Analog Integrated Circuits and Signal Processing, 14, 207–222 (1997)
© 1997 Kluwer Academic Publishers, Boston. Manufactured in The Netherlands.

Integrated Smart Sensor Calibration

GERT VAN DER HORN AND JOHAN H. HUIJSING
Electronic Instrumentation Laboratory/DIMES, Technical University Delft, Mekelweg 4, 2628 CD Delft, The Netherlands
horn@ei.et.tudelft.nl

Received August 2, 1996; Accepted December 3, 1996

Abstract. In many applications electronic sensors are used to improve performance and reliability of measurement systems. Such sensors should provide a correct transfer from the physical signal to be measured to the electrical output signal. One important step to achieve this, is to *calibrate* each sensor by applying different reference input signals and adjusting the sensor transfer accordingly. Besides expensive reference equipment the calibration process takes much time and attention per individual sensor, which means a considerable increase in sensor production costs. By including at the sensor or sensor interface chip a *programmable calibration facility* the calibration of such *smart sensors* can easily be automated and can be executed for a batch of sensors at a time, thus minimizing the calibration time and costs. This paper presents a calibration method and options for integration in the smart sensor concept, in *hardware* as well as in *software*. An advantage of the proposed method is that it does not need a large matrix of calibration data, which needs to be stored in a look-up table or converted into a correction formula, but instead it uses a step-by-step approach to correct the sensor transfer at each calibration measurement until the error is sufficiently small.

Key Words: calibration, smart sensors, linearization, analog signal processing, microsystems, microcontroller-based sensor systems

I. Introduction

A. Integrated Smart Sensors

Sensors are applied in many products to improve performance and reliability. To promote the use of sensors in an even wider range of products, especially consumer products, they should be made cheap and easy-to-use. To achieve this, sensors should preferably be manufactured in a standard IC process, because this will make it possible to equip them with smart electronics and fabricate them in mass-production [1]. The functions that need to be integrated with the sensor to simplify its application [2], are:

- *analog sensor readout and signal conditioning*
- *analog-to-digital conversion, to provide a digital output signal*
- *a bus interface, to simplify communication to microcontrollers, PC's, and other devices*
- *calibration of the sensor transfer curve, preferably a digitally programmable calibration.*

First of all, the usually weak sensor signals need to be amplified and converted into convenient electrical signals (voltage or current, rather than capacitance, resistance, etc.). Secondly, an AD-conversion is necessary to allow communication with the digital world of microcontrollers and computers. To simplify setting up systems with multiple sensors and actuators, each device will also need a bus-interface on-board. Last but not least, on-board calibration electronics will make it possible to provide customers with smart sensors with standardized output signals, allowing easy exchange of sensors and easy (custom) re-calibration [2], [3], [4]. Once smart sensors with such functionality are available at low-cost, a boost in the application of sensors in consumer products can be expected, resulting in increased performance and reliability of many of our daily-used tools. This paper concentrates on the integrated calibration facility.

B. Calibration of Smart Sensors

Calibration refers to the process of applying several physical reference signals to a sensor device, and measuring the (electrical) output signal, so that the input-output relation of that sensor can be derived with a certain accuracy. Calibration also includes the procedure of correcting the transfer of the sensor, using the reference measurements, in such a way that a specified input-output relation can be guaranteed with a certain accuracy (and under certain conditions). Besides for sensor calibration, it can be necessary to change offset and scale at the input of the sensor interface circuit to adapt the sensor signal to the signal range of the interface circuit for optimal signal-to-noise ratio and accuracy. If the sensor exhibits a *characteristic* non-linearity (for example logarithmic), the best way to compensate this is by using an analog circuit with an approximately reciprocal non-linear transfer (exponential). Next to those techniques, which will not be further discussed in this paper, it is still necessary to perform a calibration of the remaining errors and to relate the output signal precisely to physical input signal. The calibration techniques presented in this paper are intended for that purpose, considering linearity correction in the order of 10% to 50% of the full-scale.

As a first option, an analog calibration circuit is presented of which gain, offset and second- and third-order linearity are digitally controlled. This approach offers a limited calibration accuracy (8-bit, 3rd-order polynomial), but provides large bandwidth and high signal processing speed (~100kHz–1MHz).

Often sensors also need to be calibrated for cross-sensitivity, especially to temperature. This requires a so-called 2-dimensional calibration, because the sensor's sensitivity with respect to 2 (physical) input signals is measured and adjusted. As a second option, a digital calibration facility implemented in a microcontroller is presented. This digital implementation offers a two-dimensional calibration with high accuracy (16-bit, 5×5 polynomial correction) but has a limited signal processing speed (~ 10–100 samples/second) because of CPU time required for the correction of each sample. This kind of calibration is especially suited for a strain-gauge pressure sensor. A complete (stand-alone) pressure sensor system can be setup using a pressure sensor, a smart sensor interface chip with integrated temperature sensor, and a microcontroller with calibration program.

The calibration corrections can be implemented as a digitally programmable circuit and integrated with the sensor, as in the above-described smart sensor concept. It is important to realize that this does NOT eliminate the need to do calibration measurements. It is meant to make production of sensors with standardized/calibrated output-ranges easier, so that the required external signal-processing can be reduced and simplified for the customer. To reduce costs of calibration for the manufacturer, it is important to minimize the number of reference measurements, the communication, and the computation power for each sensor to be calibrated [4]. The amount of correction data and correction computations on each sensor should also be small but still provide a good linearity correction. This demands for a special calibration method, which will be explained in the next section.

II. One-Dimensional Polynomial Calibration Method

A. Calibration Principle

In this case, the sensor is considered to be sensitive only to one input variable, say pressure, hence the term 1-dimensional calibration. Most calibration methods rely on collecting a complete set of measurement data to calculate a correction formula or lookup table. The proposed calibration method however operates on the principle that each calibration measurement can be used directly to calculate one programmable coefficient in a correction function, which can then immediately be used to correct the sensor output. The next calibration makes use of this corrected sensor signal. Each succeeding correction is applied in such a way that the previous calibrations remain undisturbed. For example a first measurement is done to calibrate the offset, which means the transfer function is shifted or *translated*, as shown in Fig. 1(a). A next calibration measurement is used to correct for the gain error, without affecting the offset calibration. This is achieved by *rotating* the function around the previous calibration point, as shown in Fig. 1(b). The next calibration measurements can be used to correct linearity, which is achieved by "*bending*" the function in such a way that the previous calibration points stay fixed, as shown in Fig. 1(c). More calibrations can be done in the same manner to further linearize the sensor transfer function. This explains the basic principle of building up a polynomial correction curve. We will now describe it in more mathematical detail.

The input variable is indicated as x, the uncalibrated sensor response is denoted by $f(x)$, the desired value of the sensor response at the n^{th} calibration point by y_n, the calibration coefficients by a_n, and the calibrated transfer functions by $f_n(x)$.

The first calibration function $f_1(x)$ is given by:

$$f_1(x) = f(x) + a_1 \cdot y_{ref} \qquad (1)$$

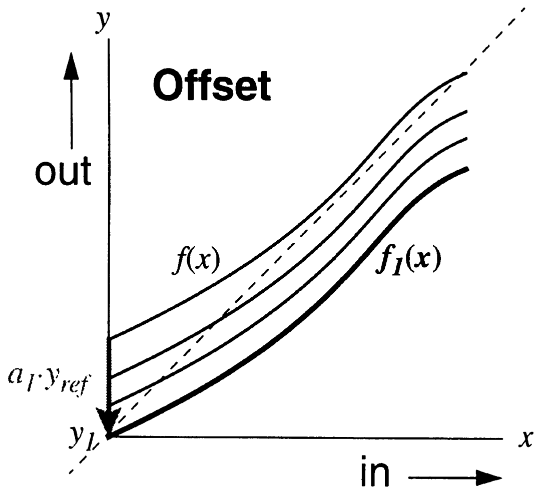

Figure 1(a). Translation of the transfer curve.

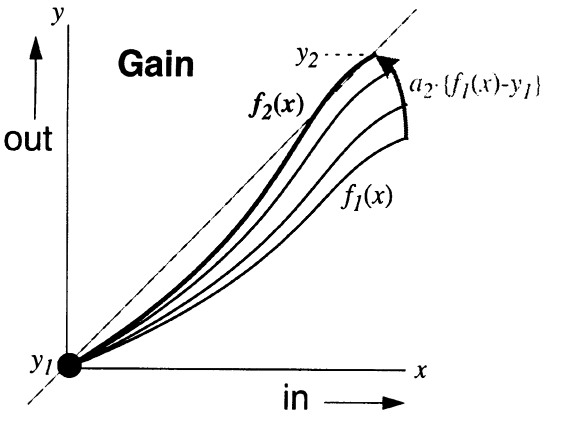

Figure 1(b). Rotation of the transfer curve.

