Error Modeling for Folding and Interpolating Analog to Digital Converters

By Ioannis Nikolaou Papantonopoulos

Thesis Advisor: Donald M. Hummels, Ph.D.

An Abstract of the Thesis Presented in
Partial Fulfillment of the Requirements for the
Degree of Master of Science (in Electrical Engineering).
August, 1995

This thesis presents in detail the Folding and Interpolating (FAI) analog to digital converter architecture. FAI converters are very efficient since they require a smaller number of components than a classical Flash architecture. Thus they reduce both power consumption and die-size, while operating at frequencies comparable to fully parallel architectures. Error mechanisms causing output distortion are identified and explained, while an effective computer simulation module is created that allows error mechanisms to be emphasized or suppressed. The state of the converter and an estimate of the input signal's slope are used in conjunction with a set of basis functions to characterize the converter. Basis functions are selected to model pertinent error mechanisms, thus allowing diagnosis of the converter. Converter output errors can be successfully traced back to specific architectural flaws through the use of methods developed in this thesis.

ERROR MODELING FOR FOLDING AND INTERPOLATING ANALOG TO DIGITAL CONVERTERS

By

Ioannis Nikolaou Papantonopoulos

A THESIS

Submitted in Partial Fulfillment of the

requirement for the Degree of

Master of Science

(in Electrical Engineering)

The Graduate School

University of Maine

August, 1995

Advisory Committee:

Donald M. Hummels, Associate Professor of Electrical and Computer Engineering, Thesis Advisor

Fred H. Irons, Castle Professor of Electrical and Computer Engineering,

Duane C. Hanselman, Associate Professor of Electrical and Computer Engineering.

ACKNOWLEDGMENTS

Numerous thanks are due to the people that have contributed to the completion of this thesis.

First and foremost I wish to thank my parents, Nikolaos and Alkmini, and my family for their love, support and dedication. I am forever grateful to them. Their strong commitment to family values and education has enabled me to come this far in my life.

I wish to thank Professors Donald M. Hummels and Fred H. Irons, for their guidance and support in this project. I express my sincere gratitude for their patience, time and enthusiasm. Their inspiring teaching has changed my life, and helped me become not only a better engineer, but a better person as well. I am proud and honored to have had the chance to work with both, and I couldn't be happier for my decision to obtain my Master's at the University of Maine. Their relentless strive for excellence will mark my life for ever.

I also wish to thank Professor Duane C. Hanselman for his constructive review of my thesis. His accurate comments were very helpful in finalizing this thesis. I appreciate his time and effort.

A lot of thanks go to all my professors here at the University of Maine who have contributed to my education and my integration as an engineer.

Finally I would like to thank Wahid Ahmed for his help with this project and all my friends and co-workers for making my stay at UMaine an unforgettable experience.

" To my parents I owe living; to my teachers living well"

-Alexander the Great

Η διατριβη αυτη ειναι αφιερωμενη στη γιαγια μου, Αναστασια Καραμιχου, που θα ειναι παντα διπλα μου, σε καθε βημα της ζωης μου.

This thesis is dedicated to my grandmother, Anastasia Karamihou, who will always be next to me, in every step of my life.

Μεσα απο αυτες τις γραμμες θελω να εκφρασω την απειρη ευγνωμοσυνη μου στους ανθρωπους που με αξιωσαν να φτασω ως εδω στη ζωη μου.
Στους γονεις μου Νικολαο και Αλκμηνη, που μεσα απο δυσκολιες φροντισαν παντα να εχω τα καλυτερα εφοδια στη ζωη μου.
Στη θεια μου Αννα, που με μεγαλωσε με την αγνοτερη αγαπη στον κοσμο.
Στον παππου μου Δημητριο, που ξερω οτι καμαρωνει απο ψηλα.
Στον αδερφο μου Σωτηρη, που η αγαπη του παντα με συντροφευει.
Στην ευρυτερη οικογενεια μου, που ειναι συνεχεια στο πλευρο μου.
Στους δασκαλους μου, και ειδικα στον Κυρ Κωστα, το σοφοτερο απο ολους.
Στους πραγματικους μου φιλους Σακη, Lubomir, Rich και Doug, που θα ειμαστε παντα μαζι.
Και τελος στη Μαρθα μου, που στολιζει με αγαπη τα καλυτερα χρονια της ζωης μου.

"Στους γονεις μου οφειλω το ζην, στους δασκαλους μου το ευ ζην"
-Μεγας Αλεξανδρος

CONTENTS

LIST OF FIGURES

CHAPTER 1
Introduction

1.1 Problem Statement

Analog to Digital Converters (ADCs) are widely used in numerous applications. Their widespread use, especially in communication devices such as radar and video receivers, has created the need for faster and more accurate converters. A new converter architecture that offers promising performance is the *Folding and Interpolating* (FAI) ADC. This architecture is more efficient than the classic Flash implementation since it requires a significantly smaller number of components [1]. The first goal of this thesis is to introduce the FAI architecture and explain its operation in detail.

The ever-increasing applications of ADCs such as the FAI converter impose higher demands on their performance. The second goal of this thesis is to develop procedures that identify prevalent error mechanisms in an FAI architecture, and examine how they affect the converter's performance. In addition, a way of tracing output distortion back to particular architectural flaws will be investigated.

1.2 Background

The concept of signal folding in the implementation of ADCs was first introduced by Arbel and Kurz [2] in 1975. The main motivation was the dramatic reduction of the number of comparators required in the design. Different ways of implementing the folding concept have been proposed since then, including the introduction of double-folding [3]. Quantizer-subtractor networks [4] or resonant tunneling diodes [5] can be used to produce folding signals, but the most popular method involves the use of coupled differential pairs (CDPs) [1, 6, 7, 8]. Almost concurrent with the introduction of the CDPs is the concept of resistive interpolation, which produces additional folding signals without requiring additional CDPs [6, 7]. The reduced hardware requirements (smaller component count) render FAI converters one of the most efficient ADC architectures in terms of power consumption and die size.

The concept of ADC compensation was introduced by Irons and Rebold [11, 12]. Since then, researchers have worked extensively in the field of compensating dynamic errors in high-speed analog-to-digital converters, developing the method of phase plane compensation [13, 14]. This method assumes that the ADC error is mainly a function of the converter's state and the input signal's slope, and focuses on creating an error correction look-up table. Different techniques for obtaining required calibration data have been introduced. They include using a multitude of different signals [15], adjacent sampling [16], or two-tone signals [17]. Although several methods of compensation have been developed [18, 19, 20], this thesis will rely heavily on the non-iterative ADC characterization procedure introduced in [21], since it allows the use of basis functions to model converter errors. An efficient algorithm that selects the most appropriate basis functions out of a set of possible candidates is also extensively utilized to produce estimates of the converter's error function [22].

The technique for performing diagnostics on an analog to digital converter presented in this thesis is believed to be novel. Although past schemes have successfully compensated ADCs, this thesis introduces selection of basis functions modeling distinct error mechanisms. This allows for the diagnosis of the error mechanisms that contribute to the converter's output distortion.

1.3 Thesis Organization

This thesis begins with the statement of the problem and its history in Chapter 1. It then introduces the main concepts of the Folding and Interpolating ADC in Chapter 2. A particular implementation of such a converter is extensively studied and analyzed.

Chapter 3 closely examines the FAI structure for possible error mechanisms. Since the main objective of this thesis is to investigate and model such error mechanisms, a computer simulation of the particular architecture studied in Chapter 2 is developed. This simulation allows the most important error mechanisms to be suppressed or emphasized, thus providing a direct means of testing the theoretical derivations.

Since the dominant error mechanisms and their affect on the converter output are identified, characterization of a FAI ADC from its output is feasible. The results presented in this thesis are based on simulations, but the technique developed can be equally successful on an actual ADC (not available at the time of completion of this project). Chapter 4 begins with the fundamentals of converter characterization and compensation and explains how basis functions can be used to model converter errors. Appropriate basis functions are then derived that model the error mechanisms investigated in Chapter 3. Simulations for the particular implementation of Chapter 2 are shown, proving that converter architectural errors can be isolated, diagnosed and compensated.

Finally Chapter 5 summarizes the results, states the main conclusions derived, and suggests future research directions for this work.

CHAPTER 2

Folding and Interpolating ADC Design Description

2.1 Motivation for FAI ADCs

The ever increasing demand for high speed analog to digital conversion has spawned the development of numerous ADC designs. The Flash Converter is an example of a high-speed architecture. This architecture utilizes a resistive ladder to produce voltage references that are used by a bank of comparators. The comparators compare the input signal against these reference voltages. Then the comparator outputs are latched and encoded to produce the resulting digital representation of the input signal. It is important to note that the number of comparators required for an n-bit converter is $2^n - 1$. Thus, as the number of bits increases, so does the complexity of the ADC. The motivation behind the Folding and Interpolating ADC is to reduce the required number of comparators. This reduces both the die size and the overall power consumption of the converter, while maintaining the very high conversion rate of a fully parallel converter. This chapter provides a detailed description of a Folding and Interpolating ADC architecture.

2.2 Folding Converters

In an ideal architecture utilizing the concept of folding, the input signal undergoes analog preprocessing by a *Folding Amplifier*. Such an amplifier has a repetitive, folded, triangular-shaped transfer characteristic, as shown in Figure 2.1. The main advantage of folding is that the input signal is converted into a signal with reduced range, thus reducing the number of comparators required for the conversion process.

An example of a folding amplifier characteristic is given in Figure 2.2. In this example, the input voltage range has been divided into 8 regions, or *subranges*

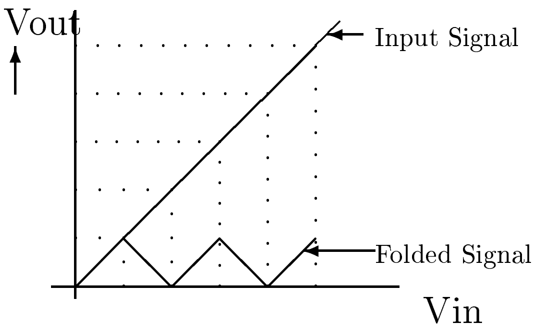

Figure 2.1: The Folding Process

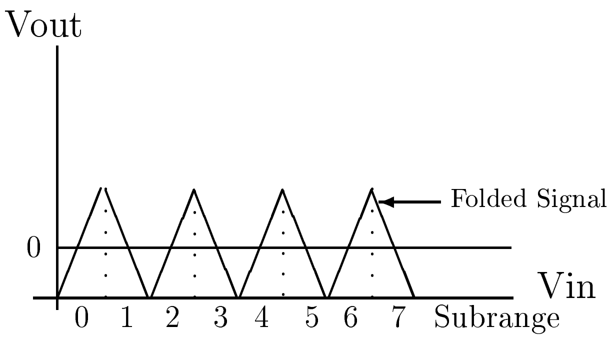

Figure 2.2: Folding Amplifier Characteristic

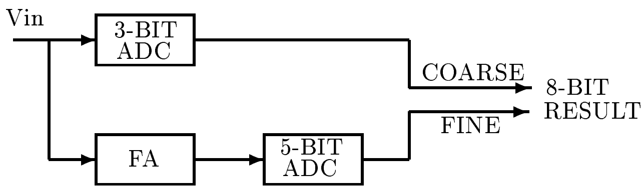

Figure 2.3: Conceptual 8-bit Folding ADC

(0 to 7). The number of subranges created is defined as the *Folding Ratio* of a folding amplifier. For the example of Figure 2.2, the folding ratio is 8.

A conceptual implementation of an ADC incorporating the folding architecture is shown in Figure 2.3. The input signal is quantized by a coarse 3-bit converter that determines the signal subrange. The same signal is also driven through a folding amplifier with a folding ratio of 8, so that a 5-bit converter can be used to obtain the least significant bits of the output sample. Here instead of a classic 8-bit direct Flash implementation that would require $2^8 - 1 = 255$ comparators, this Folding architecture requires only a total of $2^3 - 1 + 2^5 - 1 = 38$ comparators (if the 5-bit and 3-bit converters are also Flash).

Direct implementation of this scheme quickly runs into a major problem. In practice, the tips of the triangular folding waveforms at the output of the folding amplifiers tend to round off as the frequency of the input signal increases (bandwidth limitation). This introduces distortion for input signals near the transitions between subranges. In other words, the non-ideality of the folding characteristic renders the conversion ineffective, since code errors occur at the output.

One Folding architecture uses multiple folding signals with appropriate offsets to overcome the rounding problem of the triangular waveform. This ensures that there is always at least one folding signal that is not close to the dangerous (in terms of error) rounding region. The linear parts of the folding signals are the regions that are midway between the peaks of the folded signal. This is a very important observation, since in these regions the nonlinearities (rounding

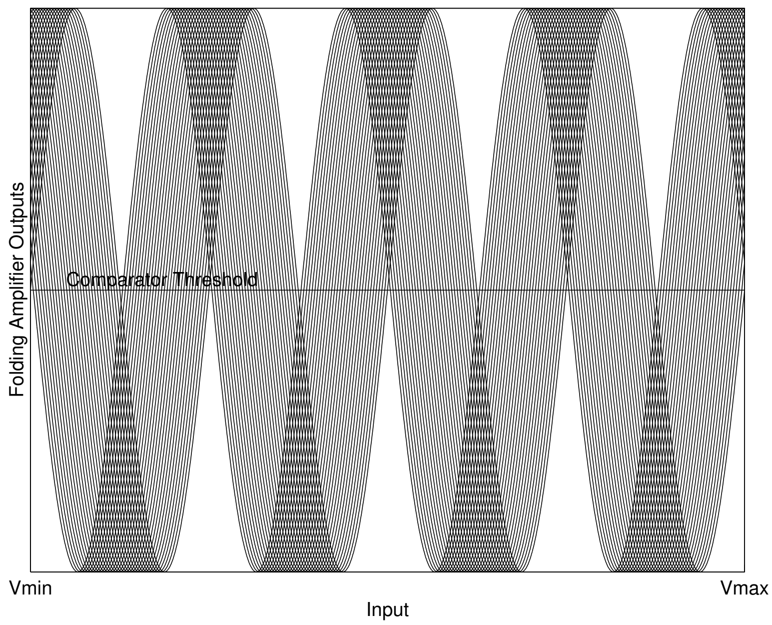

Figure 2.4: Resulting Folding Characteristics using 32 Folding Amplifiers

at the peaks) of the folding signals do not affect the conversion. For the example of Figure 2.3, instead of using one folding amplifier, a total of 32 amplifiers could be used, where each one of the 32 amplifier characteristics is offset from the previous one. This way, each subrange is now subdivided into 32 more regions. The output of each of the 32 folding amplifiers is driven through a comparator, thus allowing for the generation of the 5 least significant (*fine*) bits. Figure 2.4 illustrates 32 folding signals for amplifiers that round the folding characteristic. Note that the characteristics shown in Figure 2.4 are simulated and are used for illustration purposes. However, this implementation is almost never used, since it requires a large number of folding amplifiers. This increases the complexity of the design and counters the hardware reduction achieved by implementing the folding technique. Alternatively, *Interpolation* is implemented, which generates additional folding characteristics without building individual folding amplifiers.

One FAI implementation uses only four folding amplifiers for the analog preprocessing required by an 8-bit converter. The remaining folding amplifier characteristics are derived by means of Resistive Interpolation as shown in Figure

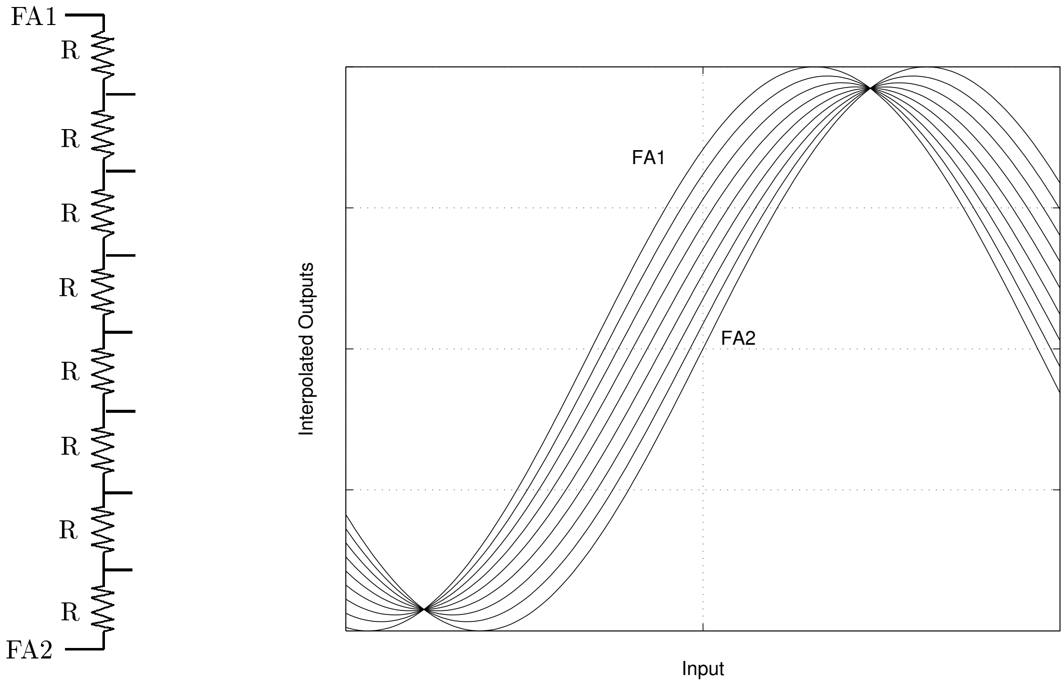

Figure 2.5: Interpolation between two Folding Amplifier Outputs

2.5. The fundamental concept behind this technique is that outputs from two consecutive folding amplifiers are used as inputs to a ladder of 8 resistors. The taps of the interpolation network generate the intermediate folding characteristics, much like a voltage divider network [7]. The fact that the amplitudes of the interpolated signals are not the same is unimportant, since only their linear portions are utilized in the conversion.

For a given set of two folding amplifier outputs, resistive interpolation produces intermediate characteristics and the corresponding threshold crossings upon which the conversion is based. The number of resistors used in one interpolation ladder is called the *Interpolation Rate*. For the example of Figure 2.5 the interpolation rate is eight. It is obvious that any deviation in interpolator resistor values leads to errors in threshold crossings (often referred to as zero crossings) and hence in code transitions [6]. This phenomenon will be studied extensively in Chapter 3.

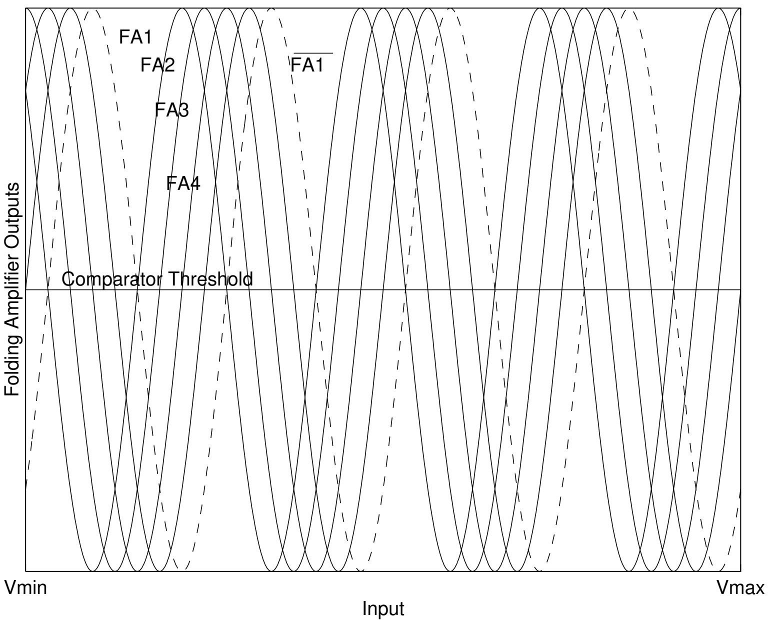

Figure 2.6: Four Folding Amplifier Characteristics

The complete set of folding amplifier outputs for an 8-bit example can be obtained as follows. Four folding amplifiers are used to generate the characteristics as shown in Figure 2.6. Note that the last curve in the interpolating pairs is $\overline{FA1}$ (the dashed line in Figure 2.6). Since the lateral shift between the four outputs is $\frac{V_{max}-V_{min}}{32}$, then a fifth folding characteristic shifted by the same amount from $FA4$ would be identical to $\overline{FA1}$. Assuming that the folding amplifiers provide complementary outputs, the fifth folding signal is available, without the need to generate it separately. It is now clear that to increase the ADC resolution more zero crossings may be obtained by interpolating between the existing 5 characteristics (labeled $FA1$,$FA2$,$FA3$,$FA4$ and $\overline{FA1}$). Four resistive ladders are required to interpolate between the pairs $FA1$&$FA2$, $FA2$&$FA3$, $FA3$&$FA4$, and $FA4$&$\overline{FA1}$. For an Interpolation Rate of 8, a total of $4 \times 8 = 32$ interpolated signals are obtained. These 32 interpolation outputs are depicted in Figure 2.7.

The rest of this chapter is devoted to analyzing the different subsystems of a Folding and Interpolation ADC. In overview, the converter produces a series of folding signals that are close enough to each other so that for any particular value

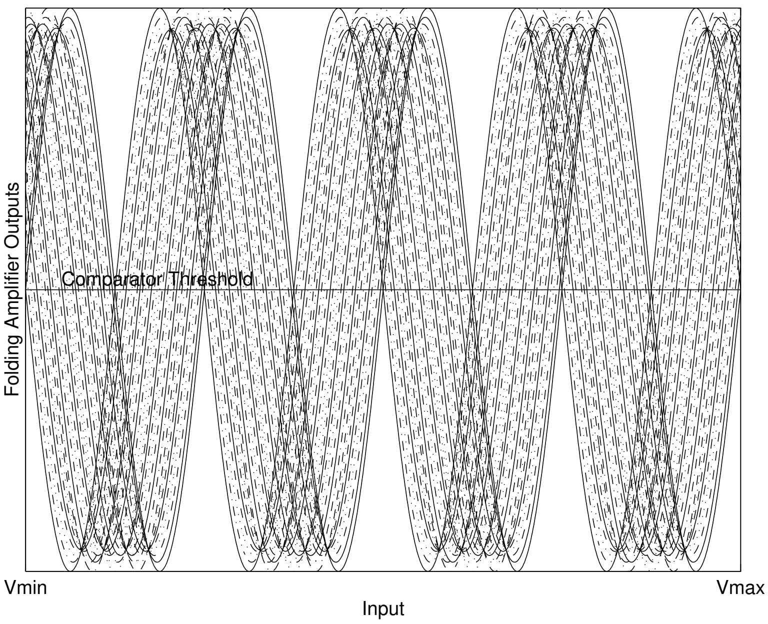

Figure 2.7: Resulting Folding Characteristics using Interpolation

of the input, at least one of them is linear. As Section 2.5 will show, some of these folding signals are used for the generation of the most significant (*coarse*) bits, while interpolation between the folding signals yields the fine bits. In practice, the converter is preceded by a Sample and Hold (S/H) network, that latches the input at discrete time intervals dictated by an external clock signal. The following sections will look closely at each step of the conversion process and analyze each converter element separately.

