

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,400

Open access books available

117,000

International authors and editors

130M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Sensors and Digital Signal Conditioning in Mechatronic Systems

Dilşad Engin and Mustafa Engin

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/67986>

Abstract

Essential part of a mechatronics system is the measurement system that senses the variations in the physical parameters, such as temperature, pressure, displacement, and so on, and converts it to voltage or current. The control of industrial processes and automated manufacturing systems requests accurate, moreover, linearized sensor measurements, where numerous sensors have nonlinear characteristics. In mechatronic systems, accurate measurement of the dynamic variables plays a vital role for the actuators to function properly. This chapter presents linearization methods and a measurement system in mechatronics consisting of temperature sensors and the signal-conditioning circuits, providing detailed information on design process of an embedded measurement and linearization system. This system uses a 32-bit microcontroller for thermocouple (T/C) cold junction compensation, amplification of low output voltage, then conversion to digital, and linearization of the type K thermocouple's output by software to output a desired signal. Piecewise and polynomial methods are used in linearization software, and the implemented embedded system for the linearization of a type K T/C is presented as a case study. The obtained results are compared to give an insight to the researchers who work on measurement systems in mechatronics.

Keywords: temperature sensors, sensor linearization, piecewise linearization, polynomial linearization, embedded system

1. Introduction

Mechatronic systems are composed of mechanical and electrical components, referred as “smart” systems because of the integration of sensors, actuators, and control systems. Essential part of a mechatronics system is the measurement system that senses the variations in the physical parameters, such as temperature, pressure, displacement, and so on, and converts it to an electrical quantity, viz. electric voltage or current. The measurement system consists of

mainly two parts: sensor and signal-conditioning circuit. A device that displays, records, or controls the measured variable utilizes the output of the measurement system.

In a measurement system, sensor detects the change in the physical parameters, but this measured variable is not usually in the form desired by the rest of the system. Signal-conditioning part may consist of sensor output amplification, analog-to-digital conversion (ADC), compensation, frequency-to-voltage conversion, and so on. The signal-conditioning interface is the part of the measurement system where the measured signal is converted to an analog or digital electrical signal that is required by the instrument.

The control of industrial processes and automated manufacturing systems requests accurate, moreover, linearized sensor measurements where numerous sensors have nonlinear characteristics. In mechatronic systems, accurate measurement of the dynamic variables plays a vital role for the actuators to function properly. An inaccurate measurement of the rotational position of a robot arm may result in a misplacement of an electronic part on a printed circuit board. The nonlinear voltage output of the temperature sensors embedded in the motor windings results in a lower temperature reading than the actual temperature, which may cause shutdown of the motor.

Emerging technologies in measurement systems have led to integrated signal-conditioning circuits within the sensors. Spreading the usage of microcontrollers or microprocessors in sensing technology has increasingly resulted in utilizing signal-processing functions embedded in the sensors, consequently accurate and linear signals are available at the outputs. For this reason, the appropriate selection of linearization technique is important while the processing time and memory usage of the processor must be limited for fast response and for maintaining the cost of the sensor reasonably priced. Linearization and calibration algorithm design with hardware and/or software is a broad research interest.

2. Sensors in mechatronics

The vital part of a mechatronics system is the sensing of physical parameters—either discrete or continuous. In Electrical Transducer Nomenclature and Terminology standard, a sensor is defined as “a device which provides a usable output in response to a specified measurand” [1]. Our focus will be on nonlinear sensors.

In mechatronic systems, one of the most utilized sensors is the capacitive-type sensor. Capacitance changes nonlinearly with the displacement of a diaphragm in a pressure sensor, linear or rotational motion of an object from the reference, or the displacement of vibration sensors. Linearization and calibration of this sensor has been an extensive research area [2].

The other utilized sensor is the temperature sensor, which measures the process temperature, monitor the temperature of the rotor windings, or in temperature compensation of sensors whose performance is affected by temperature changes. Temperature is the most measured and controlled dynamic variable in manufacturing and machine control. There are various types of these sensors, which use resistance change of metals (resistance temperature detector (RTD)) and

semiconductors (thermistors) and thermoelectric effect (thermocouples (T/Cs)) due to the change in temperature. Resistive temperature detectors and thermocouples are the most preferred ones due to high linearity of the RTDs and wide operating range of the T/Cs. Although highly nonlinear, thermistors are also used in temperature sensing due to their high sensitivity and low cost. Linearization of thermistors has found an extensive interest among researchers [3, 4].

Thermistor is a semiconductor-resistive temperature sensor made from metal oxides (negative temperature coefficient, NTC) and doped polycrystalline ceramic containing barium titanate and other compounds (positive temperature coefficient (PTC)). In NTC thermistors, the resistance decreases with the increase in temperature. NTC thermistors are widely used for a narrow range of -50 to 150°C .

