Design Strategy for a Pipelined ADC Employing Digital Post-Correction

Pieter Harpe, Athon Zanikopoulos, Hans Hegt and Arthur van Roermund
Technische Universiteit Eindhoven, Mixed-signal Microelectronics Group, EH 5.13
P.O.Box 513, 5600 MB Eindhoven, The Netherlands
Phone: +31 40 247 4703, email: p.j.a.harpe@tue.nl

Keywords— **Design strategy, Pipelined Analog-to-Digital converter, Digital post-correction**

Abstract— **This paper describes how the usage of digital post-correction techniques in pipelined analog-to-digital converters (ADC's) can be exploited optimally during the design-phase of the converter. It is known that post-correction algorithms reduce the influence of several circuit impairments on the final accuracy of the converter [1], [2]. However, until now, no models relating these circuit impairments to the final accuracy of the ADC, taking the usage of a post-correction algorithm into account, have been known to exist. To take maximum advantage of a certain correction algorithm, this model is a must. Therefore, this paper introduces a behavioral model of a pipelined ADC, including several important error mechanisms, representing possible circuit impairments like offset, gain error, harmonic distortion, etc. With this model, including the post-correction algorithm, simple design constraints for each part of the circuit can be derived such that a certain target accuracy of the ADC is achieved. In the analog design-phase these high-level constraints can be translated to implementation-dependent low-level design requirements. If these low-level requirements are fulfilled, the model guarantees that the converter will achieve its target accuracy. A design-example for a 12-bit pipelined ADC is worked out. Simulation results will be shown, validating the correctness of the presented design-method. The proposed design-strategy can be applied to all pipelined ADC's with post-correction like in [1], [2], taking maximum advantage of the benefits of the correction algorithm during the analog design-phase.**

I. Introduction

Pipelined analog-to-digital converters (ADC's) are used in many different applications like video processing, telecommunications, digital imaging, etc. The reason why this architecture is very suitable for these applications is because pipelined ADC's are able to combine high speed (in the order of $10 - 100$ MSPS) and high accuracy (in the order of $8 - 14$ bit) at the same time. However, as the target applications become more and more demanding (higher speed, higher accuracy), the design of pipelined ADC's becomes more and more complicated.

In particular, the maximum resolution achievable by pipelined converters is limited by technology properties like mismatch of components and values of parasitics. Traditionally, this problem is solved by increasing the size of all critical components until a certain matching (and hence accuracy) can be achieved. However, with this method, the required chip area grows exponentially with the required accuracy in bits. A simple but very effective solution to achieve high accuracy even though the intrinsic design from itself is not accurate, is to use digital post-correction as proposed by [1], [2]. This technique starts with an ADC having a low initial accuracy, hence physically small devices can be used. Then the post-correction algorithm is used to correct for circuit deviations, thereby improving the accuracy significantly.

Even though these digital post-correction techniques have been known and applied successfully for a long time, no models relating circuit impairments directly to the overall achievable accuracy of the converter, taking the usage of post-correction into account, have been known to exist. However, to take maximum advantage of the possibilities of the correction algorithm, this model is required definitely. In this paper, a design strategy is proposed that takes the effect of the correction algorithm into account. With this strategy, the intrinsic accuracy requirements for the analog components can be derived such that after post-correction, the targeted accuracy is achieved. Because of that, the requirements on the analog design can be kept as relaxed as possible and the digital post-correction algorithm can be exploited optimally.

In section II, an introduction to pipelined ADC's with digital post-correction is given. Section III introduces a behavioral model of the ADC including important error sources. The effect of these errors on the overall accuracy is studied in section IV. The design strategy is presented in section V and an example is worked out in section VI. Finally, conclusions are drawn in section VII.

II. Pipelined ADC with post-correction

In this section, the basic architecture of the ADC used throughout this paper is described. Also, the used digital post-correction technique is introduced. Although the synthesis method described in this paper can be applied to many different pipelined converters, as an example, a converter compromising 1.5-bit-per-stage basic cells is used here. The target accuracy of the converter is chosen to be 12 bits.

A. Pipelined architecture

The general structure of a pipelined converter is given in figure 1: it is composed of a sample-and-hold (S&H) stage, a concatenation of k basic blocks and a digital correction algorithm. The S&H-stage samples the analog input V_{in} at the sample frequency f_s. The basic blocks perform the actual analog-to-digital conversion. Each block in the pipeline has an analog input and both an analog and a digital output. The digital output gives a coarse quantization of the input voltage. The analog output represents the quantization error, and will be quantized in the consecutive stages. A digital post-correction algorithm is used to combine the separate digital outputs of each cell, correct for several circuit impairments, and to provide a valid N-bit output code. In the example considered in this paper, the output code is 12-bits large.

This paper investigates the relation between circuit impairments and overall performance, taking the influence of the post-correction algorithm into account. As the correction algorithm has influence only on the impairments in the basic blocks and not on impairments in the S&H-stage, this stage is left out of consideration here.

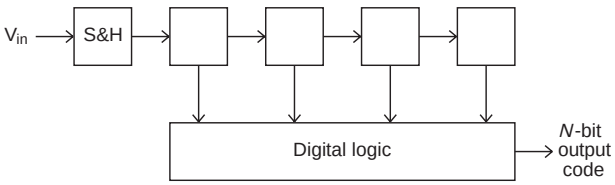

Fig. 1. Example of a N-bit pipelined ADC with digital post-correction.

