A Noniterative Polynomial 2-D Calibration Method Implemented in a Microcontroller

Khalid Fouad Lyahou, Gert van der Horn, and Johan H. Huijsing, *Fellow, IEEE*

***Abstract*— Output signal handling of a smart sensor usually involves a calibration facility to correct for input–output nonidealities which comprise offset, gain errors, nonlinearity errors, and cross sensitivity. In this paper, a calibration method is presented which features a progressive improvement in the sensor calibrated transfer toward the desired transfer as the user proceeds from one calibration point to the next. The method is based on a set of mathematical formulas whereby a calibration coefficient is calculated at a selected calibration point and used to calculate a first correction of the sensor transfer curve. Further improvement in the sensor transfer is obtained by repeating the process for a second calibration point using the transfer resulting from the first calibration, without the need to review the calibration already carried out at the first point. The process can be repeated until the desired error reduction is obtained. Step by step, the calibration method builds up a polynomial transfer correction. Simulation results are shown to demonstrate the performance of the method. A software implementation of the method for two-dimensional (2-D) calibration in an 8-bit microcontroller is presented. The microcontroller with calibration program can be incorporated in a compact smart sensor system.**

***Index Terms*—Calibration, linearization, microcontroller-based sensor systems, microsystems, pressure sensors, smart sensors.**

I. INTRODUCTION

THE MAIN requirement of an electronic sensor is to generate an electrical signal proportional to the input physical quantity. However, a practical electronic sensor does not realize the ideal signal transfer, but exhibits nonidealities like offset, gain, and nonlinearity, in addition to the possible effects of a disturbing variable which affects the desired sensor response to the physical quantity of interest. For example, the offset, gain, and nonlinearity of a pressure sensor can be dependent on the operating temperature of the sensor, in which case the sensor is said to be cross sensitive to temperature. Thus, the output of a sensor cannot be interpreted as an accurate and reliable representation of the physical quantity to be measured and certain measures must be taken to calibrate the sensor using reliable reference input signals, such that its output can be accurately related to the physical quantity. For smart sensors, calibration methods can be applied to the analog part or the digital part of the sensor signal-processing chain [1]. Analog calibration methods are based on changing the analog signal transfer in such a way as to compensate for the sensor errors, mostly through the use of trimmable resistors [2]–[4]. A second class of calibration methods within this group [5] comprises those methods where the transfer of the analog signal processing circuit is controlled and modified by a digitally programmable array of resistors or capacitors. However, higher resolution can only be achieved by increasing the number of controlling bits and the array size of the trimming elements. The disadvantages of these analog methods are circuit complexity, limited resolution, and lack of flexibility. Moreover, they have limited capabilities to handle the nonlinearity and cross sensitivity problems, and each sensor requires a lot of individual attention for the calibration, making it harder to automate the calibration process.

The digital calibration methods offer many possibilities for flexible, accurate, and programmable calibration methods. Moreover, the availability of cheap microcontrollers and smart sensors with digital, microcontroller-compatible outputs, makes integrated calibration of the smart sensor device possible. These digital calibration methods can be implemented using either a lookup table method or a calibration function method. A disadvantage of the former method is the large storage capacity needed to accommodate the large number of correction points necessary to obtain adequate calibration, and the large amount of sensor data that has to be gathered during the calibration process. Reducing the storage space at the expense of speed and accuracy can be effected by using a coarse table and an interpolation method between the points stored, but this requires additional dedicated circuitry or software to calculate the interpolated points [6], [7]. The space requirement is increased further if correction for cross sensitivity to an interfering parameter is also required. In the second class of calibration methods, a small matrix of calibration data is obtained during the calibration process which can be used to calculate the calibration coefficients to be used later in a polynomial correction function [8]–[10]. The calculation of the calibration coefficients involves extensive computations usually carried out on a PC or external computer. If the user or the calibration system decides for more calibration points to further reduce the error, then the calculations of the calibration coefficients have to be repeated all over again for the new number of calibration points.

This paper proposes a point by point calibration method, and presents the implementation of the latter in a microcontroller program. At a given calibration point (reference signal) the actual sensor output is equalized to the desired sensor output, by calculating and applying a correction coefficient according to a simple formula. This can be considered an offset calibration. The procedure is repeated at another calibration point,

Manuscript received June 3, 1996.

