DATA CONVERSION AND PROGRAMMABLE CALIBRATION FOR SMART SENSORS

P. Malcovati, C. Azeredo Leme, P. O’Leary*, F. Maloberti† and H. Baltes

Physical Electronics Laboratory
ETH Hönggerberg, HPT-H6, 8093 Zürich, SWITZERLAND
Tel. +41-1-377 20 89 Fax +41-1-371 07 21

* on leave from the Departement of Electronics, University of Pavia, ITALY
† Institute for Sensor Interfaces, Joanneum Research
Steyrergasse, 17, 8010 Graz, AUSTRIA
Tel. +43-3136-8020 231 Fax +43-3136-8020 181

Abstract - In this paper we present a new architecture for a smart sensor interface. It is based on an oversampled A/D converter associated with a small ROM containing calibration coefficients. The non-linear function desired is obtained by piecewise linear interpolation between the values stored in the ROM. This solution has the advantages of high programming flexibility, long-term stability and low area consumption. Moreover it is suitable for co-integration with sensors because of its minimum analog content. Simulation and experimental results are reported together with a detailed theoretical analysis and some design guidelines.

I. INTRODUCTION

The response of silicon integrated sensors is often non-linear. Therefore, a smart A/D interface should also calibrate this non-linearity, as shown in Fig. 1.

In the digital domain look-up tables containing the full function mapping can be used [1]. This solution is flexible and easily programmable but its speed of operation is limited and it requires a large area to hold the coefficients. Analog approaches overcome these limitations [2], [3], [4], [5]. However these are sensitive to temperature and aging. Moreover, they are difficult to program and less flexible in the functions to be implemented.

In this paper we present a new solution that combines A/D conversion and calibration. The calibration is performed by a piecewise linear function (Fig. 2), whose points are stored in a small ROM. With respect to the conventional look-up table approach, a big reduction in the chip area is achieved, while obtaining satisfactory results. Moreover, this solution has the advantages of high programming flexibility, long-term stability and, very important for sensor applications, minimum analog content.

II. ARCHITECTURE PRINCIPLE

The block diagram of the system is shown in Fig. 3. The input signal is coded by a first order sigma-delta modulator. Its output is then transformed into a \( N_T \) bits word by a “sinc” transversal filter. Because of the poor filtering action of the “sinc” filter, this signal is still affected by the shaped quantization noise. Nevertheless, it is suitable for transcoding by using a ROM, which implements the desired function \( f(x) \). We assume that \( f(x) \) is such that it produces a limited modification in the quantization noise spectrum. The coded word is then filtered and decimated.
as is required by a conventional sigma-delta modulator.

Since the output of the “sinc” filter, $x(t)$, is an $N_{TF}$ bit oversampled representation of the input signal $(Y_p)$, $f[x(t)]$ is obtained after the decimating filter. Consequently, the resulting system transfer characteristic is the piecewise linear interpolation between the values stored in the ROM.

The linear interpolation introduces an additional error (interpolation error) that will be analyzed in detail in the next section. Nevertheless, the approach proposed allows us to find the best tradeoff between accuracy and hardware complexity (i.e., the interpolation error should be slightly smaller than the quantization error).

### III. THEORETICAL ANALYSIS

There are three noise sources in the system: the interpolation error, the ROM truncation error and the sigma-delta quantization error. The design target is to keep the total noise power below the maximum allowed. In order to provide some design guidelines, in this section the contribution of each error source is calculated.

The interpolation error, $e_i = f[x(t)] - f[x(t)]$, can be written as

$$e_i(x) = f(x) - \left( f(x_0) + \frac{(x - x_0) [f(x_0 + \Delta_{TF}) - f(x_0)]}{\Delta_{TF}} \right)$$  

(1)

where $x_0$ is the previous quantized value in respect to $x$ and $\Delta_{TF}$ is the quantization step at the output of the transversal filter. They can be expressed, as a function of $N_{TF}$, as

$$x_0 = \frac{n \cdot \Delta}{2^{N_{TF}} - 1}, \Delta_{TF} = \frac{\Delta}{2^{N_{TF}} - 1}$$  

(2)

where $\Delta$ is the input signal swing and $n$ is the code corresponding to $x_0$.

Equations (1) and (2) show that $e_i$ is strongly dependent on the shape of $f(x)$, especially of its second derivative, and it decreases as $N_{TF}$ increases.