A detailed view of a basic block is given in figure 2, all blocks in the pipeline are assumed to be identical. It was decided to use a small number of bits per cell to reduce circuit complexity as much as possible. By employing 1.5-bit-per-stage blocks (instead of 1-bit), a certain redundancy is added to the coding process, providing room for the correction algorithm to solve impairments of the basic cells afterwards, without feedback to the analog part of the circuit.

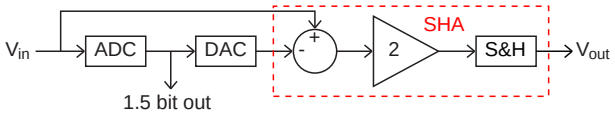

Fig. 2. Basic block of the pipelined ADC.

The two elements of the correction technique, being *1.5-bit redundancy* and *digital post-correction*, will be explained briefly in the next sections. Detailed explanations can be found in [1], [2], [3].

B. 1.5-bit redundancy

The reason for adding redundancy to the digital output of each individual basic block is to relax the accuracy requirements of the sub-ADC's. Moreover, this redundancy is required for the post-correction algorithm as will be explained in the next section.

For comparison, a system without redundancy is considered first. In this system, the sub-ADC and sub-DAC both have a 1-bit resolution while the gain of the SHA equals two. Ideally, the sub-ADC divides the input range of the basic block in two equal parts (indicated by '0' and '1' in figure 3 (left)). When, for example, the input signal is in the '1'-range, the analog output of the basic block (after subtraction of the sub-DAC output and multiplication by two) will fit exactly in the allowed output range of the block.

However, when the sub-ADC's reference level V_{ref} deviates from the ideal value (due to a static deviation or dynamic behavior of the comparator) the input range is not divided equally. Because of that, it is possible that the output signal will exceed the allowed range, resulting in a large quantization error (see figure 3 (right)).

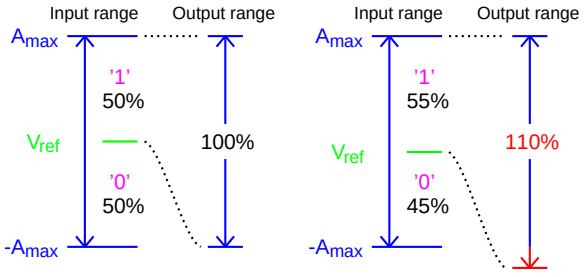

Fig. 3. Ideal behavior (left), and behavior in case of deviation of V_{ref} (right) of a 1-bit basic block.

A solution to prevent overflow of the output range in case of deviations of the comparator levels is to use *redundancy*. Redundancy is achieved when condition

1 is fulfilled. Here, n represents the number of bits in the sub-ADC and sub-DAC and A is the gain in the SHA.

$$A < 2^n \tag{1}$$

The design concept described in this paper can be used with all combinations of n and A fulfilling this condition. However, throughout this paper, $n = 1.5$ is used in combination with $A = 2$; the small number of bits per stage is used to minimize circuit complexity.

The functionality of redundancy is illustrated in figure 4. Due to the 1.5 bits, the input range is now divided in three parts. As the gain is still equal to two, normally only 67% of the output range is used (left picture). When a deviation of one of the comparator levels occurs now, the output will be still in the allowed range[1], and no distortion is introduced. It is important to notice that, due to the gain of two, the digital output of the pipeline can be still constructed according to equation 2, hence no post-correction algorithm taking the actual comparator levels into account is required in this situation. Also, the actual values of the comparator levels have no influence on the accuracy of the converter, as long as the output range is not exceeded in one of the basic blocks.

$$D_{out} = \sum_{i=0}^{k-1} 2^{-i} D(i) \text{ , where} \tag{2}$$

D_{out} is the N-bit digital output of the pipeline, $D(i)$ is the 1.5-bit output of block i (with i numbered from 0 to $k-1$).

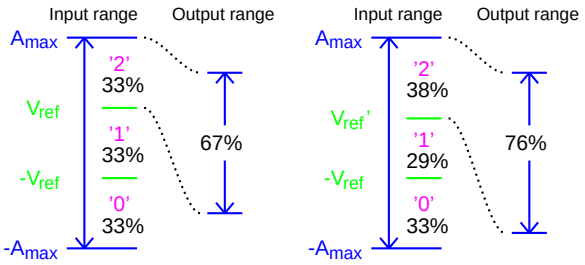

Fig. 4. Ideal behavior (left), and behavior in case of deviation of V_{ref} (right) of a 1.5-bit basic block.

C. Digital post-correction

The second correction technique applied is based on the method introduced by [1] and also described in [2], [3]. Its aim is to correct for linear and constant errors in the sub-DAC, summing-node, amplifier and S&H-circuit. Non-linear distortion is not corrected completely, but it will be shown that this method reduces the influence of non-linearity on the overall accuracy of the converter.

The basic idea of a *digital post-correction method* (see figure 5) is that most errors can be corrected *afterwards (in the digital domain)* as long as the analog output of each block remains in the allowed output range. The algorithm performs a simple mapping-function from the 1.5-bit uncorrected digital codes of the basic blocks to the corrected output code of the converter.

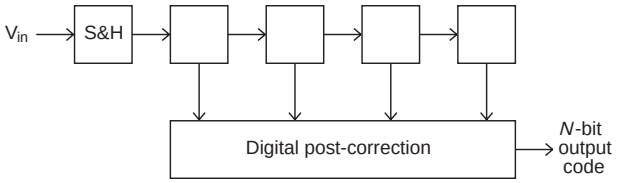

Fig. 5. Example of a pipelined AD converter with digital post-correction.