A reference signal y_{ref} is used only to give the correction factor the same unit as the functions, so that a_1 can be just a dimensionless number. From the first calibration measurement $f(x_1)$ the desired value of a_1 can be calculated, to equalize the corrected sensor output $f_1(x_1)$ to the desired output value y_1 at input x_1.

The second calibration function $f_2(x)$ uses the previously corrected function $f_1(x)$ as follows:

$$f_2(x) = f_1(x) + a_2 \cdot \{f_1(x) - y_1\} \tag{2}$$

Because of the previous equalization $f_1(x_1) = y_1$, the term with calibration coefficient a_2 is always zero in the first calibration point (x_1, y_1), thus $f_2(x_1) = f_1(x_1) = y_1$. A second calibration measurement $f_1(x_2)$ is used

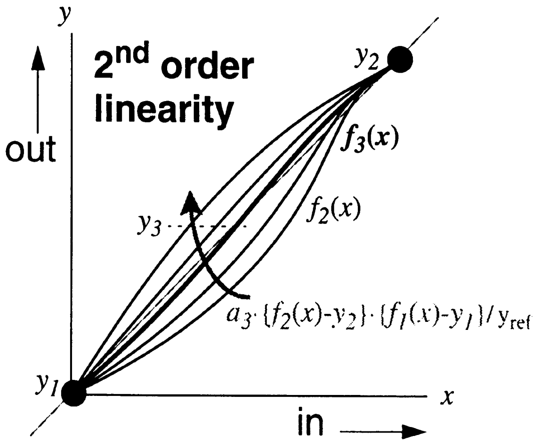

Figure 1(c). Bending of the transfer curve.

to calculate the correct value for a_2, so that the corrected sensor output $f_2(x)$ is now also equalized to the desired output y_2 at input x_2. The following calibration steps will build up a polynomial linearization function in the same successive manner. The third and fourth calibration functions are respectively given as:

$$\begin{aligned} f_3(x) &= f_2(x) + a_3 \cdot \{f_1(x) - y_1\} \\ &\quad \cdot \{f_2(x) - y_2\}/y_{ref} \qquad (3) \\ f_4(x) &= f_3(x) + a_4 \cdot \{f_1(x) - y_1\} \cdot \{f_2(x) - y_2\} \\ &\quad \cdot \{f_3(x) - y_3\}/y_{ref}^2 \qquad (4) \end{aligned}$$

A third calibration measurement $f_2(x_3)$ is used to calculate a_3 so that the calibrated function $f_3(x)$ equals the desired function at a third point (x_3, y_3). A fourth measurement $f_3(x_4)$ is used to calculate a_4 which linearizes $f_4(x)$ at a fourth point (x_4, y_4). The division by y_{ref} and y_{ref}^2 is needed to maintain the dimension of the correction factors, so that a_3 and a_4 can be plain numbers. The calibration process can be continued in the same manner, till the error signal, the difference between the desired and the calibrated transfer function, is sufficiently reduced. The advantage of the method just explained is that no mathematical iterations are required to compute the coefficients of the polynomial factors.

To demonstrate how the algorithm succeeds in reducing the error step-by-step, Fig. 2 shows an example transfer function $f(x)$, with normalized input and output, which is calibrated at $(-1, -1)$ and $(1, 1)$ and $(0, 0)$. Fig. 3 shows the corresponding uncalibrated

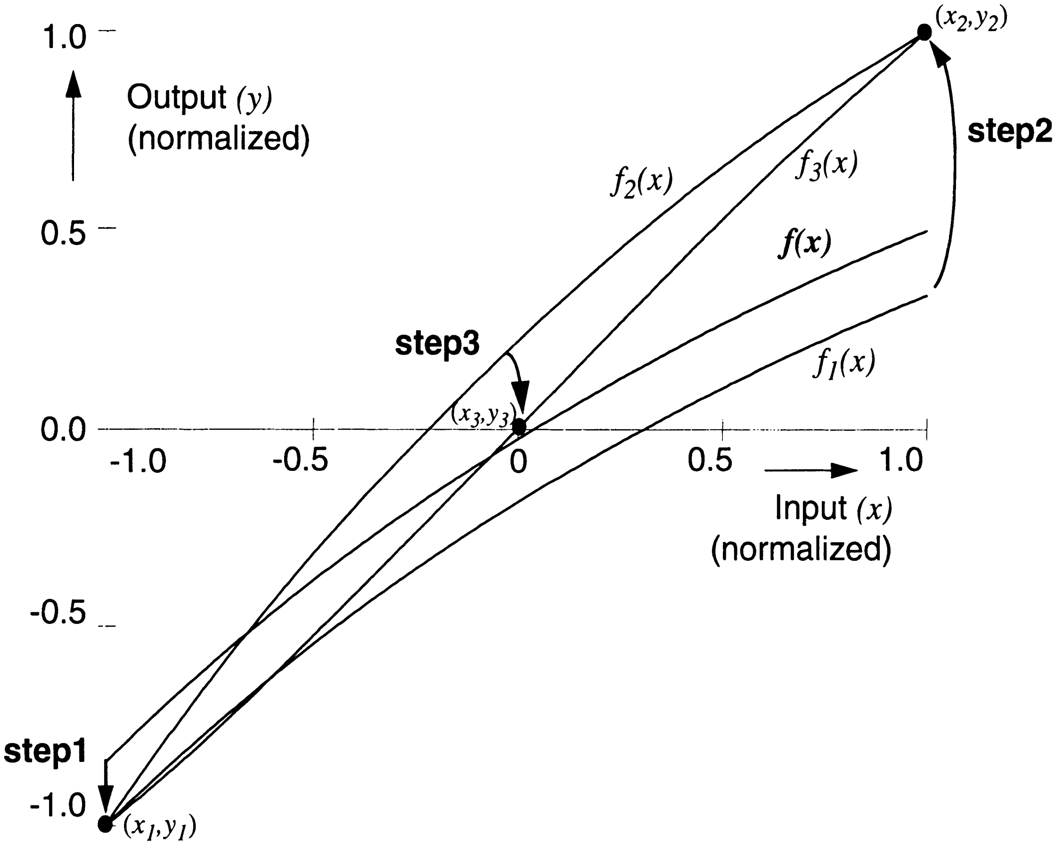

Fig. 2. Step-by-step polynomial transfer curve calibration.

error curve and the error curves after 1, 2 and 3 calibration steps, respectively $error_0$ to $error_3$. Simulation results have demonstrated that generally, for three or more calibration steps, a good linearization is obtained when the calibration points are selected in the following sequence: the first point at one end of the sensor range of operation, the second point at the other end of the range, and further calibration points halfway between two previously selected points.

B. Implementation: Analog Signal Processor for Polynomial Sensor Calibration

The signal processing explained in formulas (1) to (4) can be displayed in a flow diagram as shown in Fig. 4. The sensor output signal y, exhibiting a transfer function $f(x)$ with respect to the physical input signal x, is processed through a network of adders, multipliers and trimmable calibration factors $a_1 - a_n$. First, all calibration factors are set to zero so that the sensor signal $y = f(x)$, left in the diagram, goes straight to the output signal $f_4(x)$ at the right side of the diagram. Then a first calibration measurement is done by providing reference input x_1, and adjusting a_1 to obtain an offset calibrated signal $f_4(x) = f_1(x) = f(x) + a_1 \cdot y_{ref}$, which should be equalized to y_1 at input x_1. A second calibration is done at the next reference input x_2 by adjusting a_2, which will add a factor $a_2 \cdot \{f_1(x) - y_1\}$ to $f_1(x)$ so that a gain-calibrated function $f_4(x) = f_2(x)$ is obtained. The third part adds the term $a_3 \cdot \{f_1(x) - y_1\} \cdot \{f_2(x) - y_2\}/y_{ref}$ to the calibration function, to realize a second-order polynomial linearity correction. From there the flow diagram becomes a repetitive chain as the part in the dashed box in Fig. 4 can be repeated to increase the order of the polynomial correction. Because of the repetitive character, it can be very well implemented in a (microcontroller) software routine. But it is also possible to implement the flow-diagram in hardware, digital as