2.3 The Coupled Differential Pair

As shown in Figure 2.8, folding amplifiers are commonly implemented using a *Coupled Differential Pair (CDP)* [1].

The CDP circuit has two fixed voltage references, V_{lo} and V_{hi}, and is driven by input V_{in}. While the input is smaller than both references, T2 and T4 are conducting, making the output low ($V_{cc} - IR$). If the input is between V_{lo} and V_{hi}, then T1 is on, T2 goes off, while T3 and T4 remain off and on respectively. In this case the output is high (V_{cc}). Finally, when the input is higher than both

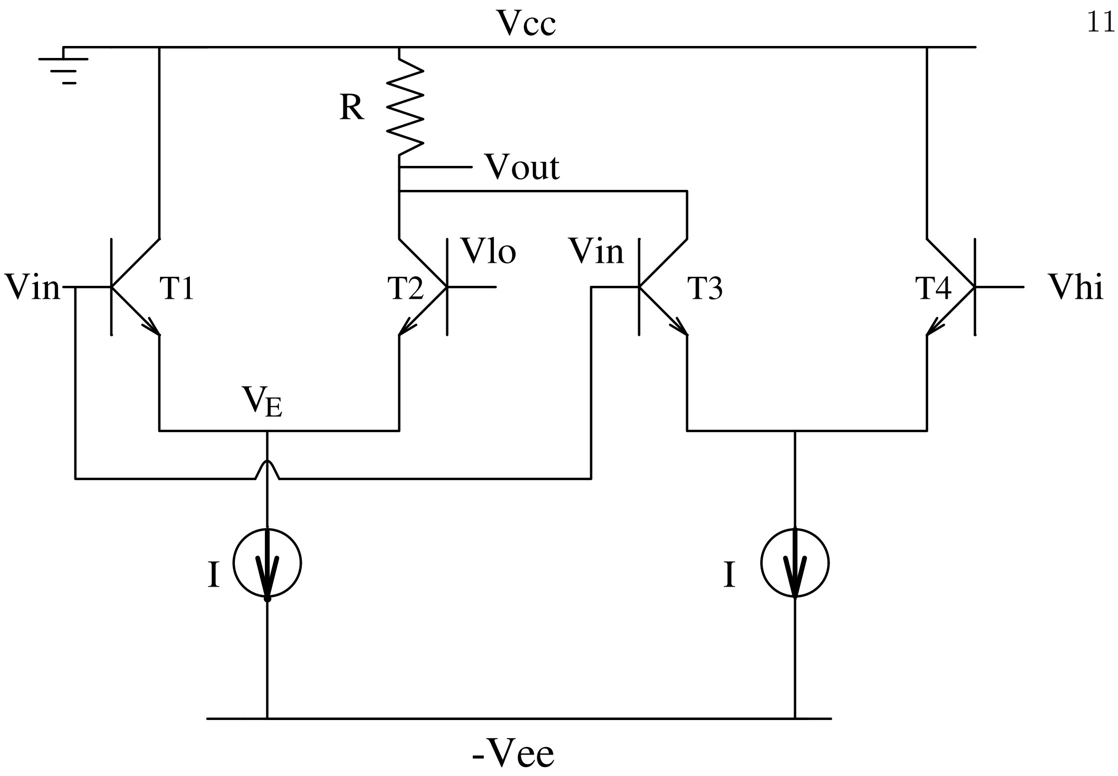

Figure 2.8: The Coupled Differential Pair

references, T3 turns on, while T4 turns off, thus bringing the output low again. In short, the CDP's output is high as long as $V_{lo} < V_{in} < V_{hi}$ and is low anywhere else. The commonly accepted symbol and transfer characteristic for a CDP are shown in Figure 2.9.

A more quantitative description of the CDP will now be developed. First assume that the common emitter mode voltage of T1 and T2 is V_E. In addition, let the currents flowing through T1, T2, T3 and T4 be denoted by i_1, i_2, i_3, i_4 respectively. Then an output equation for the coupled differential pair is derived by considering the behavior of the differential pair, T1 and T2. KCL yields Equation 2.1.

$$i_1 + i_2 = I \tag{2.1}$$

For transistors T1 and T2, the currents i_1 and i_2 are respectively given by

$$i_1 \quad = \quad I_s e^{(V_{in} - V_E)/V_T}$$

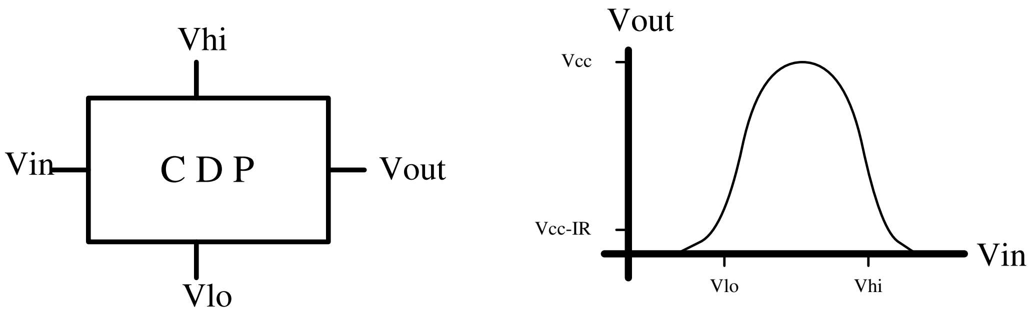

Figure 2.9: CDP Characteristic

$$i_2 = I_s e^{(V_{lo} - V_E)/V_T}$$

where V_T is the thermal voltage, and I_s is the emitter saturation current. Taking the ratio of i_1 to i_2 and using Equation 2.1 yields

$$\begin{aligned} \frac{i_1}{i_2} &= e^{(V_{in} - V_{lo})/V_T} \\ i_2 &= \frac{I}{1 + e^{(V_{in} - V_{lo})/V_T}} \end{aligned} \tag{2.2}$$

A similar analysis for the differential pair, T3 and T4 yields

$$i_3 = \frac{I}{1 + e^{-(V_{in} - V_{hi})/V_T}} \tag{2.3}$$

Finally, since the two differential pairs are independent, Equations 2.1, 2.2 and 2.3 yield an overall expression for the entire CDP

$$\frac{-V_{out}}{RI} = \frac{1}{\left(1 + \exp(\frac{(V_{in} - V_{lo})}{V_T})\right)} + \frac{1}{\left(1 + \exp(\frac{-(V_{in} - V_{hi})}{V_T})\right)} \tag{2.4}$$

2.4 Folding Amplifier Construction

Combinations of CDP's may be used to create a Folding Amplifier characteristic. Each CDP is used to create a characteristic that will be high between two reference voltages, V_{lo} and V_{hi}, and low everywhere else. Combining several CDP's with appropriate shifts in reference voltages produces an overall output resembling the desired folding characteristic. It is important to note the non-ideality of the CDP. Ideally, the characteristic of a FA should be perfectly piecewise linear, much like one of the triangles shown in Figure 2.1. However, the CDP demonstrates a non-linear transition band from high to low and backwards, and also a rounding effect at the top of the characteristic. The small transition bands are observable, and they typically span about $6V_T$, where V_T is the Thermal Voltage at a particular temperature.

In order to simplify the discussion, this thesis concentrates on a particular 8-bit implementation of the FAI converter, where the input voltage ranges from $-2V$ to $-1V$ (typical ECL levels for a high frequency device). Two CDPs can be configured to produce complementary folding amplifier patterns. In order to increase resolution (additional zero crossings), more than two CDPs are used in a CDP array. A typical implementation for a *folding block* utilizing CDPs is shown in Figure 2.10 where 8 CDPs, A_1 through A_8, are laid out in an array. The high reference voltage of each CDP is tied to the low reference of the next. Therefore, $v_1 = -2V$, $v_2 = -2 + \frac{1}{8}$ and so forth, yielding, in general, $v_{i+1} = v_i + \frac{1}{8}$ for the 8 CDPs. In this case, $v_9 = -1V$. Summing the outputs of CDPs A2, A4, A6 and A8 forms a full folding amplifier characteristic with a folding ratio of 8. Similarly, the sum of A1, A3, A5 and A7 yields the complement of the aforementioned folding signal. These characteristics are called FA1 and $\overline{FA1}$ respectively. Note that the signals shown in Figure 2.10 are for illustration purposes and are not actual folding amplifier outputs.

The complementary folding amplifier outputs are used as inputs to com-

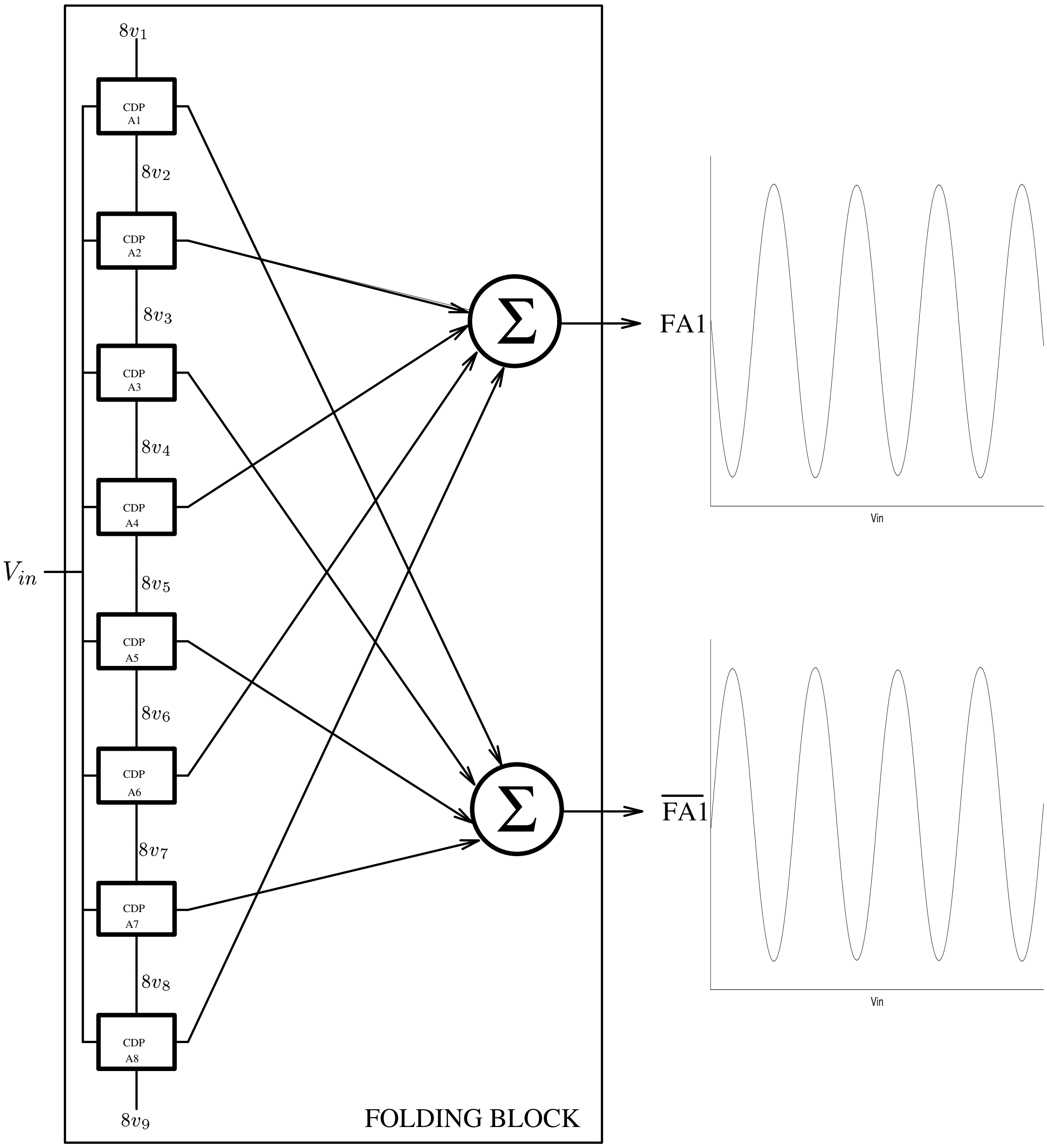

Figure 2.10: Generation of Complementary Folding Signals

parators to obtain the binary outputs required by the FAI converter. For example, $FA1$ and $\overline{FA1}$ would be connected to the positive and negative terminals of a comparator, thus producing a binary "1" whenever the *difference* signal $(FA1 - \overline{FA1})$ is above zero.

2.5 MSB Generation

Figure 2.3 illustrated the use of a low resolution converter to provide the most significant bits of the converter output. The CDP implementation of the folding amplifier provides an alternative which does not require the use of a low resolution converter. Existing CDP outputs are combined to provide signals that are used to determine the most significant output bits.

Specifically, let $b_0 b_1 \ldots b_7$ denote the output bits, where b_0 is the most significant bit of the converter. For the structure of Figure 2.3 a three bit converter would be used to form the bits b_0, b_1 and b_2. The CDP implementation allows combinations of CDP outputs to generate these bits.

Figure 2.11 shows the difference pattern obtained by $FA1 - \overline{FA1}$. This characteristic is exactly the b_2 pattern for this 8-bit converter. The input range is subdivided into 8 regions, and is entered in a comparator to obtain b_2.

Figure 2.12 shows how the same set of CDPs can be used to derive the b_0 bit. In a similar fashion, b_1 can be also derived from the same folding block. Figure 2.13 illustrates the procedure. Thus the initial block diagram of Figure 2.3 can be modified, in the sense that there is no need for a 3-bit coarse quantizer, since the folding stages also provide the required 3 MSBs.

2.6 Behavior at Voltage Range Limits

Figure 2.14 shows four folding amplifier outputs using the ideal CDP response given by Equation 2.4. It is very important to note the effect the CDP has

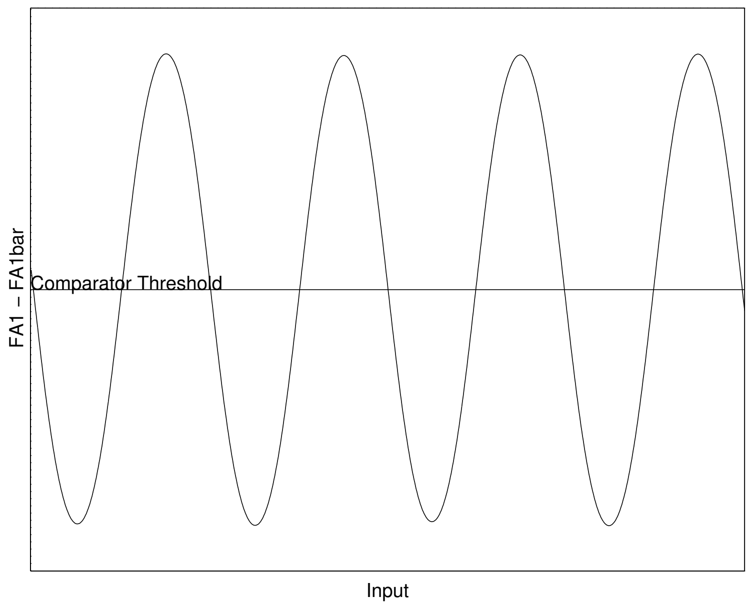

Figure 2.11: b_2 Derived from FA1 and $\overline{FA1}$

on the actual shape of the folding outputs. The peaks and valleys of the characteristics shown are rounded off (nonlinear behavior). Moreover, Figure 2.14 shows distortion at both endpoints of the input range as well (signals dropping off), since there are no additional CDPs to pull the folding signals to the correct levels.

It is clear that if these characteristics were to be used, the interpolated signal transition thresholds would exhibit nonuniform errors at both the low and high ends of the input range. In order to overcome this shortfall, additional CDPs are added to the ladder shown in Figure 2.10. An extra CDP is added on both the high and low ends of the ladder, so that the undesired behavior is pulled outside the nominal input range. For the case shown in Figure 2.10, an extra CDP is added before A1, with a low reference of $-2 - \frac{1}{8}V$ and a high reference of $-2V$, and an additional one after A8, with a low reference of $-1V$ and a high reference of $-1 + \frac{1}{8}V$. These additions pull the endpoint effects below $-2V$ and above $-1V$, which are outside the nominal input range of -2 to -1 Volt.

