

# Digital Post-Correction of Open-Loop Track-and-Hold Circuits

Pieter Harpe, Athon Zanicopoulos, Hans Hegt, Arthur van Roermund  
Mixed-signal Microelectronics Group, Eindhoven University of Technology,  
PT 5.13, P.O.Box 513, 5600 MB Eindhoven, The Netherlands  
email: [p.j.a.harpe@tue.nl](mailto:p.j.a.harpe@tue.nl), phone: +31 40 247 4703, fax: +31 40 245 5674

*Abstract*—The front-end track-and-hold (T&H) circuit is one of the most critical components of an analog-to-digital converter (ADC). Considering a pipelined ADC, it is known that the errors of each stage of the pipeline will be attenuated by the interstage amplifiers, however, the errors of the front-end T&H will be directly present in the output codes produced by the ADC. Because of this, the accuracy requirements for the front-end T&H are the most stringent of all sample-and-hold stages in the ADC.

Recently, the design of open-loop T&H circuits is getting more and more attention. Advantages of open-loop circuits include low power-consumption, high-speed operation, simple reliable design, and ability to operate at low power-supplies. However, a major disadvantage of open-loop circuits is their relatively poor linearity. Therefore, unless some correction technique is applied, the use of open-loop architectures is limited to converters with a low accuracy (up to 8-bit). Though some digital correction techniques for the improvement of the linearity of open-loop structures have been presented previously [1], [2], they are limited to specific implementations or specific types of non-linearity (e.g. third order distortion). In this paper, a general digital post-correction method is presented. This method improves the linearity of the front-end T&H circuit in the digital domain independent on the actual T&H design or the shape of the non-linearity. The method includes both a measurement procedure to determine the actual non-linearity and a correction algorithm. As it is not relying on accurate reference components, it can be integrated easily on-chip. The method was verified by simulations on a T&H circuit designed on transistor-level in a  $0.18\mu\text{m}$  technology, and shows that its linearity can be improved from 9.5-bit up to 13.5-bit, while operating at a sample frequency of 500MHz.

## I. INTRODUCTION

In most ADCs, the track-and-hold circuit (T&H) is relying on feedback to achieve a certain linearity, sufficient for the required accuracy of the overall ADC. However, in order to achieve a higher sampling frequency, a lower power consumption and the ability to operate at a lower power supply, open-loop T&H circuits would be advantageous [1], [3]. On the other

side, the relatively low linearity of basic open-loop structures limits their application to ADCs with an accuracy of up to around 8-bit. To improve the linearity, either the circuit itself can be improved (e.g. as in [4], [5]), or the actual non-linearity can be measured and corrected in the digital domain (e.g. as in [1], [2], [6]). In this work, our method for the on-chip measurement and correction of non-linearity in open-loop T&H circuits proposed previously in [7] will be discussed in detail.

In section II, the principle of self-measurement and digital post-correction of the T&H's distortion is presented. Section III covers the theory on the correction method, while section IV discusses the results on a simulated transistor-level T&H circuit. Finally, conclusions are drawn in section V.

## II. CORRECTION OF OPEN-LOOP T&H CIRCUITS

Open-loop T&H circuits are able to achieve high sample frequencies combined with a relatively low power consumption. Moreover, as these circuits typically use a small number of stacked transistors, they can operate at a low power-supply, enabling portability to future sub-micron CMOS processes. On the other hand, the intrinsic non-linearity of open-loop T&H circuits limits the accuracy that can be achieved with these topologies. A few digital post-correction techniques have been proposed recently [1], [2], [6], aiming at correction of the static non-linearity of interstage S&H circuits. However, these techniques can not be applied directly to the front-end T&H. Moreover, [1] and [2] consider third-order distortion only and they require complex digital processing. Our aim is to correct for the distortion of the front-end T&H circuit in the digital domain with a simple algorithm. Moreover, the method should be as general as possible by taking not only the third-order distortion, but also higher-order terms into account.

In the following subsections, the self-measurement of distortion components and the self-correction method will be discussed.