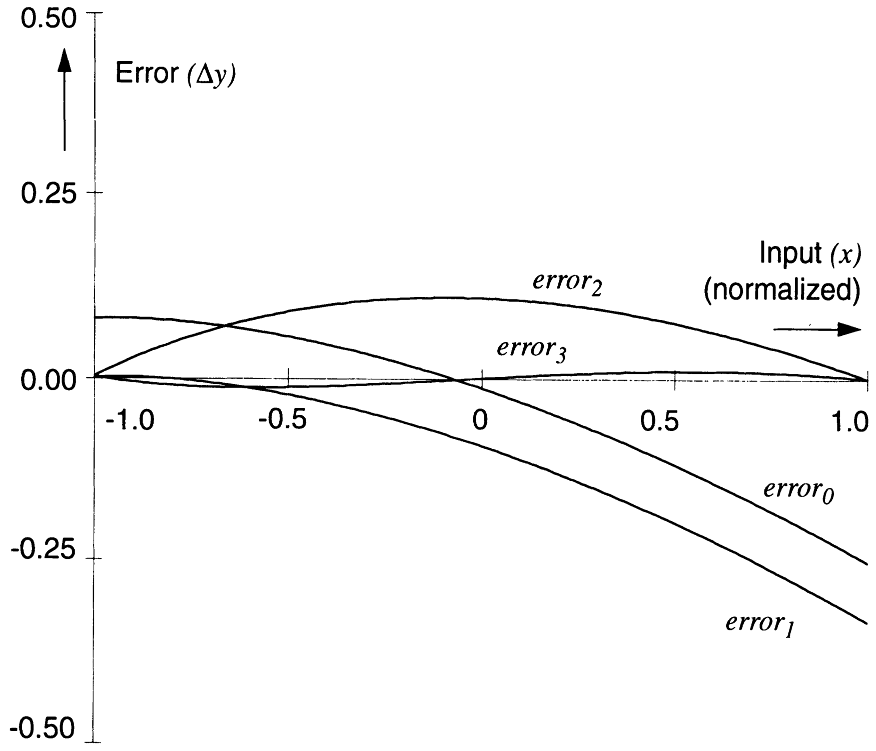

Fig. 3. Step-by-step reduction of the transfer error.

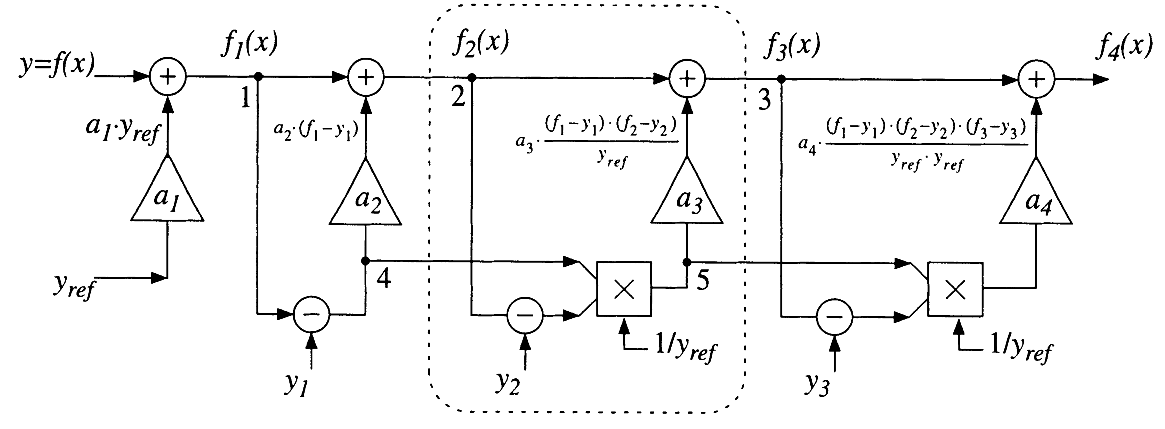

Fig. 4. Signal flow diagram of a 4-step polynomial correction.

well as analog. For example analog current signals can easily be added, subtracted (Kirchhoff) or multiplied (Gilbert [5]).

To demonstrate the feasibility of the hardware concept, the flow diagram of Fig. 4 has been implemented as a programmable analog circuit in a low-cost bipolar process. Fig. 5 shows the block diagram of the analog part, which, when carefully compared, shows great similarity with the diagram of Fig. 4. All signals are represented by differential currents and 'carried' by common-mode bias currents. Current signals $i_n(V_{IN})$ in Fig. 5 correspond to functions $f_n(x)$ in Fig. 4, and

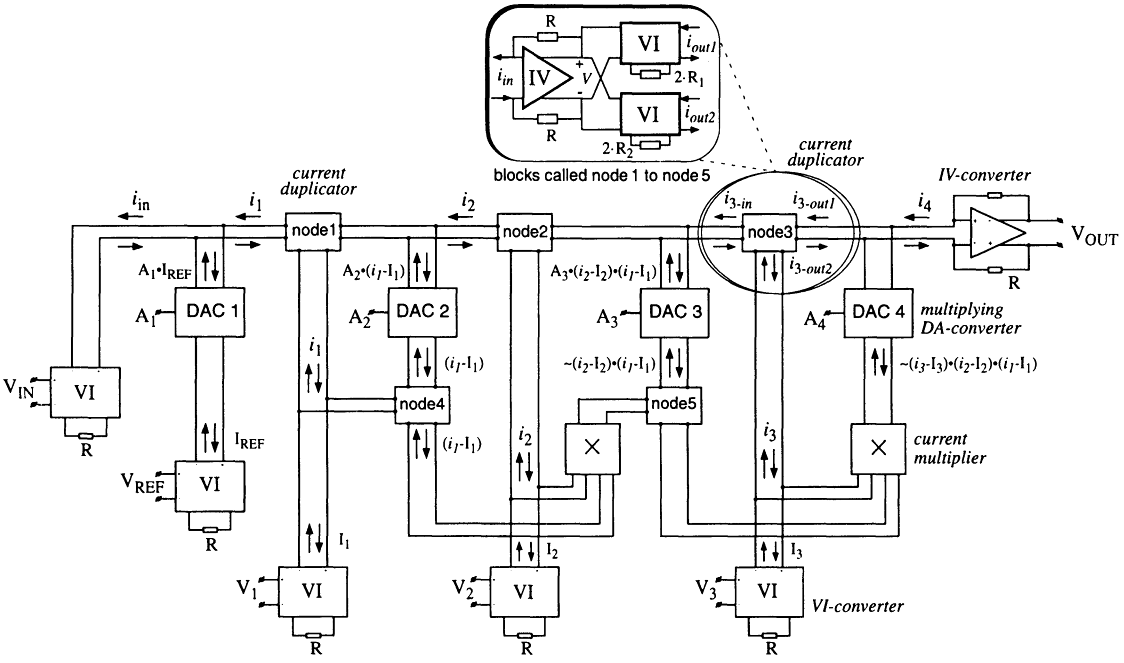

Fig. 5. Analog signal processor for 4-step polynomial calibration.

currents I_n correspond to references y_n. The input voltage V_{IN}, coming from the sensor, the voltage signal V_{ref}, and the calibration points V_1 to V_3 (desired values (constant) at calibration points) are all converted to differential current signals using VI-converters. At the output an IV-converter turns the processed current back into a differential voltage signal. The adding and subtracting is done by correctly connecting the two differential current signals. The multiplication of current signals is implemented in a Gilbert Multiplier, a well-known translinear circuit using only a few transistors [5]. The modified multiplier generates a differential output current which is proportional to the product of differential input currents divided by a reference bias current.

The circuit of Fig. 5 has a straightforward path through which the main signal is flowing: at the input the voltage signal is converted into a differential current which then goes through current duplicator blocks NODE 1, NODE 2 and NODE 3, and at the output is converted back into a differential voltage signal again, which is thus directly proportional to the input voltage. The output currents of the DACs, which can be programmed digitally, are added at several places in the forward path. When the calibration starts all **DACs** are programmed to give a zero output current. At the voltage inputs V_1, V_2 and V_3 reference signals have to be applied permanently which determine the "rotation points" of the calibration. A first reference signal is provided at the input V_{IN} of the circuit (or the connected sensor) and **DAC**1 is programmed in such a way that the output voltage V_{OUT} obtains the desired value for the first measurement. Then a second input reference can be provided and **DAC**2 can be programmed to obtain the desired output voltage for the second measurement, without the need to reprogram **DAC**1. Then a third calibration step can be done to program **DAC**3, and a fourth step for **DAC**4, just according to the calibration principle explained in the section above.