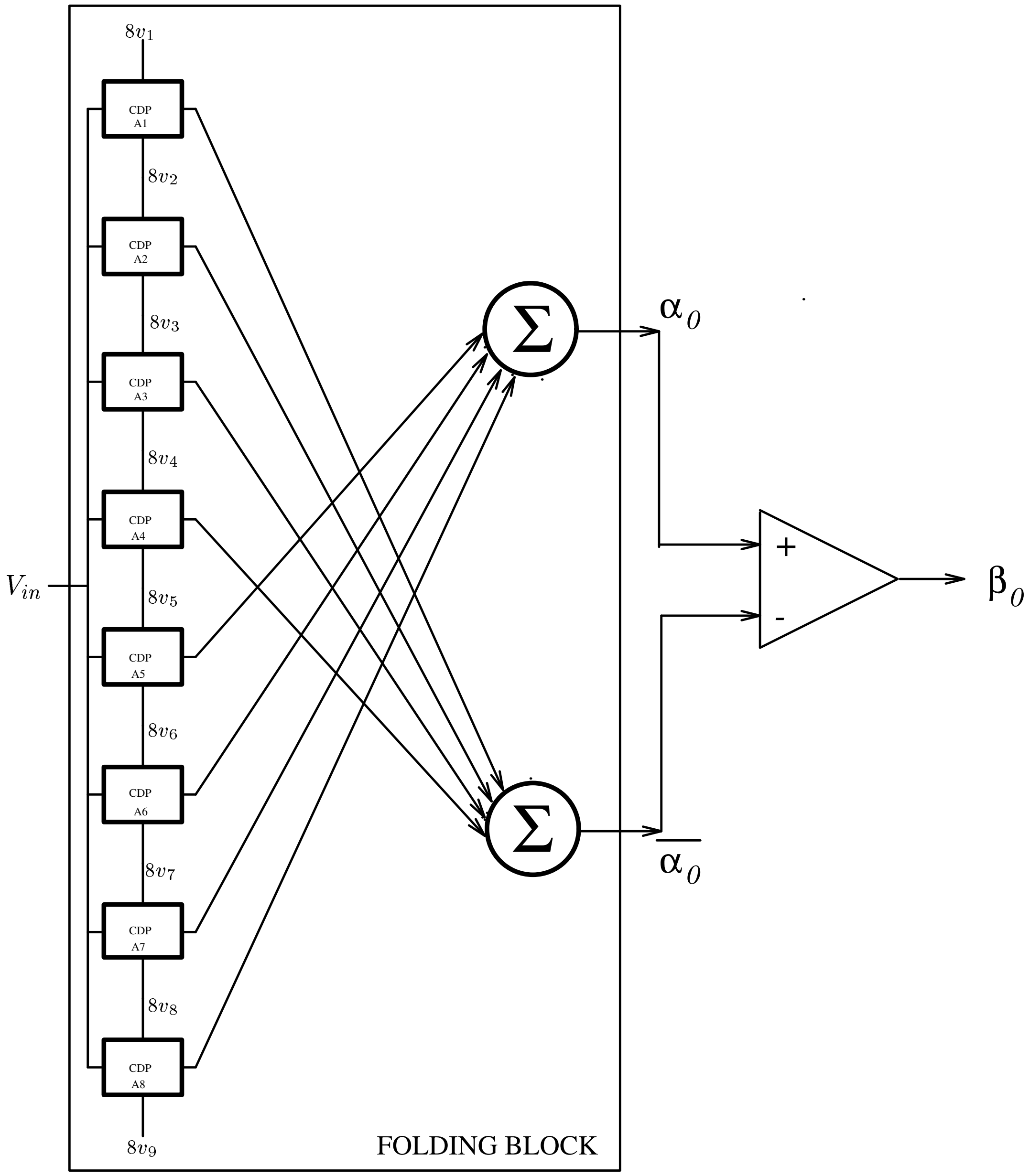

Figure 2.12: MSB Generation

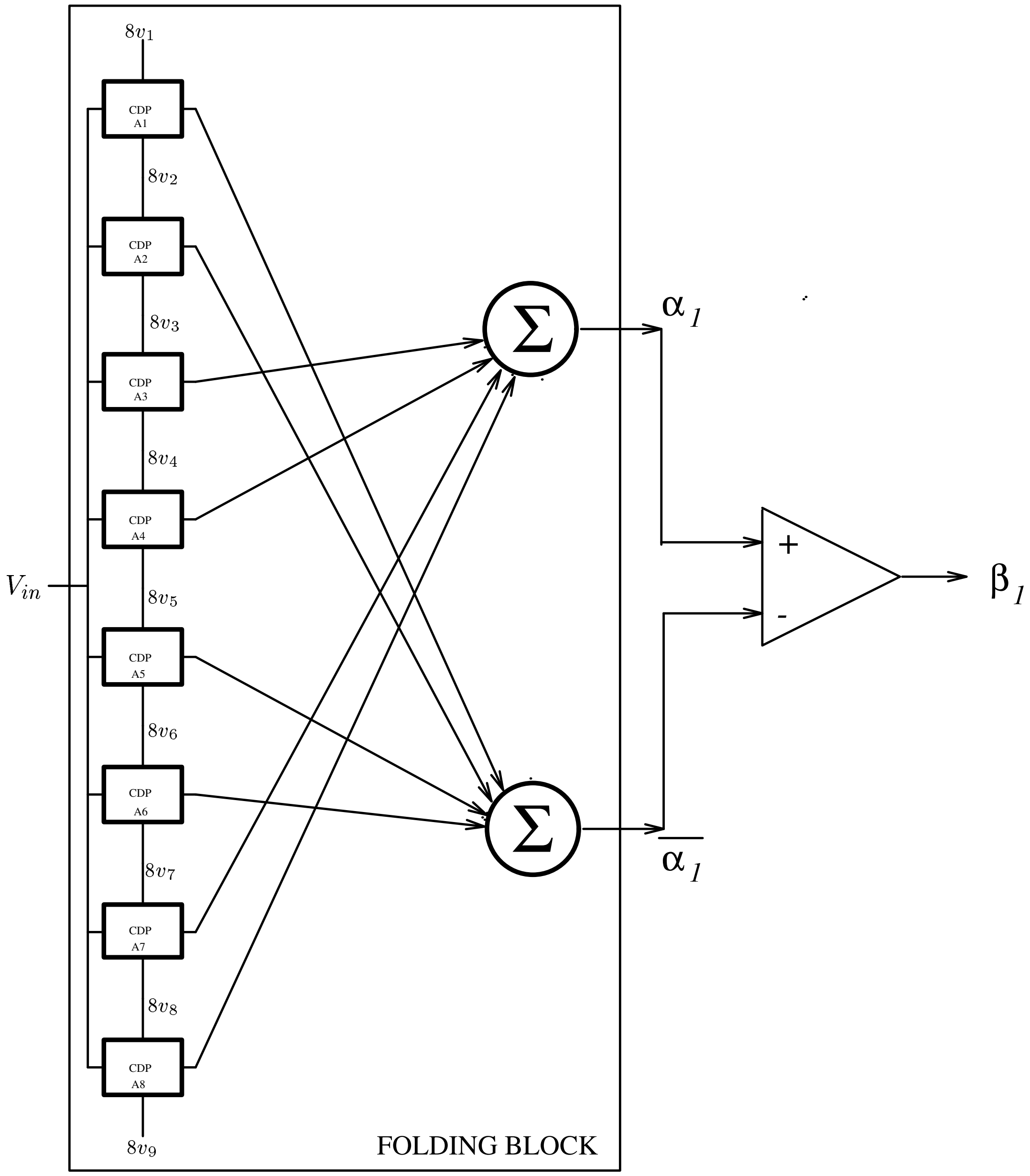

Figure 2.13: MSB-1 Generation

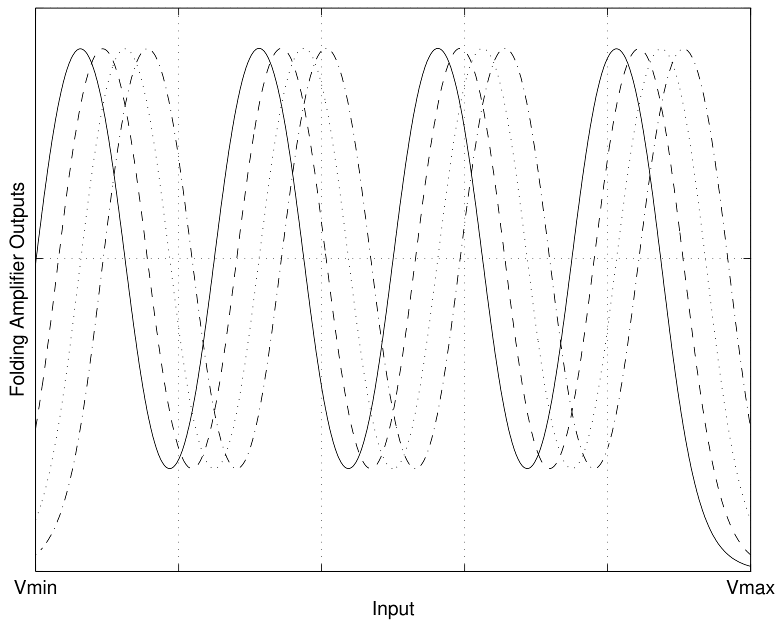

Figure 2.14: Effect of CDP nonlinearity on folding amplifier behavior

2.7 Folding Amplifier Signals

To facilitate the discussion of the formation of folding amplifier signals using CDPs consider the proposed CDP array shown in Figure 2.15. This array includes an extra CDP on both ends, and consequently the numbering on the CDPs and the corresponding reference voltages has changed accordingly.

As indicated in Figure 2.15, CDPs A0-A9 are used to generate $FA1$ and $\overline{FA1}$, CDPs B0-B9 for $FA2$ and $\overline{FA2}$, and so forth. As an example, reference v_0 is set to $-2 - \frac{1}{8}V$, while reference v_1 is $v_1 + \frac{1}{32}$. Cast in general terms, the voltage on the ith reference is given by

$$v_i = v_{min} + \frac{i}{44}(v_{max} - v_{min}) \qquad , i = 0, 2, \ldots, 43 \tag{2.5}$$

where $v_{min} = v_0 = -2 - \frac{1}{8} = -2.125V$ and $v_{max} = -1 + \frac{1}{8} + \frac{4}{32} = -0.75V$. Thus, for example, $v_4 = -2.125 + \frac{(5-1)}{44}(-0.75 - (-2.125)) = -2V$, which is the lower end of the nominal input range. The 44 reference voltages described by Equation 2.5 are created using a resistor voltage divider with 44 equal value resistors where

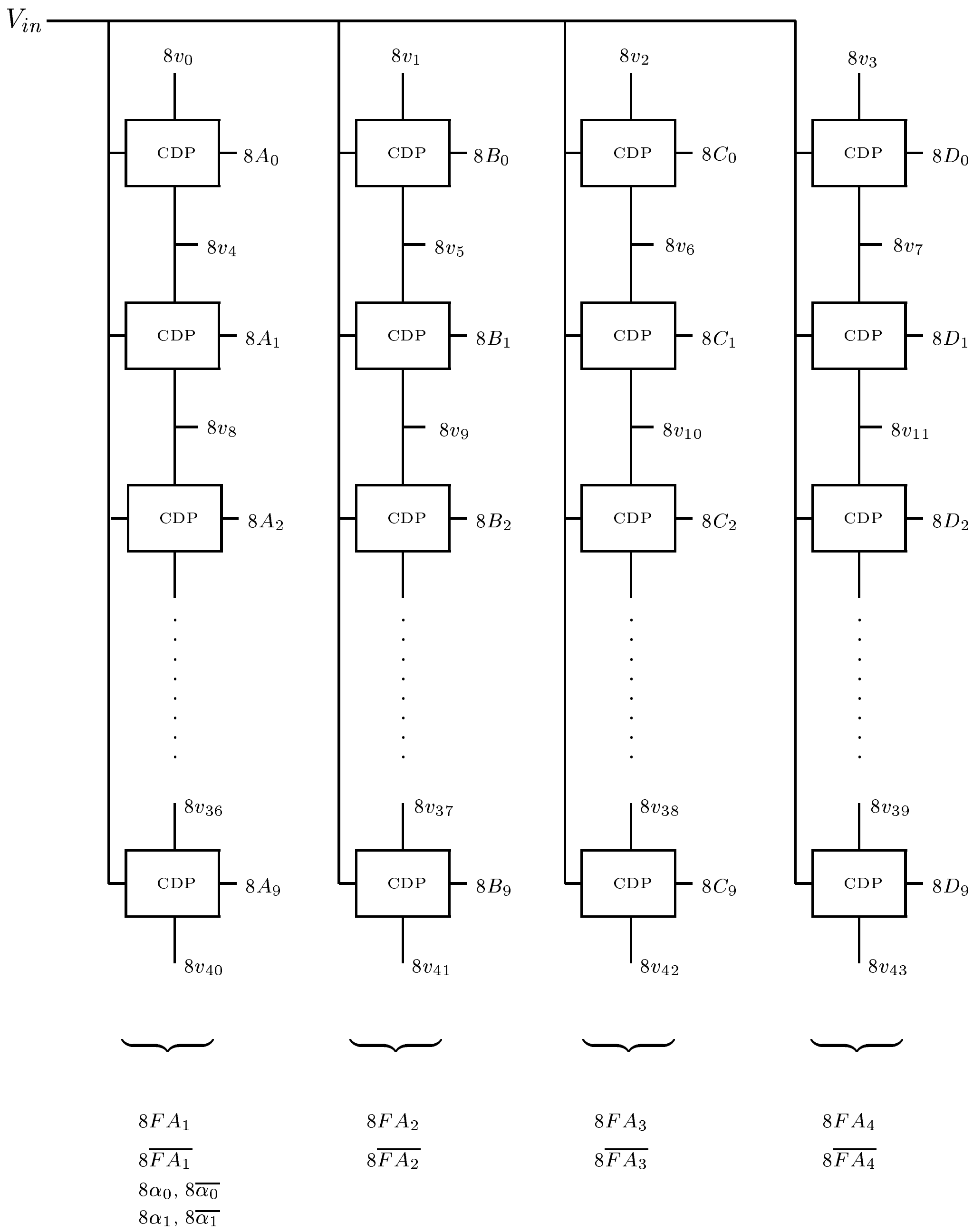

Figure 2.15: Folding Signal and Coarse Bit Generation CDP Array

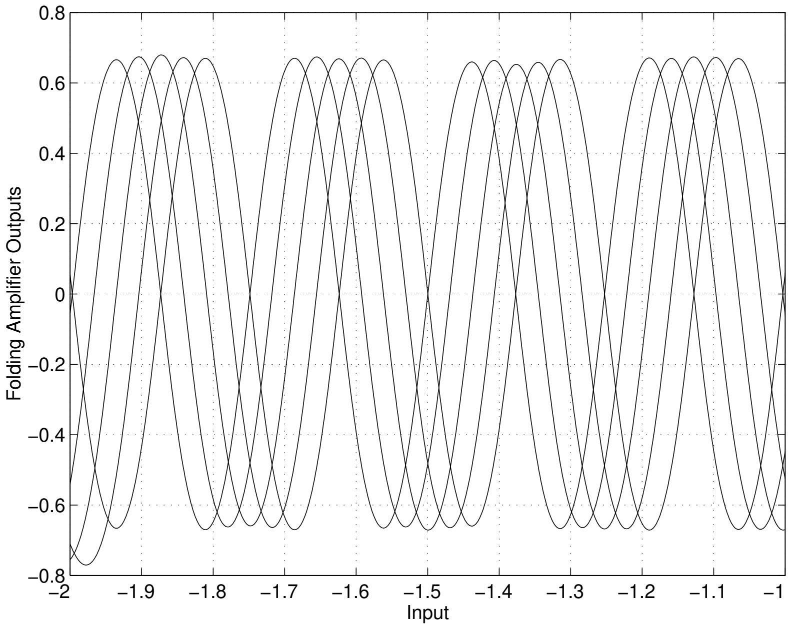

Figure 2.16: Folding Amplifier Difference Characteristics

v_{max} and v_{min} are the top and bottom voltages of the divider.

Using appropriate CDP outputs, it is possible to create the difference folding amplifier characteristics ($FA1 - \overline{FA1}$, etc.) required for the MSB generation as well as interpolation. Figure 2.16 shows these folding amplifier characteristics. To maintain consistency with the previous interpolation discussion, a fifth characteristic is shown, which is just the complement of the first difference folding amplifier output. Figure 2.16 shows that the endpoint distortion is less severe compared to Figure 2.14, due to the use of additional CDPs. Concentrating on the zero crossings of these signals, it becomes apparent that the input voltage range is subdivided into 32 equispaced intervals, which can be used for MSB extraction.

2.8 General FAI ADC Block Diagram

The general block diagram for the Folding and Interpolating (FAI) ADC model examined in this thesis is given in Figure 2.17. The input signal is first driven through a Sample and Hold (S/H) network, whose output is held at a constant level for one half of each clock cycle. The clock rate determines the overall conversion

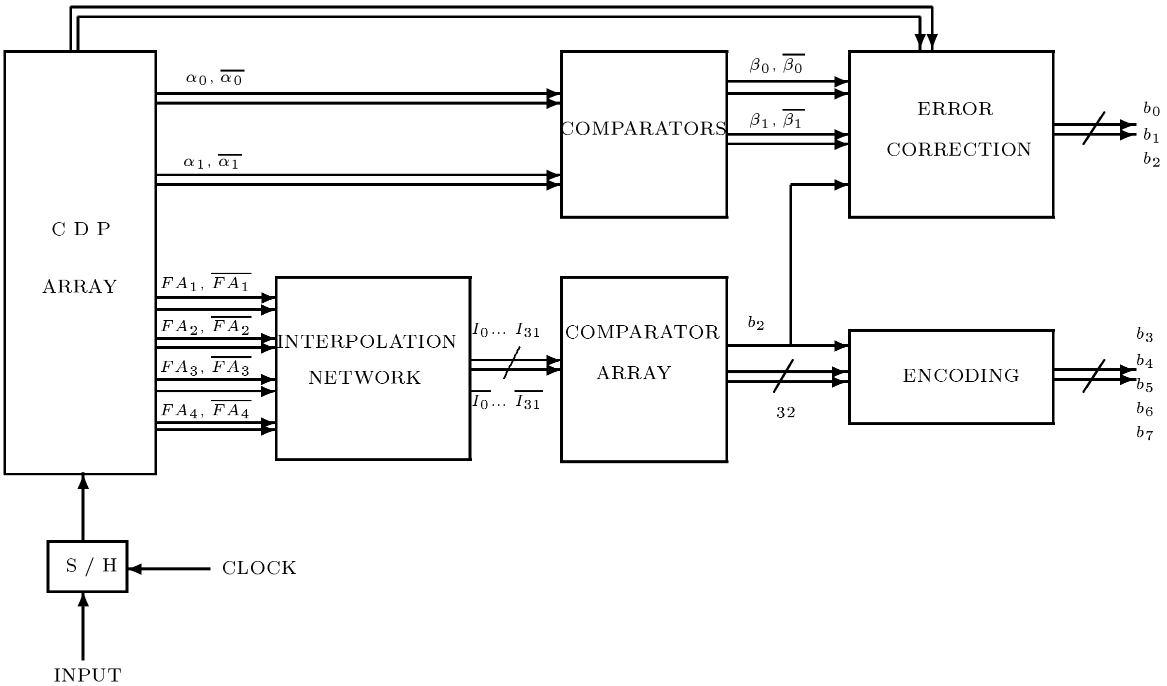

Figure 2.17: General Block Diagram for the Folding and Interpolating Converter

rate. The CDP array is portrayed in Figure 2.15. As explained earlier, the CDP array is responsible for the generation of the four folding signals ($FA1 - FA4$) and their respective complements ($\overline{FA1} - \overline{FA4}$). The CDP array is also used to obtain the signals $\alpha_0, \overline{\alpha_0}, \alpha_1$ and $\overline{\alpha_1}$. The latter signals are used for the generation of the most significant bits (MSBs) b_0 and b_1. Before deriving expressions for these signals, one should observe that the way the CDPs are located in the array allows for some overlap in their corresponding high and low regions [8]. In other words, to determine the b_0 pattern, the signal has to be between $-1.5V$ and $-1V$. So, to pick which CDP outputs to use for the b_0 determination, the CDPs in the array whose outputs are high when $-1.5 < V_{in} < -1$ have to be located. Obviously, A_5-A_8 have high outputs in that region, but so do C_5-C_7. Similarly, to formulate $\overline{b_0}$, A_1-A_4 and also C_1-C_3 can be used. In a similar fashion, the CDPs whose outputs are high when $-1.75 < V_{in} < -1.5$ and $-1.25 < V_{in} < -1$ can be located, for the formulation of b_1. The overlapping CDPs are summed to produce the corresponding signals, as shown in Equation 2.6

$$\begin{aligned}
\alpha_0 &= A_5 + A_6 + A_7 + A_8 + C_5 + C_6 + C_7 \\
\overline{\alpha_0} &= A_1 + A_2 + A_3 + A_4 + C_1 + C_2 + C_3 \\
\alpha_1 &= A_3 + A_4 + A_7 + A_8 + C_3 + C_7 \\
\overline{\alpha_1} &= A_1 + A_2 + A_5 + A_6 + C_1 + C_5 \qquad (2.6)
\end{aligned}$$

Clearly, there are more CDPs with high outputs in the appropriate regions that could be used for the determination of the signals in Equation 2.6, but that would require larger capacity summers. The C signals included in Equation 2.6 are preferred over other candidates with high outputs in appropriate regions since they are centered between the A signals used. These signals are fed into a comparator, giving approximations to the desired bits as shown in Equation 2.7.

$$\beta_0 = \begin{cases} 0 & , \alpha_0 < \overline{\alpha_0} \\ 1 & , \alpha_0 > \overline{\alpha_0} \end{cases} \quad \beta_1 = \begin{cases} 0 & , \alpha_1 < \overline{\alpha_1} \\ 1 & , \alpha_1 > \overline{\alpha_1} \end{cases} \tag{2.7}$$

Before determining the 3 MSBs, however, a very crucial problem needs to be addressed. Since the MSBs are formed virtually independently from the LSBs (different paths), there is a danger of the MSBs not being *synchronized* (valid at the same time instant) with the LSBs. Such a misalignment can cause large errors in the output word. Error correction is used to modify the MSB approximations β_0 and β_1 to give properly synchronized output bits. The intricacies of the MSB synchronization, along with some additional error correction necessary for the generation of the correct MSBs, are detailed in Section 2.9.

The generation of the least significant bits (LSBs) is now discussed. The CDP array provides four folding amplifier outputs and their complements. A total of five folding difference signals are available to interpolate between, namely $FA1-\overline{FA1}, FA2-\overline{FA2}, FA3-\overline{FA3}, FA4-\overline{FA4}$, and the complement of the first signal $\overline{FA1} - FA1$. Following the discussion that led to Figure 2.5, eight resistive interpolation networks are needed to obtain a total of 64 interpolated characteristics ($I_0 \ldots I_{31}$ and their complements). These signals are shown in Figure 2.18. For simplicity, only the difference signals $I_0-\overline{I_0}, I_1-\overline{I_1} \ldots I_{31}-\overline{I_{31}}$ are shown, and not the two components separately. The last of the interpolated characteristics, $I_{31} - \overline{I_{31}}$, is used to generate b_2. This curve is of paramount importance, since it is not only used for the MSB synchronization, but it is also used to determine the LSBs. LSB extraction is based upon identifying the number of interpolation characteristics that are above or below zero according to the state of b_2. Specifically, if b_2 is greater than zero, then the five remaining LSBs are the sum of the interpolated curves that are below zero. Conversely, if b_2 is less than zero, then the LSBs are the sum of the curves that are above zero. These facts are described

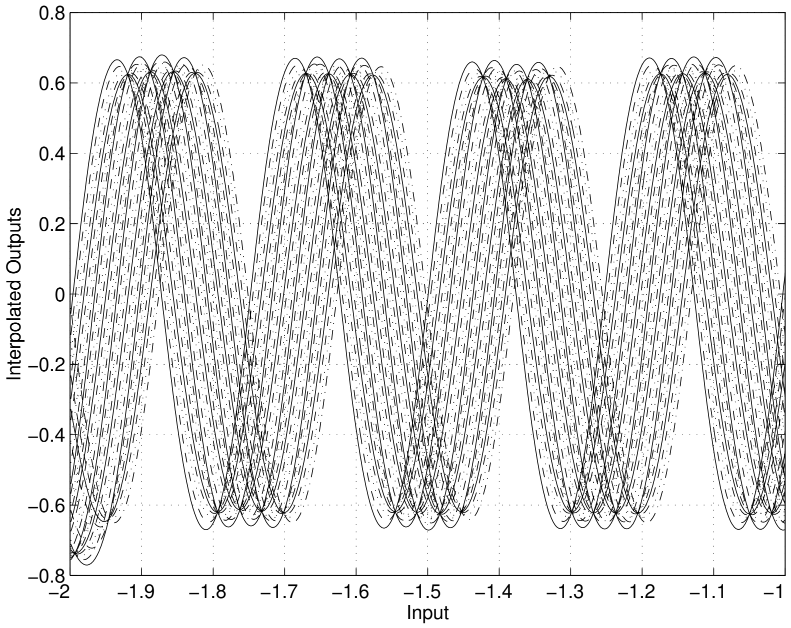

Figure 2.18: Interpolated Characteristics $I_0 - \overline{I_0}$ through $I_{31} - \overline{I_{31}}$

mathematically by Equation 2.8

$$LSB_{word} = \begin{cases} \sum_{k=0}^{k=30} \operatorname{sgn}(\overline{I_k} - I_k) & \text{if } I_{31} > \overline{I_{31}} \\ \sum_{k=0}^{k=30} \operatorname{sgn}(I_k - \overline{I_k}) & \text{if } I_{31} < \overline{I_{31}} \end{cases} \tag{2.8}$$

$$\text{where } \operatorname{sgn}(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}$$

The term LSB_{word} is the decimal equivalent value formed by the five bit binary number $b_3 b_4 b_5 b_6 b_7$. The equivalent LSB word is an integer between 0 and 31.

The final step in the LSB extraction is encoding the output so that the digital equivalent word is obtained (namely the five LSBs $b_3 ... b_7$). Exclusive OR functions are performed to determine the 0 to 1 transitions, thus yielding the output binary code.

2.9 MSB Error Correction

As discussed earlier, since the MSBs and LSBs are derived in different ways, there is a chance that they will have different delay paths. This effect, combined with existing noise and zero crossing uncertainties, can lead to erroneous output. To synchronize the MSBs with the LSBs, b_2 is used as a common reference. The signal b_2 can be derived from the folding amplifier FA1. Moreover, the LSBs are derived from interpolation between the four different folding amplifier outputs. Therefore, b_2 is in sync with the LSBs, and it can be used to ensure that the MSB and LSB code changes are concurrent. Since b_2 is in sync with the LSBs, it will be used as a reference for MSB correction. More specifically, the output is susceptible to errors when β_0 and β_1 are close to a state transition at a particular input value. To compensate for this, the outputs of certain CDPs from the CDP array that have high outputs in these dangerous input regions [8] are used. Figure 2.19 demonstrates the procedure. The curves on top correspond to β_0 and β_1, which come from the folding amplifiers. The output can have significant error in the regions where these signals change state, if they are not perfectly synchronized with the LSB changes. It can be shown that in these dangerous regions, CDPs C_2, C_4 and C_6 have high outputs as they are centered around these particular transition points. Therefore, whenever these CDPs turn high, the input is in a dangerous region, and error correction should be applied.