### A. Self-measurement

The static non-linearity of the front-end T&H circuit is measured on-chip using the setup depicted in fig. 1 at start-up. After the measurement phase, the DAC is disconnected and a digital post-correction (DPC) algorithm is added to the system to correct for the non-linearities as determined by the measurement procedure.

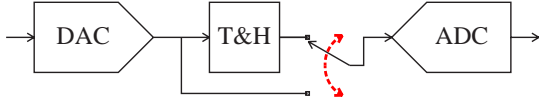


Fig. 1. Self-measurement of non-linearity of the T&H circuit.

During self-measurement, an inaccurate DAC is used to stimulate the ADC with several DC voltage levels. With a switch, the front-end T&H can be either included or omitted from the chain. The ADC digitizes the levels produced by the DAC, and an algorithm is performed to optimize the parameters of the DPC algorithm based on the measurements.

Suppose an  $M$ -bit DAC is used. In that case,  $2^M$  levels can be generated by the DAC. The accuracy of the generated levels is not important for the used measurement algorithm as long as each level remains constant during the measurement phase. Also note that  $M$  is allowed to be much smaller than the resolution of the ADC, as will be shown in sections III and IV. For each input code  $i$  (with  $0 \leq i < 2^M$ ) applied to the DAC, two output codes can be measured from the ADC: one with the T&H inserted in the system (yielding code  $C_i$ ), and one with the T&H omitted (yielding code  $D_i$ ). Together, this results in a total number of measurements equal to  $2 \cdot 2^M$ . Fig. 2 visualizes an example of the self-measurement procedure. In this case, a system composed of a 3-bit DAC and a 12-bit ADC is considered. The difference between codes  $C_i$  and  $D_i$  corresponds exactly to the non-linearity introduced by the T&H circuit. Therefore, a DPC algorithm that remaps each output code  $C_i$  to the corresponding output code  $D_i$  exactly compensates the T&H's non-linearity. This mapping function can be written as:  $D_i = R(C_i)$ , where  $R(\cdot)$  represents the mapping-function. A major advantage of the presented approach is that this mapping always gives an exact compensation independent of errors in the DAC or the ADC.

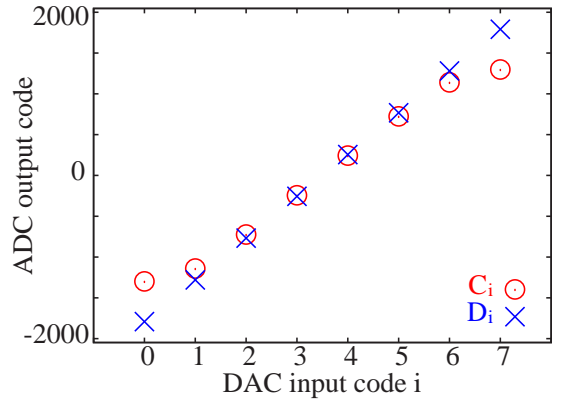


Fig. 2. Example of measured output codes  $C_i$  and  $D_i$ .

### B. Self-correction

The function  $D_i = R(C_i)$  produced by the measurement procedure does not yet provide a complete mapping, as the set of codes  $C_i$  for which the mapping is known is smaller than the set of codes that can be produced by the ADC. To generate a mapping that can be applied to all possible codes, and to reduce the algorithm's complexity, the non-linear mapping function  $R(C_i)$  is approximated by a piecewise-linear function. The set of codes  $C_i$  is split in  $S$  subsets or segments, and for each segment  $j$  (with  $0 \leq j < S$ ) the best-fit linear function is derived based on the available data pairs  $(C_i, D_i)$ . As a result of this linearization, it becomes also possible to interpolate the function for codes that were not available in the original mapping function. The more segments are used, the more digital resources will be required by the algorithm, but the more accurate the approximation can be.