The duplications of the signals at the nodes numbered 1 to 5 in the flow-diagram of Fig. 4 are implemented in the current duplicator blocks NODE 1 to NODE 5. Inside each of these blocks an IV-converter turns the current into a voltage which is then used to generate two differential current signals using two VI-converters, as explained in Fig. 5. This allows a pre-scaling of the current signals so that the error-correction-range of the calibration can be set. Fig. 6 shows the differential current signals at the inputs of the DACs as a function of the input voltage

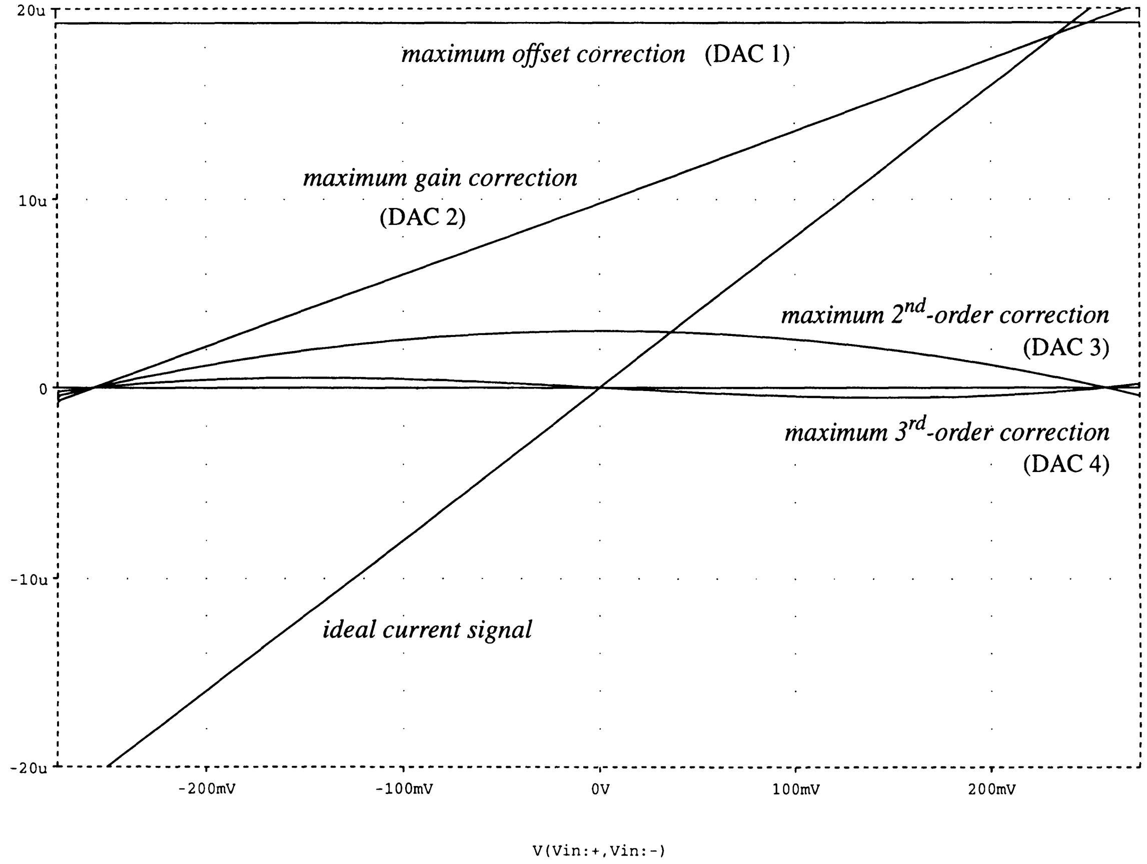

Fig. 6. Simulated differential current signals at the four DAC inputs as a function of V_{IN}.

V_{IN}, when all DACs are set to zero. The input range is set to [−250mV, +250mV], V_{ref} to 250mV, and the successive calibration points are set by voltages $V_1 = -250$mV, $V_2 = +250$mV and $V_3 = 0$mV. The corresponding current signal range is approximately [−20μA,+20μA]. The current signals in Fig. 6 represent the maximum correction signals that can be added to or subtracted from the forward signal path in Fig. 5 for respectively offset, gain and second- and third-order linearity calibrations. In this case, we scaled the currents in such a way that offset errors of maximum 95%, gain errors of up to almost 50%, and non-linearity errors of 15% of the full-scale output can be corrected.

The programmable calibration coefficients a_1 to a_4 are implemented as multiplying DACs [6]. As an example of the differential current-mode design technique, we show in Fig. 7 such an 8-bit DA-converter designed for differential current signals. The currents on the positive input I_{IN+} and on the negative input I_{IN-} are divided into 8 binary-scaled parts using cascode dividers [6]. Each binary part is switched to either the positive or the negative output using a differential pair. The circuit is configured in such a way that the differential input current $i = (I_{IN+} - I_{IN-})$ is multiplied with the 2's complement value of the 8-bit digital signal D[7-0] applied at the switches. The digital number can thus have positive as well as negative polarity.

The analog block diagram of Fig. 5 is combined with a digital circuit in low-power ECL-logic, so that the four digital calibration factors for the DACs can be programmed digitally through a serial interface and stored in on-chip 8-bit registers. Fig. 8 shows a photograph of the chip, which has been processed in a bipolar process (3GHz 40 × 50μm transistors) of the University institute **DIMES**, and which measures approximately 5 × 5mm^2. The functionality of the integrated circuit has been tested by applying constant voltage signals at the reference inputs so that

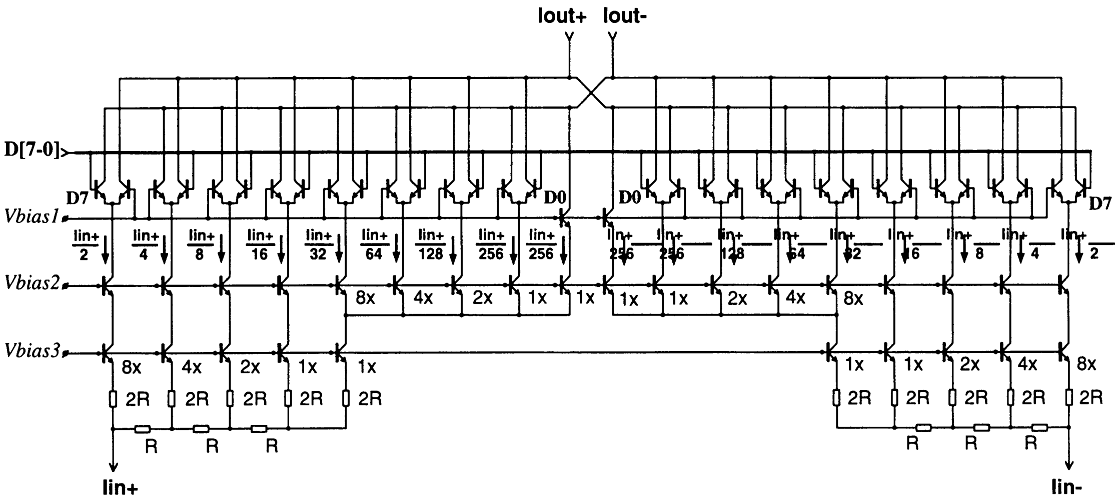

Fig. 7. Differential 8-bit DA-converter using cascode-dividers.

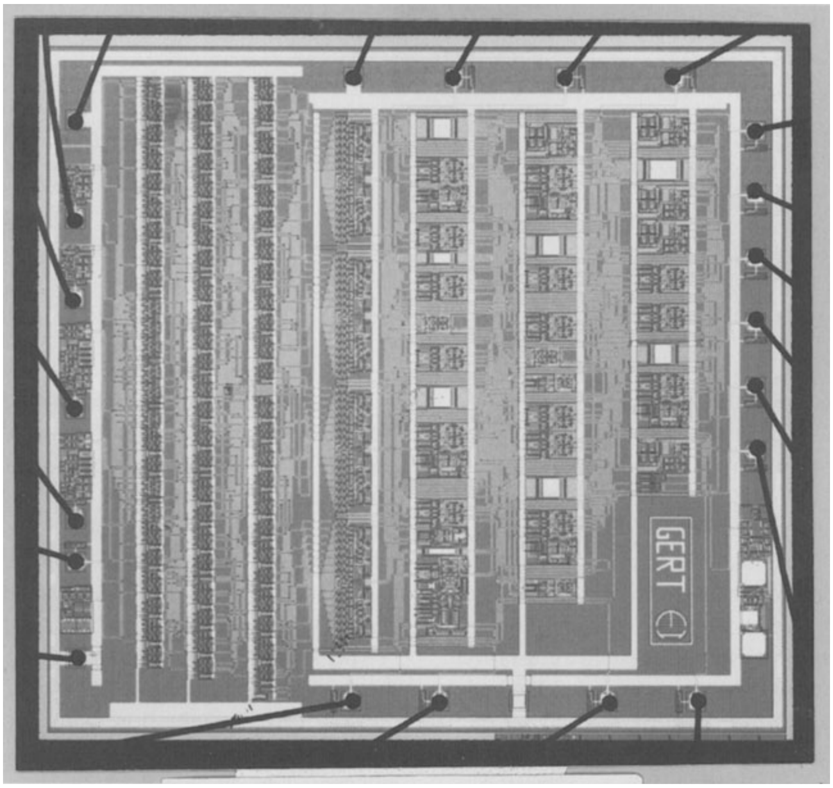

Fig. 8. Chip photograph of bipolar Integrated Circuit realization of the analog signal processor.