The NTC thermistor resistance R_T is related to temperature by the empirically developed Steinhart-Hart equation:

$$\frac{1}{T} = A + B \ln R_T + C \ln(R_T)^3 \quad (1)$$

where T is in K, and A , B , and C are the thermistor coefficients given by the manufacturers, or you can calculate these values to obtain better accuracy using least-square method with the thermistor table data [3, 4].

RTDs are metal or metal alloy resistors that exhibit an increase in resistance with the increase in temperature. The operating temperature ranges from -200 to 850°C . They are characterized by a polynomial with respect to temperature as

$$R_T = R_0(1 + \alpha_1 T + \alpha_2 T^2 + \dots + \alpha_n T^n) \quad (2)$$

where the curve-fitting coefficients are supplied for different types of RTDs.

Platinum RTDs are highly accurate (up to $\pm 0.01\%$ at 0°C), but they have relatively linear characteristic curve over large spans, but not exactly a straight line. The platinum RTD with a resistance value of 100Ω at 0°C , called Pt100, has nearly linear characteristic curve as shown in **Figure 1** compared to the nonlinear $100\text{-}\Omega$ NTC thermistor.

For platinum RTDs, according to IEC 751, the polynomial in Eq. (2) can be reduced to

$$R_T = R_0[1 + \alpha_1 T + \alpha_2 T^2 + \alpha_3 (T - 100)T^3] \quad (3)$$

in which α_3 can be applied only when $T < 0^\circ\text{C}$. The coefficients α_1 , α_2 , and α_3 for a standard sensor are specified in IEC751. For very high precision, researchers work on linearization methods for platinum RTDs [5, 6].

Thermocouple is a self-powered sensor where the ends of two different metals or metal alloys are weld bonded. The bead weld point is the hot junction, and the other point is the cold junction. A voltage in the order of millivolts is generated at this cold junction. This thermoelectric voltage is a function of the difference between the hot-junction and the cold-junction temperatures and also the composition of the metals as shown in **Figure 2**.

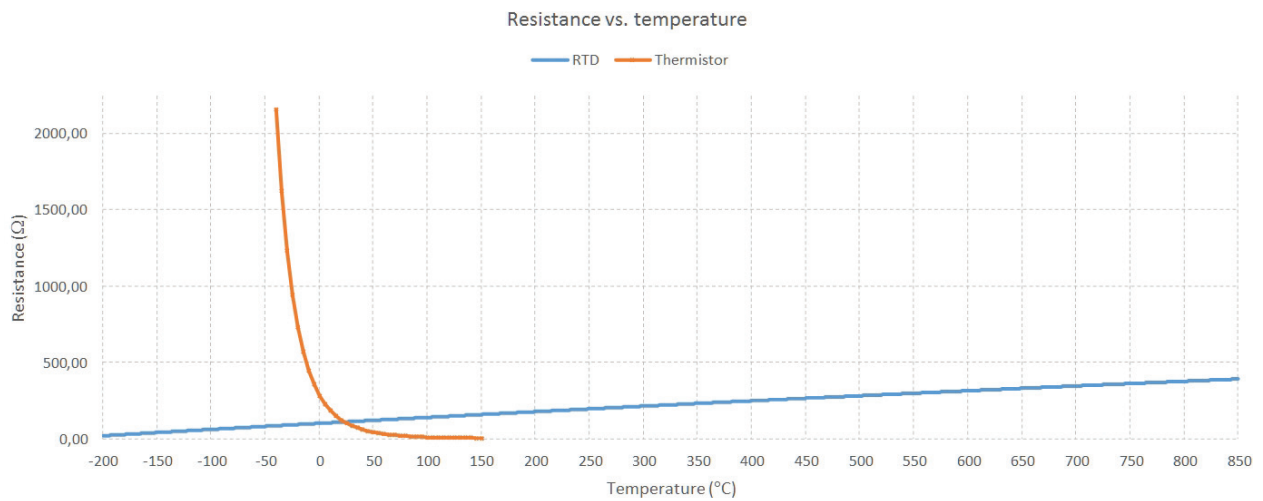


Figure 1. RTD characteristic curve, data from DIN IEC 751 temperature/resistance table compared to NTC thermistor.

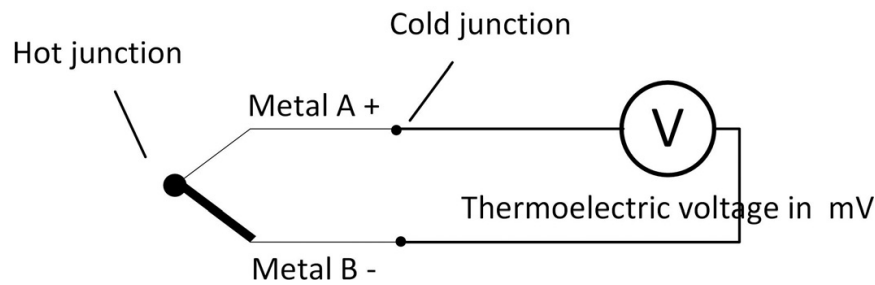


Figure 2. Thermoelectric voltage of a thermocouple.