An example of a mapping function and its piecewise-linear approximation is illustrated in fig. 3. The non-linearity of the T&H of a 12-bit ADC is measured by a 3-bit DAC. This results in a mapping function composed of 8 data points (indicated by the circles). Using  $S = 3$  segments (equally distributed over the code range  $[-2^{N-1}, +2^{N-1} - 1]$  with  $N = 12$ ) to approximate the measured function  $R(C_i)$  yields the three line segments as can be seen in the figure. Obviously, each line segment  $j$  (with  $0 \leq j < S$ ) can be described with two parameters: gain  $G_j$  and offset  $O_j$ . The corrected code  $D$  of an uncorrected code  $C$  inside segment  $j$  equals:

$$D = G_j \cdot C + O_j \quad (1)$$

### C. Implementation of the self-correction algorithm

In this section, the actual implementation of the piece-wise linear correction function in the digital do-

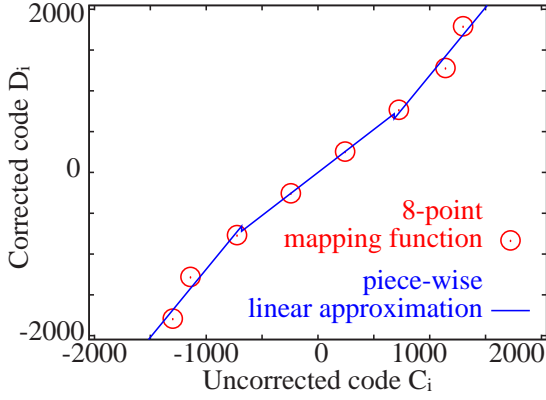


Fig. 3. 8-point mapping function and its piece-wise linear approximation.

main will be discussed. The most straightforward solution is to apply the correction function (1) directly to each code  $C$  produced by the ADC at runtime. However, this comes at the cost of a multiplication and a summation for each output sample. Especially a multiplication at a high sample frequency can be a rather costly operation. In this section, an alternative implementation is proposed that decreases the complexity of the correction algorithm. However, it can be applied only to a limited class of ADC's.

In situations where the ADC will include several different correction techniques, the complexity can often be reduced by combining the separate techniques efficiently into a single structure. Here, pipelined ADC's that use the correction method of [8] are considered. The DPC algorithm of [8] will then be combined with the technique presented in this work. The DPC algorithm of [8] can be applied to pipelined ADCs, and corrects for constant and linear errors inside the stages of the pipeline (e.g. capacitor mismatches, deviations of the sub-DAC levels, gain-errors, etc.). When the algorithm of [8] is used with a pipeline of  $K$  stages, the corrected output code  $C$  of the converter is given by a linear combination of contributions of each stage:

$$C = \sum_{t=0}^{K-1} \omega_t \quad , \quad (2)$$

where  $\omega_t$  is the contribution or *weight* of stage  $t$ . Suppose the output of each stage  $t$  is composed of  $n$  bits (with  $n$  typically between 1 and 3), then each weight  $\omega_t$  takes one out of  $2^n$  values, dependent on the output produced by the stage. The algorithm of [8] includes a self-measurement phase at start-up, during which the values of all weights are determined such that linear and constant errors in the pipeline are corrected optimally. Combining the algorithm of [8] (equation (2))

with the new algorithm for the correction of the front-end T&H (equation (1)) shows that the corrected code  $D$  can also be expressed as a summation of constant coefficients:

$$\begin{aligned} D &= G_j \cdot C + O_j = G_j \cdot \sum_{t=0}^{K-1} \omega_t + O_j \\ &= \sum_{t=0}^{K-1} \omega'_{t,j} \quad , \quad \text{with:} \end{aligned} \quad (3)$$

$$\begin{cases} \omega'_{t,j} = G_j \cdot \omega_t & \text{if } t \neq 0 \\ \omega'_{t,j} = G_j \cdot \omega_t + O_j & \text{if } t = 0 \end{cases} \quad (4)$$

As a result, the multiplication and summation for each generated output code can be omitted. At start-up, the coefficients  $\omega_t$  are determined first using [8]. Then, the measurement of the front-end T&H is performed, yielding  $G_j$  and  $O_j$ . Now, the new weights  $\omega'_{t,j}$  for each segment  $j$  can be calculated using (4) and stored into memory. From that moment on, the corrected output codes  $D$  can be calculated as linear combinations of stored coefficients (3).