The key to applying appropriate error correction is the fact that only *one* of these three CDPs can be high at a given input, and therefore there is no ambiguity as to which region the input is currently in. More specifically, if C2 is high, then b_0 must be low, and assuming that b_2 is a valid time reference, b_1 must be the complement of b_2. These error correction steps are described mathematically in Equation 2.9

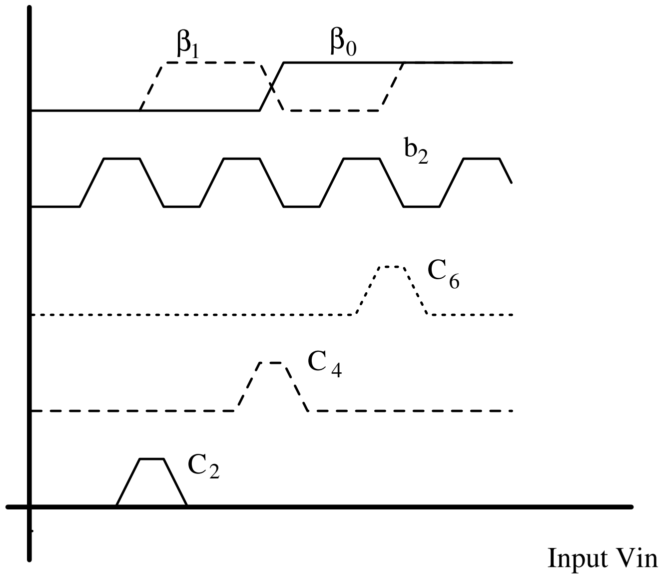

Figure 2.19: Dangerous MSB regions and corresponding CDP outputs

$$\left.\begin{aligned} b_0 &= 0 \\ b_1 &= \overline{b_2} \end{aligned}\right\} \quad \text{if } C_2 \text{ is high}$$
$$\left.\begin{aligned} b_0 &= \overline{b_2} \\ b_1 &= b_2 \end{aligned}\right\} \quad \text{if } C_4 \text{ is high}$$
$$\left.\begin{aligned} b_0 &= 1 \\ b_1 &= \overline{b_2} \end{aligned}\right\} \quad \text{if } C_6 \text{ is high} \tag{2.9}$$

After applying this error correction, there is one last step required before the conversion process is complete. A similar problem is also present at the low and high ends of the input range. To be precise, due to the CDP behavior, the MSBs tend to saturate at the end of the input range, while the LSBs wrap around. This produces major errors in the final output. To compensate for this phenomenon, the output code is limited to a maximum and a minimum word, namely 255 and 0 respectively. The same procedure as the one described in Figure 2.19 is used. C_0 and C_8 are centered around the low and high ends of the nominal input range respectively, therefore, b_2 is used to determine whether an overflow or an underflow has occurred. Equation 2.10 outlines the endpoint correction procedure.

$$\begin{aligned} &\text{If } C_0 \text{ is high and } b_2 = 1 && \Longrightarrow b_i = 0, && 0 \le i \le 7 \\ &\text{If } C_8 \text{ is high and } b_2 = 0 && \Longrightarrow b_i = 1, && 0 \le i \le 7 \end{aligned} \tag{2.10}$$

So finally, the analog to digital conversion subsystems have been described. The output bits have all been synchronized and corrected. Using this description, the inherent error mechanisms of a FAI ADC implementation are analyzed. Moreover, insight into the choice of appropriate basis functions to model these errors is presented along with methods to compensate for them.

CHAPTER 3

Error mechanisms for the FAI Converter

3.1 Simulation of the FAI converter

With the theory of the previous chapters as background, an efficient simulation for the Folding and Interpolating Analog to Digital Converter was devised. Simulation of the implementation shown in Figure 2.17 was carried out in MATLAB [9]. Inputs to the program include a vector (or matrix) of input data and the corresponding derivative (slope) of the input signal at the sample instants. The simulation returns a vector (or matrix) of digital samples of the analog input signal.

This chapter of the thesis studies error mechanisms in the converter and how they can be modeled and estimated. The ultimate goal is to develop experimental techniques to diagnose particular errors in a folding ADC using only the converter's output. Use of the simulation allows modification of converter parameters to excite or suppress various error mechanisms. Three different error sources are addressed:

- The resistive ladder used to generate the CDP reference voltages
- The interpolating resistive ladders used for LSB extraction
- The dynamic behavior of the coupled differential pairs.

A significant feature of the simulation is that the user can easily change the characteristics of any of these error mechanisms. Resistor values may either be individually specified or the user can specify nominal values and a standard deviation that will be used to assign random errors on each of the resistors.

Another important asset of the simulation is that contrary to a physical converter, the user has access to the simulated converter's input, since it is generated and stored in software. This allows the exact error function of the simulated converter to be determined. Computer routines have been developed that average the converter error as a function of the converter's output state. In other words, the *actual* converter error function can be extracted, which is useful for comparison with the estimated error characteristics.

The rest of this chapter is dedicated to investigating the errors due to variations in the CDP reference voltage resistive ladders and the interpolation resistive networks, as well as examining how the input signal frequency affects the performance of the Coupled Differential Pair.

3.2 Inherent FAI ADC error

Before examining the error mechanisms in detail, it is important to briefly discuss the inherent error associated with the FAI ADC of Chapter 2. The addition of the extra CDPs on the endpoints of the nominal input range, although effective, does not totally eliminate endpoint distortion. The setup shown in Figure 2.15, exhibits some distortion, particularly at the low end of the input range. Figure 3.1 shows the average static FAI ADC error as obtained from the simulation module.

Figure 3.1 shows that the error is centered around $-\frac{1}{2}$ LSB, which is the ideal quantizer error. The endpoint distortion is small, in the order of $\frac{2}{10}$ of an LSB. The error also demonstrates a periodic pattern, which is due to the interpolation between the non-linear CDP characteristics. These inherent FAI errors will be assumed in the following discussion, unless specifically noted otherwise. In some cases, the nonlinear CDP characteristics will be replaced with ideal CDP characteristics to eliminate these errors so that other mechanisms may be isolated.

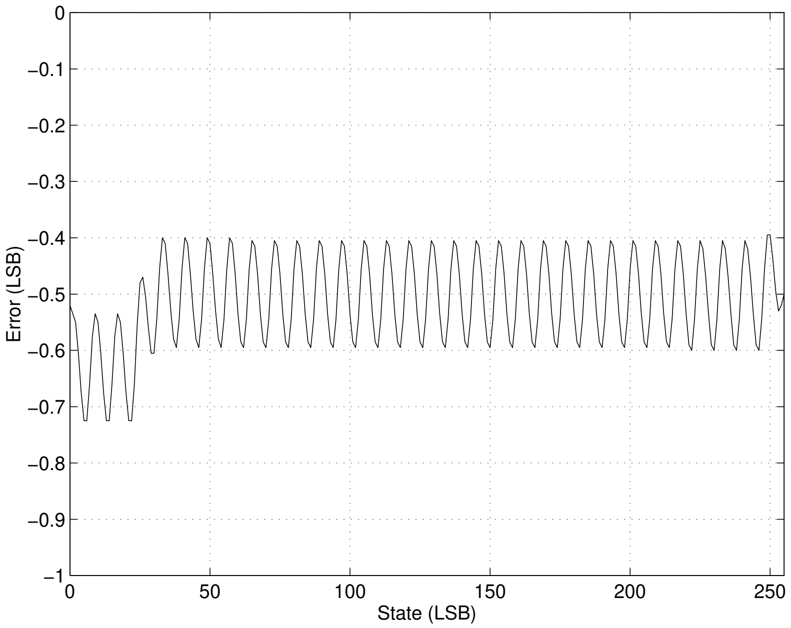

Figure 3.1: Inherent FAI ADC Average Error

3.3 Study of the CDP Voltage Reference Resistive Network

As discussed in Chapter 2, the core of the Folding converter is the CDP array. Since these CDPs are critical for the determination of the output word, their respective voltage references are crucial for the performance of the ADC. For the 8-bit FAI example introduced in Chapter 2, 44 distinct voltages are necessary for the operation of the CDPs, and they can be accurately derived from a voltage divider network with 44 resistors. Figure 3.2 shows the voltage divider network, and the corresponding nomenclature. As discussed in Chapter 2, this implementation of the FAI ADC has a nominal input voltage range of -2 to -1 Volts. Each CDP voltage reference is $\frac{1}{32}$ V apart from the next one, and the addition of extra CDPs at the endpoints of the input range calls for a maximum reference voltage of $-1 + \frac{1}{4} = -0.75$V and a minimum of $-2 - \frac{1}{8} = -2.125$V. These voltages would correspond to ADC output states 320 and -32 LSBs respectively. For simplicity, specifying values in terms of LSBs is preferable to Volts for some of the following

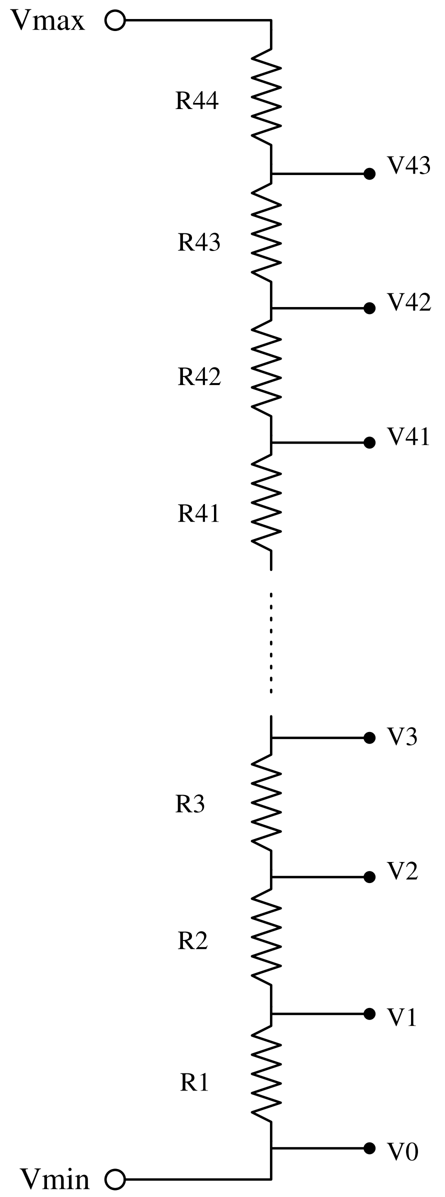

Figure 3.2: The Resistive Ladder used for the CDP Voltage References

discussions. Let these states be denoted by $x_b = 320$ LSB, and $x_a = -32$ LSB. As expected, these states are outside the nominal converter range of 0 to 255 LSB.

To model the effect of resistance deviation from nominal, an expression can be derived that relates the values of the resistors to errors in the required CDP voltage references. Let the value of each resistor in the voltage divider be

$$R_k = R_o + \epsilon_k \qquad , k = 1 \ldots M \tag{3.1}$$

where R_k is the kth resistor in the ladder, R_o is the nominal (ideal) value of the resistors, ϵ_k is the error of the kth resistor (in Ω), and M is the number of resistors in the ladder. Also let R_Σ denote the total resistance of the ladder, namely

$$R_\Sigma = MR_o + \sum_{k=1}^{M} \epsilon_k \tag{3.2}$$

Then the current I flowing through the ladder is given by

$$I = \frac{V_{max} - V_{min}}{R_\Sigma} \tag{3.3}$$

If V_μ denotes the μth output voltage, then it can be written as

$$V_\mu = V_{min} + \frac{V_{max} - V_{min}}{R_\Sigma} \sum_{k=1}^{\mu} R_k, \qquad \mu = 0, 1 \ldots M-1 \tag{3.4}$$

Now let $\mathcal{E}_\mu$ denote the error in Volts of the μth reference voltage, which is defined as the actual threshold voltage minus the ideal threshold voltage. This yields

$$\mathcal{E}_\mu = V_\mu - \left[V_{min} + \frac{V_{max} - V_{min}}{M}\mu\right] \tag{3.5}$$

Substituting Equation 3.4 for V_μ yields

$$\mathcal{E}_\mu = (V_{max} - V_{min}) \left[\frac{\sum\limits_{k=1}^{\mu} R_k}{R_\Sigma} - \frac{\mu}{M} \right] \tag{3.6}$$

Equation 3.6 describes the voltage reference error as a function of resistor values.

Insight into the expression may be gained by examining the effect of a *single* non-ideal resistor. Let R_m denote the resistor with error ϵ_m, so that all other resistors will have the nominal value R_o. For this case,

$$R_\Sigma = MR_o + \epsilon_m \tag{3.7}$$

Using this relationship in Equation 3.6 gives

$$\begin{aligned} \mathcal{E}_\mu &= (V_{max} - V_{min}) \left[\frac{\mu R_o}{MR_o + \epsilon_m} - \frac{\mu}{M} \right] & \text{if } \mu < m \\ \mathcal{E}_\mu &= (V_{max} - V_{min}) \left[\frac{\mu R_o + \epsilon_m}{MR_o + \epsilon_m} - \frac{\mu}{M} \right] & \text{if } \mu \geq m \end{aligned} \tag{3.8}$$

Here the error follows two distinct linear patterns depending on the value of the input. However, Equation 3.8 is cast in terms of μ, which is a quantized index corresponding to a particular voltage reference in the divider network. In actuality, the error is a function of the ADC input voltage x, a continuous variable. Therefore, although the two linear patterns still hold, the error will demonstrate a transition between the two linear curves for inputs that range between the discrete values of μ. Equation 3.8 predicts the error when the input voltage x is equal to a threshold voltage V_μ. For values of x between threshold voltages, the error will be an interpolated value determined in part by the interpolation network.

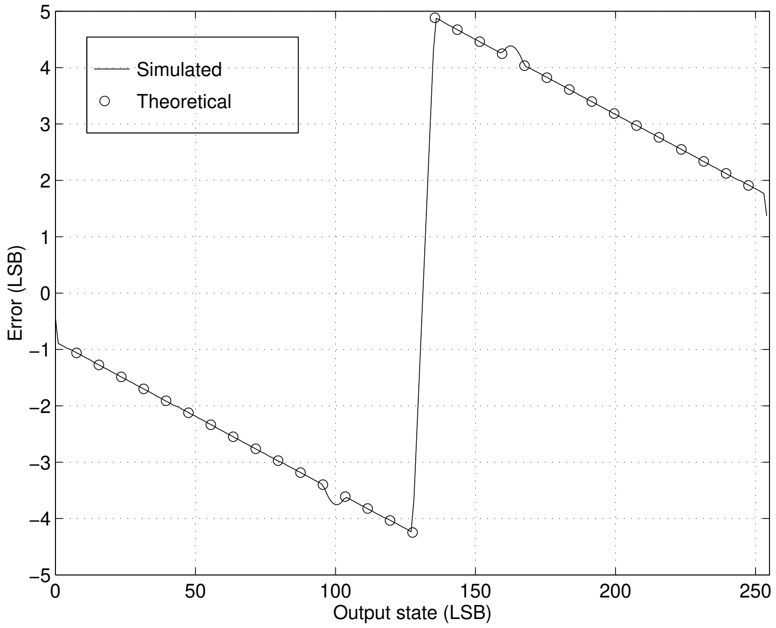

Figure 3.3: Theoretical and Simulated Average FAI ADC Error with $R_{21} = 2.2\ \Omega$

To demonstrate the above theoretical derivation for the 8-bit FAI converter of Chapter 2, a simulation was performed where the value of one resistor (R_{21}) deviated from nominal. The simulation assumed a nominal value of 1 Ω for all the resistors in the resistive ladder; whereas R_{21} was chosen to be 2.2 Ω. Since there is no frequency dependency for this kind of architectural flaw, the error is a function of the converter's input only. Figure 3.3 shows a plot of the average converter error for a given converter output state. To isolate the effect of the resistors that produce the CDP voltage references on the converter's behavior, the non-ideality of the CDP was omitted. Instead, the CDP was replaced by a perfect triangular shaped characteristic as discussed in Chapter 2. Perfect interpolation resistors were also assumed. Superimposed on the error characteristic are the discrete theoretical error values predicted by Equation 3.8. As expected, the error follows Equation 3.8 exactly. The transition between the two parts is 8 LSB wide, and accounts for equivalent values of the output that fall between states associated with $\mu = 20$ and $\mu = 21$. These correspond to states 128 to 135. It should be obvious that changing the value of a different resistor will yield a similar error function. The location of the transition band depends on the location of the non-ideal resistor in the ladder, while the amplitude and sign of the error depends on the amplitude and sign of the error for that resistor ϵ_m. Figure 3.3 also shows two small anomalies in the otherwise linear pieces of the error function. These are due to interpolation between two folding amplifier characteristics that do not have the same slope due to the non-ideal resistor.

3.4 Study of the Interpolation Network

As was discussed in Chapter 2, the fine bits of the converter's output word are determined through resistive interpolation between adjacent folding amplifier outputs. For the 8-bit FAI converter discussed in Chapter 2 there are 8 resistive interpolation networks responsible for the generation of the signals $I_0 \ldots I_{31}$ and $\overline{I_0} \ldots \overline{I_{31}}$. Figure 3.4 illustrates the interpolators and their corresponding outputs. The resistors in the network are labeled using two indices: the first corresponds to the particular interpolator (1 to 8), while the second index corresponds to the particular resistor *within* that interpolator. Each of the pairs of corresponding outputs (i.e. I_k and $\overline{I_k}$) are the inputs to a comparator.

In an approach similar to the one used in Section 3.3, assume that all resistors in one interpolator have a nominal value of R_0. Let I_i^{ideal} denote the ideal outputs of an interpolator, and I_i the actual outputs where only one resistor in the interpolator has an non-ideal value of $R(1,m) = R_0 + \epsilon_m$. Note that this discussion deals with the first interpolator, but identical equations hold for the rest. To derive a generic expression for the error in the output interpolated values, assume that each interpolator has a total of N resistors. Then in a fashion identical to the derivation of Equation 3.8, the error $\mathcal{I}_i$ in the ith interpolator output is given by

$$\mathcal{I}_i = I_i - I_i^{ideal}, \qquad i = 0 \ldots N-1 \tag{3.9}$$

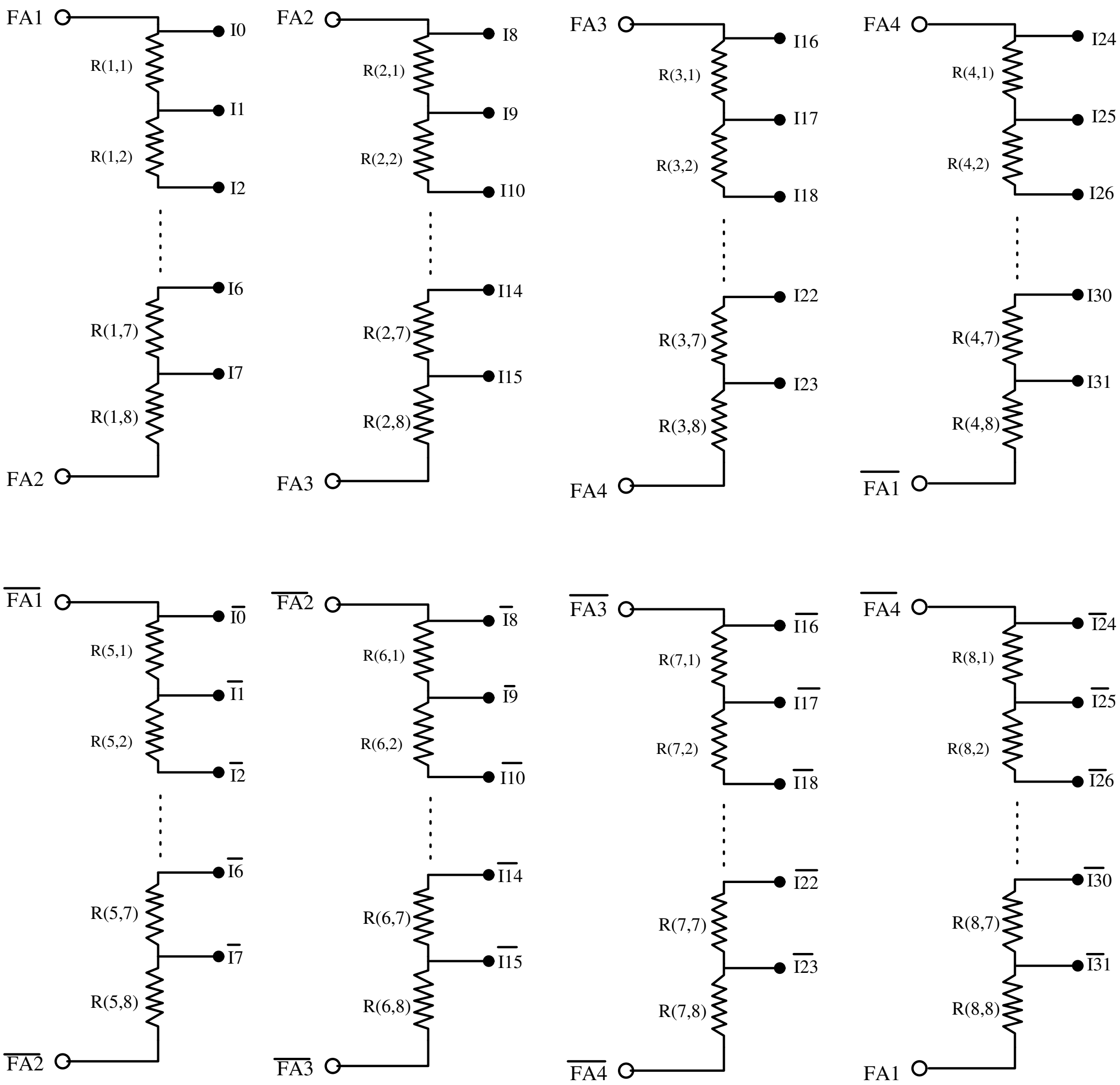

Figure 3.4: FAI Resistive Interpolation Network

$$\mathcal{I}_i = \begin{cases} (FA2 - FA1)\left[\frac{iR_0}{NR_0+\epsilon_m} - \frac{i}{N}\right], & i = 0,\ldots,m-1 \\ (FA2 - FA1)\left[\frac{iR_0+\epsilon_m}{NR_0+\epsilon_m} - \frac{i}{N}\right], & i = m,\ldots,N-1 \end{cases} \tag{3.10}$$

Similar equations hold for the other interpolators. Equation 3.10, like Equation 3.8, proves that the error is a piecewise linear function with two distinct linear patterns. However, it is important to note that unlike the CDP reference voltage resistive network, a single erroneous resistor in one interpolator will produce this piecewise linear error effect at *eight* regions of the converter's output. This is because one interpolator is responsible for producing the intermediate zero crossings between the full range of the two folding amplifiers being interpolated. This repetition is equal to the folding ratio of the converter, which is eight in this illustrative case.

