Transfer Characteristic of First Two Stages**

The residue bounds can be determined in the same manner as for partition 1/13. This time we choose $l_{13,3}$ to be 0 (since $l_{13,3}$ corresponds to $d_{13\text{-}12} = 0$), and we set $t_{13,0} = -t_{13,7}$. The solution of the system (see page 15 for solution details) relating the residue bounds to the parameters of the first subconverter is:

$$A_{13\text{-}12} = \frac{\sum_{k=0}^{6} (M_k - m_k)}{4LSB_{13-12}} \qquad \text{(EQ 4.27)}$$

$$t_{\text{13-12,k+1}} = \frac{M_k - m_k}{A_{\text{13-12}}} + t_{\text{13-12,k}} \quad , \qquad k = 0 \ldots 6 \tag{EQ 4.28}$$

$$l_{\text{13-12,k+1}} = \frac{M_k - m_{\text{k+1}}}{A_{\text{13-12}}} + l_{\text{13-12,k}} \quad , \qquad k = 0 \ldots 5 \tag{EQ 4.29}$$

Just as in Chapter 3, the units for the analog voltages will be the LSB of the first and second subconverters, for convenience.

4.5 Summary

In this chapter the architecture of a 14-stage pipeline ADC converter was presented. It was shown how this can be conceptually modeled as a two-stage ADC, by dividing the 14 stages between two subconverters. A detailed mathematical description of the residue was provided.

CHAPTER 5

Characterization and Digital Correction of Pipeline A/D Converters

It was shown previously that the operation of a pipeline ADC can be described using a pseudo-two-stage model. In this chapter, we will apply the histogram-based method of characterization to this model and we will show how this method could be turned into an adaptive on-line digital correction mechanism.

5.1 Histograms for Pipeline ADC

We always start the characterization process by sampling the data and building the bidimensional histograms. In Chapter 3 we have used histograms of code density versus the output codes of the coarse and fine subconverters. Likewise, for the pipeline ADC we need histograms of code density versus the output codes of the first and second subconverters.

The first partition analyzed here is 1/13, the output codes of which, d_{13} and $d_{12\text{-}0}$, have been defined in Section 4.3. d_{13} can take the values -1, 0, and +1, and $d_{12\text{-}0}$ can take any value between -8191 and +8191, therefore the overall bidimensional histogram will have three columns and 16383 rows. The ratio of the two dimensions is too large for the plane representation used in Figure 3.5, so that only individual column histograms will be plotted. Typical histograms are shown in Figure 5.1 (this one was obtained from

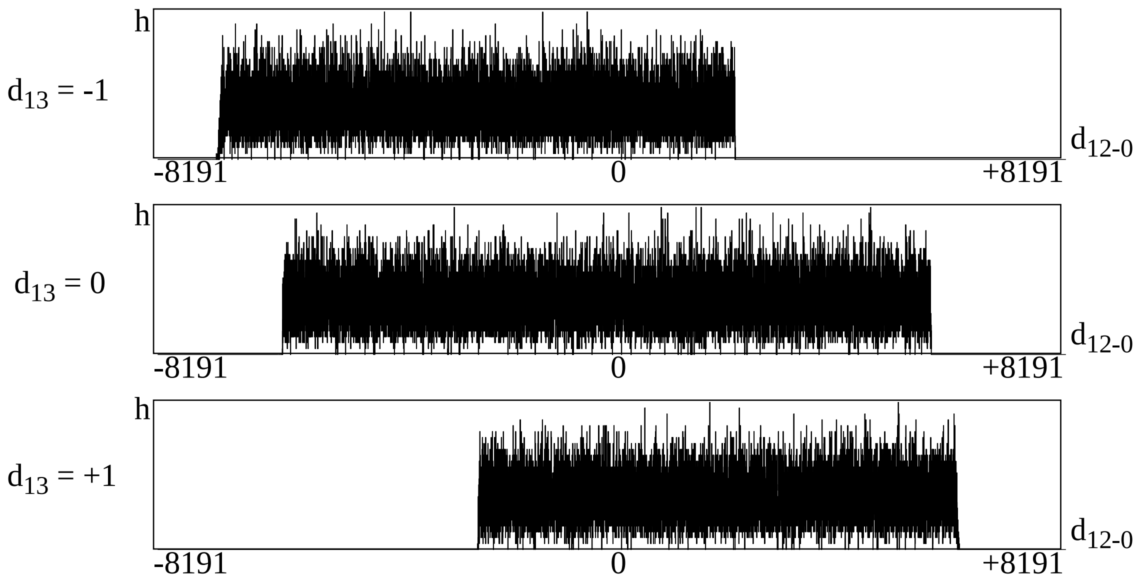


FIGURE 5.1 Histograms for Pipeline ADC with 1/13 Partition

256kSamples of a triangular waveform of peak-to-peak amplitude less than full-scale; note that the first and last histograms do not extend to -8191 or +8191, respectively). These histograms show much more variance than the ones presented in Chapter 3. This is due to the smaller length of the sample sets, and also due to the fact that this ADC has more bits and therefore larger sets would be necessary to obtain the same bin height. It was found that satisfactory results can be obtained even with these small sample sets, since error accumulation is not a problem with the pipeline ADC (see Section 5.4).

For partition 2/12 the codes of the first and second subconverter have been defined in Section 4.4 as d_{13-12} and d_{11-0}. The histograms for the same triangular waveform are presented in Figure 5.2. Since the number of possible combinations of the first two bits is nine, we could theoretically plot nine histograms, but two of them are empty and have not been included here. The reason is that the corresponding $\{D_{13}, D_{12}\}$ combinations, $\{-1,+1\}$ and $\{+1,-1\}$, never happen, proving that we were right in assuming that the residue only has seven sections.