As the new weights are dependent on the segment  $j$ , for each uncorrected code  $C$ , the correct set of weights belonging to the segment corresponding to code  $C$  has to be selected. This process can be implemented simply as illustrated in fig. 4. The division of the input range in segments is chosen identical to the quantization as performed by the first stage of the pipeline. Therefore, the output bits of the first stage (in this example a 1.5-bit stage) directly determine which set of weights has to be used for the correction of each value. This approach can be extended to more segments by combining the outputs of the first two (or more) stages to determine a segmentation. E.g. two 1.5-bit stages result in a segmentation of nine segments.

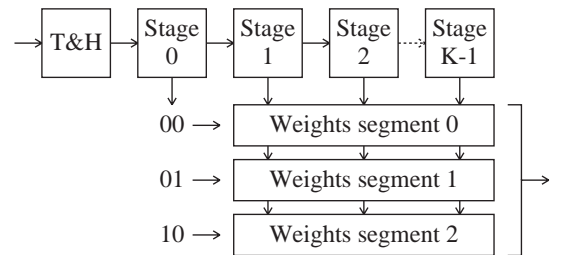


Fig. 4. DPC including piece-wise linear correction of the front-end T&H.

With the implementation described above, the non-linear errors of the front-end T&H can be corrected just at the cost of a start-up algorithm (not time-critical), some additional memory to store the new

coefficients, and some logic to select the correct set of weights for each input value. However, dependent on the properties of the ADC and the amount of segments used in the DPC algorithm, it might be more attractive to use a multiplication at runtime instead of additional memory to store the coefficients  $\omega'_{t,j}$ .

### III. THEORETICAL ANALYSIS

In this section, the effect of several design properties on the final performance will be investigated. The design properties that will be considered are: the number of segments in the piece-wise linear correction function, the resolution of the measurement DAC, and the resolution of the ADC.

#### A. Influence of the number of segments

In this section, it is investigated how large the number of segments in the piece-wise linear correction function has to be in order to achieve a certain improvement in overall linearity. To simplify the analysis, it is assumed that all segments are of equal size (i.e. the full scale range is split in  $S$  equal-size segments). Moreover, the ADC and measurement DAC are assumed to be ideal (free of mismatch and quantization errors) for the moment. Later on, the non-idealities of the ADC and DAC will be examined. Several different functions are used to model the non-linearity of the T&H in order to see the effect of the DPC on different non-linearities. The three different functions used are given in (5), describing a circuit with 3<sup>rd</sup>, 5<sup>th</sup> or 7<sup>th</sup> order distortion, respectively.

$$\begin{cases} y_3 = x - 0.1 \cdot x^3 \\ y_5 = x - 0.1 \cdot x^5 \\ y_7 = x - 0.1 \cdot x^7 \end{cases} \quad (5)$$

The input-range of the T&H is assumed to be  $-1 < x < +1$ . Then, this range is split in  $S$  segments, and for each segment, the best-fit linear approximation is derived. Finally, a sinusoid is applied as an input signal, and the output signal, based on the constructed piecewise-linear correction function, is calculated. From this output, the Signal-to-Noise-and-Distortion-Ratio (SNDR) is derived. The results for the three functions given in (5), using  $S = 1$  up to  $S = 9$  segments, are illustrated in fig. 5. Note that this figure shows the SNDR improvement, and not the absolute SNDR. Each 6dB increase corresponds to a gain of one bit of accuracy.