To demonstrate the effect of the interpolation networks on the converter one resistor in one interpolator was changed to a value other than ideal. The simulation module was used where $R(1,5)$ had a value of $1.5R_0$ Ω, while all other resistors were R_0 Ω. Figure 3.5 shows the average converter output error for this set of interpolator values. Again, a triangular characteristic was used for the CDP resulting in idealized triangular folding amplifier outputs. This simulation result validates the original assumption of a piecewise linear error characteristic, much like the results obtained for the CDP voltage reference resistive network. However, this distinct pattern is now repeated a total of 8 times across the converter output range. As discussed before, this is due to the fact that the same resistive network interpolates between its two corresponding folding amplifier characteristics at 8 distinct intervals (equal to the converter's folding ratio).

The circles in Figure 3.5 are the actual simulation data points, while the solid lines connect the data points to show the overall trend. In this case, a 50% error in an interpolator resistor creates a maximum overall error of only $\frac{1}{5}$ LSB. Clearly the interpolation resistors have a much milder effect on the converter's

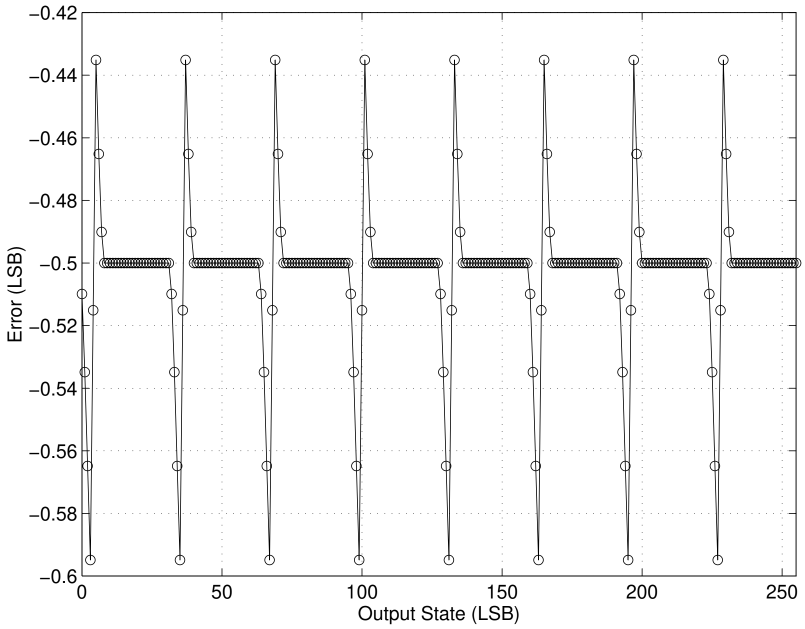

Figure 3.5: Average FAI ADC Error with $R(1,5) = 1.5\ \Omega$

behavior than the ones responsible for the CDP references. This makes sense because the interpolation resistors are responsible for generation of fine bits, as opposed to the CDP reference resistors that determine the coarse bits. Although the error patterns are similar (with a periodicity of eight in the interpolator case), the magnitude of the error is much smaller for the interpolation network.

3.5 CDP Dynamic Modeling

The distortion introduced by an analog to digital converter is a dynamic phenomenon. The Folding and Interpolating converter discussion in Chapter 2 was cast in a static sense, that is, the input signal's frequency did not affect the behavior of the converter. The scope of this section is to examine how the FAI ADC performance is affected by the dynamics of the input signal (e.g. the input signal frequency). This study is of paramount importance, since folding converters are designed specifically to operate at high frequencies.

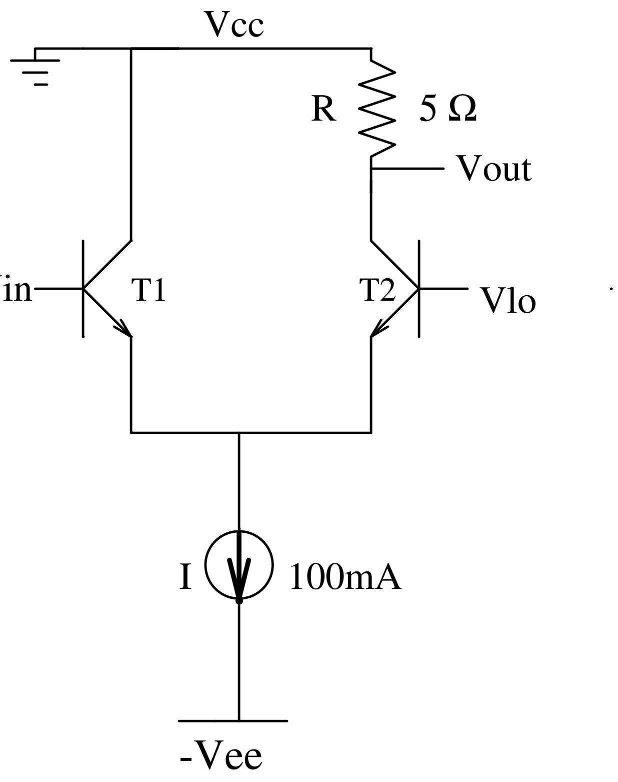

Figure 3.6: A single Differential Pair

3.5.1 Dynamic Model for the Differential Pair

Chapter 2 introduced the coupled differential pair (CDP), which is the key element of the folding converter architecture. Since the two differential pairs that form one CDP act independently of each other, one can concentrate on the study of one differential pair, and then simply subtract their outputs to obtain the composite CDP characteristic.

Figure 3.6 shows a single differential pair. Since the device is meant for high frequency, ECL levels have been adopted, where $V_{cc} = 0$ V and $-V_{ee} = -5.2$ V. Since V_{cc} is tied to ground, it provides a steady and reliable reference. For the model examined in this thesis, a nominal value of 100 mA is used for the current source I. Equation 2.2 gives the current i_2 going through transistor T2 from collector to emitter.

$$i_2 = \frac{I}{1 + e^{(V_{in} - V_{lo})/V_T}} \tag{3.11}$$

Letting $x = V_{in} - V_{lo}$, it is possible to express Equation 3.11 in an equivalent hyperbolic tangent form, namely

$$\frac{V_{out}}{R} = -i_2 = -\frac{I}{2} + \frac{I}{2}\tanh\left(\frac{x}{2V_T}\right) \tag{3.12}$$

Since a full circuit simulation of the entire FAI converter would be impractical for inclusion in the simulation module (very computationally expensive and time consuming), circuit simulation will be performed on a single CDP. Since a CDP consists of two identical differential pairs, the simulation will be limited to a single differential pair. The composite CDP output is just the difference of the two differential pair outputs. These results will then be parameterized, yielding an efficient simulation for the CDP that can be incorporated in the FAI converter simulation module.

A model of the differential pair shown in Figure 3.6 was constructed using the MicroCap circuit simulation program [10]. The resistor R was selected to be 5 Ω, while the input voltage range was $-2V$ to $-1V$. For initial simulations, the voltage reference, V_{lo}, was set to be equal to $-1.5V$ (the midrange voltage for the ADC model). The simulation used the $2N3647$ bipolar transistor with a forward beta of 150. The maximum slew rate for the input signal for this choice of transistor model was $500\frac{V}{\mu sec}$.

The transistor slew rate limits the bandwidth of the FAI converter. The bandwidth limitation may be examined by finding the slew rate requirements for a sinusoidal input signal. Since the input is chosen to range between $-2V$ and $-1V$, then an appropriate input signal is given by

$$V_{in} = -1.5 + 0.5cos(\omega t) \tag{3.13}$$

Since the input slew rate is the derivative of V_{in}, the maximum allowable input frequency is given by

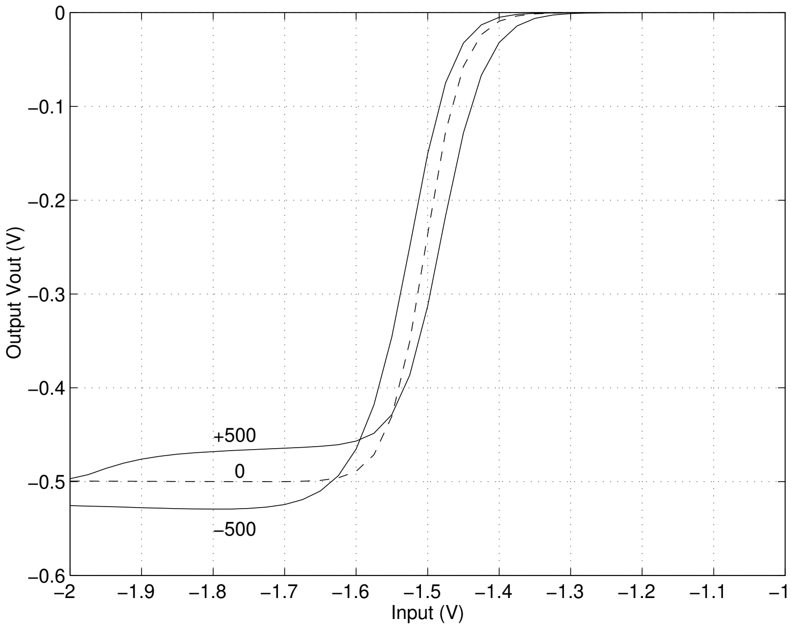

Figure 3.7: Differential Pair Characteristics for three simulated runs

$$500\frac{V}{\mu sec} = 2\pi f_{max}(0.5V)$$
$$f_{max} = \frac{500}{\pi}MHz \tag{3.14}$$

To characterize the dynamic behavior of the differential pair, the simulations utilized a set of triangle functions to set different input signal slopes, namely -500, -300, -150,-50, 0, 50, 150, 300 and $500\frac{V}{\mu sec}$. The results were then exported from MicroCap to MATLAB. Figure 3.7 shows three characteristics, obtained for $-500, 0$, and $500\frac{V}{\mu sec}$.

These results indicate that the top of the characteristics are set at zero and do not vary with frequency, because V_{cc} is tied to ground. The rest of the family of responses exhibits a horizontal and vertical shift with slope. The zero-slope characteristic (dashed) was found to be nearly identical to Equation 3.12. This shift decreases as the input slew rate decreases, as shown in Figure 3.8. This figure shows the differential pair transfer characteristic for input slew rates of $+50$ and

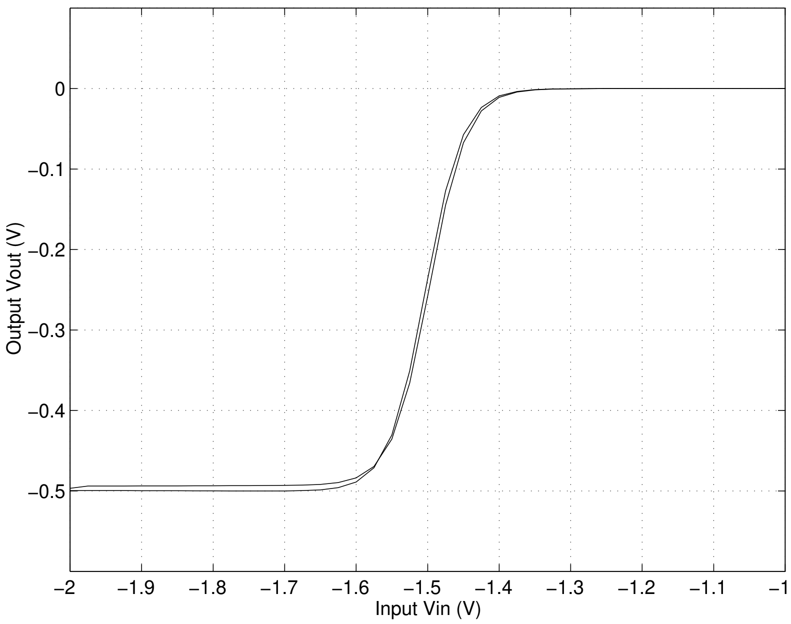

Figure 3.8: Differential Pair Characteristics for $\pm 50\frac{V}{\mu sec}$

$-50\frac{V}{\mu sec}$. Figure 3.9 shows an intuitive way to illustrate the varying behavior of the differential pair by using a 3-D plot of the various input signal amplitudes and slew rates and the respective outputs.

3.5.2 Parameterization of Simulation Results

The simulation results allow for the construction of a generic differential pair response. Figure 3.10 shows a typical differential pair transfer function. The following study of the dynamics of the differential pair will concentrate on how this characteristic varies with signal dynamics. The goal is to be able to predict the CDP response in terms of the input signal value and its rate of change. It is important to note that since V_{cc} is tied to ground, the top part of the output response will always approach to zero asymptotically for large V_{in}. However, the midpoint and the end of the curve change depending on the rate of change of the input signal. As shown in Figure 3.10, the lateral drift of the midpoint is signified by the parameter c_2. This parameter shifts the curve left or right, while another parameter, c_1 is used to raise or lower the bottom of the curve. Equation 3.12 may

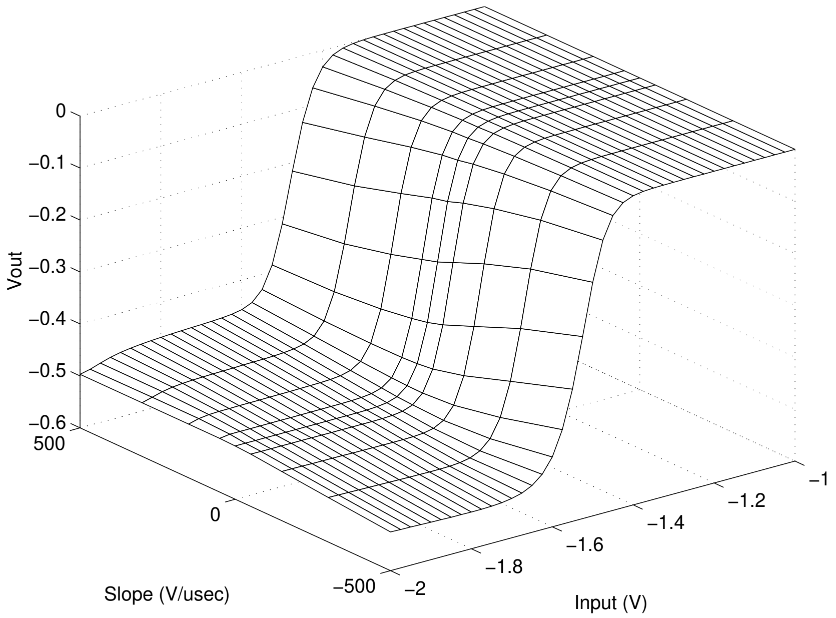

Figure 3.9: Differential Pair Characteristics for different inputs

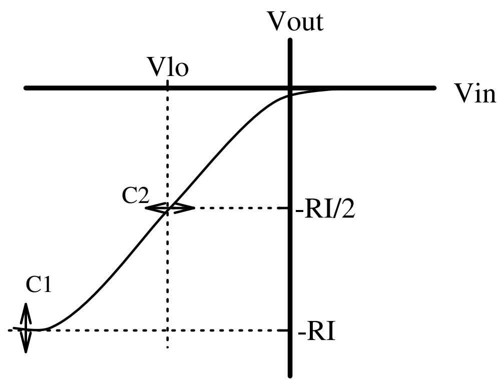

Figure 3.10: Typical Differential Pair characteristic

be modified to include these parameters.

$$-i_2 = c_1 \left[-\frac{I}{2} + \frac{I}{2} \tanh\left(\frac{x - c_2}{2V_T} \right) \right] \tag{3.15}$$

For a 5 Ω resistance, the output voltage is given by $-5i_2$, or for $I = 100$ mA and $V_T = 26$ mV,

$$V_{out} = c_1 \left[\frac{1}{4} \tanh\left(\frac{x - c_2}{0.052} \right) - \frac{1}{4} \right] \tag{3.16}$$

For a static input, $c_1 = 1$ and $c_2 = 0$, Equation 3.16 becomes the theoretical behavior predicted in Equation 3.12. To create an efficient MATLAB simulation, the relationship between the input signal slope and the value of c_1 and c_2 must be determined.

3.5.3 Determination of c_1

In determining the characteristic-shaping parameter, c_1, it is useful to find a region where the other parameter, c_2, is not affecting the overall shape of the curves. Referring back to Figure 3.9 we see that around $V_{in} = -1.8V$, and regardless of the input's slope, c_1 is the only parameter that comes into play, by raising the asymptote of the curves up and down. It is therefore at this input level that an empirical estimate for the value of c_1 is made. To achieve this, a curve fit was applied to the simulation data.

More specifically, solving Equation 3.16 for $V_{in} = -1.8V$, $V_{lo} = -1.5V$, and further assuming that for this input amplitude $c_2 = 0$, we obtain

$$V_{out} = c_1 \left[0.25 \tanh\left(\frac{(-1.8 - (-1.5))}{0.052} \right) - 0.25 \right] \tag{3.17}$$

The hyperbolic tangent evaluates to -1, so that

$$
\begin{aligned}
V_{out} &= c_1(-0.5) \\
c_1 &= -2V_{out} \qquad \text{for small } V_{in}
\end{aligned}
\tag{3.18}
$$

The procedure assumes that c_2 has no effect on the asymptote of the hyperbolic tangent for large negative input. It is important to note that Equation 3.18 only holds for the V_{out} data collected through simulation for the particular set of inputs specified above. If we now define y to be the slope of the input signal

$$
y = \frac{dx}{dt} \tag{3.19}
$$

then we can fit a curve to the data to estimate coefficients of a polynomial in y that will fit data from Equation 3.18. The polynomial in question for this model was picked to be first order. The empirical solution for c_1 is given by

$$
c_1 = -1.3436 \times 10^{-4} y + 0.9939V \qquad \text{for} -500 \leq y \leq +500\frac{V}{\mu sec} \tag{3.20}
$$

As a validation of the approximation given by Equation 3.20, the coefficient c_1 for the conditions of Equation 3.18 is plotted in Figure 3.11 for different slopes. Superimposed on the plot are c_1 estimates as given by Equation 3.20. Clearly, the first order approximation of Equation 3.20 is sufficiently accurate for modeling c_1.

3.5.4 Determination of c_2

With c_1 determined, an expression for c_2 can be derived. In an analogous manner, a region is located for the input amplitude where the differential pair characteristic changes the most. Referring to Figure 3.9 it is clear that if the input signal is at the voltage reference, V_{lo}, or $V_{in} = -1.5V$, then the parameter

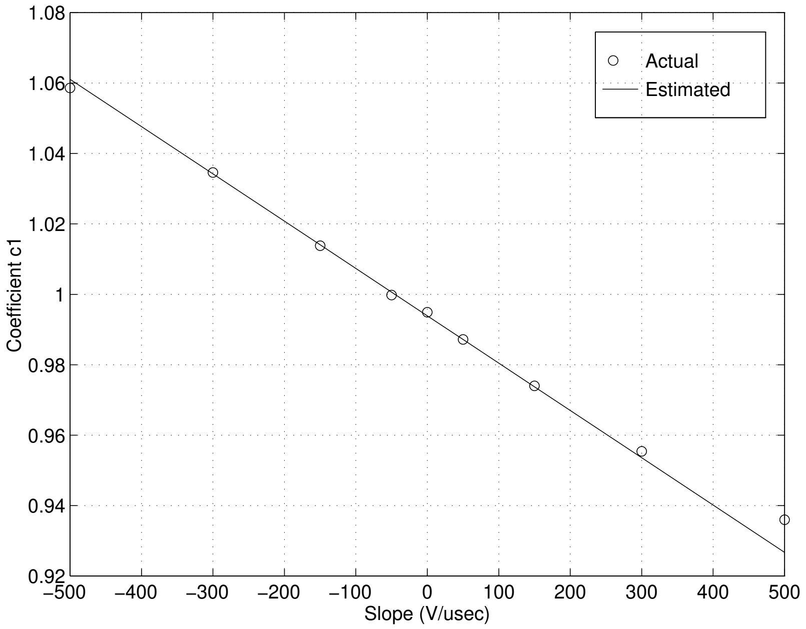

Figure 3.11: Actual and Estimated c_1 for different input signal slopes

c_2 can be efficiently estimated. Solving Equation 3.16 for c_2 at $V_{in} = -1.5V$ and $x = V_{in} - V_{lo} = 0$, we obtain

$$\begin{aligned} V_{out} &= c_1 \left[0.25 \tanh \left(\frac{-c_2}{0.052} \right) - 0.25 \right] \\ c_2 &= -0.052 \tanh^{-1} \left(1 + \frac{4V_{out}}{c_1} \right) \end{aligned} \tag{3.21}$$

As before, this equation only holds for the particular choices of input conditions. Curve fitting was used again to produce the coefficients of a third order polynomial in y that fits the MicroCap simulation data for these inputs. Prior to this determination of polynomial coefficients, the estimate of c_1 given in Equation 3.20 was substituted into Equation 3.21. All the processing was done by computer, and the resulting polynomial expression for c_2 is given by

$$c_2 = -3.1125 \times 10^{-11} y^3 - 9.7390 \times 10^{-9} y^2 + 5.0243 \times 10^{-5} y - 4.7094 \times 10^{-4} V \tag{3.22}$$

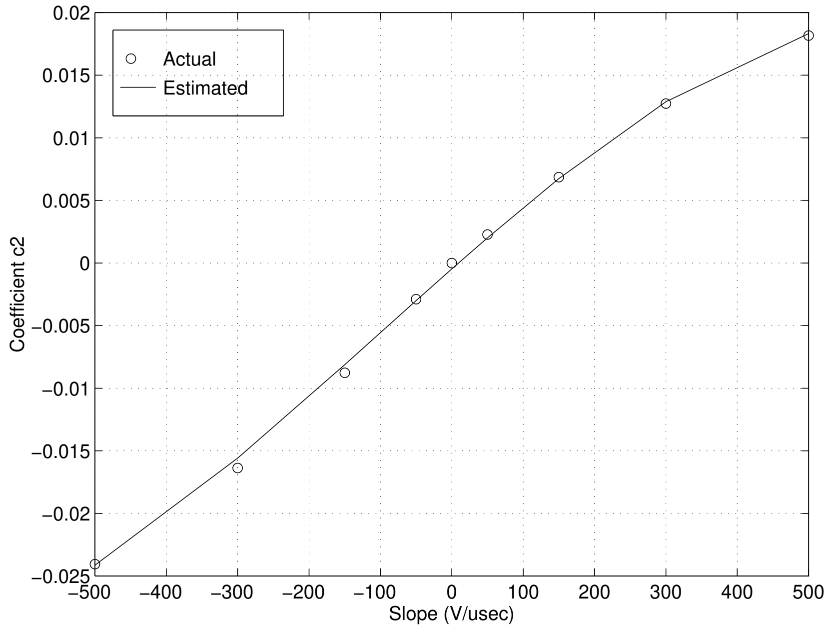

Figure 3.12: Actual and Estimated c_2 for different input signal slopes

In a fashion similar to the validation of the c_1 parameterization, Figure 3.12 shows the actual coefficient c_2 for the conditions of Equation 3.21 for a variety of input signal slopes. Superimposed on this plot is the coefficient c_2 as estimated by Equation 3.22. The results confirm that the third order approximation of c_2 in Equation 3.22 is an accurate model.