The authors are with Electronic Instrumentation Laboratory/DIMES, Technical University Delft, 2628 CD Delft, The Netherlands (e-mail: horn@ei.et.tudelft.nl).

Publisher Item Identifier S 0018-9456(97)06501-7.

where the actual sensor output is again directly equalized to the desired output, without disturbing the previous equalization. This results in a gain calibration, while the offset calibration is preserved. The calibration can be repeated for any number of reference signals, building up, step by step, a polynomial correction curve, which can be applied to the sensor signal immediately. After four calibration points, for example, a third-order polynomial correction is achieved. Each correction is applied in such a way that the previous calibrations are not disturbed.

The method does not require the collection of a matrix of sensor data and then using it to calculate the calibration coefficients through matrix inversion. Instead, the calculation of each calibration coefficient involves the solution of one equation with one unknown only, plus the calculation of the resulting corrected sensor response. Each time, a newly calculated calibration coefficient is used to add a new term to the calibration function already at hand from previous calibrations, which then forms a new calibration function resulting in further correction of the sensor response. After each calibration step, the sensor response and error can be examined at a new calibration point, and if the error reduction is not yet satisfactory, further corrections can be effected. Moreover, the method can be readily extended for 2-D calibration to calibrate for cross sensitivity to another variable, which can be measured separately (such as temperature).

In Section II, the mathematics of the one-dimensional (1-D) point-by-point calibration will be explained. Then, the two-dimensional (2-D) calibration will be discussed, followed by the implementation in a microcontroller for the calibration of pressure sensors (esp. piezoresistor-bridges).

II. 1-D Calibration

In this section, the principle of the proposed method as applied to 1-D calibration will be discussed. In this case, the sensor is considered to be sensitive only to the input desired variable, hence the term 1-D calibration.

The method is based on a number of calibration equations where each equation is built up from the previous one and with each new equation involving one new calibration coefficient. The input variable is indicated as x, the uncalibrated sensor response is denoted by $f_0(x)$, the desired value of the sensor response at the nth calibration point by p_n, the calibration coefficients by a_n, and the calibration functions by $f_n(x)$. The first calibration function $f_1(x)$ is given by the simple formula

$$f_1(x) = f_0(x) + a_1.$$

From the first measurement $f_0(x_1)$ the desired value of a_1 can be calculated, to equalize the corrected sensor output $f_1(x_1)$ to the desired output value p_1 at input x_1.

The second calibration function $f_s(x)$ uses the previously corrected function $f_1(x)$ as

$$f_2(x) = f_1(x) + a_2 \times \{f_1(x) - p_1\}.$$

The correction term with calibration coefficient a_2 is zero in the first calibration point (x_1, p_1). A second calibration measurement $f_1(x_2)$ is used to calculate the correct value for a_2, so that the corrected sensor output $f_2(x_2)$ at input x_2 is now equalized to the desired output p_2. Also the previous calibration $f_2(x_1) = p_1$ is still valid.

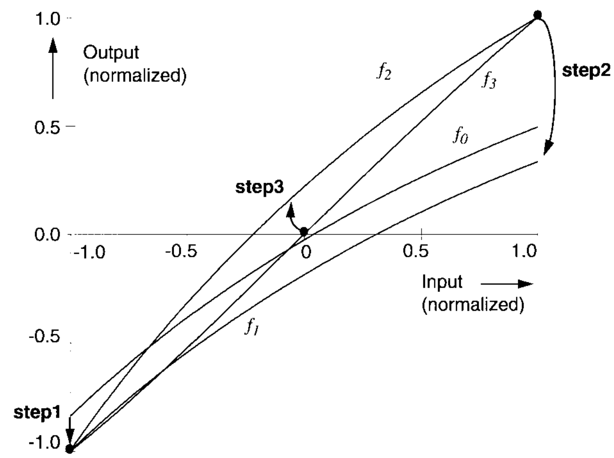


Fig. 1. Transfer curves before and after calibration.