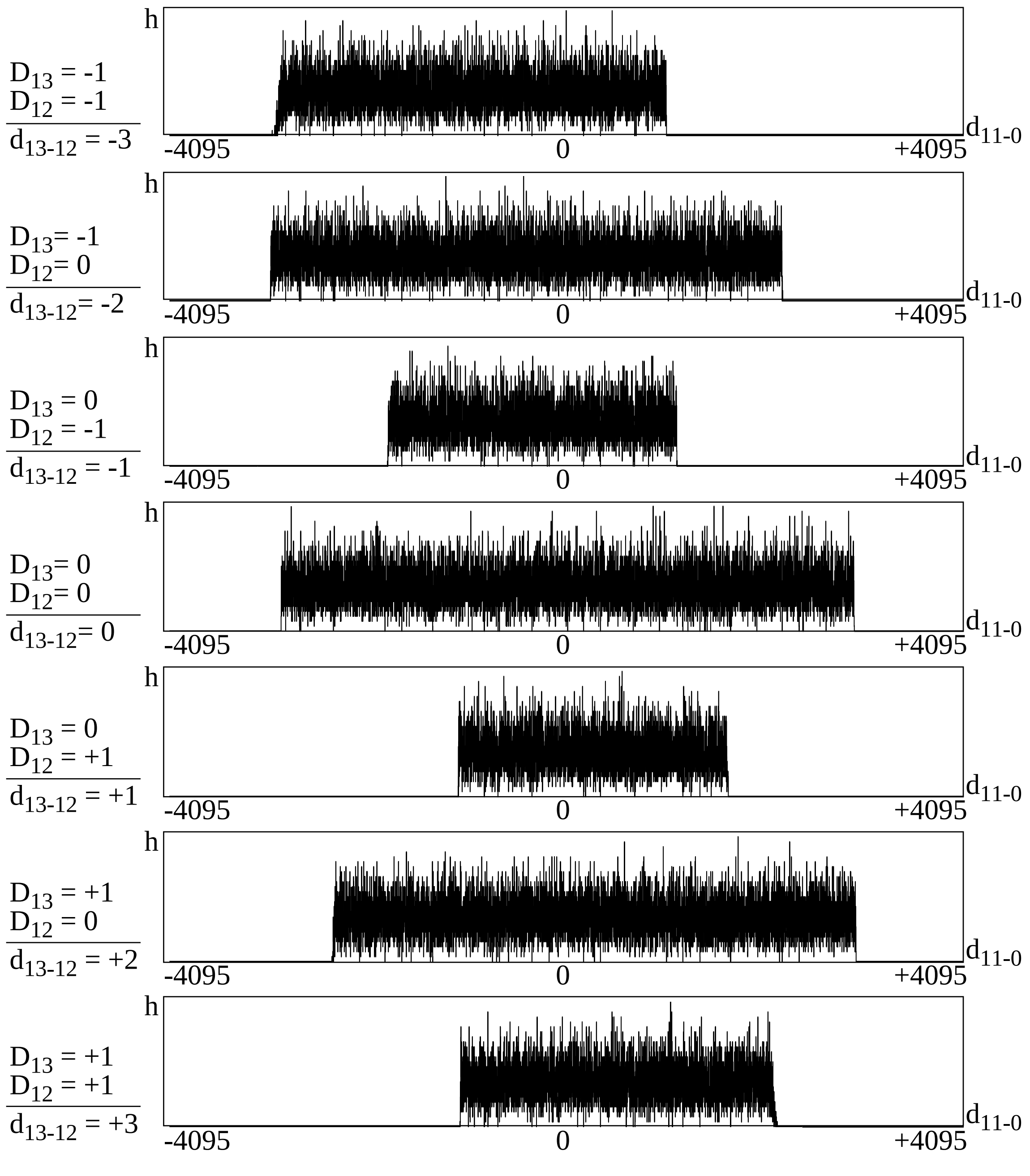


FIGURE 5.2 **Histogram for Pipeline ADC with 2/12 Partition**

An interesting observation is that for this ADC, under certain conditions of bias, all nine combinations occur. The explanation is simple: the bias currents affect the thresholds of the comparators, which in turn determine the limits of each histogram. Sometimes the thresholds that define a section of the residue get too close to each other or even switch sides, which means that the corresponding combination will never occur. This is not a problem in any way: all the possible values of the output code d occur and the linearity of the converter is not affected.

The parameters of the first subconverter, gain (A), thresholds (t_k), and normalized output levels (l_k), can be calculated from the bounds of these histograms, which are estimated in the following section.

5.2 Bound Estimation

For bound estimation we could simply use the same method as for the AD9042. Nevertheless, we will present here a less accurate but simpler method, which has the advantage that it can actually be implemented in hardware as a digital correction algorithm. In Section 5.4, after explaining the new method, we will compare it to the other one. The point that we are trying to make is that the same idea, histogram-based characterization, can be customized to particular requirements of complexity and accuracy.

One of the major difficulties in estimating the bounds of the histograms is the nonuniformity in the pdf of the input signal. In Section 3.7 we solved this problem by normalizing the histogram with respect to the estimated pdf. Another possible solution is to use for calculations only very small regions of the histogram. For instance, we can assume that the pdf is nearly uniform over intervals of $\Delta = 64\ LSB$, which represent less than 0.2% of the full-scale of the pipeline ADC ($FS = 32767\ LSB$).

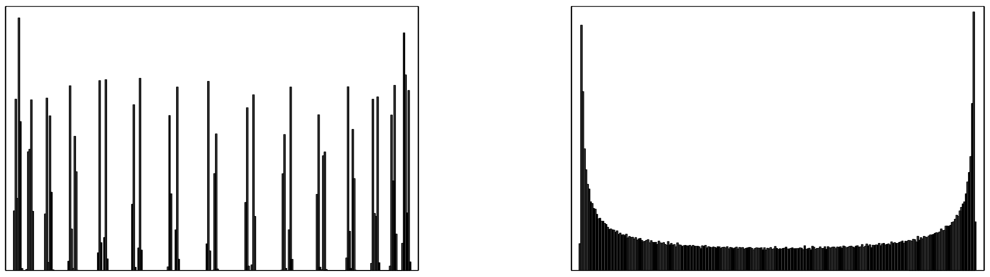


FIGURE 5.3 **Nonuniform and Locally Uniform Histograms**

Of course, there are classes of signals for which this assumption is not true. Suppose the input signal is a sinewave, or any periodic signal whose frequency can be expressed as $f_{signal} = \frac{m}{n} f_{clock}$ where m and n are small mutually prime integers. Then, the histograms will be just a collection of spikes; the bins between the spikes will be empty and in the vicinity of the spikes the pdf will be highly nonuniform (Figure 5.3 left). However, any signal that does not satisfy this condition and covers all the $64 \cdot LSB$ intervals (Figure 5.3 right) where calculations are made should lead to satisfactory results.

Another difficulty is the presence of noise which makes the transition (between the region where codes occur and the region where codes do not occur) extend over an interval of about 16 *LSB*. MATLAB simulation plots showing the shape of the transition region for different noise variances σ_n can be seen in Figure 5.4.

As we might expect, the transition region becomes wider as the noise level increases. Nevertheless, for a fixed arbitrary reference (r_k^+, an integer) in the flat region (see Figure 5.5) and a fixed upper bound (M_k), the area to the right of the reference is the same, regardless of the noise level. Hence we could define a statistic based on this area to estimate M_k, by noting that the area added by noise to the right of M_k is equal to the area removed by noise to the left of M_k:

$$\hat{M}_k = r_k^+ + \frac{\sum_{j=r_k^+ + 1}^{8191} h_j}{\bar{\hat{h}}_k^+} + 0.5 \qquad \textbf{(EQ 5.1)}$$

In the same manner an estimate of the lower bound can be defined as

$$\hat{m}_k = r_k^- - \frac{\sum_{j=-8191}^{r_k^- - 1} h_j}{\bar{\hat{h}}_k^-} - 0.5 \qquad \textbf{(EQ 5.2)}$$

Note that in the equations above we used two different average bin heights: one for the upper bound ($\bar{\hat{h}}_k^+$) and one for the lower bound ($\bar{\hat{h}}_k^-$). Both are calculated in intervals of length Δ in the immediate vicinities of the corresponding transition regions, to ensure that the pdf nonuniformity of the input signal does not affect the accuracy of the estimations:

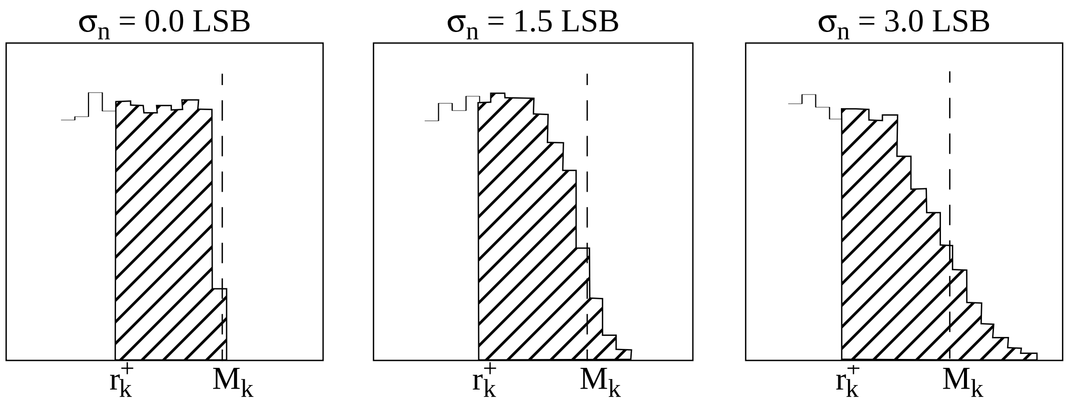


FIGURE 5.4 **Effect of Noise Level on Shape of Histogram**

$$\hat{\bar{h}}_k^- = \frac{\sum_{j=r_k^-}^{r_k^-+\Delta-1} h_j}{\Delta}$$ (EQ 5.3)

$$\hat{\bar{h}}_k^+ = \frac{\sum_{j=r_k^+-\Delta+1}^{r_k^+} h_j}{\Delta}$$

The regions used for estimation may be seen in Figure 5.5. The average values are calculated in the reference regions and the sums at the numerator of the middle term in (EQ 5.1) and (EQ 5.2) in the transition regions.