Since both Equations 3.20 and 3.22 were obtained by using data that span a particular set of frequencies, they only hold for input slew rates between $\pm 500 \frac{V}{\mu sec}$.

3.5.5 Validation of Parameterization

In confirmation of the effectiveness of c_1 and c_2 in simulating the CDP behavior, a test was performed to compare a MicroCap simulated CDP response to the response predicted by the parameterized model. A full CDP circuit simulation was performed in MicroCap, and the response to a 50 MHz sinusoidal input signal was obtained. The input was a sinewave so that the converter could be tested for a variety of different slopes. Using the same input signal, parameters c_1 and c_2 were employed to produce an estimated response. The responses are plotted in Figure

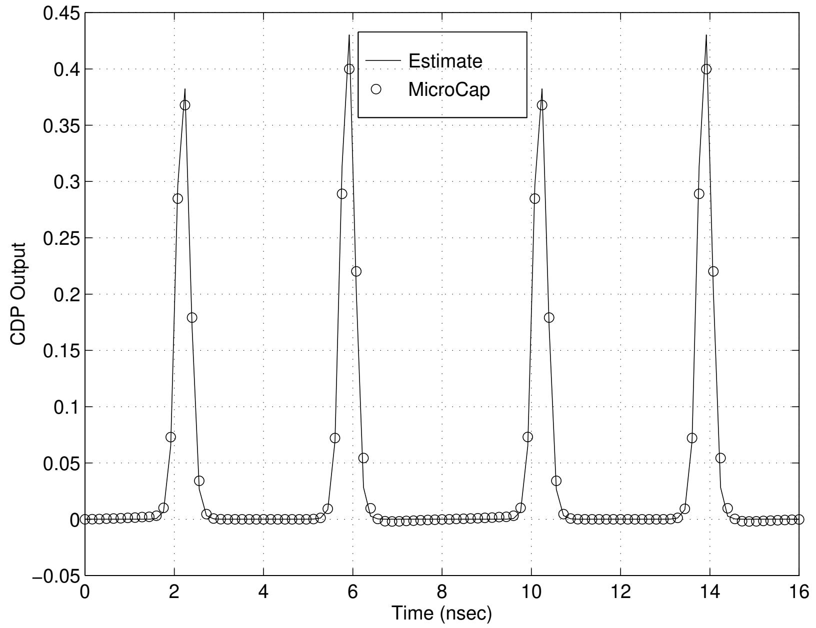

Figure 3.13: Comparison of MicroCap CDP Response and Estimated Response using parameters c_1 and c_2

3.13. The two responses are nearly identical, proving that the coupled differential pair can be accurately modeled using the parameterized model given by Equation 3.16.

3.5.6 Incorporation of Dynamic Model in FAI ADC Simulation

The empirical determination of the parameters c_1 and c_2 allows for the incorporation of dynamic (frequency-dependent) effects in the FAI ADC simulation module. A simulation was performed using the dynamic model for the CDP, while all resistors in the voltage divider networks were ideal. Since the error is now a function of both the ADC state and the input signal's slope, a 3-D graph of the error function is necessary. Figure 3.14 shows the error caused by the dynamic behavior of the coupled differential pair to be a fairly smooth function of state and slope. As expected, the effect is much more severe for high frequencies. One should observe that a dominant part of the error function shown in Figure 3.14

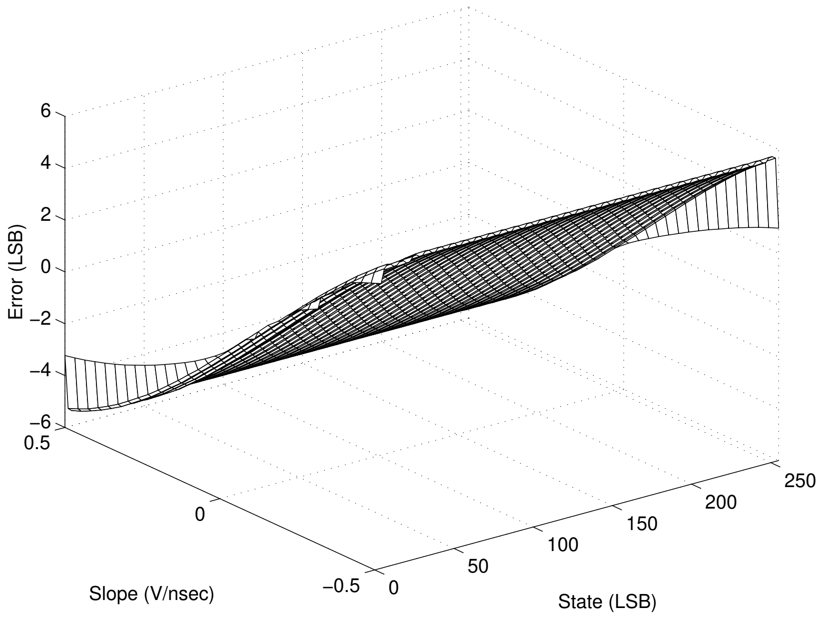

Figure 3.14: Simulated FAI ADC Error due to CDP Dynamic Behavior

has a constant slope, i.e. it is a plane. This plane corresponds to a constant time delay in the converter response. Such a delay, however, is a linear error (much like a constant gain) that does not introduce harmonic distortion to the converter's output.

CHAPTER 4

FAI Characterization

4.1 ADC Characterization Fundamentals

In a device as complicated as an Analog to Digital Converter, simple input/output measurements do not always fully describe its behavior. A more precise indicator of the converter's performance is its Spurious Free Dynamic Range (SFDR). This section explains one way to characterize the dynamic error of an Analog to Digital Converter, and describes test procedures that have been developed to facilitate the characterization.

4.1.1 Spurious Free Dynamic Range

The distortion introduced by ADCs is a dynamic phenomenon, and a good measurement of this distortion is the SFDR [11]. The SFDR is measured by driving the converter using a pure (no harmonics) sinusoidal input, and measuring the harmonic distortion introduced in the sampling process. The spurious free dynamic range is defined as the separation (in dB) between the signal fundamental and the largest magnitude harmonic in the converter's output spectrum [11]. Figure 4.1 shows a simulated converter output spectrum, plotted against frequency, up to *Nyquist* (one half the sampling frequency).

Clearly the fundamental, the highest tone in the spectrum, is near 68 dB, while the next highest component is at 31 dB, thus yielding an SFDR of $68 - 31 = 37$ dB. ADC manufacturers typically limit distortion power levels to yield SFDR of approximately $6n$ dB, where n is the number of bits in the converter. SFDR in practice means that if the converter is driven by a full scale sinusoidal input, the converter will introduce spurious harmonics at a maximum power level of $6n\ dB$ below the input power level.

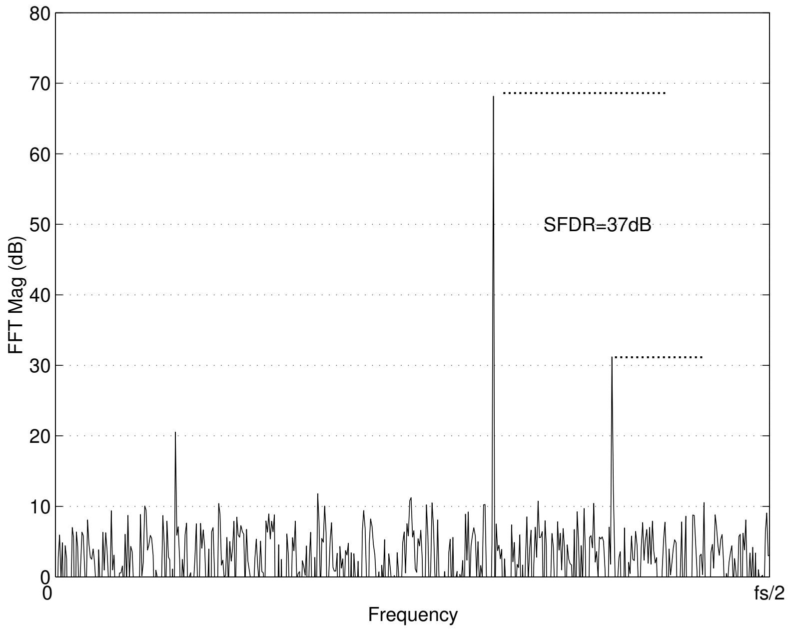

Figure 4.1: Example of SFDR calculation

4.1.2 Dynamic Converter Characterization

Recent developments in ADC characterization have introduced new procedures for characterizing ADCs and increasing their dynamic range [17, 20, 21]. All of these techniques are based on the assumption that the error introduced by the converter is dependent not only on the state of the converter, but on the derivative of the input signal as well. Since the converter-caused distortion is a complex dynamic phenomenon, a simple fixed correction at the output would only improve the SFDR near the calibration frequency. To overcome the distortion frequency dependence, dynamic compensation has been introduced. This technique bases the determination of the output error correction on the current ADC output and an estimate of the slope of the input signal at the instant that the sample was obtained. Figure 4.2 illustrates the approach. The key element of this procedure is the creation of a table whose entries contain the correction values for a particular set of state and derivative estimates. This error table is typically stored in a memory accessible by the state and slope of the input signal. The slope can be determined either by processing the output sequence, or by including extra analog

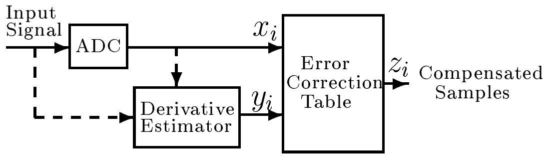

Figure 4.2: ADC Compensation Architecture

circuitry or even converters so that the slope can be determined directly from the input.

For the compensation scheme to work, an ADC calibration procedure is required. This procedure determines the entries of the error correction table. To successfully calibrate a converter, a multitude of calibration signals (various amplitudes and frequencies) is required so that the device is densely excited in the state-slope plane, which is often referred to as the *phase-plane*. The formation of the calibration signal can be achieved in at least two ways. The first involves selecting a set of input signals with different frequencies and amplitudes so that most pairs of state and slope are included, while the second only requires the use of a single two-tone calibration signal, where the two signals intermodulate to produce a dense set of state-slope excitations [17].

4.1.3 Compensation Formulation

Maintaining the primary assumption that the error introduced by the ADC is a function of the converter's state and the slope of the sampled signal, one can formulate the compensation procedure in a straightforward manner.

Assume that x_i represents the ith sample obtained from the ADC, and y_i the estimate of the slope of the input signal used to access the table. Let the compensated signal be denoted by z_i, where

$$z_i = x_i - e(x_i, y_i) \tag{4.1}$$

The goal is to linearize the output of the converter so that z_i is as close to the original input sequence as possible. As discussed above, a set of input signals is required for converter calibration. However, to facilitate the discussion, it will be initially assumed that there is only one calibration signal. This signal is a pure sinusoid with known frequency f_0, whose exact amplitude and phase are generally *unknown*, since the various gains and delays in the input signal path cannot be accounted for. Let $\vec{x} = [x_1\ x_2\ \ldots\ x_N]^T$ be a vector of ADC samples collected by sampling the sinusoidal input at a sampling frequency, f_s, and let $\vec{y} = [y_1\ y_2\ \ldots\ y_N]^T$ denote the vector of the signal derivative estimates. Using this terminology, Equation 4.1 becomes

$$\vec{z} = \vec{x} - \vec{\mathrm{e}}(\vec{x}, \vec{y}) \tag{4.2}$$

Since the ultimate goal is to linearize the converter output, the distortion in $\vec{z}$ must be minimized. The calibration procedure minimizes the energy present in the harmonics of the converter's output spectrum. In solving for the error function, $e(x, y)$ is expressed as a linear combination of L known basis functions $b_i(x, y)$.

$$\mathrm{e}(x, y) = \sum_{i=1}^{L} \alpha_i b_i(x, y) \tag{4.3}$$

The optimization problem is now focused on selecting the best values for the coefficients $\alpha_1,\ \alpha_2,\ \ldots,\ \alpha_L$. Evaluation of the ith basis function at each of the calibration pairs (x_i, y_i) yields the vector $\vec{b}_i(\vec{x}, \vec{y})$, where

$$\vec{b}_i(\vec{x}, \vec{y}) = \begin{bmatrix} b_i(x_1, y_1) \\ b_i(x_2, y_2) \\ \vdots \\ b_i(x_N, y_N) \end{bmatrix} \tag{4.4}$$

In this way, the vector of error table values becomes

$$\vec{e}(\vec{x},\vec{y}) = \sum_{i=1}^{L} \alpha_i \vec{b}_i(\vec{x},\vec{y}) \tag{4.5}$$

$$= \widetilde{B}\vec{\alpha} \tag{4.6}$$

where $\widetilde{B}$ is a $N \times L$ matrix with (i,j) element given by $b_j(x_i,y_i)$, and $\vec{\alpha} = [\alpha_1\ \alpha_2\ \ldots\ \alpha_L]^T$. The basis function representation of the error function $e(x,y)$ lends itself to a least-squares solution [21]. It is important to note that although the above discussion was cast in terms of a single calibration signal, it is easy to extend it to the number of signals used in practice. As mentioned above, the motivation behind using multiple calibration signals is to excite as many state-slope pairs (x,y) as possible. In such a case, each of the calibration signals gives a system of equations according to Equation 4.6. Efficient algorithms have been developed to determine the appropriate coefficients α_i [21, 17].

The successful compensation of a converter yields a good estimate of its error function. This provides a good model for the mechanisms in a device that cause distortion. Therefore the compensation scheme can be exploited to produce an effective characterization or diagnosis of a converter. If the basis functions selected have the shape of possible error functions, then the dominant error mechanisms present in the converter can be identified and traced back to particular architectural flaws.

4.2 Selection of Basis Functions

This section focuses on deriving appropriate basis functions to account for the error mechanisms described in Chapter 3. The basis functions are necessary for the implementation of the ADC characterization procedure developed in Section 4.1.3. Selection of such basis functions enables the converter error function to be expressed in the form given by Equation 4.3. Once L basis functions, $b_i(x,y)$, are

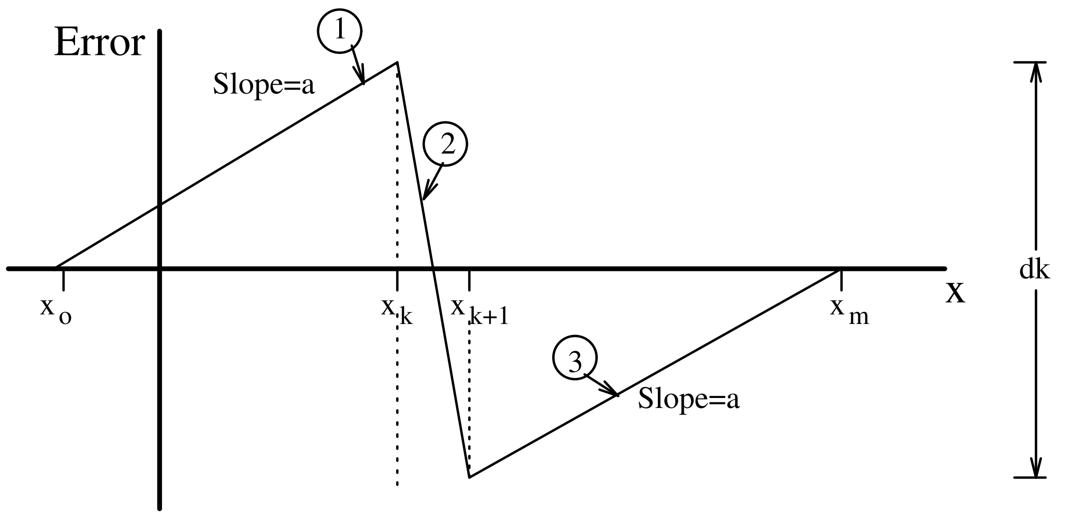

Figure 4.3: Typical Error caused by a single CDP reference voltage network resistor

selected, computer algorithms are utilized to obtain corresponding coefficients, α_i, that minimize the harmonic content in the sample sets for each calibration signal.

4.2.1 CDP Reference Voltage Resistor Network

The study of the resistive network in Chapter 3 that produces CDP reference voltages along with the corresponding simulations, showed that the overall converter error caused by a single non-ideal resistor in the voltage divider network is a piecewise linear function of the output state with no signal slope dependence. Figure 4.3 shows a typical error function caused by a single resistor. In Figure 4.3, x represents the converter's state, and x_o and x_m the states corresponding to the minimum and maximum levels respectively at the bottom and top of the voltage divider circuit of Figure 3.2. The point at which the converter error changes over from one linear pattern to the next one is called x_k. The width of the transition region is equal to the interpolation rate of the converter, denoted R_I. For the 8-bit FAI converter modeled in this thesis, the transition period between the two linear pieces of the error function is 8 LSB long. It is clear that the basis function necessary to account for such an error is a function like the one shown in Figure 4.3. Since the calibration algorithms return one coefficient for each basis function, the basis function will be cast in terms of the unknown coefficient d_k. This variable corresponds to the height of the basis function and depends on the magnitude of the error in the particular resistor. If d_k is positive, then the resistor in question is smaller than its nominal value, and the error function has the exact same form as Figure 4.3. If d_k is negative, then the resistor is greater than its nominal value and the error function is inverted.

For a voltage division network containing M resistors, M different basis functions should be chosen to account for possible errors in all of the resistors. All of these functions will have the same shape, but the value of x_k will be different depending on which resistor is in error. There are, therefore, M distinct values for x_k, which are given by Equation 4.7

$$x_k = x_o + i(\frac{x_m - x_o}{M}) \qquad i = 0 \ldots M-1 \tag{4.7}$$

Figure 4.3 divides the basis function in three distinct pieces, namely regions 1, 2, and 3. As Equation 3.8 shows, the two linear pieces have the same slope, denoted here by a. Let $\mathcal{F}(x)$ denote the basis function in question. Then $\mathcal{F}(x)$ in the three different regions becomes

$\underline{REGION1}$ $(x_o \leq x < x_k)$

$$\mathcal{F}(x) = a(x - x_o) \tag{4.8}$$

$\underline{REGION2}$ $(x_k \leq x < x_k + R_I)$

$$\mathcal{F}(x) = -\frac{d_k}{R_I}(x - x_k) + a(x_k - x_o) \tag{4.9}$$

$\underline{REGION3}$ $(x_k + R_I \leq x < x_m)$

$$\mathcal{F}(x) = a(x - x_m) \tag{4.10}$$

Taking advantage of the right triangles formed in Figure 4.3, the slope a can be expressed in terms of the height d_k, as dictated by Equation 4.11

$$a = \frac{d_k}{x_m - x_o - R_I} \tag{4.11}$$

Substituting Equation 4.11 in Equations 4.8, 4.9, and 4.10 yields

$$\mathcal{F}(x) = \begin{cases} d_k \left[\frac{x - x_o}{x_m - x_o - R_I}\right] & \text{for } x_o \leq x < x_k \\ d_k \left[-\frac{x - x_k}{R_I} + \frac{x_k - x_o}{x_m - x_o - R_I}\right] & \text{for } x_k \leq x < x_k + R_I \\ d_k \left[\frac{x - x_m}{x_m - x_o - R_I}\right] & \text{for } x_k + R_I \leq x < x_m \end{cases} \tag{4.12}$$

The algorithm will return the value of d_k for each of the M basis functions that minimizes the harmonic content of the converter calibration data.

4.2.2 Interpolation Resistive Networks

As Figure 3.5 proves, the error caused by a single resistor in an interpolation ladder is a piece-wise linear function with a periodicity equal to the folding ratio (8 in this case). Since the LSBs of the output word are determined by driving the signals I_k and $\overline{I_k}$ into a comparator, it is impossible to distinguish if an error appearing at the output is due to an erroneous resistor in the interpolator responsible for the generation of I_k or the one responsible for $\overline{I_k}$. However, it is possible to derive a set of basis functions that will model all possible errors caused by the interpolation resistor networks. Figure 4.4 shows the appropriate basis function modeling the error caused by an erroneous $I_k - \overline{I_k}$ pair. It is clear that there should be a distinct basis function for each of the $I_k - \overline{I_k}$ pairs.