The interpolation error, as well as the quantization error, can be considered a zero-mean noise. Therefore its power ($P_i$) corresponds to the variance of $e_i$. If the probability density function of the input signal is uniform between $-\Delta/2$ and $\Delta/2$, $P_i$ can be written as

$$P_i = \int_{-\Delta/2}^{\Delta/2} e_i^2(x) \cdot \frac{1}{\Delta} \cdot dx$$  

(3)

As expected, for a quadratic calibrating function, $f(x) = a + b \cdot x + c \cdot x^2$, equation (3) becomes

$$P_i = \frac{c^2 \cdot \Delta^4}{30 \cdot (2^{N_{TF}} - 1)^4}$$  

(4)

The truncation error depends on the word length of the ROM $(N_R)$. Since the quantization step at its output is $\Delta/ (2^{N_R} - 1)$, the truncation noise power results

$$P_t = \frac{\Delta^2}{12 \cdot (2^{N_R} - 1)^2}$$  

(5)

The interpolation noise and the truncation noise are residual non-linearities. This leads to a strongly colored spectrum. Therefore, unfortunately, their contributions are not attenuated by the decimating filter.

Finally, the quantization noise power in the base-band for an Lth order sigma-delta modulator, in first approximation, is given by [6]
\[ P_q = \frac{\Delta^2}{12} \cdot \frac{\pi^{2L}}{2L + 1} \cdot \frac{1}{M^{2L + 1}} \]  

where \( M \) is the oversampling ratio.

It can be observed that the three noise contributions \( P_i, P_f \) and \( P_q \) are controlled by different design parameters: \( N_T, N_R \) and \( M \). Therefore, it is quite easy to design the system in order to have, for example, \( P_q = P_i = P_f = P_{\text{max}}/3 \).

IV. IMPLEMENTATION AND RESULTS

The proposed system was simulated with MATLAB and implemented in FPGA. A full integrated version with sensor on chip is being designed. The optimum design parameters to achieve at least 10 bits are summarized in Tab. 1, the decimator used is a "sinc\(^2\)" filter.

The signal to noise ratio of the system (S/N) can be calculated using equations (4), (5), (6) and assuming the three contributions to be uncorrelated. This leads to S/N = 63 dB. Simulation results and experimental measurements are summarized in Tab. 2. They are slightly different from the calculated value, because of some correlation between \( P_f \) and \( P_i \) which, in the case considered, partially cancels the two effects. However, the results obtained show that the simple analysis proposed is adequate for the system design.

The input signal, used for simulations and experimental verifications, emulates the sine-wave response of a non-linear sensor. Fig. 4 and Fig. 5 respectively show the spectra of the signals at the input and at the output of the ROM. In Fig. 4 a strong harmonic distortion, produced by the non-linearity, can be observed. It disappears in Fig. 5 because of the calibration. Moreover, the DC component, also produced by the distortion, is strongly reduced at the output of the ROM.

V. CONCLUSIONS

In this paper we presented a new architecture for a smart sensor interface, based on oversampling and noise shaping techniques. A ROM look-up table is driven by an oversampled data stream, obtained by first order filtering the output of a sigma-delta modulator. The placement of the ROM before the decimating filter allows us to considerably reduce its size, with respect to a conventional digital look-up table approach.

The system transfer characteristic obtained is the piecewise linear interpolation between the values stored in the ROM. The ensuing interpolation error was investigated and a FPGA prototype was implemented and tested.

REFERENCES

Fig. 1 - Example of calibration system transfer characteristic

Fig. 2 - Example of piecewise linear calibrating function

Fig. 3 - Block diagram of the system

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Calculated</th>
<th>Simulated</th>
<th>Measured</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L$</td>
<td>1</td>
<td>0.25 $10^{-6}$</td>
<td>0.19 $10^{-6}$</td>
<td>0.22 $10^{-6}$</td>
</tr>
<tr>
<td>$M$</td>
<td>256</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\Delta$</td>
<td>2 V</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$N_{TF}$</td>
<td>4</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>$N_R$</td>
<td>11</td>
<td>63.0</td>
<td>64.2</td>
<td>63.6</td>
</tr>
<tr>
<td>$f(x)$</td>
<td>$x + c x^2$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$c$</td>
<td>0.1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Tab. 1 - Design parameters

Tab. 2 - Calculated, simulated and experimental results

Fig. 4 - Measured spectrum of the signal before the ROM

Fig. 5 - Measured spectrum of the signal after the ROM