The following calibration steps will build up a polynomial linearization function in the same successive manner. The third, fourth, and fifth calibration functions are, respectively, given as

$$f_3(x) = f_2(x) + a_3 \times \{f_1(x) - p_1\} \times \{f_2(x) - p_2\}$$

$$f_4(x) = f_3(x) + a_4 \times \{f_1(x) - p_1\} \times\{f_2(x) - p_2\} \times \{f_3(x) - p_3\}$$

$$f_5(x) = f_4(x) + a_4 \times \{f_1(x) - p_1\} \times\{f_2(x) - p_2\} \times \{f_3(x) - p_3\} \times \{f_4(x) - p_4\}.$$

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 three points (x_1, p_1), (x_2, P_2), and (x_3, p_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, p_4). A fifth measurement $f_4(x_5)$ is used to calculate a_5 which linearizes $f_5(x)$ at a fifth point (x_5, p_5). The process can be continued until the error, the difference between the desired and the calibrated transfer function, is sufficiently reduced.

To demonstrate how the method works to successively reduce the errors in sensor response, starting with the gain and offset errors and proceeding to the nonlinearity errors, reference is made to Figs. 1–3. Fig. 1 shows the uncalibrated sensor response together with the resulting responses after 1, 2, and 3 calibration steps for the case where the sensor input and output are normalized to $\{-1, 1\}$. Fig. 2 shows the initial uncalibrated error and the error values after 1, 2, and 3 calibration steps. For clarity, the error values after 3, 4, and 5 calibration steps are shown separately on a different scale in Fig. 3. In those figures, f_0 and $error_0$ refer to the uncalibrated sensor response and the uncalibrated error, while f_n and $error_n$ refer to the response and error after n calibration steps.

Simulation results involving other function calibration methods in the literature namely the three-point method in [10], and the polynomial fitting method in [8], showed that the

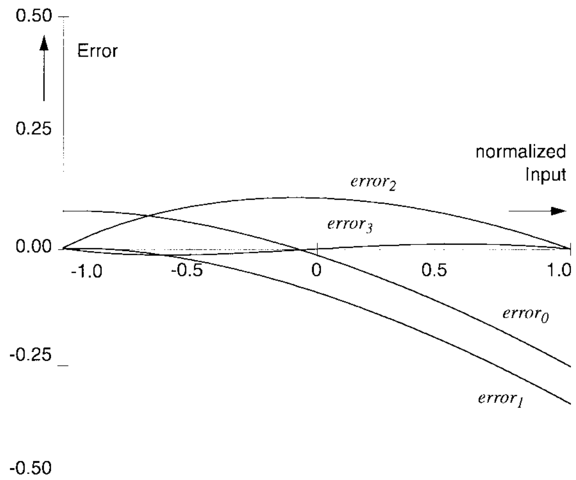


Fig. 2. Error curves before and after calibration.

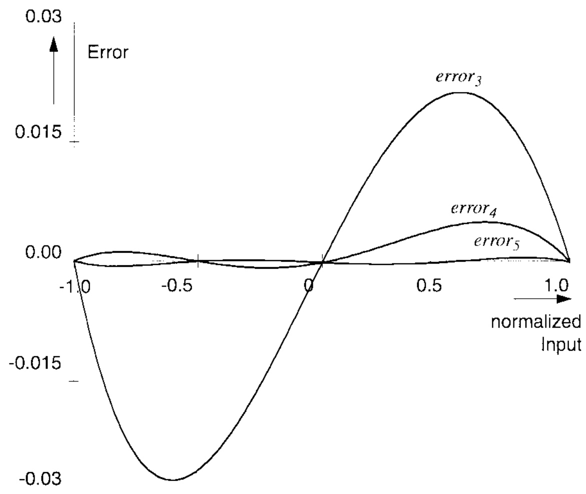


Fig. 3. Error curves for three successive calibration steps.

performance of this method is good: the error reduction per calibration step is comparable, and for certain sensor curves, better than those methods.