Let us verify that (EQ 5.1) gives the right result in the noiseless case. We can express M_k as $M_k = r_k^+ + n + \alpha + 0.5$, where n is an integer and $0 \le \alpha < 1$ (any number can be expressed in this way). The thresholds of the second subconverter are located at $i + 0.5$, where i is an integer between -8192 and +8191. We are only interested in the bins from $r_k^+ + 1$ to +8191, since these are the ones summed in (EQ 5.1) . Out of these bins, those for which both thresholds (each bin is defined by two thresholds and the bin is assumed to be located in the middle of them) are less than M_k will be full, i.e. their height will be approximately $\hat{\bar{h}}_k^+$. The index of these bins must therefore meet the following conditions:

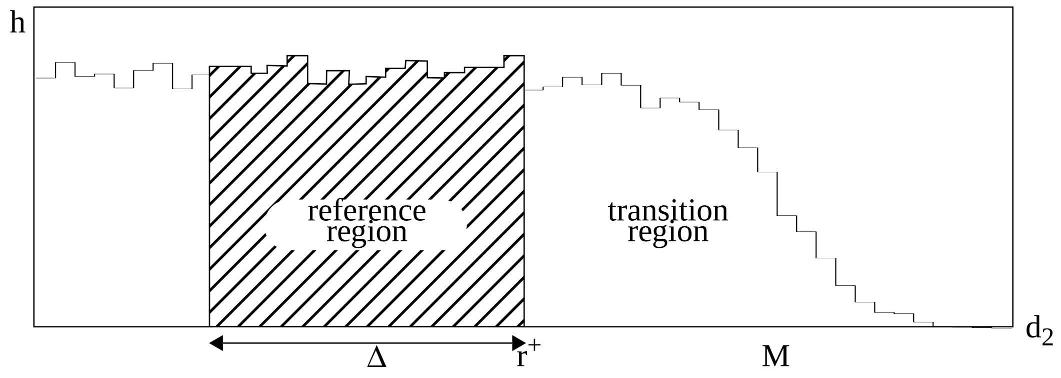


FIGURE 5.5 Regions Used in Estimation

$$\overset{+}{r_k} \le i \qquad \& \qquad i + 1 + 0.5 < \overset{+}{r_k} + n + \alpha + 0.5 \tag{EQ 5.4}$$

It is easy to see that there are n full bins (i.e. n integer values of i that satisfy the equation above).

Those bins for which both thresholds are greater than M_k will be empty. The bin with one threshold less than M_k and one greater than M_k will be partially full. This bin is the one defined by the thresholds $\overset{+}{r_k} + n + 0.5$ and $\overset{+}{r_k} + (n+1) + 0.5$, since

$$\overset{+}{r_k} + n + 0.5 \le \underbrace{\overset{+}{r_k} + n + \alpha + 0.5}_{M_k} < \overset{+}{r_k} + (n+1) + 0.5 \tag{EQ 5.5}$$

and its height will be $\alpha\hat{\overline{h}}{}_k^{+}$. Now we can apply (EQ 5.1) :

$$\hat{M}_k = \overset{+}{r_k} + \frac{n\hat{\overline{h}}{}_k^{+} + \alpha\hat{\overline{h}}{}_k^{+}}{\hat{\overline{h}}{}_k^{+}} + 0.5 = \overset{+}{r_k} + n + \alpha + 0.5 = M_k \tag{EQ 5.6}$$

which is the right result.

This approach also solves the last major difficulty in estimating the histogram bounds, to wit the errors introduced by the nonlinearity of the second subconverter, which are averaged out over the reference region and the transition region. This solution is not as accurate as the one demonstrated for the two-stage architecture, but it is extremely simple, and we will see later that it could be implemented with a little digital circuitry.

The bounds can now be calculated from (EQ 5.1) , (EQ 5.2) , and (EQ 5.3) . The results are presented in Table 5.1 and Table 5.2, for the histograms shown in Figure 5.1 and Figure 5.2.

TABLE 5.1 Histogram Bounds for Partition 1/13

Bound	m_1	m_2	M_0	M_1
Value ($LSB_{12\text{-}0}$)	-5940.5	-2411.3	2220.8	5747.7

TABLE 5.2 Histogram Bounds for Partition 2/12

Bound	m_1	m_2	m_3	m_4	m_5	m_6
Value ($LSB_{11\text{-}0}$)	-3051.1	-1839.6	-2945.4	-1115.8	-2403.9	-1097.3
Bound	M_0	M_1	M_2	M_3	M_4	M_5
Value ($LSB_{11\text{-}0}$)	1029.7	2224.5	1135.0	2963.5	1656.9	2979.1

Note that the first lower bound and the last higher bound are always known. m_0 and M_2 for partition 1/13 are -8191.5 and +8191.5, and m_0 and M_6 for partition 2/12 are -4095.5 and +4095.5, respectively. This is because the lowest / highest possible codes of the second subconverter are ± 8191 and ± 4095, respectively, and 0.5 *LSB* must be added or subtracted since the thresholds that define a histogram bin are located 0.5 *LSB* to the left and to the right of the bin position.

5.3 Component Characterization

We have seen the relationship between the parameters of the first stage and the bounds of the residue in Section 4.3 for partition 1/13, and the bounds were estimated in Section 5.2. The next step is to calculate the parameters, using (EQ 4.18) , (EQ 4.19) , and (EQ 4.20) , in which the actual values of the bounds are replaced by the estimated values.

The parameters can be easily calculated:

$$\hat{A}_{13} = 1.9960$$
$$\hat{t}_{13,k} = \{-8191.5, -2974.6, 2879.2, 8191.5\} \qquad \text{(EQ 5.7)}$$
$$\hat{l}_{13,k} = \{-4088.86, 0, 4087.71\}$$

These numbers must be translated into physical quantities for a reality check. The model parameters used in the previous equation were defined in Section 4.3 in terms of physical parameters, and the correspondence between estimated $(\hat{\ })$ and ideal $(\tilde{\ })$ values can be readily derived:

$$\hat{V}_{comp} = \frac{\hat{t}_{13,k}}{8191.5}\frac{\tilde{V}_{FS}}{2} \qquad \text{(EQ 5.8)}$$

$$\hat{V}_{ref} = \frac{\hat{A}_{13}\hat{l}_{13,k}}{8192}\tilde{V}_{ref} \qquad \text{(EQ 5.9)}$$

where we assume that there are no overall gain and offset errors (we have no other option, since the proposed characterization method is insensitive to such errors and they are anyway less important than the nonlinearity errors we are trying to estimate). Replacing (EQ 5.7) in (EQ 5.8) and (EQ 5.9) , we obtain:

$$\hat{A}_{13} = 1.9960$$
$$\hat{V}^{-}_{\text{comp},13} = 0.3631\frac{\tilde{V}_{FS}}{2} \qquad \hat{V}^{+}_{\text{comp},13} = 0.3515\frac{\tilde{V}_{FS}}{2} \qquad \text{(EQ 5.10)}$$
$$\hat{V}^{-}_{ref,13} = 0.9963\ \tilde{V}^{-}_{ref,13} \qquad \hat{V}^{+}_{ref,13} = 0.9960\ \tilde{V}^{+}_{ref,13}$$

In order to interpret these results, we must remember that real circuits are plagued by various offsets and mismatches.

Threshold errors are not important, since they do not affect the overall linearity. The thresholds of the comparators (V^{-}_{comp}, V^{+}_{comp}) are in this case set by a deliberate comparator offset (the transistors in the input differential pair of the comparator have different widths). Gain and output level errors, on the other hand, can affect the shape of the overall characteristic.