In this section, the correction method is described on system level. First, a system without post-correction will be described, after which the post-correction algorithm is added.

C.1 System without post-correction

When it is assumed that the transfer function of each 1.5-bit basic block is ideal, like in figure 6, post-correction is not needed. The N-bit output code is generated simply by a summation of constant *weights*. Each stage adds a single weight, dependent on the produced code of that stage ('0', '1' or '2'), and the place of the stage in the pipeline. Table I shows the weights for a 1.5-bit per stage converter with 5 stages. For example, when the pipeline produces code '01220', the output code becomes $-x + 0 + \frac{1}{4}x + \frac{1}{8}x - \frac{1}{16}x$.

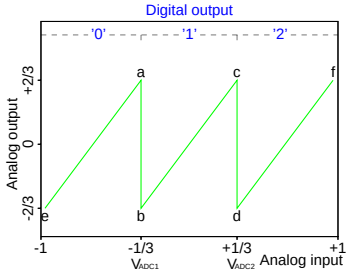

Fig. 6. Ideal transfer function of a basic block.

[1]Except when the deviation is larger than 8% of the full-scale range.

Stage	0	1	2	3	4
Code					
'0'	$-x$	$-\frac{1}{2}x$	$-\frac{1}{4}x$	$-\frac{1}{8}x$	$-\frac{1}{16}x$
'1'	0	0	0	0	0
'2'	x	$\frac{1}{2}x$	$\frac{1}{4}x$	$\frac{1}{8}x$	$\frac{1}{16}x$

TABLE I

EXAMPLE OF THE WEIGHTS OF EACH STAGE IN THE PIPELINE, x IS AN ARBITRARY CONSTANT.

C.2 System with post-correction

Now, consider the situation that the transfer function of each basic block shows a certain deviation from the ideal curve. Moreover, each block can have a different deviation due to mismatch, process spread, etc. In figure 7, two examples of non-ideal curves are shown. In the following, it is assumed that the non-ideal curves are always linear. Non linearities are left out of consideration for the moment, but will be included in section III and IV.

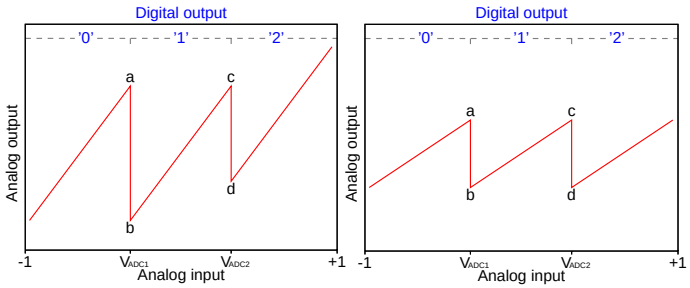

Fig. 7. Non-ideal transfer functions of a basic block. Curve with deviation of the DAC-level for code 2 (left), and deviation of the gain stage (right).

To make clear that it is possible to correct all deviations of the transfer function afterwards, the transfer function of each stage is described with four parameters:

1. The slope of the three linear parts of the curve. (As the same amplifier is used in all regions, the three linear parts always have the same slope.)
2. A constant offset value, equal to the output voltage when the input voltage equals 0 and the digital code equals 1.
3. The transition height of the curve around input level V_{ADC1}, the first comparator level.
4. The transition height of the curve around input level V_{ADC2}, the second comparator level.

With these parameters, the transfer function can be reconstructed without ambiguity. The first two parameters don't have influence on the overall accuracy of the pipelined converter. The only effect is that they produce a single input referred offset and gain error of the pipelined ADC. Only the deviations of the two transition heights from their ideal value results in loss of overall accuracy. Luckily, the deviations of the transfer heights can be corrected by making use of variable weights, instead of using fixed weights. The correction algorithm measures the transition heights first, and then it derives the optimal values of the weights. It is assumed that the deviations of the transfer function are static, i.e. the deviations are constant during the time of operation. In that situation, it is sufficient to determine the correct weights once at the beginning of operation.

Before explaining the post-correction algorithm, some definitions are introduced:

- the pipelined converter consists of k stages, numbered from 0 (first stage) to $k-1$ (last stage),
- the output code of stage i is indicated with $u(i) \in \{\text{'0'}, \text{'1'}, \text{'2'}\}$,
- $U(i)$ is the concatenation of the output codes from stage i up to stage $k-1$:

$$U(i) = u(i)u(i+1)\cdots u(k-2)u(k-1) \qquad (3)$$

- the variable weights of stage i are indicated with $\omega_0(i)$, $\omega_1(i)$ and $\omega_2(i)$ for code '0', '1' and '2' respectively. As the correction algorithm only needs two degrees of freedom per stage (there are two transition heights that need correction), one weight can be fixed. For simplicity $\omega_1(i) = 0$ will be used,
- $M(\cdot)$ is the mapping function from the output code of the pipeline to the corrected output code of the converter:

$$M(U(i)) = \sum_{j=i}^{k-1} \omega_{u(j)}(j) \qquad (4)$$

Before the pipelined converter can be used, a measurement procedure is performed to determine the optimal weights for each stage. The measurement procedure measures each stage one by one. Suppose, one would like to measure the weights of stage i. The measurement procedure for $\omega_0(i)$ is as follows:

- the analog input of stage i is set to V_{ADC1}, the first reference level of the sub-ADC as indicated in figure 6,
- the input of the sub-DAC is forced to code '0'. Mark a in figure 6 indicates the output voltage of the stage in this situation, named v_a. The output code of the converter from stage i up to $k-1$ now equals:

$$\text{'0'}U_a(i+1) \qquad (5)$$

$U_a(i+1)$ is the digital code, produced by the part of the pipelined converter following stage i, when the input voltage of stage $i+1$ equals v_a,
- the input of the sub-DAC is now forced to code '1'. Mark b in figure 6 indicates the output voltage of stage i. The output code equals:

$$\text{'1'}U_b(i+1) \tag{6}$$

This procedure yields two codes ('0'$U_a(i+1)$ and '1'$U_b(i+1)$), both describing the same analog input voltage V_{ADC1}. So, after post-correction, the codes should be equal to each other:

$$\begin{aligned} M(\text{'0'}U_a(i+1)) &= M(\text{'1'}U_b(i+1)) \\ \omega_0(i) + M(U_a(i+1)) &= \omega_1(i) + M(U_b(i+1)) \\ \omega_0(i) + M(U_a(i+1)) &= 0 + M(U_b(i+1)) \\ \omega_0(i) &= M(U_b(i+1)) \\ &- M(U_a(i+1)) \end{aligned} \tag{7}$$

Likewise, the second weight can be measured by applying V_{ADC2} in combination with code '1' and '2', yielding:

$$\omega_2(i) = M(U_c(i+1)) - M(U_d(i+1)) \tag{8}$$

So, $\omega_0(i)$ and $\omega_2(i)$ can be determined in the digital domain, by making use of the blocks following stage i. Yet, the weights of stage i can be determined only when the weights of the stages following stage i are already known. Therefore, the measurement algorithm starts at the back-end of the pipeline. The last stage $(k-1)$ can not be measured, as there are no consecutive stages anymore, so the values of this stage are fixed to $\omega_0(k-1) = -1$ and $\omega_2(k-1) = 1$. Then, one by one, stages $k-2$, $k-3$, $\cdots$, 0 are measured.

In most calibration systems, the accuracy of the calibration algorithm limits the achievable accuracy of the calibrated converter. However, in this case, the accuracy of the calibration algorithm increases for each additional block that is calibrated. Because of that, the accuracy of the calibration algorithm is not a limit for the achievable accuracy of the pipelined ADC.

III. Behavioral model

To investigate the influence of errors on the converter's accuracy, the model of the basic block (figure 8) is extended with several error sources. The model of each component is described below. Only static and quasi-static deviations are taken into account, as the aim of this paper is to show the influence of the two exploited correction techniques, and they are aimed at (quasi-)static errors only.

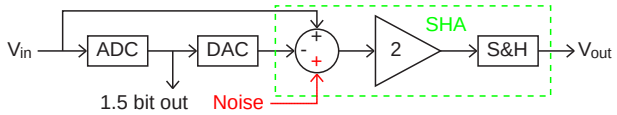

Fig. 8. Basic block for a pipelined converter.

It is assumed that all error sources are *stochastic* and *independent* on other error sources. Systematic errors, as opposed to stochastic errors, can be prevented by proper design and layout. If required, the formulas derived in this section can be adapted to describe systematic errors as well.

The effect of clock jitter is not taken into account by the presented model, even though it can be a major limitation for the accuracy of the converter. The reason not to include this effect is that the clock jitter will have by far the most influence on the front-end sample-and-hold stage (placed before the first block of the pipeline) and the design of this stage is beyond the scope of this paper.

For simplicity the input and output range of each basic block are normalized to $[-1,+1]$. In practice, any arbitrary range can be used when all values given here are adjusted accordingly.

A. Sub-ADC

The sub-ADC is a 1.5 bit AD converter, having two comparator levels (V_{ADC1} and V_{ADC2}). Due to transistor mismatch, these levels show a stochastic deviation δ_{ADC} from the ideal value:

$$\begin{cases} V_{ADC1}(i) &= -\frac{1}{3} + \delta_{ADC1}(i) \\ V_{ADC2}(i) &= +\frac{1}{3} + \delta_{ADC2}(i) \end{cases} \tag{9}$$

δ_{ADC1} and δ_{ADC2} represent the stochastic deviation, using a normal distribution with mean 0 and variance σ^2_{ADC}. Due to the symmetry of the two comparator levels, the same variance is used for both levels. As the deviation is stochastic, each basic block will have different reference levels. Thus, an index i is used to indicate each individual basic block.

B. Sub-DAC

The 1.5 bit sub-DAC has three reference levels. As was the case with the sub-ADC, these levels show a stochastic deviation due to mismatch of components:

$$\begin{cases} V_{DAC0}(i) &= -\frac{2}{3} + \delta_{DAC0}(i) \\ V_{DAC1}(i) &= \delta_{DAC1}(i) \\ V_{DAC2}(i) &= +\frac{2}{3} + \delta_{DAC2}(i) \end{cases} \tag{10}$$

The deviation of the ideal values is indicated with δ_{DAC}, and is unique for each reference level of each basic block. It is assumed that the deviations are constant during the time of operation and that they have the same variance σ^2_{DAC}.