$V_{ref} = -V_1 = V_2 = +250\text{mV}$, $V_3 = 0\text{mV}$, and a triangle-wave signal at the input V_{IN}. The oscilloscope graph of Fig. 9 shows the input signal and several possible output signals, which have been generated by programming different calibration factors in the chip. It demonstrates that the offset, gain and linearity of

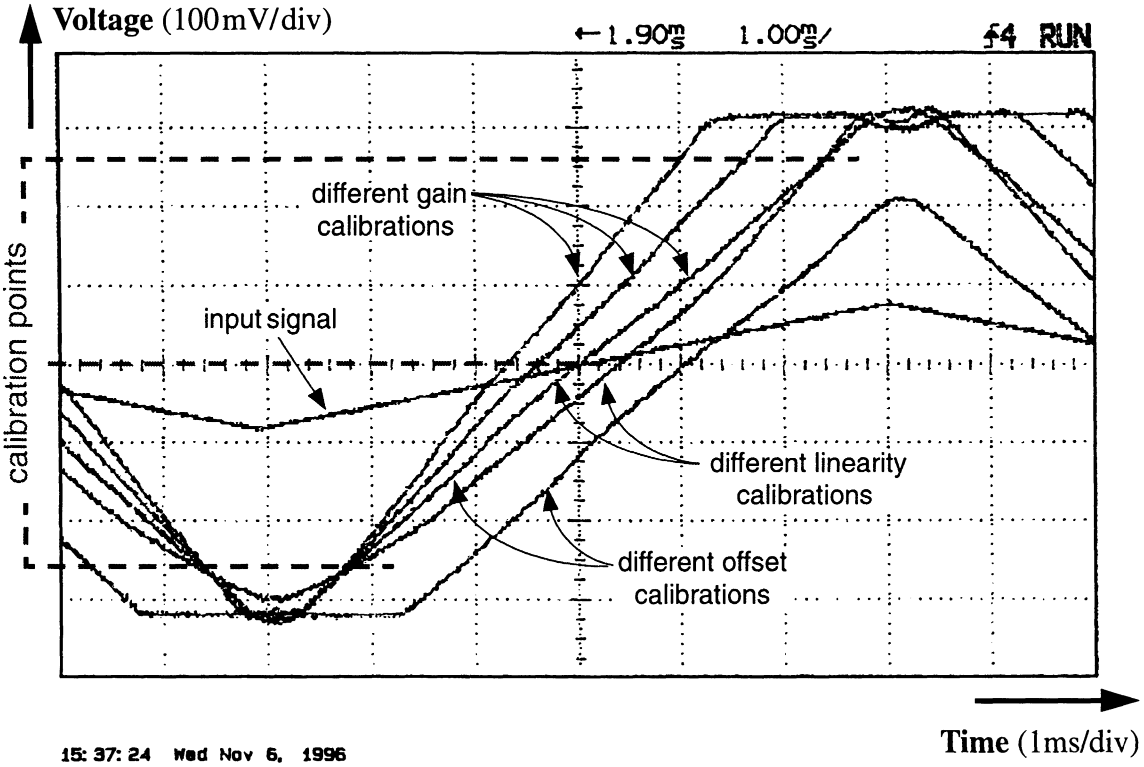

Fig. 9. Oscilloscope graph showing the triangular input signal and the output signal for several different calibration factors programmed in the calibration chip.

input-output transfer of the chip can be programmed, according to the presented calibration principle.

Because the analog signal goes through a cascade of local-feedback amplifiers (IV- and VI-converters) with relatively large bandwidth, the bandwidth of the complete circuit can be kept large as well. So, after the sensor transfer is calibrated using low-frequency or static reference signals, the correction also works for analog signals of high frequencies. Depending on the power invested in the (common-mode) bias currents of the different circuit blocks, bandwidths in the order of 100kHz to 1MHz can be achieved.

III. Two-Dimensional Polynomial Calibration Method

A. Calibration Principle

The output of for example a pressure sensor is determined not only by the pressure applied to the sensor but it is also affected to a certain extent by the operating temperature of the sensor, in such a way that the sensor errors, offset, gain and non-linearity, are temperature dependent [7], [8], [9], [10]. The sensor response in this case can be represented graphically by a surface in a 3-dimensional picture, see for example Fig. 10. Such a sensor has to be calibrated for both pressure and temperature, hence the term 2-dimensional calibration. The calibration method explained in section II.A, can be extended to handle 2-dimensional calibration [11]. The basic principle is to select axes in one dimension with fixed values for one input variable and then proceed along each selected axis with different values for the other variable as in the 1-dimensional calibration. The values along the first variable are also to be selected according to the same procedure as for the 1-dimensional calibration. The results are best (regarding speed and error reduction) when calibration is carried out in such a sequence that each error (offset, etc.) and its temperature coefficient are corrected before proceeding to correct for the next error (gain, etc.). Calibration should thus be carried out along axes with

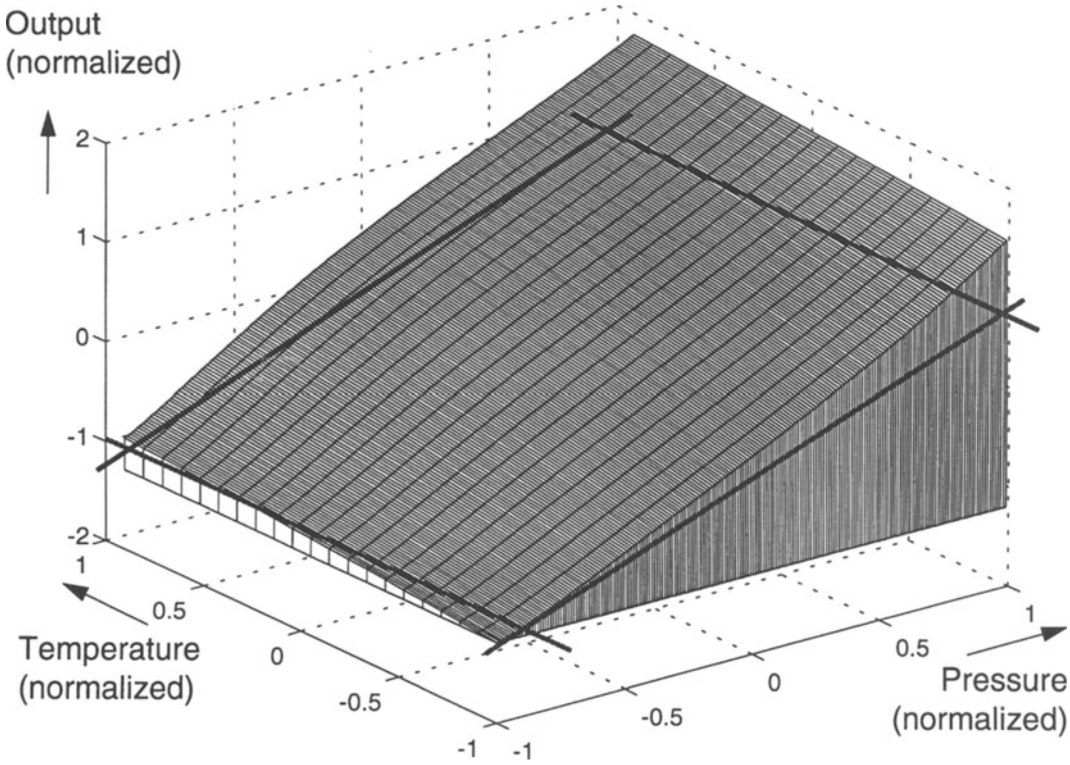

Fig. 10. Normalized transfer surface.

fixed well-known pressure values and proceeding along each selected pressure axis with different temperature values.

To explain this principle mathematically, we will use input variables p for pressure and T for temperature. We will denote the calibration points as (p_i, T_j), the original sensor transfer function by $f(p, T)$ and the calibration functions by $f_{ij}(p, T)$. For the case of an $m \times n$ calibration (m represents the number of pressure points and n the number of temperature points), we write:

- Calibration for offset and its temperature coefficient along the pressure axis $p = p_1$, taking n measurements at temperatures $T = T_1$ to $T = T_n$

$$\begin{aligned} f_{11}(p,T) &= f(p,T) + a_{11} \cdot y_{ref} \\ f_{12}(p,T) &= f_{11}(p,T) + a_{12} \cdot (T - T_1) \cdot y_{ref} \\ f_{13}(p,T) &= F_{12}(p,T) + a_{13} \cdot (T - T_1) \\ &\quad \cdot (T - T_2) \cdot y_{ref} \\ &\dots \\ f_{1n}(p,T) &= f_{1\,n-1}(p,T) + a_{1n} \cdot (T - T_1) \\ &\quad \cdot (T - T_2) \dots \cdot (T - T_{n-1}) \cdot y_{ref} \end{aligned} \tag{5}$$

At each calibration measurement at $T = T_j$ one calibration coefficient a_{1j} is adjusted to equalize the output $f_{1j}(p, T_j)$ to the desired output value y_1.