It can be seen that in all cases, a higher number of segments improves the performance. Though there

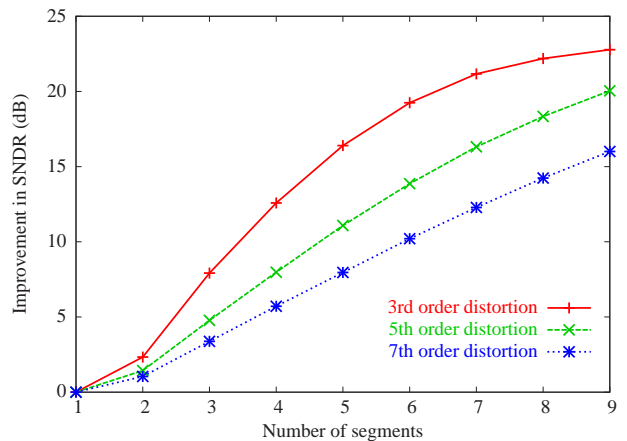


Fig. 5. SNDR improvement as a function of the number of segments for different types of non-linearity.

are differences between the different types of distortion, the same trend can be observed in all situations. However, the higher the order of the distortion component, the more segments will be required to achieve a certain improvement. The reason behind this is that distortion components of a high order are more difficult to approximate by piece-wise linear functions. Nevertheless, the most likely situation in practice is that the 3<sup>rd</sup> order distortion is the dominant component, followed by the 5<sup>th</sup> order distortion. In such a case, as shown by the results in fig. 5, it should be possible to achieve a linearity improvement of around 3 to 4-bit using a 9-segment correction function. Using more than 9 segments does not seem to be attractive as the required resources continually increase, while the added value (in practice) is probably very low.

#### B. Influence of the measurement DAC

As stated before in section II-A, the accuracy of the measurement DAC should not influence at all the precision of the measurement of the T&H. The only constraint on the DAC is that it provides enough information to optimize the parameters of the piece-wise linear correction function. This means that the DAC should provide at least 2 levels within each segment of the piece-wise linear function to determine the two parameters of this segment (see equation (1)). In other words: for a DPC using  $S$  segments, the DAC should generate at least  $2S$  levels, hence the resolution  $M$  of the DAC should be at least:

$$M \geq 1 + \log_2 S \quad (6)$$

In practice, the DAC will suffer from mismatch errors and non-linearities, causing the output-levels to be not exactly evenly distributed. Also, the segments

of the DPC will not be evenly distributed. Therefore, the DAC will need a slightly higher resolution than expected based on (6).

Simulations were performed to verify the expectations. While the ADC was fixed to 16-bit resolution, the number of segments was varied from 1 up to 9. The resolution of the measurement DAC was set to 4, 6, 8 and 10-bit, respectively. Mismatch with a  $3\sigma$ -level of 0.5LSB was added to both to the ADC and the DAC. The non-linearity of the T&H circuit was modelled as:

$$y = 0.9x - 0.01x^3 + 0.006x^5, \text{ where} \quad (7)$$

$x$  is the input and  $y$  the output of the T&H. The DAC was used to perform a self-measurement on the T&H, based on which the parameters of the DPC algorithm were optimized. Then, a sinusoid was applied to the input of the T&H, and the SNDR of the digital output of the ADC (including the DPC) was determined. The results are shown in fig. 6. It can be seen that the performance is essentially the same for the situations with a DAC resolution of 6, 8 or 10-bit. The 4-bit DAC is useful only for small segmentations ( $S < 8$ ), as expected. Knowing from the analysis in the previous section that in practice the segmentation will be limited to  $S \leq 9$ , it can be concluded that a 6-bit DAC is sufficient for all practical situations.

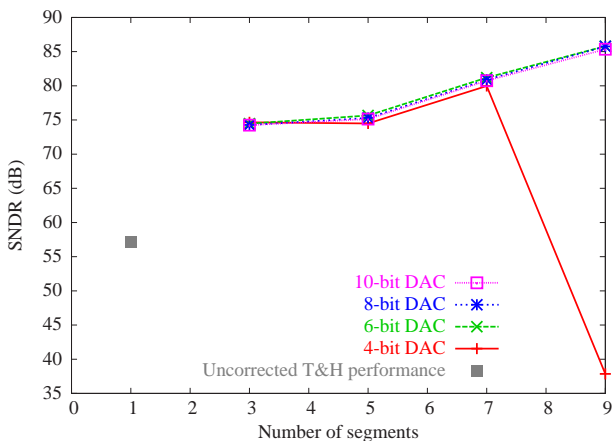


Fig. 6. SNDR as a function of the number of segments for different DACs. The ADC is fixed to 16-bit.