C. SHA

The summing node, amplifier and sample-and-hold stage are combined in the sample-and-hold-amplifier (SHA). A white noise source is added to the SHA to represent thermal noise, produced by the circuit itself. The SHA is modelled as follows:

$$\begin{aligned} V_{out}(i) &= A(i)V_x(i) - A_3V_x(i)^3 \text{ , with} \quad (11)\\ V_x(i) &= V_{in}(i) + V_{off}(i) - V_{DAC}(i) + V_n(i,t) \end{aligned}$$

with:

$$\begin{aligned} A(i) &= 2(1+\delta_A(i)) \quad (12)\\ V_{off}(i) &= \delta_{off}(i) \quad (13)\\ V_n(i,t) &= \delta_n(i,t) \quad (14) \end{aligned}$$

The gain A of the SHA (normally 2) and the input-referred offset voltage V_{off} are modelled with a random deviation with variance σ^2_A and σ^2_{off} respectively. The white noise source V_n is modelled with a normal distribution with variance σ^2_n, and is a function of time t. The non-linearity of the SHA is modelled with a third-order distortion component A_3. As the implemented circuit will be fully differential, it is assumed that the third-order component is the dominant source of non-linear distortion, therefore only this component is taken into account. A_3 is equal for all basic blocks, as our goal is to see the influence of harmonic distortion, and not to see the influence of deviations of the harmonic distortion.

IV. Relation between errors and performance

In this section, mathematical expressions are derived, showing the relation between each individual error source and the accuracy of the pipelined converter. This is done for a 1.5-bit per stage pipelined converter with post-correction algorithm employed. The number of stages equals k, and the required overall resolution is N bit. For comparison, the results for the same converter without correction are given as well.

A. Noise

To derive an upper bound for the amount of noise allowed for a certain accuracy, we use the rule that the average input referred thermal noise power is of the same magnitude as the quantization noise power according to N bits accuracy.

Each stage contains a noise source with an average noise power of σ^2_n. The average input referred thermal noise power is the sum of all noise sources, corrected by the intermediate gain stages:

$$P_{tn} = \sum_{i=0}^{k-1} \frac{1}{2^{2i}}\sigma^2_n \approx \frac{4}{3}\sigma^2_n \quad (15)$$

The input range of the first stage is $[-1,1]$, for N bits accuracy this means:

$$\frac{1}{2}V_{lsb} = 2^{-N} \quad (16)$$

When the probability distribution of the quantization error is uniform, the average quantization noise power is:

$$P_{qn} = \int_{-\frac{1}{2}V_{lsb}}^{\frac{1}{2}V_{lsb}} \frac{1}{V_{lsb}}x^2dx = \frac{1}{3}\left(\frac{1}{2^N}\right)^2 \quad (17)$$

Combining the equations for thermal and quantization noise power yields:

$$\begin{aligned} P_{tn} &\leq P_{qn}\\ \sigma_n &\leq 2^{-(N+1)} \quad (18) \end{aligned}$$

This relation is valid both for converters without post-correction and converters with post-correction.

B. Constant and linear deviations

In the previous section, the following constant and linear error mechanisms were modelled:
- deviations in the sub-ADC (modelled with σ_{ADC}),
- deviations in the sub-DAC (modelled with σ_{DAC}),
- offset in the SHA (modelled with σ_{off}),
- gain error in the SHA (modelled with σ_A).

For this class of errors, the usage of post-correction has a great influence. Therefore, the two situations will be discussed separately, starting with a converter employing post-correction.

B.1 With post-correction

As the post-correction algorithm corrects constant and linear errors completely, they will not affect the overall accuracy of the converter. Nevertheless, their values should be such that their combined influence does not exceed the 'error budget' created by the 1.5-bit redundancy, i.e. the transfer function of each basic block should not exceed the allowed output range $[-1,+1]$.

An example of a transfer function of a basic block was given before in figure 6. Mathematically, this function is given by the following relations[2]:

$$V_{out} = A(V_{in} + V_{off} - V_{DACi}) \text{ , with:}$$

$$\begin{cases} i = 0 \text{ if } & V_{in} \leq V_{ADC1} \\ i = 1 \text{ if } V_{ADC1} < & V_{in} \leq V_{ADC2} \\ i = 2 \text{ if } V_{ADC2} < & V_{in} \end{cases} \tag{19}$$

To prevent that the allowed input range of the next stage is exceeded, the output voltage should be bounded by:

$$-1 \leq V_{out} \leq 1 \tag{20}$$

Looking at figure 6, this means that the six corners (marked with a up to f) should be in this range. Using the definitions from equations 9, 10, 12 and 13 this leads to the set of constraints:

$$\begin{cases} \frac{1}{3} & -\delta_{off} + \delta_{DAC0} \leq \frac{1}{2(1+\delta_A)} \\ \frac{1}{3} & +\delta_{ADC1} + \delta_{off} - \delta_{DAC0} \leq \frac{1}{2(1+\delta_A)} \\ \frac{1}{3} & -\delta_{ADC1} - \delta_{off} + \delta_{DAC1} \leq \frac{1}{2(1+\delta_A)} \\ \frac{1}{3} & +\delta_{ADC2} + \delta_{off} - \delta_{DAC1} \leq \frac{1}{2(1+\delta_A)} \\ \frac{1}{3} & -\delta_{ADC2} - \delta_{off} + \delta_{DAC2} \leq \frac{1}{2(1+\delta_A)} \\ \frac{1}{3} & +\delta_{off} - \delta_{DAC2} \leq \frac{1}{2(1+\delta_A)} \end{cases} \tag{21}$$

The stochastic deviations (δ_{ADC}, δ_{DAC}, δ_{off} and δ_A) are limited by 3σ with great certainty (99.74%), so it is assumed that the following conditions are valid:

$$\begin{aligned} -3\sigma_{ADC} \leq \delta_{ADC} &\leq 3\sigma_{ADC} \\ -3\sigma_{DAC} \leq \delta_{DAC} &\leq 3\sigma_{DAC} \\ -3\sigma_{off} \leq \delta_{off} &\leq 3\sigma_{off} \\ -3\sigma_A \leq \delta_A &\leq 3\sigma_A \end{aligned} \tag{22}$$