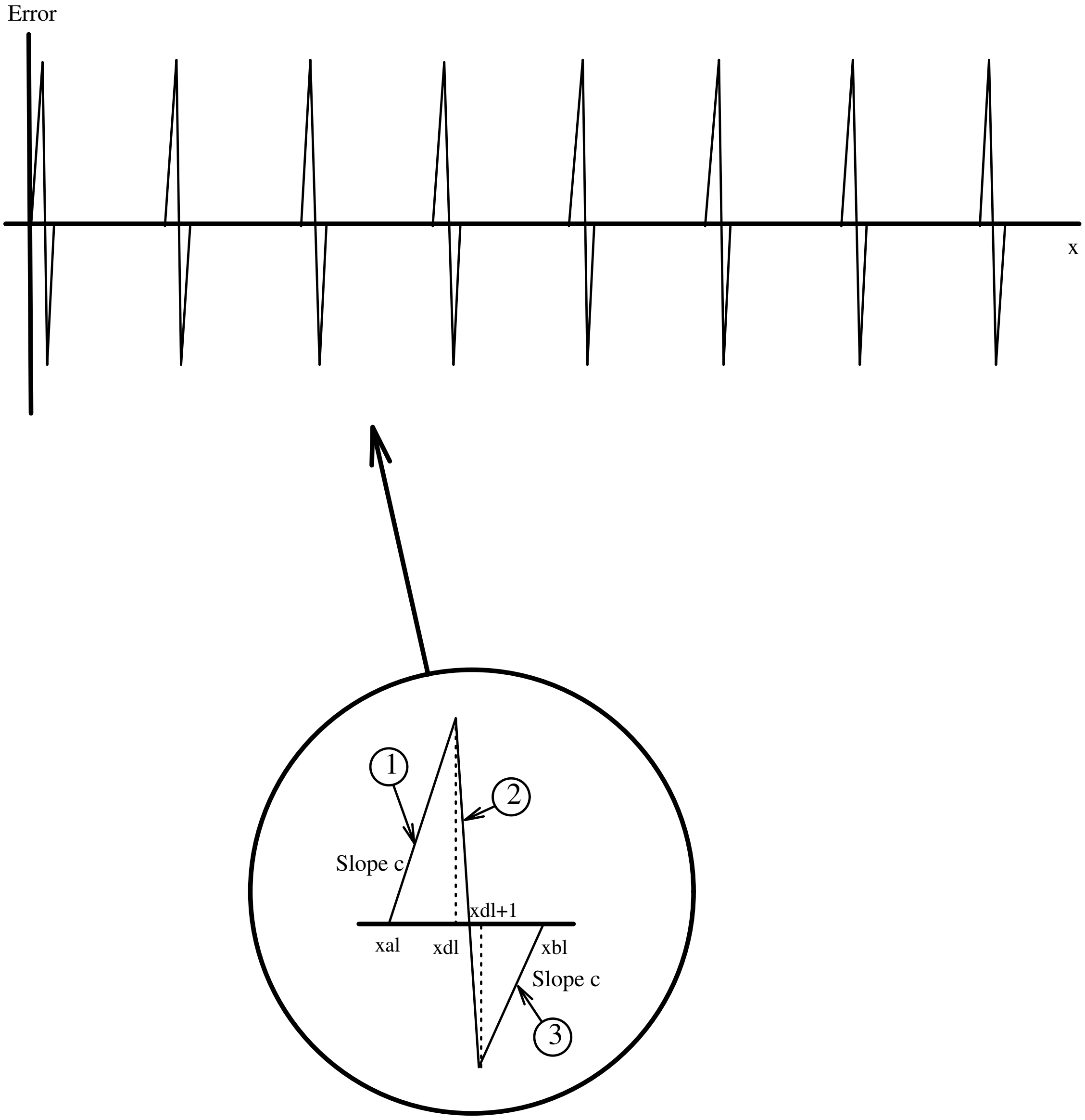

Figure 4.4: Basis Function modeling the error in an $I_k - \overline{I_k}$ pair

Since an error in $I_k - \overline{I_k}$ will result in an error function with periodicity F_R, let $l = 0, 1, \ldots, F_R - 1$ be defined as that periodicity index. Then for a given k, a set of F_R distinct x_a values exist. Let x^k_{al} denote the lth value of x^k_a, for $l = 0, 1, \ldots, F_R - 1$ and x^k denote the kth converter's output (in LSBs). Then an expression determining x^k_{al} can be written as

$$\begin{aligned} x^k_{al} &= R_I p + 32l \\ p &= \text{floor}\ (\frac{k}{R_I}) \end{aligned} \tag{4.13}$$

The variable p corresponds to the particular pair of folding amplifiers being interpolated by the kth resistor. The corresponding value of x_b is given by

$$x^k_{bl} = x^k_{al} + R_I \tag{4.14}$$

For example, the 8-bit FAI converter introduced in Chapter 2 has an interpolation rate, $R_I = 8$, yielding a total of 32 interpolation signals $I_k - \overline{I_k}$, $k = 0, \ldots, 31$. The folding ratio is $F_R = 8$, making each of the interpolation error signals have periodicity 8 (for example, see Figure 3.5). Equation 4.13 gives values for $p = 0, 1, 2$ and 3, indicating which pair of the four folding amplifier signals is being interpolated by the kth resistor. The index l corresponds to which of the 8 "pulses" of the error characteristic is being considered ($l = 0, 1, \ldots, 7$). The lth pulse begins at x^k_{al} given in Equation 4.13 and ends at x^k_{bl} given in Equation 4.14. The location of the discontinuity for the lth pulse is given by Equation 4.15

$$\begin{aligned} x^k_{dl} &= x^k_{al} + q \\ q &= k \bmod R_I \end{aligned} \tag{4.15}$$

The basis function will now be cast in terms of the unknown slope c. This slope is the unknown coefficient that the ADC calibration algorithm must determine. Equation 4.16 gives the expressions for the three different regions in the kth basis function, denoted by $\mathcal{G}_k(x)$

$$\mathcal{G}_k(x) = \begin{cases} c(x - x_{al}^k) & x \leq x_{dl}^k \\ c(q - (R_I - 1)(x - x_{dl}^k)) & x_{dl}^k < x < x_{dl+1}^k \\ c(x - x_{bl}^k) & x \geq x_{dl+1}^k \end{cases} \tag{4.16}$$

It is understood that Equation 4.16 gives the expression for the basis function for values of x between x_{al}^k and x_{bl}^k for each value of l. The basis function is assumed to be zero elsewhere.

4.2.3 Coupled Differential Pair Dynamics

The simulation shown in Figure 3.14 suggests that the error caused by the dynamic behavior of the CDPs is a smooth function of state and slope. It is clear that an appropriate basis function to account for such an error phenomenon can be a polynomial in state and slope (x, y). The order of the polynomial is not critical, however, it must be large enough to cover the 3rd order effects imposed by the coefficient c_2.

For this particular simulation, a 3rd order in state and 5th order in slope was selected. This gives a total of $(3+1) \times (5+1) = 24$ unknown coefficients. This polynomial is denoted by $\mathcal{H}(x, y)$, and its general form is given in Equation 4.17.

$$\begin{aligned} \mathcal{H}(x, y) \;=\; & \alpha_{00}x^0y^0 + \alpha_{10}x^1y^0 + \ldots + \alpha_{30}x^3y^0 \\ & +\alpha_{01}x^0y^1 + \ldots + \alpha_{31}x^3y^1 \\ & \vdots \\ & +\alpha_{05}x^0y^5 + \ldots + \alpha_{35}x^3y^5 \end{aligned} \tag{4.17}$$

The polynomial selected for $\mathcal{H}(x, y)$ yields 24 basis functions whose coefficients are estimated by the ADC calibration algorithm.

Table 4.1: FAI Simulation Parameters

Parameter	Value
CDP Reference Resistors	2% standard deviation
Interpolation Resistors	2% standard deviation
Coefficient c_1	Equation 3.20
Coefficient c_2	Equation 3.22

4.3 Simulated Results

Sections 4.2.1, 4.2.2 and 4.2.3 outlined appropriate choices of basis functions to model three distinct error phenomena as described in Chapter 3. For this particular configuration a total of $44 + 32 + 24 = 100$ basis functions are used, resulting in 100 unknown coefficients. The simulation module was modified to include all three error mechanisms. Table 4.1 shows the simulation parameters used to model errors. An efficient way to solve for the coefficients of the basis functions is the *Fast Orthogonal Search* documented in[22]. This technique chooses only those basis functions, out of the candidate ensemble (100 in this case), that significantly reduce the overall solution error. The user specifies the maximum number of functions to be selected, and the algorithm picks the ones that contribute the most to error minimization. Using this algorithm, an error table (function of state and slope) was constructed that minimized the ADC harmonic distortion of a set of converter samples. These samples were obtained by driving the FAI ADC simulation module with a set of pure sinewaves of different frequencies and amplitudes, and storing the resulting digital outputs. These converter outputs are referred to as the *calibration data.* Figure 4.5 shows the resulting error table. Although the error is a continuous function of state and slope, the user is limited to a discrete value error table. The error table in Figure 4.5 has a total of 128 possible entries for the state and a total of 127 slope entries for each state. Therefore the size of the error table is referred to as 128×127.

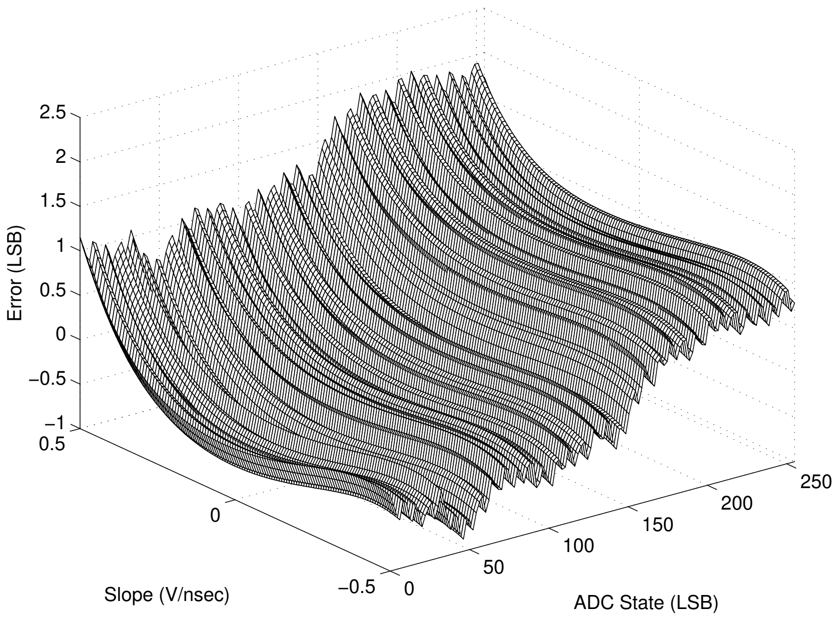

Figure 4.5: Error Table for FAI ADC using selected basis functions

In verifying the validity of the error table created, a set of test data from the converter was used to compare its harmonic content before and after the error correction. Figure 4.6 shows an uncompensated output sample spectrum, for samples of a pure sinusoidal signal at approximately 160 MHz. The composite effect of all three error mechanisms distorts the converter's output, yielding a Spurious Free Dynamic Range of about 55dB. Next, the error table of Figure 4.5 was used to compensate each data point of the test sample sequence. The resulting spectrum for the compensated samples is shown in Figure 4.7, where it is seen that the distortion has disappeared from the spectrum. The compensated data spectrum shows that the error table generated by using appropriate basis functions accurately represents distortion introduced by the converter. The compensated SFDR is approximately 65 dB, yielding a total improvement of around 10 dB over the uncompensated case. Compensation clearly eliminated all important harmonics of the signal.

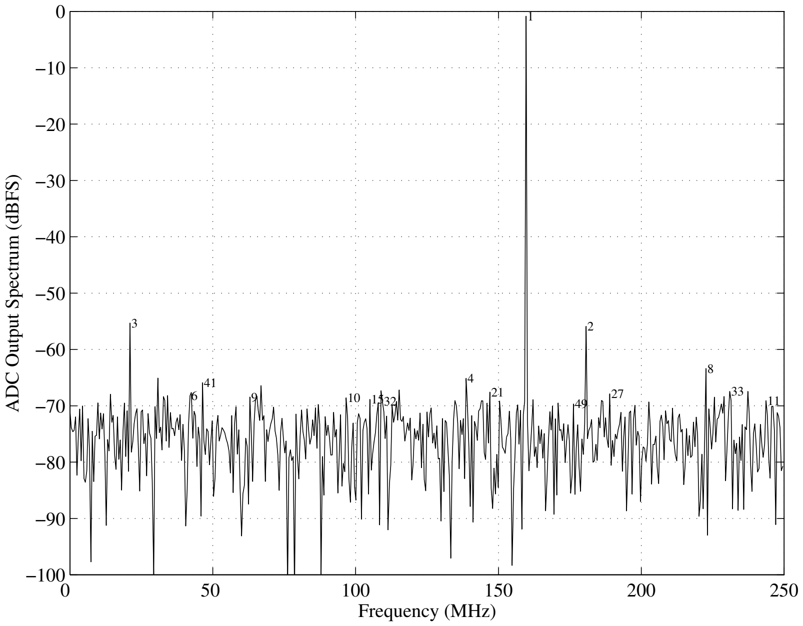

Figure 4.6: Uncompensated FAI ADC Output Spectrum

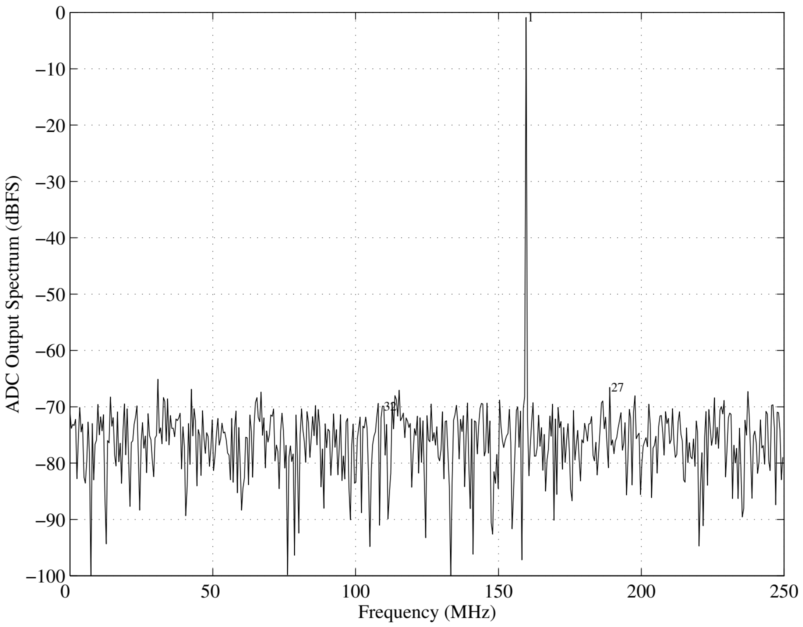

Figure 4.7: Compensated FAI ADC Output

4.3.1 Isolation of Error Phenomena

Figure 4.7 showed that an error table generated for this particular example is accurate and effective in minimizing the converter's harmonic distortion. Although the basis functions selected as candidates corresponded to particular error phenomena, one still needs to determine whether the combined effect of all three different error sources acting together impedes the basis functions from only extracting the errors they represent. Ideally, the basis functions should be impervious to cross coupling from other error mechanisms (with different error functions). For example, the CDP reference voltage basis functions should only isolate that effect even though the interpolators and the CDP's exhibit non-idealities simultaneously.

CDP Reference Voltage Functions

To determine how well the basis functions derived in Equation 4.12 match the errors caused by non-ideal resistors in the voltage divider network, an expression relating the coefficient that the compensation algorithm solves for with the error in a particular resistor is sought. Equation 3.8 gives the error in a particular output tap of the CDP reference voltage resistive network, while Equation 4.12 gives the best choice for a basis function modeling such an error. Equating the first clauses of Equation 3.8 and 4.12 yields

$$(V_{max} - V_{min})\mu \left[\frac{R_o}{MR_o + \epsilon_m} - \frac{1}{M}\right] = d_k \left[\frac{x - x_o}{x_m - x_o - R_I}\right] \tag{4.18}$$

The ADC output corresponding to a given value of μ is $x = x_o + R_I\mu$. Then Equation 4.18 yields

$$(V_{max} - V_{min})\mu \left[\frac{R_o}{MR_o + \epsilon_m} - \frac{1}{M}\right] = d_k \left[\frac{R_I\mu}{x_m - x_o - R_I}\right] \tag{4.19}$$

Now if Equation 4.19 is applied to the particular implementation studied in this thesis, where the actual values for M, R_I, V_{min} and V_{max} are substituted, it can be

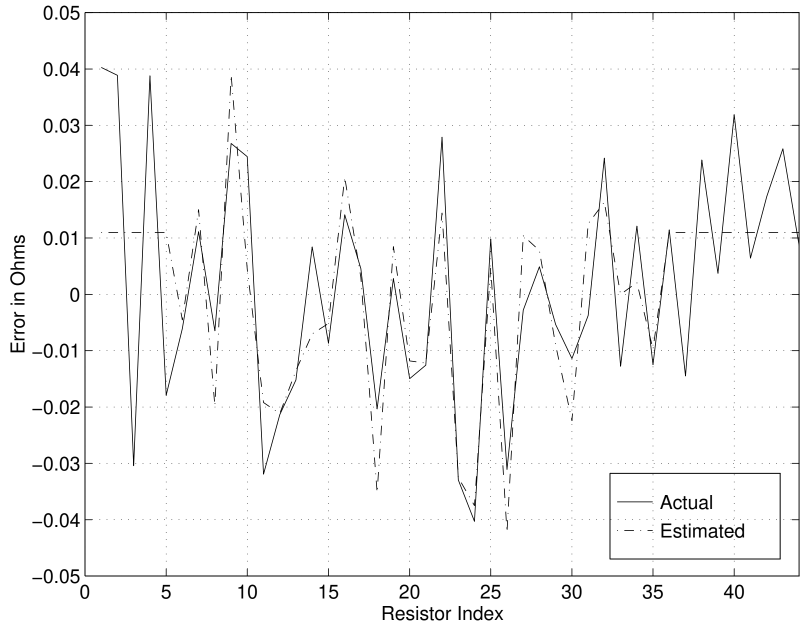

Figure 4.8: Actual vs Estimated Errors in CDP Reference Resistors

shown that

$$\frac{\epsilon_m}{R_o} = \frac{15136}{d_k + 344} - 44 \tag{4.20}$$

Equation 4.20 is of paramount importance. It proves that there is a one-to-one relationship between the coefficients for the basis functions chosen and the ratios of the particular resistors to their nominal values. Therefore one is able to determine the ratio $\frac{\epsilon_m}{R_o}$ strictly by processing the output of a FAI ADC with this architecture.

To demonstrate this point, the coefficients returned by the Fast Orthogonal Search algorithm were retained, and the 44 that corresponded to the CDP voltage reference resistive network were applied to Equation 4.20. For this example, the nominal value of the resistors was known (1Ω), and the actual errors on each one of the 44 resistors was also known (the actual resistor values were stored in memory). Figure 4.8 shows the actual error in Ω for each one of the 44 resistors and the error from the dynamic compensation method. The plots in Figure 4.8 show that the original assumption of being able to diagnose exact errors in the resistive ladder responsible for the CDP reference voltages is accurate. The estimated errors match the actual errors remarkably well. It is important to note that the basis functions corresponding to the top and bottom of the ladder were not chosen, since there is no calibration data available for these points because they fall outside the converter's useful range of -2 to -1 V.

Interpolation Network Basis Functions

The procedure followed in Section 4.3.1 provided a way of tracing actual resistor errors by processing the corresponding basis function coefficients estimated by standard calibration procedures. However, this technique cannot be applied to the interpolation network resistors. This is because the intermediate zero crossings used to determine the LSBs are formed by feeding the complimentary signals I_k and $\overline{I_k}$ into a comparator. The complementary signals originate from different interpolation networks, and their combined effect is used to determine the zero crossings. Therefore it is impossible to distinguish whether an error in the kth zero crossing is due to an erroneous resistor in the I_k ladder or the $\overline{I_k}$ ladder. This ambiguity hinders the diagnosis of the exact erroneous resistor, rendering the technique of Section 4.3.1 unfeasible. However, since the interpolation networks are responsible for the LSB generation, the errors on the converter output due to the interpolation resistors are insignificant compared to the other sources of errors. Figure 4.9 shows the actual error of the FAI ADC used in this thesis where the interpolation network resistors were deviating from their nominal value with a standard deviation of 2%. All other error mechanisms for this converter were suppressed in this run. The resulting errors are on the order of a few *hundredths* of an LSB, and therefore not easily detectable in the presence of errors on the order of LSBs.