- Calibration for gain and its temperature coefficient along the pressure axis $p = p_2$, taking n measurements at temperatures $T = T_1$ to $T = T_n$

$$\begin{aligned} f_{21}(p,T) &= f_{1n}(p,T) + a_{21} \cdot \{f_{1n}(p,T) - y_1\} \\ f_{22}(p,T) &= f_{21}(p,T) + a_{22} \cdot (T - T_1) \\ &\quad \cdot \{f_{1n}(p,T) - y_1\} \\ f_{23}(p,T) &= f_{22}(p,T) + a_{23} \cdot (T - T_1) \\ &\quad \cdot (T - T_2) \cdot \{f_{1n}(p,T) - y_1\} \\ &\dots \end{aligned}$$

$$\begin{aligned} f_{2n}(p,T) &= f_{2n-1}(p,T) + a_{2n}\cdot(T-T_1) \\ &\quad \cdot(T-T_2)\ldots\cdot(T-T_{n-1}) \\ &\quad \cdot\{f_{1n}(p,T)-y_1\} \end{aligned} \tag{6}$$

At each calibration measurement at $T = T_j$ one calibration coefficient a_{2j} is adjusted to equalize the output $f_{2j}(p, T_j)$ to the desired output value y_2.

- Calibration for 2nd-order non-linearity and its temperature coefficient along the pressure axis $p = p_3$, taking n measurements at temperatures $T = T_1$ to $T = T_n$

$$\begin{aligned} f_{31}(p,T) &= f_{2n}(p,T) + a_{31}\cdot\{f_{1n}(p,T)-y_1\} \\ &\quad \cdot\{f_{2n}(p,T)-y_2\}/y_{ref} \\ f_{32}(p,T) &= f_{31}(p,T) + a_{32}\cdot(T-T_1) \\ &\quad \cdot\{f_{1n}(p,T)-y_1\} \\ &\quad \cdot\{f_{2n}(p,T)-y_2\}/y_{ref} \\ f_{33}(p,T) &= f_{32}(p,T) + a_{33}\cdot(T-T_1) \\ &\quad \cdot(T-T_2)\cdot\{f_{1n}(p,T)-y_1\} \\ &\quad \cdot\{f_{2n}(p,T)-y_2\}/y_{ref} \\ &\ \ \ldots \\ f_{3n}(p,T) &= f_{3n-1}(p,T) + a_{3n}\cdot(T-T_1) \\ &\quad \cdot(T-T_2)\ldots\cdot(T-T_{n-1}) \\ &\quad \cdot\{f_{1n}(p,T)-y_1\} \\ &\quad \cdot\{f_{2n}(p,T)-y_2\}/y_{ref} \end{aligned} \tag{7}$$

At each calibration measurement at $T = T_j$ one calibration coefficient a_{3j} is adjusted to equalize the output $f_{3j}(p, T_j)$ to the desired output value y_3.

This procedure can be continued to calibrate for higher-order non-linearities and their temperature coefficients, along the axes $p = p_4$, $p = p_5$ to $p = p_m$ until the final corrected output value $f_{mn}(p, T)$ is obtained. As is clear from formulas (5) ... (7), at each pressure calibration step a (1-dimensional) polynomial temperature function is build-up around the temperature calibration points T_1 to T_n, so that the calibration function is equal to the desired function in all temperature calibration points. At succeeding pressure calibration points a polynomial pressure function is build-up, which is equal to the desired output at all previous calibration points (p_1, T_1) to (p_i, T_j). Fig. 10 shows the uncalibrated transfer surface of a fictive pressure sensor, with normalized pressure on the x-axis, normalized temperature on the z-axis, and a normalized output on the y-axis. The ideal sensor would show a flat surface with a zero slope along the temperature-axis and a slope of 1 along the pressure axis, as indicated by the frame in Fig. 10. Fig. 11 shows the uncalibrated error surface of the sensor which represents the difference between the actual and the ideal sensor surface. The transfer surface and the error surface after a 5 × 3 calibration are shown in Fig. 12 and Fig. 13. The error surface clearly reveals the remaining third-order polynomial error curve along the temperature-axis, and the fifth-order polynomial error along the pressure-axis. It can be seen that the error value is zero at all 15 calibration points $(-1,-1)$, $(-\frac{1}{2},-1)$, $(0,-1)$, $(\frac{1}{2},-1)$, $(1,-1)\ldots(\frac{1}{2},1)$, $(1,1)$.

Many simulations have been done with a mathematical computer program to verify the calibration method, for different fictive sensor characteristics similar to the one in Fig. 10, and different numbers of calibration steps. In all cases, up to a 7 × 7 calibration, a steady reduction of the linearity error could be obtained. In most cases the best result was obtained when the first 2 calibration points are selected at opposite edges of the input-range, and further calibrations at points halfway in between two previous calibration points, as indicated in Fig. 13. To prevent escalation[1] of the polynomial factors in the formulas, it is important to first correct for the cross-sensitivity variable (temperature in this case), before linearizing the sensitivity for the input variable (pressure), as indicated in formulas (5) to (7). This gives a practical problem in the calibration measurements because it implicates that at each pressure point, the temperature is swept through the whole temperature range. However the temperature cycles require much more time than the pressure measurements, and it is thus advantageous to minimize the number of temperature sweeps. This can be done for the proposed calibration method, even though the calculation of the corrected output (after calibration) needs to be done in the same order as in equations (5) to (7), the calculation of the calibration coefficients a_{ij} (during calibration) can be done in a different order. When looking carefully at formula (5) one sees that for all reference temperatures $T = T_1, T = T_2, \ldots, T = T_n$ the calibration functions are equal, namely $f_{1n}(p, T_j) = f_{1n-1}(p, T_j) = \cdots = f_{12}(p, T_j) = f_{11}(p, T_j)$, and from formula (6) we can derive the same for $f_{2j}(p, T_j)$ and from formula (7) for $f_{3j}(p, T_j)$, etc. This means that the calibration coefficients can be calculated in a different order, namely: at temperature T_1, we can first apply p_1 to calculate a_{11}, then apply p_2 to calculate a_{21}, then apply p_3 to calcu-

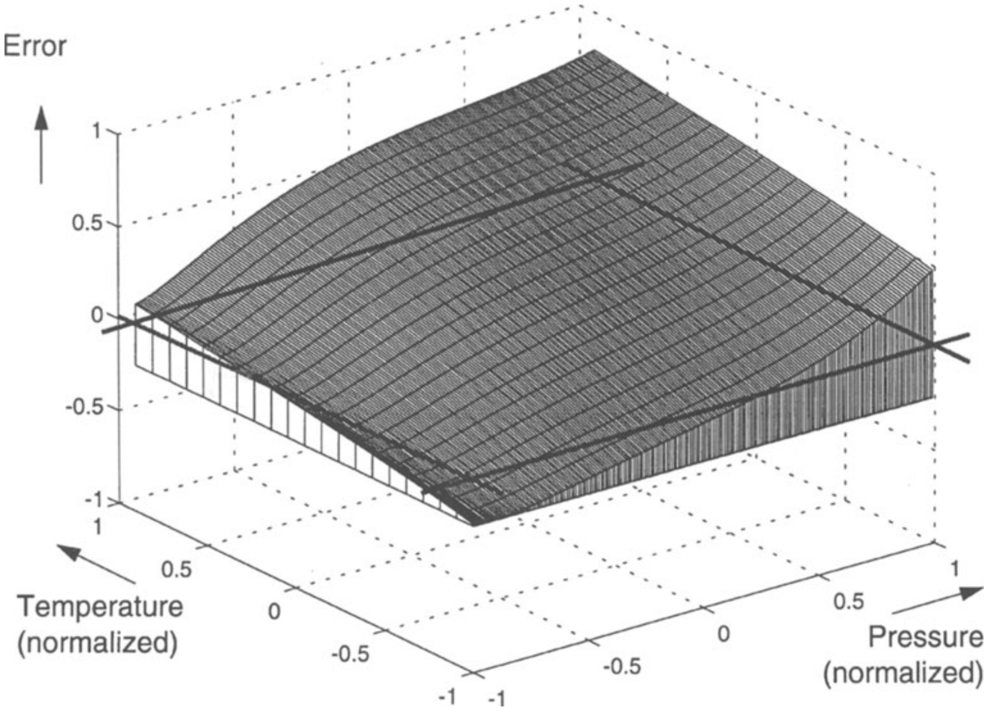

Fig. 11. Uncalibrated error surface.

late a_{31}, etc.; at the next temperature T_2 we apply p_1 to calculate a_{12}, then apply p_2 to calculate a_{22}, knowing that $f_{1n}(p, T_2) = f_{11}(p, T_2)$, then apply p_3 to calculate a_{32}, knowing that $f_{2n}(p, T_2) = f_{21}(p, T_2)$, etc. This minimizes the number of temperature changes needed during (batch) calibration, and therewith the required calibration time.