For this specific BiCMOS ADC, the most likely source of nonlinearity is capacitor mismatch. From Figure 4.3 it can be easily proved that

$$V_{out} = \left(1 + \frac{C_2 + C_3}{C_1 + C_4}\right)V_{in} - \left(\frac{C_2 + C_3}{C_1 + C_4}\right)V_{ref} \qquad \textbf{(EQ 5.11)}$$

In the ideal case ($C_1 = C_2 = C_3 = C_4$) this simplifies to

$$V_{out} = 2V_{in} - V_{ref} \qquad \textbf{(EQ 5.12)}$$

but in real circuits we can expect to have some mismatches between capacitors (around 0.1%-1%), which will lead to a gain (A_{13}) error, and an output level ($V^{-}_{ref,13}$, $V^{+}_{ref,13}$) error, due to the first and the second terms, respectively, in (EQ 5.11) .

Another possible problem is the op amp gain. It can be shown that (EQ 5.12) becomes for finite gain

$$V_{out} = \frac{1}{1 + \frac{2}{A}}(2V_{in} - V_{ref}) \qquad \textbf{(EQ 5.13)}$$

However, for the gain to cause the reduction in the SFDR measured in the real chip, it would have to be very low (about 500 according to MATLAB simulations). The op amp was designed to have a much higher gain (>10,000 according to HSpice simulations), so that, in all likelihood, its contribution to the overall nonlinearity is smaller than that due to capacitor mismatches.

Unfortunately, there are nonidealities that are impossible to detect with this characterization method, for instance nonlinear capacitors, nonlinear op amps, and input-dependent charge injection and sampling times. This is because the model we are using for the basic conversion stage does not take into account these phenomena. A more elaborate model could take them into account but the analysis would be much more difficult.

A similar analysis can be performed for partition 2/12. The equations to be used are (EQ 4.27) , (EQ 4.28) , and (EQ 4.29) . The resulting parameters are

$$\hat{A}_{13\text{-}12} = 3.9834$$

$$\hat{t}_{13,\mathrm{k}} = \{-4095.5, -2809.48, -1485.07, -738.30, 745.07 \\ 1441.15, 2792.50, 4095.5\} \quad \text{(EQ 5.14)}$$

$$\hat{l}_{13,\,k} = \{-3069.09, -2044.63, -1024.35, 0, 1024.07 \\ 2043.52, 3066.86\}$$

The correspondence between these parameters and physical parameters was presented in (EQ 4.26) . Equations similar to (EQ 5.8) and (EQ 5.9) can be employed, with 4095.5 (4096) instead of 8191.5 (8192), and we can use the value of A_{13} previously estimated. The end results are:

$$A_{13\text{-}12} = 3.9834 \qquad A_{12} = 1.9957$$

$$\hat{V}^{-}_{\mathrm{comp},13} = 0.3626\frac{\tilde{V}_{FS}}{2} \qquad \hat{V}^{+}_{\mathrm{comp},13} = 0.3518\frac{\tilde{V}_{FS}}{2}$$

$$\hat{V}^{-}_{\mathrm{comp},12} = 0.3598\frac{\tilde{V}_{FS}}{2} \qquad \hat{V}^{+}_{\mathrm{comp},12} = 0.3631\frac{\tilde{V}_{FS}}{2} \quad \text{(EQ 5.15)}$$

$$\hat{V}^{-}_{ref,13} = 0.9964\tilde{V}^{-}_{ref,13} \qquad \hat{V}^{+}_{ref,13} = 0.9958\tilde{V}^{+}_{ref,13}$$

$$\hat{V}^{-}_{ref,12} = 0.9962\tilde{V}^{-}_{ref,12} \qquad \hat{V}^{+}_{ref,12} = 0.9959\tilde{V}^{+}_{ref,12}$$

Please note the relatively good agreement between these results and those obtained for the partition 1/13 (0.01% of $\tilde{V}^{+/-}_{ref,13}$ for $\hat{V}^{+/-}_{ref,13}$, and 0.05% of $\frac{\hat{V}_{FS}}{2}$ for $\hat{V}^{+/-}_{\mathrm{comp},13}$).

5.4 Discussion of Characterization Method

The method we used for the pipeline ADC is quite different from the one used for the two-stage ADC. A legitimate question is why we did not use the same method to characterize both ADCs. This issue will be addressed in the present section.

The two methods are essentially the same. Both use bidimensional histograms and go through bound estimation and parameter calculation. The latter step is necessarily different from architecture to architecture, but for a given circuit design there is only one way it can be done: by solving the system of equations that relates the parameters to the bounds. This is straightforward and should not pose any problems.

On the other hand, different methods can be imagined for the extraction of the bounds from histograms (this includes all the steps from normalization to bound estimation). Some of them are more elaborate and also more accurate, like the one presented in Chapter 3, while others are simpler but not as accurate, like the one presented in Section 5.2. Probably the best approach to finding the optimum trade-off between accuracy and algorithm complexity is to start with the most accurate method and then check which steps can be discarded without significant loss of accuracy.

If the input signal has a uniform distribution, or is at least locally uniform in the vicinity of the bounds, we may skip the histogram normalization step (see Section 3.5), and calculate the average bin height in a reference region next to the bound.

For the method explained in Chapter 3, fine ADC characterization would be the next step. For pipeline ADC, this step is more or less irrelevant, because all 14 conversion stages are identical, so that we only need to characterize one or two to get a feel for what might be the main cause of nonlinearity.

Histogram smoothing may also be skipped if error accumulation is not a problem. For the two-stage ADC analyzed in Chapters 2 and 3, this is a serious concern, because there are 63 thresholds and 63 output levels to calculate progressively, and an error in any of these would show up in all the following parameters. For the pipeline ADC, there are only two thresholds and two output levels to calculate for partition 1/13, and six of each for partition 2/12 with only moderate error accumulation, but for higher-order partitions, error accumulation may become a problem. We may also skip this step if the integral nonlinearity of the fine subconverter is small enough in the vicinity of the bounds.

The last step before parameter calculation is the actual bound estimation (the previous steps were just preparation). Two approaches have been presented here: one used the fact that the bin-height at the bound should be half of the average bin height in the flat region, and the other was based on the observation that, regardless of the noise level, the area of the transition region delimited by a fixed reference in the flat region is constant. If the number of samples is large enough, the results provided by these two approaches are almost identical. On the other hand, as we will see in Section 5.6, the latter approach is more amenable to hardware implementation. It is hard to imagine that the former could actually be turned into a circuit, because it is very complicated: it requires LMS and polynomial equation solving. Of course, in the case of diagnosis, the computations are performed in software so that it does not matter which one is used: either of them takes only a few seconds (consequently the version presented in Chapter 3 should be used, because it is more accurate).

We can now answer the question we asked at the beginning of the chapter. The first method is general and can be used for any circuit, and in the case of the pipeline ADC it provides very similar results to the second method. On the contrary, the second method leads to gross errors when applied to the AD9042, mainly because the histograms are not normalized. With normalization, or with uniform (over the whole range) inputs, the errors are much reduced. The test engineer will have to decide, from case to case, if a simpler method can be used instead of the one presented in Chapter 3.

The main reason why we presented the second method of bound estimation is that it is very easy to implement, as it will be shown in the following sections.

5.5 Prior Art of Digital Correction

From characterization to digital correction there is only one small step. If we know the errors with sufficient accuracy, we should be able to make corrections for them, as long as they do not cause deadbands (in this circuit they do not). It was proved in Section 2.5 that if we use

$$d = \frac{A_{13} l_{13,k}}{LSB_{13}} + d_{12-0} \tag{EQ 5.16}$$

instead of

$$d = 2^{13} d_{13} + d_{12-0} \tag{EQ 5.17}$$

the characteristic becomes linear (assuming an ideal second subconverter), although the gain may still be off. Consequently, for partition 1/13 we could use a small look-up table to store $(A_{13} l_{13,0})$ and $(A_{13} l_{13,2})$ (note that $(A_{13} l_{13,1})$ is always zero). d_{13} would address the table, the output of which would be added to $d_{12\text{-}0}$.