CDP Dynamic Functions

In investigating how well the polynomial basis functions given by Equation 4.17 isolate the errors due to the dynamic CDP behavior, the basis function coeffi-

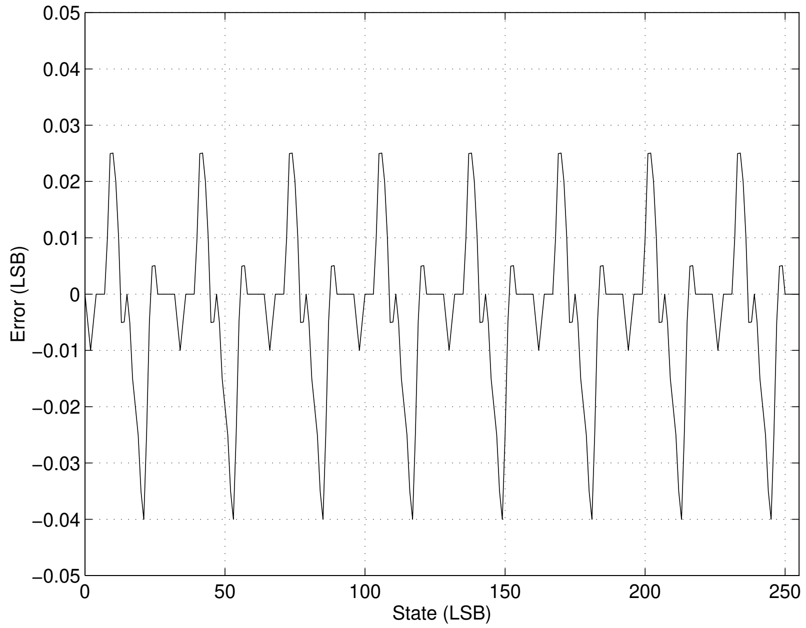

Figure 4.9: FAI ADC Error due to 2% standard deviation errors in Interpolation Network Resistors

cients for the case of all three error mechanisms acting together were retained. All coefficients not related to the polynomial basis functions of Equation 4.17 were set to zero, and the remaining 24 polynomial coefficients (α's) were used to construct an error table of size 128×127 as discussed before. The resulting error table is plotted in Figure 4.10

To compare the error table of Figure 4.10 with the actual error table obtained when the CDP dynamic effects are the only sources of error, there is a necessity to remove the slope from the actual error table. The actual error table can be obtained for this simulation since the exact input is known and stored in memory. The constant slope present in error tables does not contribute to harmonic distortion on the converter output, and can therefore be removed. Figure 4.11 shows the actual error table minus the constant plane. For clarity purposes, the endpoints of the state range are not shown, since the dominant error there is the inherent non-ideality of the CDPs. It is important though to note that the polynomial basis functions will account for the inherent sinusoidal FAI error described in Section 3.2. As a final check, the two error functions are compared by

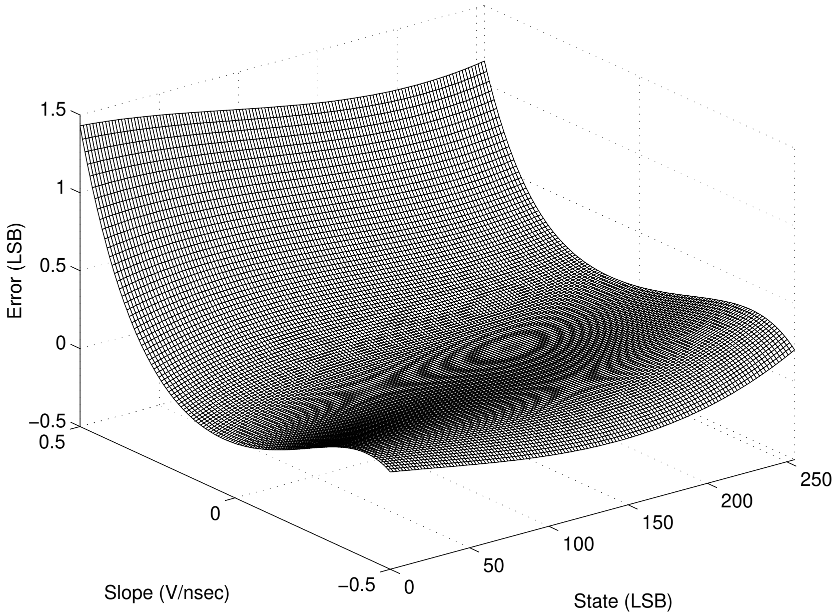

Figure 4.10: FAI ADC Error Table using only the polynomial basis functions coefficients

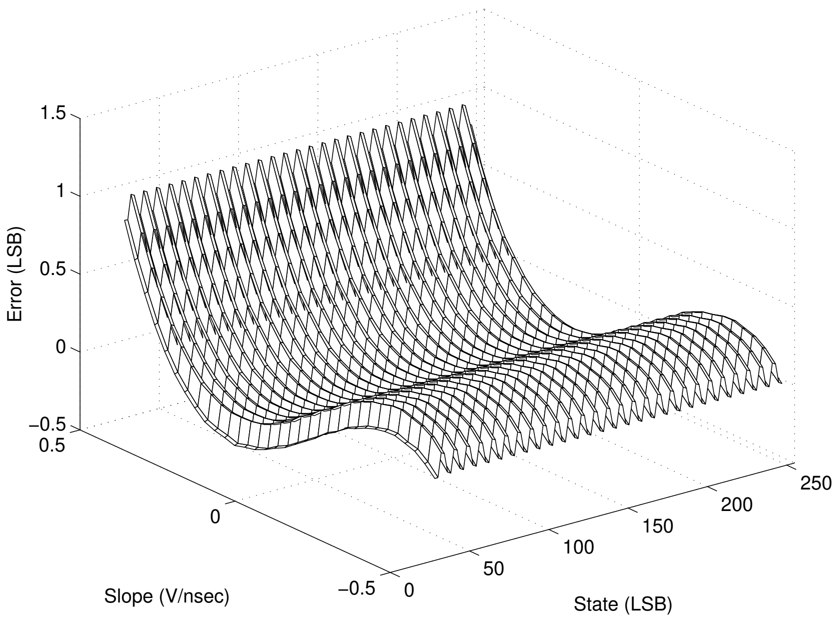

Figure 4.11: Actual FAI ADC Error Table due to CDP dynamic effects with plane subtracted

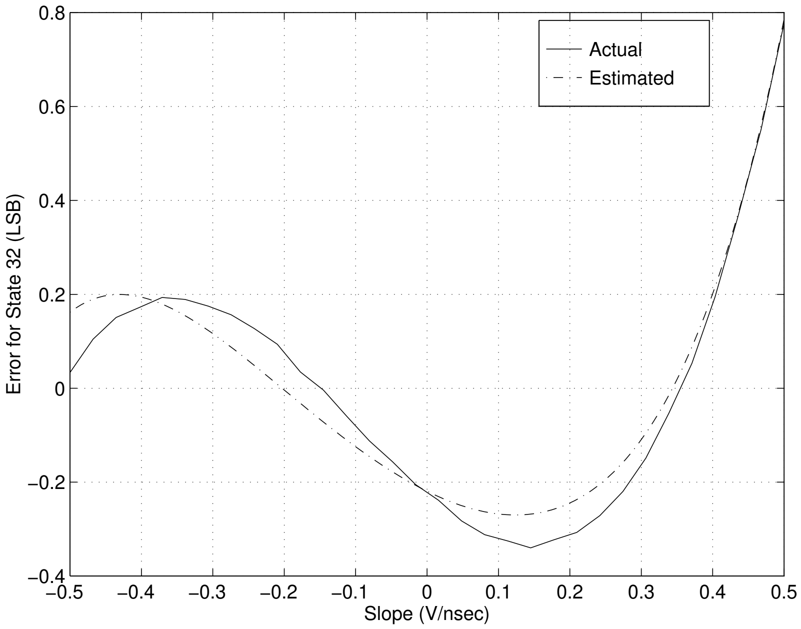

Figure 4.12: Comparison of Actual and Estimated Errors due to CDP Dynamic Behavior for different signal slopes

focusing on one particular state (32 in this example) and plotting the errors for all possible slopes. Figure 4.12 shows that the estimated error fits the actual error accurately for all different signal slopes. The polynomial basis functions were able to isolate the errors due to the dynamic behavior of the CDPs while other error mechanisms were contributing to the converter output error.

CHAPTER 5

Conclusions

This thesis had a twofold goal: first to explain the Folding and Interpolating Analog to Digital Converter Architecture, and second to introduce a technique for diagnosing particular errors in a FAI converter's architecture by selecting appropriate basis functions that model converter errors. Some of the most important conclusions derived include:

1. The Folding and Interpolating ADC architecture is an efficient alternative:
   - It requires less comparators and components in general.
   - It is low-power, and die-size efficient.
   - It is suited for high frequency applications.
2. Three error mechanisms for the FAI ADC were examined and accounted for:
   - Errors due to non-ideal resistors in the voltage divider network responsible for the generation of the CDP reference voltages are by far the most significant source of distortion.
   - The CDPs produce significant distortion when the slope of the input signal increases.
   - Errors in the interpolation network resistors contribute minimally to the converter's distortion.
3. Appropriate selection of basis functions allows for converter output distortion to be traced back to particular architectural errors:
   - The ratio of the error in a CDP reference voltage network resistor to its nominal value can be successfully determined by processing the converter's output.

- The distortion caused by the dynamic behavior of the CDPs can be successfully detected at the output of the converter.
- The procedure developed enables the basis functions to isolate particular error mechanisms acting simultaneously.

5.1 Future Work

Although several error mechanisms for the FAI converter were examined, there are still more mechanisms that could be addressed. This thesis developed a consistent and coherent method of diagnosing architectural flaws that can be easily extended. Possible candidates might include comparator and sample and hold errors.

This thesis analyzed the FAI converter and derived basis functions to model particular errors in the architecture. The same procedure can be applied to different converter architectures. Possible candidates include the Delta-Sigma and the Subranging ADCs.

Part of the motivation for this project was to provide a procedure for performing diagnostic tests on industry converters and provide feedback on their dominant sources of error. Modification of the simulation module to mimic particular industry converters would be a worthwhile future direction.

REFERENCES

[1] Johan van Valburg and Rudy J. van de Plassche, "High Speed Folding ADCs," *Analog Circuit Design*, Johan H. Huijsing, Rudy J. van de Plassche and Willy Sansen (editors), Dordrecht: Kluwer Academic Publishers, pp. 163-183, 1993.

[2] A. Arbel and R. Kurz, "Fast ADC," *IEEE Transactions on Nuclear Science*, vol. NS-22, pp. 446-451, February 1975

[3] Rob E.J. van De Grift and Rudy J. van de Plassche "A Monolithic 8-Bit Video A/D Converter," *IEEE Journal of Solid-State Circuits*, vol. SC-19, No. 3, pp. 374-378, June 1984.

[4] Rudy J. van de Plassche and Rob E.J. van De Grift "A High-Speed 7 Bit A/D Converter," *IEEE Journal of Solid-State Circuits*, vol. SC-14, No. 6, pp. 938-943, December 1979.

[5] Tai-Haur Kuo, Hung Chang Lin, Robert C. Potter, and Dave Shupe "A Novel A/D Converter Using Resonant Tunneling Diodes," *IEEE Journal of Solid-State Circuits*, vol.26, No. 2, pp. 145-149, February 1991.

[6] Rob E.J. van De Grift, Ivo W.J.M. Rutten and Martien van Der Veen "An 8-bit Video ADC Incorporating Folding and Interpolation Techniques," *IEEE Journal of Solid-State Circuits*, vol. SC-22, No. 6, pp. 944-953, December 1987.

[7] Johan van Valburg and Rudy J. van de Plassche "An 8-b 650-MHz Folding ADC," *IEEE Journal of Solid-State Circuits*, vol. 27, No. 12, pp. 1662-1666, December 1992.

[8] Rudy J. van de Plassche and Peter Baltus "An 8-bit 100-MHz Full-Nyquist Analog-to-Digital Converter," *IEEE Journal of Solid-State Circuits*, vol. 23, No. 6, pp. 1334-1344, December 1988.

[9] *MATLAB Reference Guide.* Massachusetts: The MathWorks, Inc. , August 1992.

[10] Martin S. Roden, *The Student Edition of MICRO-CAP IV.* California: The Benjamin/Cummings Publishing Company, Inc. , 1993.

[11] F.H. Irons and T.A. Rebold, "Characterization of high frequency analog to digital converters for spectral applications," MIT Lincoln Laboratory, Lexington, MA, Project Report AST-2, Nov. 1986.

[12] T.A. Rebold and F.H. Irons, "A phase plane approach to the compensation of high speed analog to digital converters," in *Proc. IEEE Int. Symp. Circuits and Systems*, pp. 455-458, May 1987.

[13] D. Asta and F.H. Irons, "Dynamic error compensation of analog to digital converters," The Lincoln Laboratory Journal, vol. 2, 1989.

[14] J.P. Deyst, "Wideband distortion compensation for bipolar flash analog-to-digital converters," S.M. thesis, MIT, Cambridge MA, 1990.

[15] F.H. Irons, D.M. Hummels, and S.P. Kennedy, "Improved Compensation for analog-to-digital converters," *IEEE Trans. on Circuits and Systems*, vol. 38, No. 8, pp. 958-961, Aug 1991.

[16] D.M. Hummels, F.H.Irons, and S.P. Kennedy, "Using adjacent sampling for error correcting analog-to-digital converters," in *Proc. IEEE Int. Symp. Circuits and Systems*, p. 589, May 1992.

[17] I.N. Papantonopoulos, D.M. Hummels and F.H. Irons "Two-tone Characterization of Analog to Digital Converters," Accepted for presentation at the Midwest Symposium on Circuits and Systems, May 1994.

[18] D.M. Hummels and S.P. Kennedy, "Improved dynamic compensation of ADCs using an iterative estimate of the ADC input calibration signal," in *Proc. Midwest Symp. Circuits and Systems*, pp. 68-71, Aug. 1992.

[19] S.P. Kennedy, "Improved ADC phase plane error compensation," M.S. Thesis, University of Maine, Orono Maine, May 1992.

[20] D.M. Hummels, R.W. Cook, F.H.Irons, "Discrete-time dynamic compensation of analog-to-digital converters," in *Proc. IEEE Int. Symp. Circuits and Systems*, pp. 1144-1147, May 1993.

[21] D.M. Hummels, F.H Irons, R. Cook, I. Papantonopoulos, "Characterization of ADCs Using a Non-Iterative Procedure," in *Proceedings of International Symposium on Circuits and Systems (ISCAS '94)* (London, U.K.), vol. 2, pp. 5-8, May 29 - June 2, 1994.

[22] Wahid Ahmed " Fast orthogonal search for training radial basis function neural networks," M.S. Thesis, University of Maine, Orono Maine, August 1994.

Appendix A
PROGRAM LISTING

A.1 Folding and Interpolating ADC MATLAB Simulation Module

```
function [adc_out] = fai(adc_in,inpdot)
%FAI     Folding and Interpolating Analog to Digital Converter Simulation.
%        FAI(adc_in,inpdot) is a simulation module that takes a set of
%        input data (adc_in) and its corresponding derivative (inpdot)
%        and returns the digital 8-bit equivalent word for the input data.
%        The simulation parameters can be modified internally.  By default,
%        adc_in is assumed to range between -2V and -1V, and the maximum
%        input signal slope (inpdot) should not exceed +/-0.5 V/nsec.
%*************************************************************************
%*  FOLDING AND INTERPOLATING ANALOG TO DIGITAL CONVERTER SIMULATION MODULE  *
%*-----------------------------------------------------------------------*
%*  Author : Ioannis Papantonopoulos                     | Date: May 11, 1995 *
%*************************************************************************

adc_out = zeros(size(adc_in));

    %
    %% Define ADC Parameters
    %
    fold_ratio = 8;     % Cycles in one folding amp characteristic
    num_fa = 4;         % Number of Folding amps.
    num_cdps = num_fa*(fold_ratio+2);  % Total # of CDPs.
    std1= 0/100;        % The random error std on the volt. ref. resistors
    std2= 0/100;        % The random error std on the interpolation resistors
    vt = 0.026;         % Thermal voltage at room temperature
    fix_thresh = 0.25;  % Threshold for correction of an MSB
    num_interp = 8;     % Number of interpolated characteristics per folding amp

    %
    % CDP Parms
    %
    vmax = -1;
    vmin = -2;

    %
    %% Now create the voltage divider that generates the CDP voltage references
    %
```

```
vmaxlad=vmax+2*(vmax-vmin)/fold_ratio;  %Maximum voltage at top of ladder
vminlad=vmin-(vmax-vmin)/fold_ratio;    %Minimum voltage at bottom of ladder
old_seed = randn('seed');               %Retain old number generator seed
randn('seed',1.101867880e+09);          %Set seed to repeatable value
rfa= ones(1,num_cdps+num_fa)+std1*randn(1,num_cdps+num_fa);
rfa = [0 rfa];                          %Zero pad to make code easier
total=sum(rfa);                         %Get the total divider resistance
v_low = zeros(num_fa,fold_ratio+2);     %Initialize the matrices to hold
v_high = zeros(num_fa,fold_ratio+2);    %the high and low CDP references

for i=1:num_cdps,                       %Now generate the references
    v_low(i) = vminlad +(sum(rfa([1:i])/total))*(vmaxlad-vminlad);
    v_high(i) = vminlad +(sum(rfa([1:i+num_fa])/total))*(vmaxlad-vminlad);
end

%
%% Parameters for the Dynamic model for the CDP
%
c1_coeff = [-1.3436e-4 0.9939];
c2_coeff = [-3.1125e-11 -9.7390e-9 5.0243e-5 -4.7094e-4];
c1 = polyval(c1_coeff,inpdot/1e6);      %Need slope in V/usec
c2 = polyval(c2_coeff,inpdot/1e6);      %Need slope in V/usec

%
%% Set up the interpolator networks as a matrix (res)
%
randn('seed',1.998022768e+09);          %Set seed to repeatable value
res = ones(2*num_fa,num_interp) + std2*randn(2*num_fa,num_interp);
randn('seed',old_seed);                 %Restore seed to original value
total2 = sum(res');                     %Sum resistance of each interpolator
rtap = zeros(size(res));                %Initialize the tap 'weights'
res = ([zeros(2*num_fa,1) res]);        %Zero pad for easier code
for k=2:num_interp,
    rtap(:,k) = sum(res(:,[1:k])')';    %Generate the 'weight' at each tap
end

%
%% Interpolation Matrix Creation.  This part of the code will generate a
%% matrix.  It assumes that we have a folding ratio of 8 and 4 folding
%% amplifiers.  This matrix (called interp) will postmultiply the
%% 1x8 vector of folding amplifier signals [FA1 ... FA4 FA1bar ... FA4bar]
%% and their product will contain the 32 values of Ik-Ikbar.
%
for i_row = 1:num_fa
    interp(i_row,((i_row-1)*num_interp+1):((i_row)*num_interp)) = ...
```

```
                                     1-rtap(i_row,:)/total2(i_row);
    interp(i_row+num_fa,((i_row-1)*num_interp+1):((i_row)*num_interp))= ...
                      -(1-rtap(i_row+num_fa,:)/total2(i_row+num_fa));
end
interp(1,((num_fa-1)*num_interp + 1):num_fa*num_interp) = ...
                                     -rtap(2*num_fa,:)/total2(2*num_fa);
interp(num_fa+1,((num_fa-1)*num_interp + 1):num_fa*num_interp) = ...
                                     rtap(num_fa,:)/total2(num_fa);
for i_row = 2:num_fa
    interp(i_row,((i_row-2)*num_interp+1):((i_row-1)*num_interp)) = ...
                                     rtap(i_row-1,:)/total2(i_row-1);
   interp(i_row+num_fa,((i_row-2)*num_interp+1):((i_row-1)*num_interp))= ...
               -rtap(i_row+num_fa-1,:)/total2(i_row+num_fa-1);
end

%
%% Convert the input samples one at a time
%
for i = 1:size(adc_in,1)
    for j=1:size(adc_in,2)

        x = adc_in(i,j);                %x contains the current input

        %
        % Find the CDP outputs
        %
        cdp_out = c1(i,j)*(0.25*tanh((x-v_low-c2(i,j))/(2*vt)) -0.25) - ...
                  c1(i,j)*(0.25*tanh((x-v_high-c2(i,j))/(2*vt)) -0.25) ;

        %
        % Calculate the folding amplifier outputs
        %
        fa = sum(cdp_out(:,2:2:fold_ratio+2)');     %FA1 through FA4
        fa_bar = sum(cdp_out(:,1:2:fold_ratio+2)'); %FA1bar through FA4bar

        %
        %% Get the MSB'S
        %% This part assumes fold_ratio = 8, num_fa = 4
        %
        msb_2 = (fa(1) - fa_bar(1)) <=0;

        %
        %% Preset MSB's if we are in 'dangerous' regions
        %
        if (cdp_out(3,3) > fix_thresh)
```

```
            msb = 0;
            msb_1 = ~msb_2;
        elseif (cdp_out(3,5) > fix_thresh)
            msb = ~msb_2;
            msb_1 = msb_2;
        elseif (cdp_out(3,7) > fix_thresh)
            msb = 1;
            msb_1 = ~msb_2;
        %
        %% Derive  MSB's in other regions by summing up appropriate CDP's
        %
        else
            msb = ( sum(cdp_out(1,6:9)) + sum(cdp_out(3,6:8)) ...
                   -sum(cdp_out(1,2:5)) - sum(cdp_out(3,2:4)) ) >= 0;
            msb_1 =(cdp_out(1,4)+ cdp_out(1,5)+ cdp_out(1,8)+ ...
                    cdp_out(1,9)+ cdp_out(3,4)+ cdp_out(3,8) -  ...
                   (cdp_out(1,2)+ cdp_out(1,3)+ cdp_out(1,6)+ ...
                    cdp_out(1,7)+ cdp_out(3,2)+ cdp_out(3,6)  )) >=0;
        end;

        %
        %% Now derive the fine bits, by summing up Ik's with the same
        %% sign depending on the state of MSB-2
        %
        interp_out = [fa fa_bar] * interp;   %Find the Ik's
        if (msb_2 == 1)
            fine = sum(interp_out<=0)-1;
        else
            fine = sum(interp_out >= 0) -1;
        end;

        %
        % Convert this to the final ADC output, in LSB's
        %
        adc_out(i,j) = fine + 32*msb_2 + 64*msb_1 + 128*msb;

        %
        %% Check for overflow and underflow and correct
        %
        if (cdp_out(3,1) > fix_thresh & msb_2==1) adc_out(i,j)=0; end
        if (cdp_out(3,9) > fix_thresh & msb_2==0) adc_out(i,j)=255; end

    end;
end;
```

BIOGRAPHY OF THE AUTHOR

Ioannis Nikolaou Papantonopoulos was born in Thessaloniki, in the province of Macedonia, Greece, on May 26, 1971. He obtained the "Apolitirion Lykeiou" Diploma of Higher Education in 1989 from Anatolia College in Thessaloniki.

Ioannis enrolled in the University of Maine in the fall of 1989, where he obtained his Bachelor of Science in Electrical Engineering with highest distinction in May of 1993. As an undergraduate he performed research in the Communications Laboratory in the field of Analog to Digital converters.

In the fall of 1993 Ioannis entered the graduate program at the University of Maine, Orono. He has served as a Research Assistant in the Department of Electrical and Computer Engineering. His current research interests include high-speed Analog to Digital Conversion, communications and signal processing, and modern filters. He has co-authored articles on converter characterization in conference proceedings. He is a member of HKN, TBΠ, ΦΚΦ, and ΠΜΕ honor societies. Ioannis is a candidate for the Master of Science Degree in Electrical Engineering in August, 1995.

LIBRARY RIGHTS STATEMENT