### C. Influence of the ADC

The final design parameter to study is the resolution of the ADC. Several errors of the ADC influence the final performance of the corrected system, like quantization errors, mismatch errors, non-linearities, etc. In a good design, the overall ADC error is in the order of

0.5LSB. However, in the presented system, the ADC error will appear in two different ways:

- during the quantization of each sample at runtime,
- during the quantization in the measurement phase, resulting in a suboptimal correction function.

As a result, the total error will be a small multiple of the original error of the ADC. Conclusively, the accuracy of the ADC limits the final ENOB to a level which is slightly worse than the ADC's performance.

Simulations were performed using a similar setup as described in section III-B, but now the DAC was fixed to 8-bit resolution, and the resolution of the ADC was varied between 8 and 16-bit. Mismatch with a  $3\sigma$ -level of 0.5LSB was added to both to the ADC and the DAC, and the T&H circuit was modelled as before (7). The results in terms of achieved SNDR are shown in fig. 7. For the 8, 10 and 12-bit ADCs, the performance is actually limited by the inaccuracy of the ADC itself, and not by the non-linearity of the T&H, as their performance is clearly approximating the theoretical SNDR of 6dB per effective bit. For the 14 and 16-bit ADC, performance is limited by the non-linearity of the T&H that remains after DPC. Increasing the segmentation could further enhance the performance, however, it is doubtful whether this would be realistic in practice. Nonetheless, the achieved improvement of 30dB using a 16-bit ADC with 9 segments is already a major improvement.

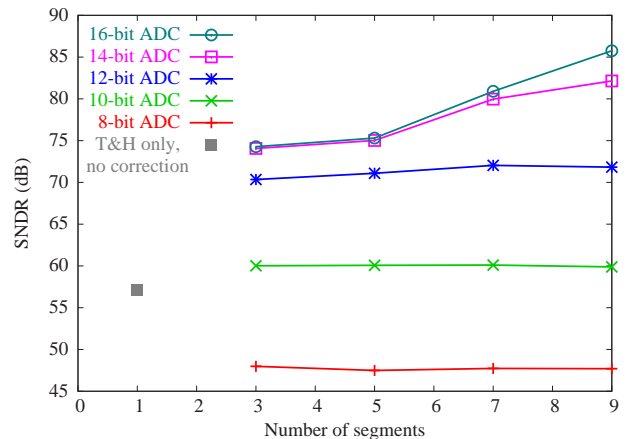


Fig. 7. SNDR as a function of the number of segments for different ADCs. The DAC is fixed to 8-bit.

## IV. TRANSISTOR-LEVEL SIMULATION RESULTS

In this section, the presented correction technique is verified by applying it to a transistor-level designed T&H circuit. After a brief discussion on the circuit design, simulation results will be given.



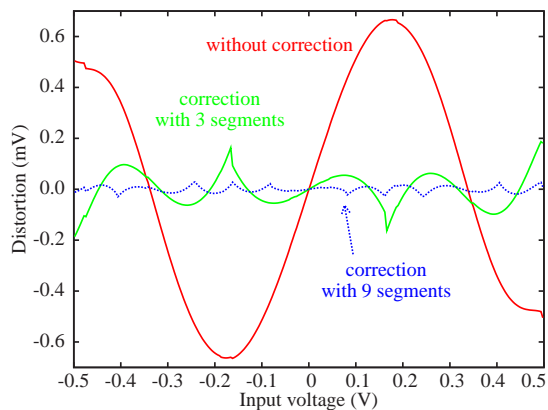


Fig. 9. Simulated static performance without and with DPC.