NIST ITS-90 thermocouple database [7] presents this generated voltage for temperatures below 0°C as

$$E = \sum_{i=0}^n d_i (t_{90})^i \tag{4}$$

and for temperatures above 0°C as

$$E = \sum_{i=0}^n d_i (t_{90})^i + b_0 e^{b_1 (t_{90} - b_2)^2} \tag{5}$$

where the coefficients, b_0 , b_1 , b_2 , and d_i , of reference equations provide the thermoelectric voltage, E , as a function of temperature, t_{90} , for the indicated temperature ranges, and $b_0 = 1.185976 \times 10^{-1}$, $b_1 = -1.183432 \times 10^{-4}$, and $b_2 = 126.9686$.

This generated voltage by T/C's cold junction can be obtained by measuring the junction temperature with a thermistor, or an IC temperature sensor fixed on an isothermal block. The measured voltage corresponds to the cold-junction temperature and it is utilized in computing

the hot-junction temperature. The voltage output of the cold-junction temperature can be fetched from the T/C reference tables according to IEC 584-1. The difference between this voltage and the measured voltage is the voltage generated by hot junction. Temperature value, which corresponds to the computed voltage, can also be fetched from the T/C temperature/mV output table. This process is the software compensation.

There are a variety of T/C types ranging from -270 to 2300°C . They are small in size, thus fast in response, and low in cost. Unfortunately, they have poor linearity and low sensitivity. The lead effect is so high that it must be compensated. Linearization, amplification of the low output, and compensation of the lead effect of temperature sensors can be carried out by analog circuit design [8, 9], by developed software embedded in a microcontroller [10, 11], or a circuit design where the sampled data stored in an SRAM is transferred to a computer for linearization process [12]. Anyway, temperature measurements with T/Cs are challenging, particularly when the temperature measurement is below 0°C .

We discuss the measurement system in mechatronics consisting of thermocouples as temperature sensors and the signal-conditioning circuits, providing detailed information on design process of an embedded measurement and linearization system. This system uses 32-bit microcontroller for T/C cold-junction compensation, analog-to-digital conversion, and T/C temperature sensor output linearization by software.

3. Methods for linearization

To overcome the poor linearity of T/Cs, linearization circuits are developed. Complicated analog electronic circuits are designed to cope with T/C's nonlinear and low output voltage problem [13, 14]. These circuits amplify the low-voltage output to a desired level and linearize the output to obtain the intended accuracy in the operating range of the system [6]. Therefore, in linearization with hardware precise assignation of the circuit elements is considerably important to achieve the essential accuracy.

Software linearization techniques are also preferred among researchers. Wei et al. first amplified the T/C output, converted it to digital by on-chip analog-to-digital converter of the microcontroller, and then linearized the output using least-squares method [14]. Sarma and Boruah amplified type K T/C output, then converted to digital by 12-bit ADC, and finally linearized the output with an eight-bit microcontroller using a piecewise polynomial of ninth degree [15]. Engin used an eight-bit microcontroller, its on-chip programmable gain amplifier and 24-bit ADC to amplify and digitize the measured type T T/C output, and built-in temperature sensor for compensation, finally linearized the output by first- and second-degree polynomials, and piecewise linear interpolation methods [11]. Some researchers utilized a T/C amplifier for amplification and cold-junction compensation and linearized the output by look-up table (LUT) embedded in the microcontroller [11, 16, 17]. Wang et al. used B-spline method for linearizing the output of nonlinear sensors [18]. The sensor linearization process comprises complex mathematical computations that an eight-bit microcontroller cannot achieve. Therefore, many researchers had rather realized calibration algorithm on a computer

through I/O interface cards. Danisman et al. initially amplified the T/C output with an instrumentation amplifier, then used an ADC for conversion to digital, and transmitted this digitized measurement to a computer where they applied artificial neural network (ANN) calibration algorithm by means of a virtual instrument [10]. Researchers who utilized low-cost microcontrollers for linearizing the sensor outputs limited the sensor's input range to a part of the full scale.

In this section, we give a brief description of the linearization methods. Mathematical models and implementation by software will be the main scope.

3.1. Least-squares regression method

A set of measured values from a sensor output needs to be fitted to a curve in order to obtain a mathematical representation of the sensor output. Linear least-squares regression is considerably the most used modeling method. This method utilizes linear algebra to determine the "best-fit" line for a data set by minimizing the sum of the squares of the vertical residuals of the data points to a modeling curve. The sum of the squares of the residuals is preferred because this warrants continuously differentiable residuals at every point contrary to the absolute error differentiation.