III. 2-D Calibration

The output of a pressure sensor is not only determined 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 themselves are temperature dependent. Then, such a pressure sensor has to be calibrated for both pressure (desired variable) and temperature (undesired interfering variable), hence the term 2-D calibration. The sensor response in this case can be represented graphically by a surface in a three-dimensional (3-D) picture. The calibration method can be extended to handle 2-D calibration by succeedingly selecting axes with fixed values for one input variable and then proceeding along each selected axis with different values for the other variable as in the 1-D calibration. A separate temperature sensor is of course needed to calculate and apply a temperature dependent correction on the pressure sensor output. The values along the first variable are also to be selected following the same order as that for 1-D calibration. The results are best (as regards speedy and steady error reduction) when calibration is carried out in such a sequence that each error (such as offset) and its temperature coefficient are corrected before proceeding to correct for the next error (in this case, gain error) and its temperature coefficient. Calibration should, thus be carried out along axes representing pressure values (the desired variable) and proceeding along each selected pressure axis with different temperature values. Denoting the measured temperature value by T, and the reference temperature at the calibration point (P_i, T_j) by T_j, the 2-D calibration functions $f_{ij}(p, T)$ for the case of m by n calibration points (m represents the number of pressure points and n represent the number of temperature points), are given below.

- Calibration for offset and its temperature coefficient along the pressure axis (p_1)

$$
\begin{aligned}
f_{11} &= f_0 + a_{11}\\
f_{12} &= f_{11} + a_{12}(T - T_1)\\
f_{13} &= f_{12} + a_{13}(T - T_1)(T - T_2)\\
&\cdots\\
&\cdots\\
f_{1n} &= f_{1(n-1)} + a_{1n}(T - T_1)(T - T_2)\cdots(T - T_{n-1}).
\end{aligned}
$$

- Calibration for gain and its temperature coefficient along the pressure axis (p_2)

$$
\begin{aligned}
f_{21} &= f_{1n} + a_{21}(f_{1n} - p_1)\\
f_{22} &= f_{21} + a_{22}(f_{1n} - p_1)(T - T_1)\\
f_{23} &= f_{22} + a_{23}(f_{1n} - p_1)(T - T_1)(T - T_2)\\
&\cdots\\
&\cdots\\
f_{2n} &= f_{2(n-1)} + a_{2n}(f_{1n} - p_1)(T - T_1)(T - T_2)\\
&\quad \cdots(T - T_{n-1}).
\end{aligned}
$$

- Calibration for second-order nonlinearity and its temperature coefficient along the pressure axis (p_3)

$$
\begin{aligned}
f_{31} &= f_{2n} + a_{31}(f_{1n} - p_1)(f_{2n} - p_2)\\
f_{32} &= f_{31} + a_{32}(f_{1n} - p_1)(f_{2n} - p_2)(T - T_1)\\
f_{33} &= f_{32} + a_{33}(f_{1n} - p_1)(f_{2n} - p_2)(T - T_1)(T - T_2)\\
&\cdots\\
&\cdots\\
f_{3n} &= f_{3(n-1)} + a_{3n}(f_{1n} - p_1)(f_{2n} - p_2)\\
&\quad \times (T - T_1)(T - T_2)\cdots(T - T_{n-1}).
\end{aligned}
$$

Calibration for higher order nonlinearities and their temperature coefficients can be further carried out along the axes $(p_4, p_5, \ldots, p_m)$, following the same procedure, where a_{ij} is the calibration coefficient to be determined at the calibration point P_i, T_j, f_{ij} is the ith calibration function at the jth temperature value, and p_i is the desired output value for the function f_{ij} at the calibration point (P_i, T_j).

To illustrate how the method works, Fig. 4 shows the uncalibrated transfer surface and Fig. 5 the uncalibrated error surface

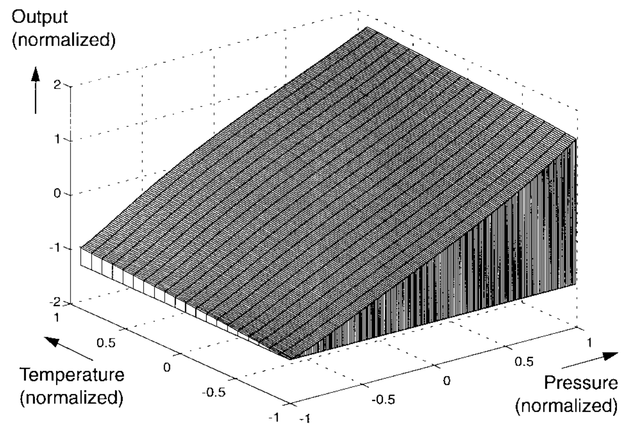


Fig. 4. Transfer surface before calibration.