Using this assumption, the set of constraints can be simplified to the single restriction:

$$\frac{1}{3} + 3\sigma_{ADC} + 3\sigma_{off} + 3\sigma_{DAC} \leq \frac{1}{2(1+3\sigma_A)} \tag{23}$$

B.2 Without post-correction

When no post-correction is applied, condition 23 has to be fulfilled in order to prevent overflow in one of the blocks. Moreover, the deviations of the sub-DAC levels and the gain error of the SHA have a direct influence on the accuracy of the converter. The deviations of the sub-ADC have no influence in the overall performance as they are corrected by the 1.5-bit redundancy automatically. The offset of the SHA has also no influence, as it can be represented as an input-referred offset. The requirements on the accuracy of the sub-DAC and gain error of the SHA for an overall accuracy of N bits will be discussed now.

The input-referred error of a deviation $\delta_{DAC}(i)$ in one of the levels of the sub-DAC in the i^{th} stage of the pipeline equals:

$$V_{inp,err}(i) = 2^{-i}\delta_{DAC}(i) \text{ ,} \tag{24}$$

when all deviations $\delta_{DAC}(i)$ are bounded by $3\sigma_{DAC}$, the total input-referred error is estimated by:

$$V_{inp,err} = \sum_{i=0}^{k-1} V_{inp,err}(i) < 6\sigma_{DAC} \tag{25}$$

Equating this value with $\frac{1}{2}V_{lsb}$ results in the constraint:

$$\left.\begin{aligned} \tfrac{1}{2}V_{lsb} &= 2^{-N} \\ V_{inp,err} &\approx 6\sigma_{DAC} \end{aligned}\right\} \Rightarrow \sigma_{DAC} \leq \frac{1}{6}2^{-N} \tag{26}$$

The gain error of the SHA in stage i is modelled by $\delta_A(i)$. When other errors are neglected, the transfer function of a basic block is given by:

$$V_{out} = 2(V_{in} - V_{DAC}) + 2\delta_A(V_{in} - V_{DAC}) \tag{27}$$

The maximum absolute deviation is achieved for maximum $V_{in} - V_{DAC}$, as this signal is bounded by $\frac{1}{3}$ (due to the 1.5-bit redundancy), the maximum input-referred error is given by:

$$V_{inp,err}(i) = 2^{-i}\frac{1}{3}\delta_A \tag{28}$$

As opposed to the deviations in the sub-DAC's, the deviations due to gain error are correlated from one stage to the other. As the gain error in the first stage has the most influence on the input-referred error, the situation is considered where $V_{in} - V_{DAC} = \frac{1}{3}$ for the first stage. In that situation, the output of the first stage (and hence the input for the second stage) equals $\frac{2}{3}(1 + \delta_A(0)) \approx \frac{2}{3}$. This means that for the second and following stages $V_{in} - V_{DAC} \approx 0$, so their gain errors have minor influence. Therefore, the total input-referred error is estimated by the error made by the first stage only. Bounding $\delta_A(0)$ with $3\sigma_A$ and equating this value with $\frac{1}{2}V_{lsb}$ yields the requirement:

$$\left.\begin{aligned} \tfrac{1}{2}V_{lsb} &= 2^{-N} \\ V_{inp,err} &\approx \sigma_A \end{aligned}\right\} \Rightarrow \sigma_A \leq 2^{-N} \tag{29}$$

[2]Note that here the non-linearity of the SHA is not taken into account.

C. Non-linear distortion

As was the case with gain error, the errors due to non-linearity of the SHA are correlated from one stage to the other. When the first stage experiences the maximum error, the errors in all other stages are negligible. Therefore, the total error due to harmonic distortion can be estimated quite accurate by the distortion of the first SHA only.

The situation of a converter without post-correction is considered first. Assuming that all other components are ideal, the distortion of the first SHA is given by (derived from equation 11):

$$V_{hd}(i) = A_3\Big(V_{in}(i) - V_{DAC}(i)\Big)^3 \tag{30}$$

Maximum distortion is achieved when the maximum input signal is applied ($[V_{in} - V_{DAC}]_{max} = \frac{1}{3}$). Making this distortion input-referred yields:

$$V_{hd,in,max} = \Big|\frac{1}{2}A_3\frac{1}{3}^3\Big| = \frac{|A_3|}{54} \tag{31}$$

The maximum allowed distortion for N bits accuracy now becomes:

$$\left.\begin{array}{rcl} \frac{1}{2}V_{lsb} & = & 2^{-N} \\ V_{hd,in,max} & = & \frac{|A_3|}{54} \end{array}\right\} \Rightarrow |A_3| \leq 54 \cdot 2^{-N} \tag{32}$$

In case of a converter with post-correction applied, it appears that the requirement on the linearity of the SHA becomes less strict. As before, the distortion of the first stage only is considered.

Figure 9 (left) shows an example of an ideal transfer function, and an example of a transfer function including 3^{rd} order harmonic distortion. In a system without post-correction, it is assumed that the transfer function equals the ideal curve, thus the maximum deviation occurs at the highest absolute output-level. In case of a system with post-correction (right picture), the correction algorithm reconstructs the transfer function such that the actual curve and its reconstruction exactly match around the transitions of the transfer curve. The picture shows that in this situation, the maximum distortion occurs not at the maximum absolute output-level, but somewhere in the middle.

Figure 10 shows the error due to harmonic distortion in case of a system without and a system with post-correction. Obviously, the same amount of distortion in the SHA has less influence when post-correction is applied.