The idea of using look-up tables is by no means new. What is new is the manner of calculating the values in the look-up table. So far, the most widespread technique has been to measure off-line the correction coefficients by reconfiguring the converter by means of switches [Cra90], [Kar93], [Kar96], [Lee94], [Oli96], [Soe95], and [Suz90]. The analog input of conversion stage j is set to zero, and its digital output is set to *k*. As a result, the input to the next stage, *j*-1, is exactly ($A_j l_{j,k}$) and can be measured by the stages *j*-1 to 0. The procedure starts from a low-order stage and goes on until the MSB stage. For better precision extra stages beyond the LSB can be provided, or a circular structure may be formed, so as to correct errors in stage 0 stage using stages MSB to 1, errors in stage 1 using stages 0 and MSB to 2, and so on. It is also recommended to measure the coefficients several times, and average the results, in order to remove the effect of noise. The only implementation known to us that can be used on-line is based on an algorithm that skips conversions randomly to do some correction and uses interpolation to estimate the value that was skipped [Moo97]. However, this implementation requires a reduced frequency range for the input signal, to ensure that the interpolation results are accurate.

Another solution is to compare the outputs of the converter under test with the outputs of a very accurate reference converter for the same input signal, [Rib91], [Tie90]. The reference ADC would provide the coefficients for the look-up table.

A more elaborate approach is phase-plane compensation [Hum92a], [Hum92b], [Hum93], [Hum96a], [Hum96b], [Iro91], [Iro96], [Mou89], and [Reb87]. This requires that clean sinewaves should be applied at the input of the ADC and uses sine-fitting to estimate the errors for each possible code. Dynamic errors can be taken into account, by considering the error coefficients dependent on both the current state and the previous state (or the current state and the slope). The method is very accurate, but also very laborious and time consuming. In its more elaborate and efficient implementations, a large number of signals of different frequencies and amplitudes must be sampled in order to cover as much as possible of the phase plane (i.e. the current-state/slope plane).

Another approach is based on the analysis of the fft of the digital output for a sinusoidal input. The magnitude of the harmonics can be used to calculate the coefficients of a nonlinear (polynomial) model [Eva86] or even a nonlinear dynamic model [Eva90]. The values in the look-up tables are then calculated from this model.

In the next sections we propose a new adaptive digital correction algorithm, derived from the characterization method presented before, and we discuss its advantages and shortcomings.

5.6 Adaptive Digital Correction

The formulae used for the characterization of the pipeline ADC are simple enough to warrant an investigation into the possibility of turning it into an adaptive algorithm.

Suppose we have to measure the lower bound of the histogram. Let r_k^- be an adjustable reference level. We define a frame containing two windows, as indicated in Figure 5.6. Window A extends down from r_k^- to the bottom of the scale, while window B extends from r_k^- to $r_k^-+\Delta$, and thus has fixed length (Δ).

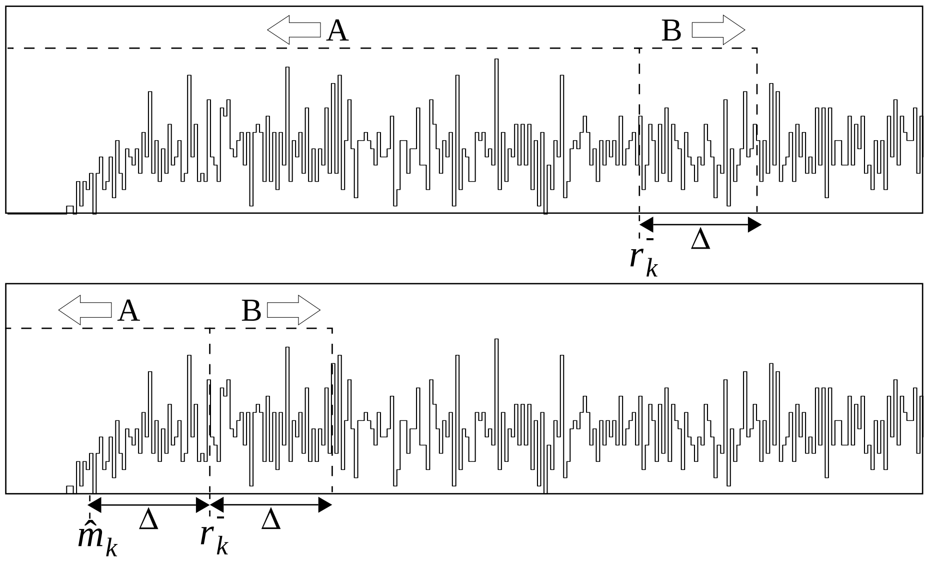


FIGURE 5.6 Finding the Histogram Bounds

For every code occurrence in window A, the frame is to be moved a small step (δ) to the left, and for every code occurrence in window B, the frame is to be moved (by δ) to the right. Mathematically, if *j* is the current code, we can write:

$$r_k^- = \begin{cases} r_k^- - \delta, & j < r_k^- \\ r_k^- + \delta, & r_k^- < j < r_k^- + \Delta \end{cases} \qquad \text{(EQ 5.18)}$$

r_k^- must be initially located somewhere in the “flat” region. Since the frequency of codes occurring in window A is higher than in B, the frame will tend to move left, and it will stop only when the frequency is equal in the two windows. From that moment, r_k^- will simply oscillate around a fixed position, i.e. it will be locked to it. We can calculate this position by noting that after convergence

$$\sum_{j=-8191}^{r_k^-} h_j \cong \sum_{j=r_k^-}^{r_k^-+\Delta} h_j \qquad \text{(EQ 5.19)}$$

where it is assumed that *j* only takes the integer values in the intervals indicated by the limits of the sums. Let us estimate $\hat{m}_k$ using (EQ 5.2) and (EQ 5.3) :

$$\begin{aligned}\hat{m}_k &= r_k^- - \frac{\sum_{j=-8191}^{r_k^-} h_j}{\hat{\bar{h}}} - 0.5 \\ &= r_k^- - \frac{\sum_{j=-8191}^{r_k^-} h_j}{\left(\sum_{j=r_k^-}^{r_k^-+\Delta} h_j / \Delta\right)} - 0.5 \\ &= r_k^- - \Delta - 0.5\end{aligned} \quad \text{(EQ 5.20)}$$

Since Δ is fixed and r_k^- can be found through the algorithm presented above, it follows that (EQ 5.20) can be used as a statistic for the lower bound.

Needless to say, the algorithm can be applied for the higher bound as well, by placing the variable-length window at the right of the fixed-length window. The estimate of the higher bound will be

$$\hat{M}_k = r_k^+ + \Delta + 0.5 \quad \text{(EQ 5.21)}$$

The algorithm converges indeed to the right values, as indicated in Figure 5.7, where the estimates of M_1, M_0, m_2, and m_1 (from top to bottom) have been plotted versus time (for a sampling frequency of 200kHz). At first, all the estimates start from the middle of the range, and then they move toward the bounds.

The adjustment step δ was chosen so that the randomness in the estimates after convergence has been attained be less than 2 *LSB*. It is apparent that it takes quite a long time (10 seconds) for the estimates to reach the desired values. This time would be shorter if we provided the algorithm with a starting point closer to the actual values.

If this time is to be reduced, δ should be increased at the beginning, when r_k^+ and r_k^- are far from the bounds of the histograms, and reduced after convergence has been achieved, to minimize the random noise in the estimates. These opposing requirements could be met by a mechanism of “gear-shifting”, to adapt the step.

The gear shifting could be implemented, for example, using a shift register. Each time a code occurs in either window A or window B, 0 or 1, respectively, are applied at the input and the register is shifted one position. The numbers of 0’s and 1’s are constantly compared to each other, and if the difference between them exceeds a given limit, the step δ is increased. An illustration of the operation of the algorithm is provided in Figure 5.8, using steps of 1 $LSB_{12\text{-}0}$ and 1/8 $LSB_{12\text{-}0}$. A significant improvement in speed with no increase in the random noise after convergence can be noticed.