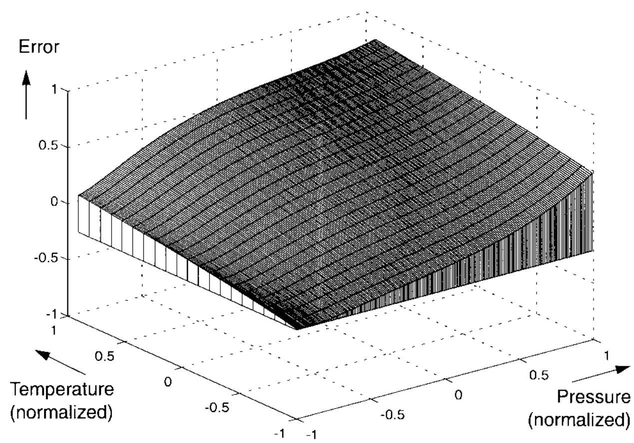


Fig. 5. Error surface before calibration.

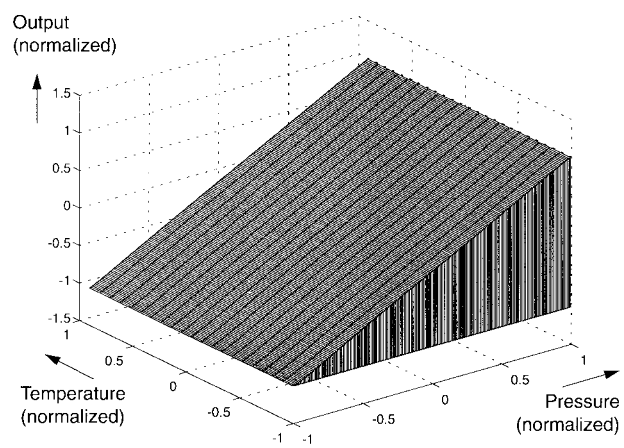


Fig. 6. Transfer surface after 5 × 3 calibration steps.

of an example sensor which is sensitive to the desired as well as to an undesired input variable (pressure and temperature, respectively). The transfer surface and the error surface after five pressure by three temperature calibration steps are shown in Figs. 6 and 7.

The absolute maximum error values for six different sensor characteristics after different numbers of calibration steps and

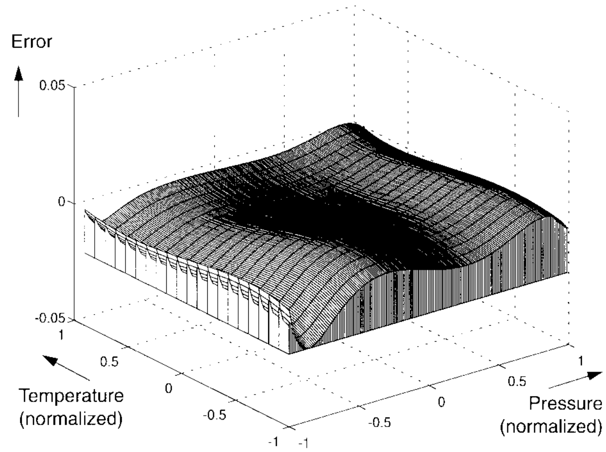


Fig. 7. Error surface after 5 × 3 calibration steps.

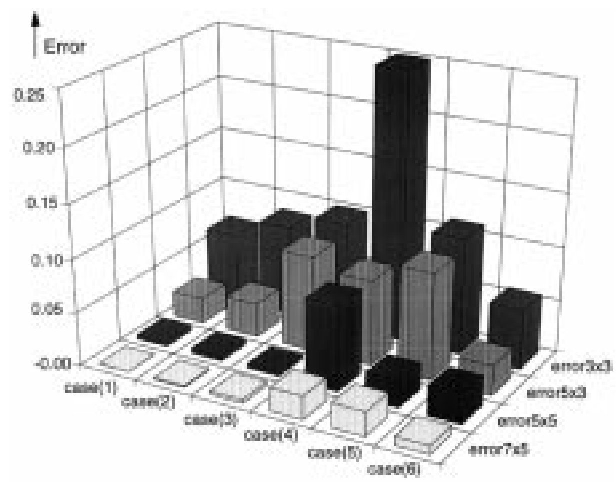


Fig. 8. Error values for different sensor characteristics.

for input pressure and temperature values normalized to (−1, 1) are shown in Fig. 8. The six different sensor characteristics investigated are referred to by the corresponding case numbers: error3×3 refers to the error values after 3×3 calibration steps and error5×3 to the error values after 5×3 calibration steps, whereas the error values after 5×5 and after 7×5 calibration steps are referred to by error5×5 and error7×5, respectively. The initial maximum nonlinearity error for all considered cases was approximately 25% of full-scale.