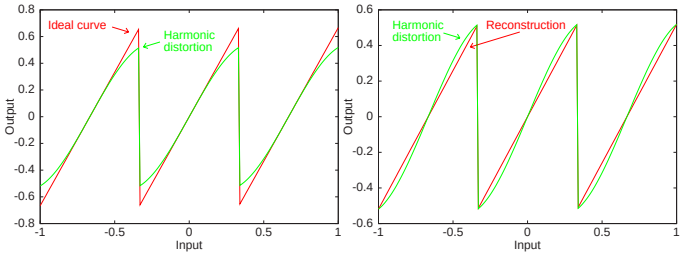

Fig. 9. Reconstruction of a transfer function with harmonic distortion in case of a system without post-correction (left) and with post-correction (right).

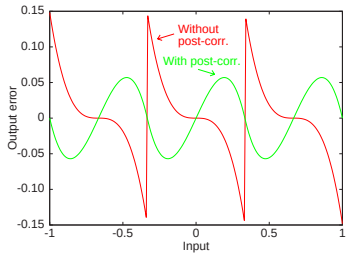

Fig. 10. Introduced deviation due to non-linearity in the SHA for systems without and with post-correction applied.

A simple analysis shows the improvement after application of the post-correction algorithm. First, consider the system without post-correction. The actual curve is given by:

$$y_{act} = 2x - a_3x^3 \tag{33}$$

For simplicity, variables x and y_{act} are used instead of the actual names of the signals. Moreover, only the middle part of the transfer function is dealt with, as the other two parts show exactly the same behavior.

The converter uses the reconstruction function:

$$y_1 = 2x \tag{34}$$

Thus, the resulting *input referred* deviation is:

$$y_{dev1} = \frac{1}{2}\Big(y_{act} - y_1\Big) = -\frac{1}{2}a_3x^3 \tag{35}$$

If the maximum input signal is m, the maximum absolute deviation is:

$$y_{dev1,max} = \frac{1}{2}|a_3|m^3 \tag{36}$$

In case of a system with post-correction, the reconstruction function equals:

$$y_2 = bx \tag{37}$$

b is chosen such that for the maximum input m, y_2 equals y_{act}:

$$y_2(m) = y_{act}(m) \Rightarrow b = 2 - a_3m^2 \tag{38}$$

The input referred deviation now becomes:

$$y_{dev2} = \frac{1}{2}\Big(y_{act} - y_2\Big) = \frac{1}{2}a_3 m^2 x - \frac{1}{2}a_3 x^3 \qquad (39)$$

The maximum deviation is now achieved for:

$$\frac{\partial y_{dev2}}{\partial x} = 0 \Rightarrow y_{dev2,max} = \frac{1}{9}\sqrt{3}|a_3|m^3 \qquad (40)$$

The improvement of accuracy after application of the post-correction algorithm is determined by the ratio between $y_{dev1,max}$ and $y_{dev2,max}$:

$$\log_2\left(\frac{y_{dev1,max}}{y_{dev2,max}}\right) = \log_2\left(\frac{3}{2}\sqrt{3}\right) \approx 1.38 \text{ bit} \qquad (41)$$

Where the non-linearity of the SHA for a converter without post-correction was restricted to relation (32), in systems with post-correction, the restriction is:

$$|A_3| \leq 81\sqrt{3} \cdot 2^{-N} \qquad (42)$$

V. Design strategy

In this section, the requirements to achieve a certain overall accuracy as derived in the previous section, are summarized briefly. A simple design strategy on system level for the various components of the basic block is given, based on these requirements. As the usage of digital post-correction has significant influence on these requirements, converters without and with post-correction are discussed separately.

A. Without post-correction

When a pipelined ADC without post-correction but with 1.5-bit redundancy has to be designed, relations 18, 23, 26, 29 and 32 have to be fulfilled in order to achieve N bits overall accuracy. For convenience, the requirements are repeated below:

$$\sigma_n \leq 2^{-(N+1)} \qquad (43)$$

$$\frac{1}{3} + 3\sigma_{ADC} + 3\sigma_{off} + 3\sigma_{DAC} \leq \frac{1}{2(1+3\sigma_A)} \qquad (44)$$

$$\sigma_{DAC} \leq \frac{1}{6}2^{-N} \qquad (45)$$

$$\sigma_A \leq 2^{-N} \qquad (46)$$

$$|A_3| \leq 54 \cdot 2^{-N} \qquad (47)$$

B. With post-correction

When a pipelined ADC with post-correction has to be designed, relations 18, 23 and 42 have to be fulfilled in order to achieve N bits overall accuracy. In short, the requirements are:

$$\sigma_n \leq 2^{-(N+1)} \qquad (48)$$

$$\frac{1}{3} + 3\sigma_{ADC} + 3\sigma_{off} + 3\sigma_{DAC} \leq \frac{1}{2(1+3\sigma_A)} \qquad (49)$$

$$|A_3| \leq 81\sqrt{3} \cdot 2^{-N} \qquad (50)$$

Comparing these requirements with the requirements in case of a design without post-correction (section V-A), one can see that the requirements on the following properties have been relaxed significantly:
- sub-DAC level accuracy,
- SHA gain accuracy,
- SHA linearity.

When these effects of the post-correction algorithm are taken into account during the design-phase of the converter, the abilities of the algorithm can be exploited optimally. The relations given in this section (eq. 48, 49 and 50) represent the design strategy for a pipelined ADC with digital post-correction at system-level. During the transistor-level design phase, these requirements can be translated to requirements on component level, like matching of capacitors and transistors, linearity of the amplifier and minimum capacitor sizes with respect to thermal noise.