TABLE I

EFFECTIVE ACCURACY BASED ON STATIC PERFORMANCE

	Maximum deviation (mV)	Effective accuracy (bits)
Without correction	0.67	9.5
Correction with three segments	0.17	11.6
Correction with nine segments	0.04	13.5

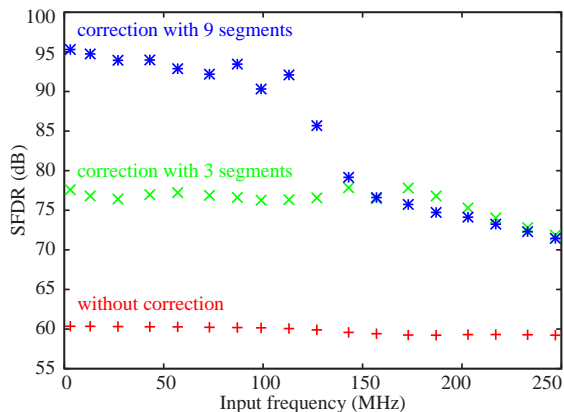


Fig. 10. Simulated dynamic performance without and with DPC using a sample frequency of 500 MHz.

## V. CONCLUSION

In this paper, a digitally post-corrected open-loop front-end T&H system was presented. A new on-chip measurement method determining the non-linearity of the T&H was proposed as well as a method for digital post-correction (DPC) of this non-linearity. The algorithm can be implemented efficiently when combined with an existing algorithm correcting for the errors inside the ADC. An open-loop T&H circuit was designed in CMOS  $0.18\mu\text{m}$ . Clock boosting, cross coupling and resistive source degeneration were used to improve the intrinsic linearity. Without post-correction, 9.5 bit static linearity, a SFDR of 60 dB,

and 22 mW power consumption were achieved when operating at a 500 MHz sample frequency. With DPC employed, the performance improved to 13.5 bit static accuracy, 90 dB SFDR up to 100 MHz input signals, and 72 dB SFDR up to 250 MHz input signals. The presented DPC method is flexible as the achievable performance is determined by the amount of digital resources used in the correction circuit. Therefore, the approach is especially suitable for flexible mixed-signal platforms.

## REFERENCES

- [1] B. Murmann and B. E. Boser, "A 12 b 75 MS/s pipelined ADC using open-loop residue amplification," in *proc. IEEE ISSCC 2003*, 2003, pp. 328 – 497.
- [2] A. Ravindran, A. Savla, and J. Leonard, "Digital error correction and calibration of gain non-linearities in a pipelined ADC," in *proc. IEEE ISCAS 2004*, May 23 – 26, 2004, pp. I-1 – I-4.
- [3] S. M. Louwsma, E. J. M. van Tuijl, M. Vertregt, P. C. S. Scholtens, and B. Nauta, "A 1.6 GS/s, 16 times interleaved track&hold with 7.6 ENOB in  $0.12\mu\text{m}$  CMOS," in *proc. ES-SCIRC 2004*, Sept. 21 – 23, 2004, pp. 343 – 346.
- [4] S. Ouzounov, E. Roza, J. A. Hegt, G. van der Weide, and A. H. M. van Roermund, "A CMOS V-I converter with 75-dB SFDR and  $360\text{-}\mu\text{W}$  power consumption," *IEEE J. Solid-State Circuits*, vol. 40, no. 7, pp. 1527 – 1532, July 2005.
- [5] H. Voorman and H. Veenstra, "Tunable high-frequency Gm-C filters," *IEEE J. Solid-State Circuits*, vol. 35, no. 8, pp. 1097 – 1108, Aug. 2000.
- [6] D.-L. Shen and T.-C. Lee, "A linear-approximation technique for digitally-calibrated pipelined A/D converters," in *proc. IEEE ISCAS 2005*, May 23 – 26, 2005, pp. 1382 – 1385.
- [7] P. Harpe, A. Zanicopoulos, H. Hegt, and A. van Roermund, "Digital post-correction of front-end track-and-hold circuits in ADCs," in *proc. IEEE ISCAS 2006*, May 21 – 24, 2006, pp. 1503 – 1506.
- [8] 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, no. 12, pp. 1207 – 1214, Dec. 1993.
- [9] A. M. Abo and P. R. Gray, "A 1.5-V, 10-bit, 14.3-MS/s CMOS pipeline analog-to-digital converter," *IEEE J. Solid-State Circuits*, vol. 34, no. 5, pp. 599 – 606, May 1999.