Other solutions are certainly possible. For instance, a one-speed implementation followed by a low-pass digital filter could provide a very smooth output.

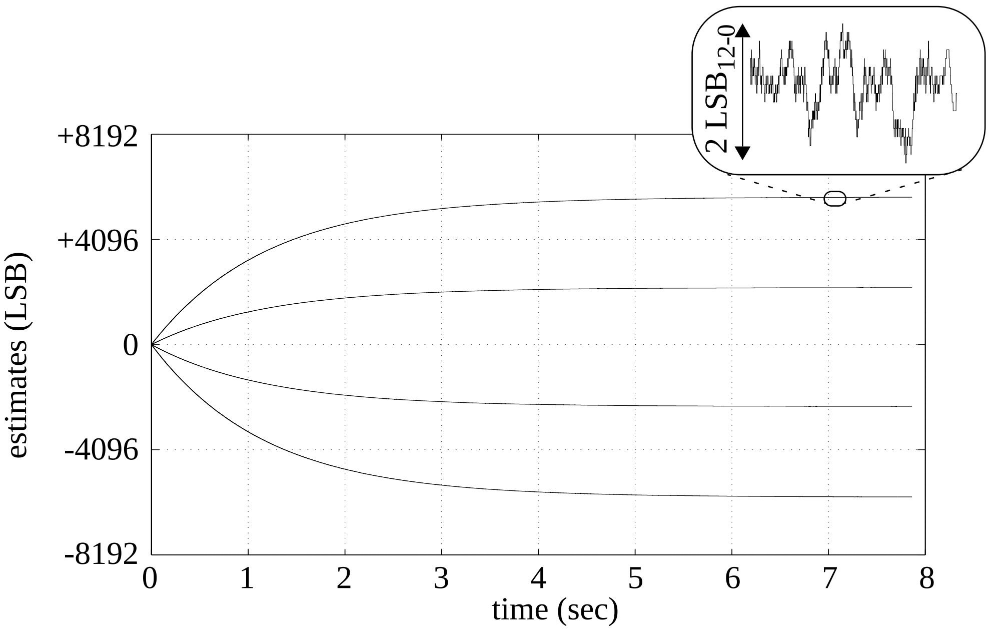


FIGURE 5.7 Algorithm Convergence

Once the bound are estimated, (EQ 4.21) gives us the two parameters we need for correction:

$$(A_{13} l_{13,0}) = -\hat{M}_0 + \hat{m}_1$$
$$(A_{13} l_{13,2}) = \hat{M}_1 - \hat{m}_2 \qquad \text{(EQ 5.22)}$$

Since δ may be less than $1LSB_{12-0}$, the two parameters calculated above and also the corrected digital output (EQ 5.16) will have a fractional part. For instance, if the smallest δ we use is 1/8 $LSB_{12\text{-}0}$, three extra bits would have to be provided in addition to the number of bits used for the uncorrected digital output. Alternatively, digital filtering can be used to round the output without significant loss of accuracy.

Although a circuit implementation is not provided here, we believe this would require little digital circuitry. A minimal circuit to track one bound would include:

- one up/down counter for r_k^+ (or r_k^-) and one adder for $\hat{M}_k$ (or $\hat{m}_k$),
- two comparators, to check whether the current code falls within one of the two windows, and

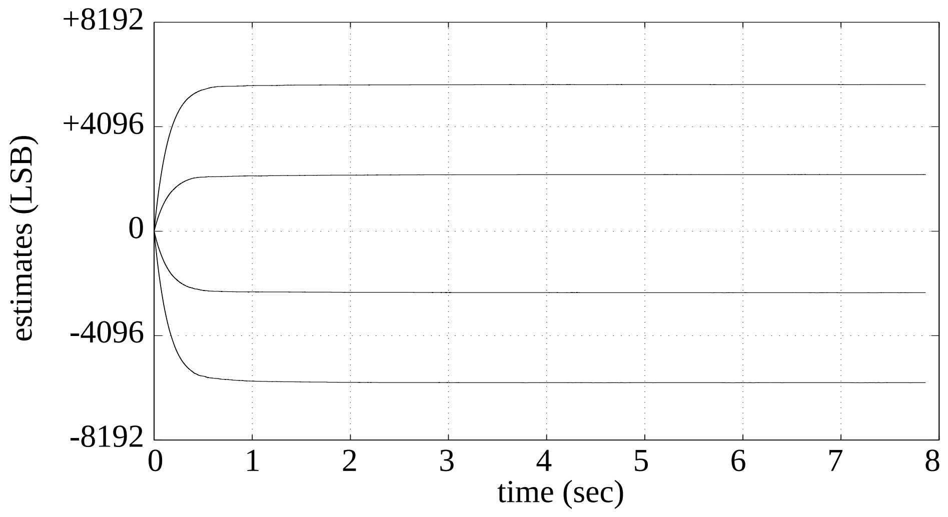


FIGURE 5.8 Algorithm Convergence with Two Steps

- logic, to determine the direction of counting, up or down.

The algorithm was tested with different calibration signals (triangular and sinusoidal waveforms) of different frequencies (between DC and $f_{clock}/2$), and different test signals (sinusoidal waveforms), in the same range of frequencies. If both the calibration and the test frequency are less than $f_{clock}/4$, the distortion harmonics are reduced by more than 6dB (Figure 5.9).

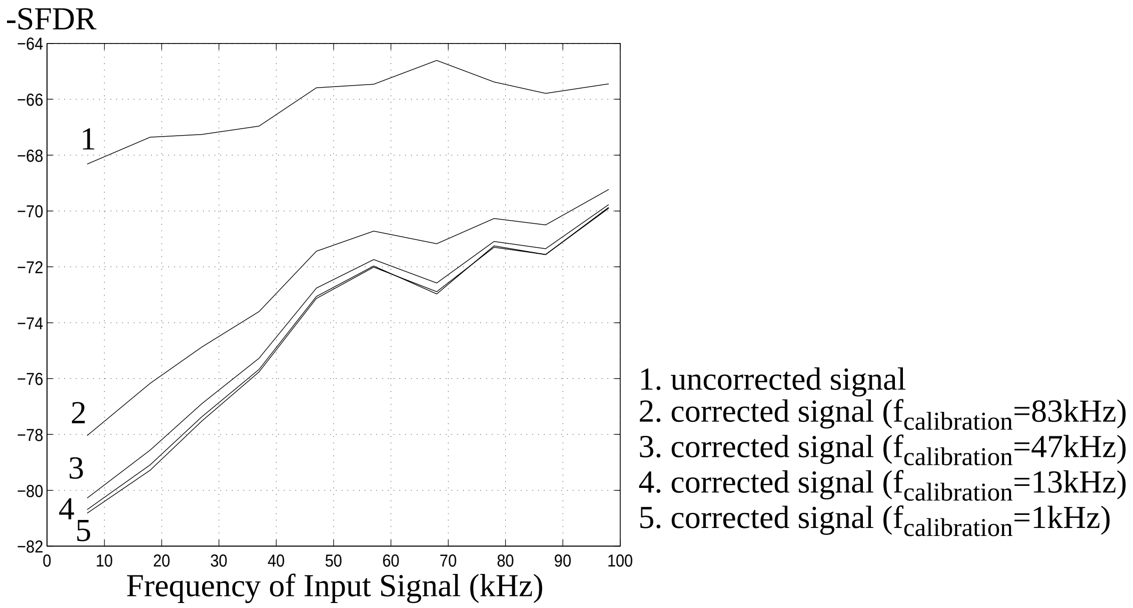


FIGURE 5.9 Improvement in SFDR with Digital Correction

Some more improvement (1 to 2dB at low frequencies) can be obtained if we use the algorithm presented in Chapter 3 instead of the adaptive algorithm presented here. However, the former is more complicated and the calculations have to be performed off-line on a computer.