Simulations carried out to compare the performance of this method with that of the 2-D calibration polynomial using matrix inversion [8], showed a better performance of the proposed method as regards the speed of error reduction per calibration step, in addition to the fact that progressive error reduction always resulted with this method, as the number of calibration steps were increased.

IV. Microcontroller Implementation

Fig. 9 shows the block diagram of the proposed smart-sensor microcontroller-based system using a smart sensor bus [11] to read out integrated temperature and pressure sensors. The output of the pressure sensor (S_p) is determined by

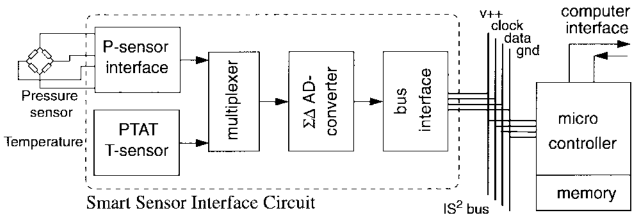


Fig. 9. Microcontroller-based pressure sensor system setup.

the applied pressure and the operating temperature of the sensor. A temperature sensor (S_T) is also used to give the calibration setup an indication of the operating temperature. The calibration program and the calibration coefficients are saved in the memory.

Through the user terminal, the necessary calibration data asked for by the microcontroller can be entered. The microcontroller used is the 80c552 which is an 8-bit microcontroller driven by 12 MHz clock. The system software consists of a program written in C, which is compiled, assembled and then downloaded into the system memory. The program consists of two parts: the *calibration part* to calculate the calibration coefficients and the *measurement part* to calculate the corrected values of sensor output using the values of calibration coefficients already calculated and saved in memory during the calibration cycle. It is an interactive program which, during the calibration process sends the necessary messages to inquire the user about the reference data required at each stage of the calibration process. Once the correct response has been received, 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.

It was earlier discussed that the 2-D calibration algorithm requires scanning across temperature values for as many pressure values as required to reduce the error to the specified limit. This represents a practical disadvantage during the calibration cycle, since it is easier and faster to change the pressure values applied to a sensor than to change the temperature. Accordingly, it is more convenient to select the order in which the calibration process is carried out such that it requires the minimum number of changes in temperature values. This fact has been accounted for in the program used where functions have been used to enable the program to carry out the calibration cycle along axes of fixed temperature values for different pressure values, whereas the correction process during the measurement cycle is carried out along fixed pressure values as was explained earlier.

The program was operated and tested using the fictive sensors data already used in the simulations. Actual results from the microcontroller 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 on the average 30 ms based on 5×5 calibration steps. This is fast enough for sensors with a signal bandwidth lower than 15 Hz.

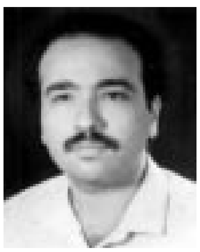
V. Conclusions

A technique to correct for sensor offset, gain, nonlinearity, and cross sensitivity, using an 8-bit microcontroller has been developed. Calibration methods based on mathematical correction functions implemented in a microcontroller have become increasingly popular because of their flexibility and effectiveness and also because of the continuing reduction of size and price of microcontrollers. Already it is possible to be incorporate them in the sensor package to form a smart sensor with calibration facility. With the proposed calibration method, it is possible to let such a smart sensor itself carry out both the calculation of the calibration coefficients and the calculation of the measurement correction, thus providing an accurate ready-to-use sensor system. Moreover, the calibration method, and the calibration system based upon it provide the user with a means to enter calibration points data until a certain error reduction is achieved. At each calibration step, a calibration coefficient is calculated according to a simple formula, and the newly calculated calibration coefficient is used to add a new term to correction function resulting from the previous calibration. A steady reduction in error always resulted when successively applying this method to different types of sensor response curves. The number of summations and multiplications required to calculate a single corrected value is comparable to that required by other methods based on polynomial correction using matrix inversion. Simulation results have also shown that the performance of the proposed method (as regards error correction) is comparable to and for most of the cases considered, better than that of the other method referred to. The advantage of the proposed method is the progressive user-controlled error reduction, in addition to the ease with which the calibration coefficients can be calculated. The calculation of the corrected values, now totalling 30 ms per sample, can be speeded up by using a 16-bit or 32-bit microcontroller. The software part can be further developed, such that the calibration cycle becomes more automated with minimum interference from the part of the user. In this way, the resulting smart sensor setup with the microcontroller makes a fast, automated, and thus cheap calibration of sensors in production possible, and also provides for fast and cheap customer recalibration [2]. Furthermore, since the smart sensor has all the calibration facilities on board, a minimum attention per sensor is needed, namely the reference data can be sent to many sensors at a time in a batch calibration process. This will reduce the production costs of accurate sensors considerably.