B. Implementation: Microcontroller-Based Pressure Sensor System with Digital Calibration

Fig. 14 shows the block diagram of the proposed smart sensor system using a smart sensor interface chip with **IS**^{2} bus interface [12] to read-out the integrated PTAT temperature sensor and the external pressure sensor. The output of the pressure sensor is determined by the applied pressure and the operating temperature of the sensor, therefore a temperature sensor is needed to give the calibration program an indication of the operating temperature. The temperature sensor should of-course be near to the pressure sensor, which can be achieved by including the interface chip with temperature sensor in the same package or instrument as the pressure bridge. A multiplexer controlled by the bus interface makes it possible to readout the temperature sensor and the pressure sensor using only one sigma-delta AD-converter [13], [14]. The calibration program and the calibration coefficients, as well as a bus-interface program are stored in the memory of the microcontroller. It is possible to integrate all the components of Fig. 14, including the microcontroller, in one multi-chip module or package, but it is also possible to keep the microcontroller external and use its computation power also for other sensors connected to the same smart sensor bus [12].

The microcontroller we used for evaluation of the calibration method is the 80c552 which is a cheap 8-bit microcontroller driven by a 12MHz clock. The system software consists of a program written in C, which is compiled, assembled and then downloaded into the microcontroller memory. The program consists of two parts: the *calibration part* to calculate the calibration coefficients when applying reference signals to the sensors and measuring the output signals, and the *measurement part* to calculate the corrected values of the pressure output, according to equations (5) to (7), using

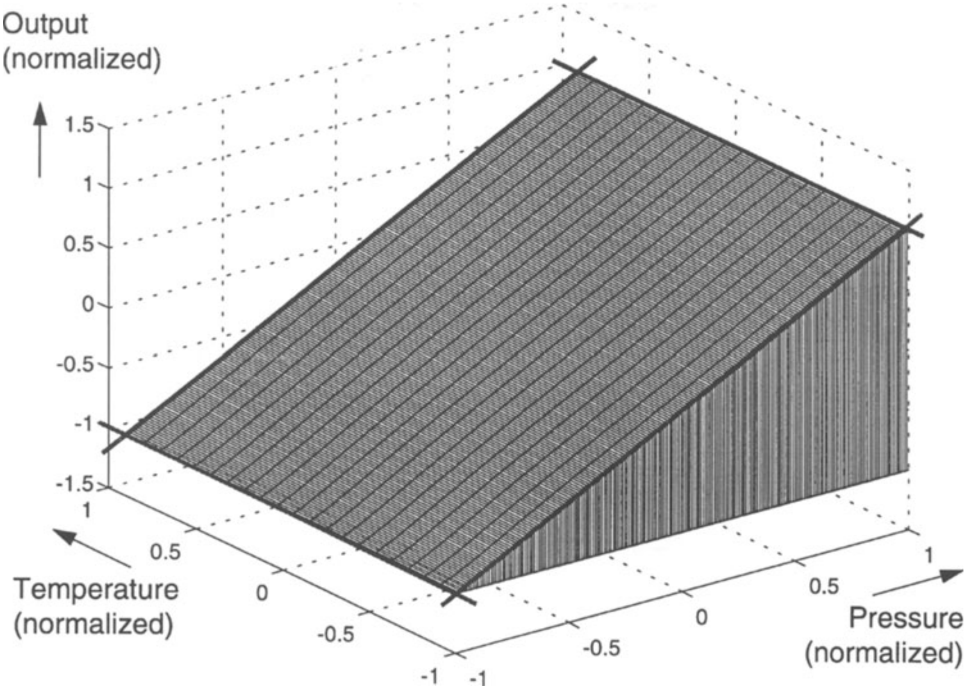

Fig. 12. Normalized transfer surface, after a 5 × 3 calibration.

the sensor outputs and the values of calibration coefficients already calculated during the calibration cycle. Currently, it is an interactive program which, during the calibration process inquires the user about the reference data (desired output signal) required at each stage of the calibration process. Then the microcontroller proceeds with the necessary calculations, which involves calculating a new calibration coefficient and checking the error limit supplied by the user. Once this limit has been reached, entering more calibration data can stop, and the measurement cycle can start. Once the optimal number of calibration points and the reference signals have been chosen, the system can be made autonomous by storing the program and the reference data in the microcontroller PROM and using hardware interrupts to indicate that certain reference signals (pressure and temperature) have been applied for calibration. During batch calibration, many of such pressure sensor systems can then be connected to the same interrupt signals, and each system independently calculates and stores its own calibration coefficients. After the calibration calculations, the actual pressure value measured by the pressure sensor can easily be calculated by the microcontroller by applying the correct multiplication factor, depending on the desired pressure unit (Pascal, bar, etc.).

The microcontroller program was operated and tested using the fictive sensor data already used in the simulations. Actual results from the microcontroller program were found to coincide with the simulation results. With the microcontroller type already mentioned, and 32-bit precision floating point arithmetic, it was found that the time required to calculate one corrected value during the measurement cycle was approximately 30ms based on 5 × 5 calibration steps. This is fast enough for sensors with a signal bandwidth lower than 15Hz. Of course speed and accuracy can be exchanged. To increase speed one could for example reduce calibration accuracy to 5 × 3 or 3 × 3 steps, chose a lower arithmetical resolution, or chose a faster 16-bit (more expensive) microcontroller. Maximum speed can then be in the order of 100–1000 samples per second.

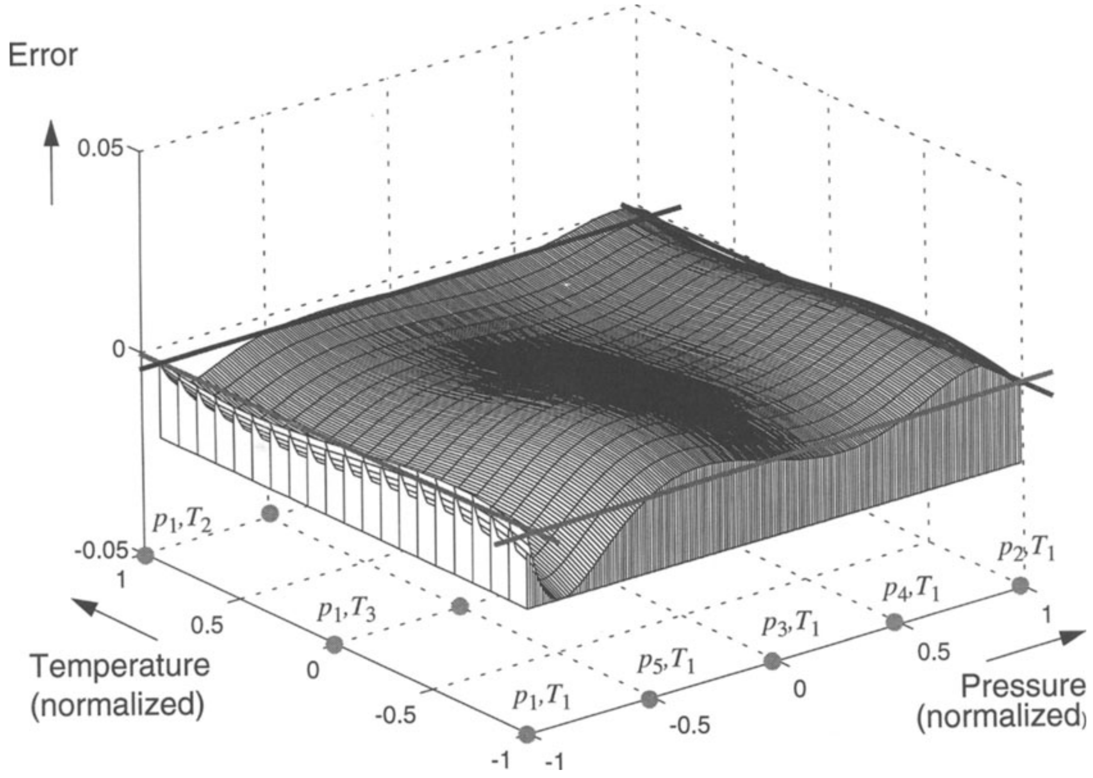

Fig. 13. Error surface, after a 5 × 3calibration.