The performance of the algorithm, just like the performance of the uncorrected converter, seems to degrade at higher frequencies. This means that other types of nonlinearity are present, aside from those included in the model developed in Section 4.3, only these are dynamic (either frequency- or slope-dependent). Dynamic errors usually result from input-dependent sampling instants, which may be due to voltage-dependent “on” resistance of MOS switches, or misalignment of complementary clock signals. Another possible explanation would be input-dependent charge injection. These phenomena are unfortunately difficult to model and investigate.

So far we have only talked about digital correction for partition 1/13. The same idea can also be applied to partition 2/12. The corrected digital output would be in this case

$$d = \frac{A_{13\text{-}12} l_{13\text{-}12,k}}{LSB_{13\text{-}12}} + d_{11\text{-}0} \quad \text{(EQ 5.23)}$$

However, no further improvement was obtained for this partition. On the contrary, the performance is slightly worse, except at low frequencies (Figure 5.10), possibly because of error accumulation.

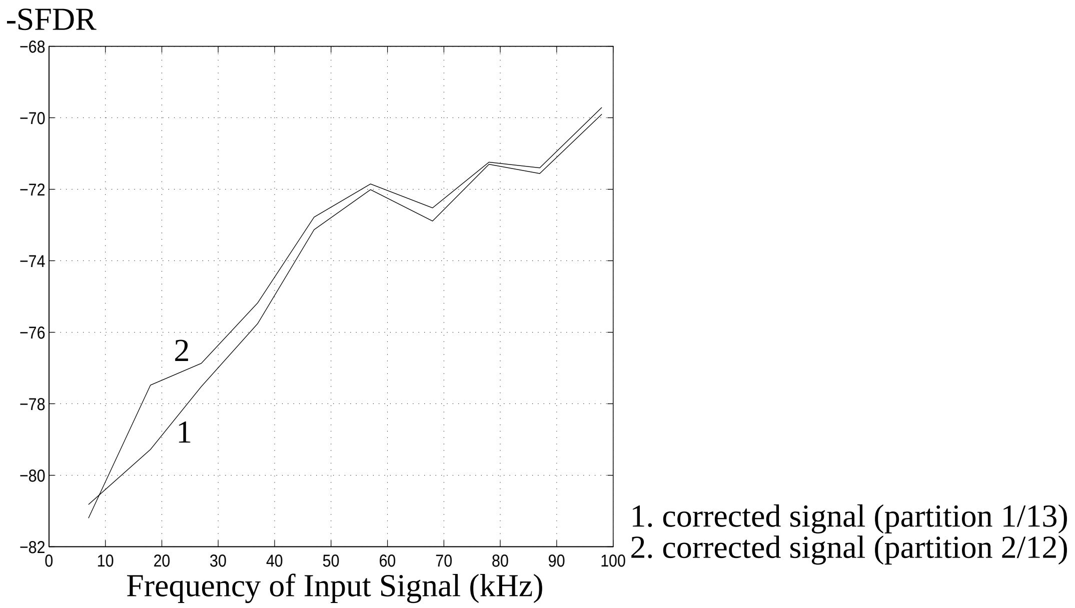


FIGURE 5.10 **Digital Correction for Partitions 1/13 and 2/12**

5.7 Discussion of Correction Method

After presenting in detail the proposed adaptive digital correction algorithm, it is necessary to compare it to existing algorithms, an overview of which has been provided in Section 5.5, to determine its advantages and limitations.

We will begin with the advantages. First, this algorithm does not require any additional analog circuitry, not even an extra switch aside from what is already in the uncorrected ADC, since the correction is done entirely on the digital side.

Second, it is adaptive, which means that it can track nonlinearity errors even when these change in time due to variations in bias, power supply, or temperature.

Third, it is completely on-line: a major limitation of existing correction methods is that they need an off-line calibration phase. For some particular signals, there is an interval during which the input signal is not of interest, for instance the retrace interval in television, and this can be used for off-line calibration. In other cases, for instance in a cellular base station, there is no such interval, and an on-line algorithm which uses the input signal (as opposed to a special calibration signal) to correct the errors would be very useful.

On the negative side, the proposed method is rather picky about the input signal. It is necessary that this signal should satisfy the requirement of local pdf uniformity in the vicinity of the bounds. We have seen in Section 5.2, that some signals are not suited for this, therefore it cannot be used in all-purpose ADCs. However, there are applications in which this condition is easily met, for instance in radio circuits, which use signals containing a number of independent modulated sinusoids, and possibly a large dither. If the requirement cannot be met, calibration can still be done off-line at a low frequency and then turned off during normal operation.

Another limitation is that it does not correct dynamic errors, unlike the phase-plane compensation method. One must keep in mind, however, that this method is rather complicated and, to our knowledge, seldom used.

In terms of performance, a disadvantage of our method is that it only corrects the errors in the first conversion stage(s). There are off-line methods that can correct the errors in all the stages, and therefore have better performance. On the other hand, we are talking here about pipeline ADCs, in which the discontinuities in the overall transfer characteristic caused by errors in a conversion stage decrease by a factor of two for each stage, from the MSB to the LSB. The improvement obtained by correcting the error in conversion stage 14-j is only $\frac{1}{2^j}$ of what is obtained by correcting the error in the first stage.

5.8 Summary

In this chapter another version of the histogram-based characterization method has been applied to a pipeline ADC. This version is slightly less accurate than the previous one, presented in Chapter 3, but much simpler.

An adaptive digital correction mechanism derived from the characterization method mentioned above was also presented and compared to existing techniques. This mechanism was shown to add one bit (6dB) to the SFDR of an experimental ADC over most of the bandwidth.

CHAPTER 6

Contributions and Future Work

6.1 Contributions of Research

The thesis brings two major contributions: a characterization method which provides information about the performance of the blocks that make up a multi-stage ADC, and a fully adaptive and fully on-line digital correction algorithm. To our knowledge, both contributions are the first of their kind. Other contributions include the analysis of nonidealities in ADCs and the redesign of a pipeline ADC, which was fabricated and tested successfully.

As opposed to existing methods which only characterize the ADCs at the system level, the method proposed in this thesis characterizes the ADCs at the internal block level, thus offering much more insight into what limits the performance of the ADC. One can use this method to study the nonlinearity of the A/D and D/A subconverters, the gain of the interstage amplifiers, component mismatches, hysteresis, and other phenomena. None of the existing characterization methods can provide the same range of information. Also, the software and hardware requirements are comparable to those of the other methods: simple programs in any high-level language and one buffer (in the case of the two-stage ADC or the pipeline ADC in partition 1/N-1) or two buffers (for the pipeline ADC in partition 2/N-2) would suffice.

The digital correction algorithm was developed in Chapter 5 as a spin-off of the histogram-based characterization and led to a 6dB improvement in the SFDR over most of the bandwidth. Compared to existing algorithms it enjoys significant advantages. First, the calibration can be performed using the normal input signal; no special calibration signals or calibration time periods are required. Second, it tracks the errors in an adaptive fashion, adjusting in a matter of seconds if there is any change in the circuit parameters. Finally, the correction is done completely on the digital side and no analog components whatsoever are necessary. However, the algorithm has a few shortcomings, for instance it only converges if the pdf of the input signal is nearly uniform, which means that it cannot be applied in a general-purpose ADC. Also, it cannot correct dynamic errors such as slope-dependent errors, and its performance is probably not as good as for some existing techniques because it only corrects the errors in the first stage or the first two stages. However, this approach is only at its first version and it has not said its last word yet.

The proposed characterization method and digital correction algorithm are based on the concept of bidimensional histogram, which is a code density table as a function of the digital outputs of the A/D subconverters of the multi-stage converter. This type of histogram, as well as most of the procedure required to analyze it, has been introduced in this thesis for the first time.

6.2 Proposals for Future Research

There are several ways in which the histogram-based characterization method introduced in this thesis could be improved.

Future work should probably start from the modeling stage, since a better model of the ADC would enable us to extract more information out of the histograms. Consideration should be given especially to other types of nonlinearity, aside from subconverter and inter-stage gain errors, and to dynamic effects such as slope dependence. This would imply a model with more parameters, and would require a different, more complex approach than the one we have used, which was based on calculating the parameters from the bounds, since this is already fully exploited - the number of unknown variables is equal to the number of equations.