References

[1] J. H. Huijsing, F. R. Riedijk, and G. van der Horn, "Developments in integrated smart sensors," *Sens. Actuators*, vol. A43, pp. 276–288, 1994.
[2] J. Bryzek *et al.*, *Silicon Sensors and Microstructures*, Nova Sensor, 1990.
[3] S. Ansermet *et al.*, "Cooperative development of a piezoresistive pressure sensor with integrated signal conditioning for automotive and industrial applications," *Sens. Actuators*, vol. A21–23, pp. 79–83, 1990.
[4] J. Gakkestad *et al.*, "A front end CMOS circuit for a full bridge piezoresistive pressure sensor," *Sens. Actuators*, vol. A25–27, pp. 859–863, 1991.

[5] F. Schnatz *et al.*, “Smart CMOS capacitive pressure transducer with on-chip calibration capability,” *Sens. Actuators*, vol. A34, pp. 77–83, 1992.
[6] W. Gopel, J. Hesse, and J. Zemel, *Sensors: A Comprehensive Study, Vol. 1: Fundamentals and General Aspects*, VCH, 1989.
[7] P. Malcovati *et al.*, “Smart sensor interface with A/D conversion and programmable calibration,” *IEEE J. Solid State Circuits*, vol. 29, pp. 963–966, Aug. 1994.
[8] P. N. Mahana *et al.*, “Transducer output signal processing using an eight-bit microcomputer,” *IEEE Trans. Instrum. Meas.*, vol. IM-35, pp. 182–186, June 1986.
[9] S. B. Crary *et al.*, “Digital compensation of high performance silicon pressure transducers,” *Sens. Actuators*, vol. A21–23, pp. 70–72, 1990.
[10] W. T. Bolk, “A general digital linearising method for transducers,” *J. Physics, Vol. E: Sci. Instrum.*, pp. 61–64, 1985.
[11] F. R. Riedijk *et al.*, “Sensor interface environment based on sigma-delta conversion and serial bus interface,” *SENSORS, J. Appl. Sens. Technol., Sensors Expo Issue,* Apr. 1996.



Khalid Fouad Lyahou was born in Baghdad, Iraq, on July 18, 1958. He received the B.Sc. and M.Sc. degrees in electrical engineering from the University of Baghdad, in 1980 and 1987, respectively. He received the Degree of Chartered Design Engineer in March 1996, from the Technical University of Delft, Delft, The Netherlands.

He was a Teaching and Research Assistant at the University of Baghdad, from 1980 until 1991. In 1993, he joined the Faculty of Electronic Engineering, the Technical University of Delft. His main interests are digital signal processing and microcontroller programming.

Gert van der Horn was born on July 4, 1968, in Ede, The Netherlands. He studied electrical engineering at the Delft University of Technology, Delft, 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 is now researching smart sensor calibration techniques and pursuing the Ph.D. degree.

His research interests include instrumentation systems, analog signal conditioning, AD conversion, and amplifier design.

Johan H. Huijsing (SM’81–F’97) was born in Bandung, Indonesia, on May 21, 1938. He received the M.Sc. degree in electrical engineering from the Delft University of Technology, Delft, The Netherlands, in 1969, and the Ph.D. degree from Delft University of Technology in 1981 for work on operational amplifiers (thesis: “Integrated circuits for accurate linear analogue electric signal processing,” supervised by Dr. I. 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.