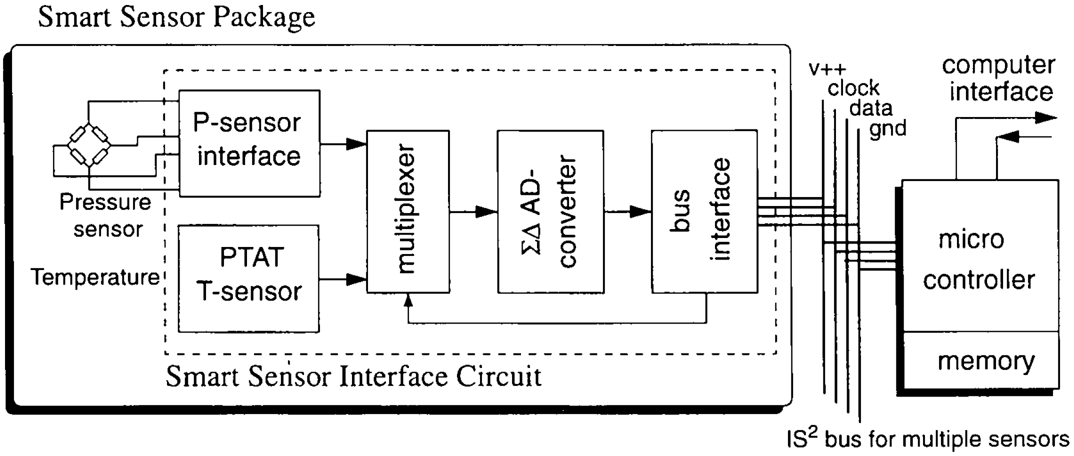

Fig. 14. Pressure sensor micro-system, consisting of a sensor, an interface IC, and a microcontroller.

IV. Conclusion

A new calibration method has been presented, which is based on building up a polynomial correction function for the sensor transfer, in a step-by-step approach by calculating an additional correction term at each calibration measurement. It does not have the disadvantages of methods based on lookup tables, such as the need for a large memory and a large amount of calibration data [14], [15], [16]. Compared to other function correction methods, it does not have the disadvantage of collecting a matrix of data first and then inverting the matrix or use an iterative method to obtain the polynomial correction factors [15], [17], [18], which needs to be completely redone when the number of calibration points needs to be increased. Instead, the proposed method uses each calibration measurement to obtain one specific correction factor, at the first calibration to correct the offset, at the second to correct gain, at the third to correct second-order non-linearity, etcetera. Each calibration results in an additional correction term which is constructed in such a way that all previous calibrations remain undisturbed. It has been shown that the calibration approach can also be used for a 2-dimensional calibration for sensors that suffer from cross-sensitivity to another parameter (often temperature), which is the case for a strain-gauge pressure sensor.

We showed that the construction of each correction factor is in fact the same (see Fig. 4). Thus the calculation of the error correction (after calibration) is a repetitive procedure, which can be implemented in software as a small subroutine, or in hardware as a cascade, pipeline or a loop of the same "building block". As examples, an implementation of the calibration method in a programmable analog circuit and a microcontroller-based calibration for a pressure sensor microsystem have been presented. The software implementation offers advantages in flexibility and in price (because of mass-production, microcontrollers are available at low-price). The hardware implementation offers the possibility to optimize performance with respect to signal processing speed (esp. analog implementations) or with respect to resolution, but has the disadvantage of requiring the development of an ASIC. This will be less of a disadvantage, when the calibration facility is combined with other sensor readout electronics as proposed in the smart sensor concept [2] or, even better, with a general sensor readout interface circuit [19]. The integration of a smart calibration facility on the sensor will make it possible to produce calibrated, thus more-accurate and easy-to-use sensors at a lower price.

Acknowledgements

The authors like to thank the people of DIMES, especially RuudKlerks and Wimv.d.Vlist, for processing and packaging the ICs, and the Dutch Technology Foundation STW for the financial support of this research project.

Notes

[1] In the situation described in equations (5) and (6) all the terms containing temperature polynomials are used to "flatten" the temperature-sensitivity of each function $\underline{f_{mn}}$ to zero. Therefore the remaining temperature error in the correction terms in equation (7) containing a product of functions $\underline{f_{1n}}$ and $\underline{f_{2n}}$ will still be small. Whereas in the case that pressure errors are calibrated first, the correction terms contain products of the functions $\underline{f_{mn}}$ which not yet have a "flattened" temperature-dependence and thus result in large multiplicative temperature errors to be corrected for.

References

1. S. Middelhoek and S. Audet, *Silicon Sensors*. Academic Press: London, 1989.
2. J. H. Huijsing, F. R. Riedijk and G. v. d. Horn, "Developments in integrated smart sensors," *Sensors and Actuators A* 43, 1994, special issue on Transducers'93.
3. J. E. Brignell, "Digital compensation of sensors," *Journal of Physics* E: Scientific Instrumentation, pp. 1097–1102, 1987.
4. J. Bryzek et al., *Silicon Sensors and Microstructures*. Nova Sensor, 1990.
5. B. Gilbert, "A precision four-quadrant multiplier with subnanosecond response," *IEEE Journal of Solid-State Circuits* SC-3, pp. 365–373, Dec. 1968.
6. R. v. d. Plassche, *Integrated Analog-to-Digital and Digital-to-Analog Converters*. Kluwer Academic Publishers: Boston/Dordrecht/London, 1994.
7. S. Ansermet et al., "Cooperative development of a Piezoresistive pressure sensor with integrated signal conditioning for automotive and industrial applications," *Sensors and Actuators A*, 21–23, pp.79–83, 1990.
8. S. B. Crary, W. G. Baer, J. C. Cowles and K. D. Wise, "Digital compensation of high performance silicon pressure transducers," *Sensors and Actuators A* 21–23, pp. 70–72, 1990.
9. J. Gakkestad et al., "A front end CMOS circuit for a full bridge Piezoresistive pressure sensor," *Sensors and Actuators A*, 25–27, pp. 859–863, 1991.
10. F. V. Schnatz et al., "Smart CMOS capacitive pressure transducer with on-chip calibration capability," *Sensors and Actuators A* 34, pp. 77–83, 1992.
11. K. F. Lyahou, G. v. d. Horn and J. H. Huijsing, "A non-iterative polynomial 2-dimensional calibration method implemented in a microcontroller," *Proceedings IMTC'96*, pp. 62–67, June 1996.
12. F. R. Riedijk and J. H. Huijsing, "Sensor interface environment based on sigma-delta conversion and serial bus interface," *SENSORS, Journal of Applied Sensing Technology*, Sensors Expo issue, April 1996.
13. J. C. Candy and G. C. Temes, *Oversampled Delta-Sigma Data Converters*. IEEE Press: New York, 1992.
14. P. Malcovati, C. A. Leme, P. O'Leary, F. Maloberti and H. Baltes, "Smart sensor interface with A/D conversion and programmable calibration," *IEEE Journal of Solid State Circuits* 29(8), pp. 963–966, August 1994.
15. P. N. Mahana and F. N. Trofimenkoff, "Transducer output signal processing using an eight-bit microcomputer," *IEEE Transactions on Instrumentation & Measurement* IM-35(2), pp. 182–186, June 1986.

16. J. E. Brignell, "Software techniques for sensor compensation," *Sensors and Actuators A* 25–27, pp. 29–35, 1991.
17. W. T. Bolk, "A general digital linearising method for transducers," *Journal of Physics* E: Scientific Instrumentation, pp. 61–64, 1985.
18. D. Patranabis and D. Gosh, "A novel software-based transducer linearizer," *IEEE Transactions on Instrumentation and Measurement* 36(6), December 1989.
19. F. M. L. vanderGoes and G. C. M. Meijer, "A novel low-cost and accurate multi-purpose sensor interface with continuous auto-calibration," *Proceedings IMTC'96*, pp. 782–786, June 1996.

Gert van der Horn was born on July 4, 1968 in Ede, The Netherlands. He studied electrical engineering at the Delft University of Technology in The Netherlands. At the Electronic Instrumentation Laboratory he worked on input stages with common-mode extension and on low-voltage comparator design, before he received the M.Sc. degree in 1991. He received the ESSCIRC'92 Best Paper Award for a presentation on input stages with common-mode extension beyond-the-rail. He is now researching smart sensor calibration techniques and working towards the Ph.D. degree. His research interests include measurement and instrumentation systems, analog signal conditioning, AD conversion and amplifier design.

Johan H. Huijsing was born in Bandung, Indonesia, on May 21, 1938. He received his M.Sc. in Electrical Engineering from the Delft University of Technology, Delft, the Netherlands, in 1969, and his Ph.D. from this University in 1981 for work on operational amplifiers (thesis: "Integrated Circuits for Accurate Linear Analogue Electric Signal Processing", supervised by Prof. Dr. Ir. J. Davidse). Since 1969 he has been a member of the Research and Teaching Staff of the Electronic Instrumentation Laboratory, Department of Electrical Engineering, Delft University of Technology, where he is now Professor of Electronic Instrumentation. He teaches courses on Electrical Measurement Techniques, Electronic Instrumentation, Operational Amplifiers and Analog-to-Digital Converters. His field of research is Analog Circuit Design (operational amplifiers, analog multipliers, etc.) and Integrated Smart Sensors (signal conditioning on the sensor chip, frequency and digital converters which incorporate sensors, bus interfaces, etc.). He is the author or co-author of some 100 scientific papers and has filed 15 patents.