Improvements could probably be achieved in the estimation procedure too, for instance by calculating the confidence intervals of the estimates, or by using ultralinear signal generators for better histogram normalization or for input S/H characterization.

More work should be done to better interpret the results, in order to identify the sources of nonlinearity at the physical level. We have pointed out in Section 2.6 that different errors (V_{be}, resistor, and current mismatches) have different effects on the DNL/INL of the subconverters so that it should be possible to recognize them.

The procedure could be easily automated. Modern test equipment can calculate and plot DNL and INL for the overall ADC. The same equipment, without any modification, can be used to sample data for histogram-based characterization. This data could be transferred to a computer or even processed locally if a microprocessor with mathematical capabilities is available in the test equipment.

We feel that the area of digital correction should also be looked into, either by developing off-line calibration procedures or by implementing the adaptive on-line algorithm introduced in Chapter 5.

In conclusion, the histogram-based multi-stage ADC characterization method has a great potential, and the research presented in this thesis is only the first step in what seems to be a very promising direction.

References

[AD95] Analog Devices data sheet, “AD9042: 12-bit, 41 MSPS Monolithic A/D Converter”, 1995

[Bla94] Blair, Jerome, “Histogram Meaurement of ADC Nonlinearities Using Sine Waves“, IEEE Transactions on Instrumentation and Measurement, vol. 43, No. 3, pp. 373, 1994

[Cap95] Capofreddi, Peter D. and Bruce A. Wooley, “The Use of Linear Models for the Efficient and Accurate Testing of A/D Conveters“, International Test Conference, pp. 54, 1995

[Chi96] Chiorboli, Giovanni, Giovanni Franco, and Carlo Morandi, “Analysis of Distortion in A/D Converters by Time-Domain and Code-Density Techniques”, IEEE Transactions on Instrumentation and Measurement, vol. 45, No. 1, pp. 45, 1996

[Cra90] Craiglow, Robert L., “Method and Apparatus for Self-Calibration of Subranging A/D Converter”, United States Patent 4,896,155, 1990

[Dem91] Demler, Michael J., “High-Speed Analog-to-Digital Conversion”, Academic Press, 199

[Eva86] Evans, William P., “Harmonic Distortion Reduction Technique for Data Aquisition”, United States Patent 4,612,533, 1986

[Eva90] Evans, William P. and Thomas K. Lisle, “A/D Converter with Error Correction and Calibration Apparatus and Method”, United States Patent 4,903,024, 1990

[Hum92a] Hummels, D. M., F. H. Irons, and S. P. Kennedy, “Using Adjacent Sampling For Error Correcting Analog-to-Digital Converters“, ISCAS 1992, pp.589

[Hum92b] Hummels, D. M. and S. P. Kennedy, “Improved Dynamic Compensation of ADC’s Using an Iterative Estimate of the ADC Calibration Signal”, Proc. Midwest Symposium on Circuits and Systems 1992, pp. 68

[Hum93] Hummels, D. M., R. W. Cook, and F. H. Irons, “Discrete-Time Dynamic Compensation of Analog-to-Digital Converters“, ISCAC 1993, pp. 1144

[Hum96a] Hummels, D. M., Y. N. Papantonopoulos, F. H. Irons, and C. A. Zoldi, “Identification of Error Mechanisms in a Folding and Interpolating ADC”, poster demo at ISCAS 1996

[Hum96b] Hummels, D. M., J. J. McDonald II, and F. H. Irons, “Distortion Compensation for Time-Interleaved Analog to Digital Converters”, IEEE Instrumentation and Measurement Technology Conference, pp. 728, 1996

[Iro91] Irons, F.H., D. M. Hummels, and S. P. Kennedy, “Improved Compensation for Analog-to-Digital Converters”, IEEE Transactions on Circuits and Systems, vol. 38, No. 8, pp. 958, 1991

[Iro96] Irons, F. H., D. M. Hummels, I. N. Papantonoulos, and C. A. Zoldi, “Analog-to-Digital Converter Error Diagnosis”, IEEE Instrumentation and Measurement Technology Conference, pp. 732, 1996

[Kar93] Karanicolas Andrew N., Hae-Seung Lee, and Kantilal L. Bacrania, “A 15-b 1-Msample/s Digitally Self-Calibrated Pipeline ADC”, IEEE Journal on Solid-State Circuits”, vol. 28, No. 12, pp. 1207, 1993

[Kar96] Karanicolas, Andrew N. and Hae-Seung Lee, “Digitally Self-Calibrating Piepline Analog-to-Digital Converter”, United States Patent 5,499,027, 1996

[Lee94] Lee, Hae-Seung, “A 12-b 600ks/s Digitally Self-Calibrated Pipelined Algorithmic ADC“, IEEE Jornal on Solid-State Cicruits, vol. 29, No. 4, pp. 509, 1994

[Man96a] Mantyniemi, Antti, Timo Rahkonen, and Antti Ruha, “A Low Power 10-bit 300kS/s RSD Coded Pipeline A/D Converter”, ISCAS 1996, pp. 79

[Man96b] Mantyniemi, Antti, Timo Rahkonen, and Antti Ruha, “A 6-mW 10-bit 300ksamples/s Pipeline A/D Converter”, Proceedings of the 39th Midwest Symposium on Circuits and Systems, vol. 1, pp. 205, 1996

[Moo97] Moon, Un-Ku and Bang-Sup Song, “Background Digital Calibration Techniques for Pipeline ADC’s”, IEEE Transactions on Circuits and Systems - II: Analog and Digital Signal Processing, vol. 44, No. 2, pp. 102, 1997

[Mor94] Morandi, Carlo and Luca Niccolai, “An Improved Code Density Test for the Dynamic Characterization of Flash A/D Converters”, IEEE Transactions on Instrumentation and Measurement, vol. 43, No. 3, pp. 384, 1994

[Mou89] Moulin, D., “Real-Time Equalization of A/D Conveter Nonlinearities”, ISCAS 1989, pp.262

[Mur95] Murden, Frank, and Roy Gosser, “12b 50MSample/s Two-Stage A/D Converter”, ISSCC 1995, pp. 278, slide supplement pp. 216

[Oli96] Oliveira, J. P., J. Vital, and J. E. Franca, “A Digitally Calibrated Current-Mode Two-Step Flash A/D Converter“, ISCAS 1996, pp. 199

[Raz95] Razavi, Behzad, “Principles of Data Conversion System Design”, IEEE Press, 1995

[Reb87] Rebold, T. A. and F. H. Irons, “A Phase-Plane Approach to the Compensation of High-Speed Analog-to-Digital Converters”, ISCAS 1987, pp.455

[Rib91] Ribner, David B., “Digital Error Correction System for Subranging Analog-to-Digital Converters”, United States Patent 5,047,772, 1991

[She86] (edited by) Sheingold, Daniel H., “Analog-Digital Conversion Handbook”, Prentice Hall, 1986

[Soe95] Soenen, Eric G. and Randall L. Geiger, “An Architecture and An Algorithm for Fully Digital Correction of Monolithic Pipelined ADC’s”, IEEE Transactions on

Circuits and Systems - II: Analog and Digital Signal Processing, vol. 42, No.3, pp. 143, 1995

[Suz90] Suzuki, Kaoru, “Multi-Stage Analog-to-Digital Converting Device”, United States Patent 4,973,974, 1990

[Tie90] Tiemann, Jeomre J., William E. Engeler, and Kenneth B. Welles, “Architecture for High Sampling Rate, High Resolution Analog-to-Digital Converter System”, United States Patent 4,903,026, 1990

[Wag91] Wagdy, Mahmoud Fwazy and Selim S. Awad, “Determining ADC Effective Number of Bits Via Histogram Testing”, IEEE Transactions on Instrumentation and Measurement, vol. 40, No. 4, pp. 770, 1991