VI. Design example

To verify the design strategy, a design example for a 12-bit ADC is worked out for a converter without and a converter with post-correction.

A. Without post-correction

To achieve 12-bits accuracy with an ADC without post-correction, the requirements given in section V-A have to be fulfilled. For $N = 12$, a possible solution, fulfilling these constraints marginally, is:

$$\begin{cases} \sigma_n &= 1.2 \cdot 10^{-4} \\ \sigma_{ADC} &= 3.0 \cdot 10^{-2} \\ \sigma_{off} &= 2.5 \cdot 10^{-2} \\ \sigma_{DAC} &= 4.0 \cdot 10^{-5} \\ \sigma_A &= 2.4 \cdot 10^{-4} \\ A_3 &= 1.3 \cdot 10^{-2} \end{cases} \qquad (51)$$

Using these values, system-level simulations were performed. The results will be shown in section VI-C.

B. With post-correction

When post-correction is employed, the requirements given in section V-B have to be fulfilled. For $N = 12$, a possible solution, fulfilling these constraints marginally, is to use the following values:

$$\begin{cases} \sigma_n &= 1.2 \cdot 10^{-4} \\ \sigma_{ADC} &= 2.0 \cdot 10^{-2} \\ \sigma_{off} &= 2.0 \cdot 10^{-2} \\ \sigma_{DAC} &= 1.0 \cdot 10^{-2} \\ \sigma_A &= 1.0 \cdot 10^{-2} \\ A_3 &= 3.4 \cdot 10^{-2} \end{cases} \tag{52}$$

Using these values, system-level simulations were performed. The results will be shown in the next section.

Comparing these values with the values for the converter without correction (51), one can see that the requirements on σ_{ADC} and σ_{off} actually become more severe. However, the requirements remain of the same order of magnitude (e.g.: σ_{ADC} goes from $3.0 \cdot 10^{-2}$ to $2.0 \cdot 10^{-2}$). On the other hand, the requirements for DAC accuracy (σ_{DAC} from $4.0 \cdot 10^{-5}$ to $1.0 \cdot 10^{-2}$) and gain-accuracy (σ_A from $2.4 \cdot 10^{-4}$ to $1.0 \cdot 10^{-2}$) are relaxed several orders of magnitude, giving major advantages in the design of these blocks.

C. Simulation results

The two converter designs (without and with post-correction) were implemented in C^{++}, according to the model described in section III. The parameters of the modelled error-sources were set according to (51) and (52) for the converter without and with post-correction respectively. As these parameters (except for the harmonic distortion parameter A_3) are stochastic, a Monte-Carlo Code-Density-Test (CDT) analysis was performed on the converters to determine the achieved accuracy. The CDT was repeated 1000 times to obtain satisfactory results. By deriving the INL from the CDT, the accuracy was determined. The results are given in fig. 11.

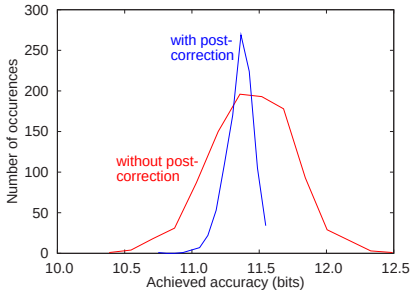

Fig. 11. Simulated accuracy without and with digital post-correction employed.

The achieved average accuracy of both designs is slightly less than the expected 12-bit. This is because the combined influence of several error sources can be more severe than the influence of a single error source only. To achieve true 12-bit performance, the design strategy should be targeted at a value slightly higher than 12-bit. One can see that the average accuracy of the system with post-correction is slightly lower than that of the converter without post-correction. Nevertheless, the spread of the accuracy of the converter without correction is larger. When a yield of (for example) 99% is required, the guaranteed accuracy of the converter with post-correction will be better than that of the converter without correction. The reason why the spread of the accuracy of the converter with post-correction is much smaller, even though the spread of its error-sources is larger, is because its accuracy is dependent only on the non-linearity of the SHA, and this parameter is modelled with a fixed value. The other error sources have no influence on the final accuracy as they can be corrected completely by the post-correction algorithm.

VII. Conclusion

In this paper, a design strategy for pipelined ADC's employing digital post-correction is presented. By modelling important error sources and deriving their relation with the overall accuracy, accuracy requirements for each component of the converter can be derived. To make the theory generally applicable, the model is implemented on system level. During the actual transistor-level design of the converter, the system-level requirements can be translated directly to low-level constraints.

By making use of the presented strategy, the requirements on the accuracy of the analog blocks can be minimized while the correction technique guarantees the desired accuracy target.

Finally, simulation results were given to verify the presented strategy with an example of a 12-bit pipelined ADC.

References

[1] A. N. Karanicolas, H.-S. Lee, and K. L. Bacrania, "A 15-b 1-MSample/s digitally self-calibrated pipeline ADC," *IEEE J. Solid-State Circuits*, vol. 28, pp. 1207–1214, Dec. 1993.

[2] H.-S. Lee, "A 12-b 600 ks/s digitally self-calibrated pipelined algorithmic ADC," *IEEE J. Solid-State Circuits*, vol. 29, pp. 509–515, Apr. 1994.

[3] P. J. A. Harpe, "Design of a high-speed, high-resolution pipelined AD converter," Master's thesis, Technische Universiteit Eindhoven, Eindhoven, The Netherlands, Jan. 2004.