Our aim is to find the "best fit" particular to any finite linear combinations of the identified function instead of the best-fit line. Consequently, we can write a general equation including functions f_1, \dots, f_n , and then find the coefficients a_1, \dots, a_n such that the linear combination

$$\hat{y} = a_1 f_1(x) + \dots + a_n f_n(x) \quad (6)$$

is considered to be the best approximation to the data.

We have a set of n data points $\{(x_1, y_1), \dots, (x_n, y_n)\}$ with n residuals $\{(y_1 - \hat{y}_1), \dots, (y_n - \hat{y}_n)\}$. If our model \hat{y} is a good fit to the set of data points, then the mean should be small and the variance will be a measure of how the model fits the data. For this set of measured data, we define the error as the sum of the squares of the residuals as

$$E(a, b) = \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (7)$$

We must determine the coefficients a and b , which minimize the error. Then, we should calculate the partial derivatives of the error such that

$$\frac{\partial E}{\partial a} = 0 \text{ and } \frac{\partial E}{\partial b} = 0 \quad (8)$$

For linear approximation to the data, we can write that our model is $\hat{y} = ax + b$, and if this is the best fit, then the residual must be $y - \hat{y} = 0$. An example of linear least-squares regression analysis for the given set of data in **Table 1** is implemented using Excel regression analysis, and the linear approximation plot is given in **Figure 3**.

x	0	1	2	2.5	3	3.5	4	4.75	5
y	2	1	0.5	1.125	1	1.125	0.75	0.5	0

Table 1. Example data for linear least-squares approximation.

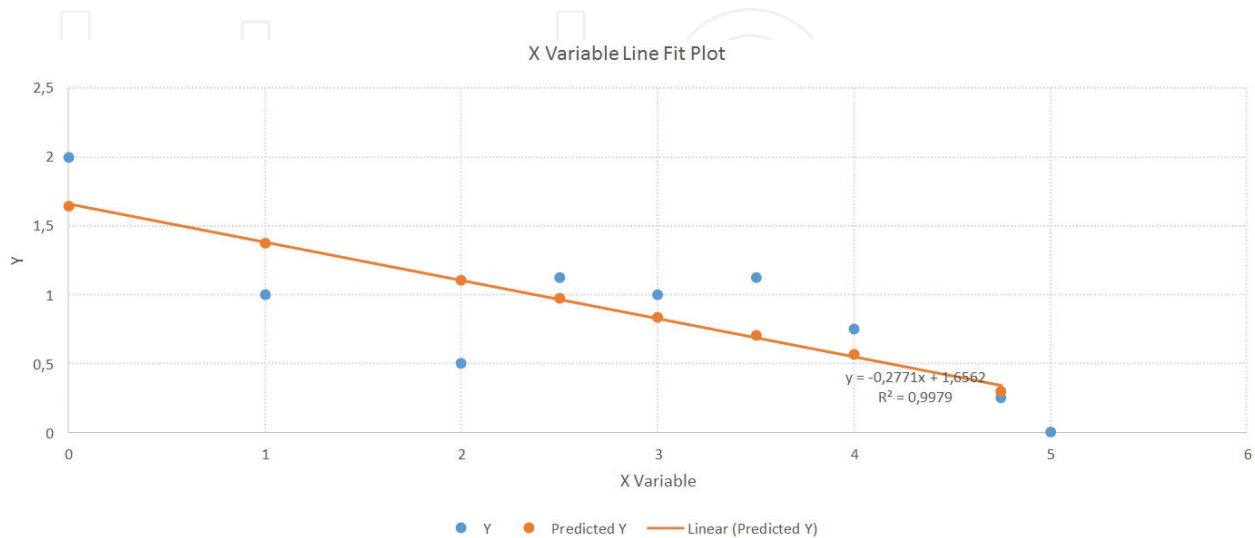


Figure 3. Linear least-squares regression plot for the given set of data.

Linear least-squares regression is the main instrument for process modeling since it is effective in finding a model that best fits, especially, a small set of data. Although there are sets of data that are better defined by nonlinear-coefficient functions, numerous practices in engineering can be described by linear models due to the fact that these processes are linear in nature or they can be approximated by a linear model within narrow ranges. In software-based sensor linearization, it provides minimum code size and consumes the lowest power.

On the other hand, for inherently nonlinear processes, it is more difficult to find a linear model to fit the set of measured data, particularly for wide range. Moreover, the computation time for linearization process run by a computer or a microcontroller will increase as the explanatory variables increase. The sensitivity to the outliers caused by improper measurements can also seriously deflect the “best-fit” line; therefore, model validation becomes critical to acquire accurate responses to the demands stimulating the construction of the model [19].

3.2. Piecewise polynomial interpolation

Curve fitting of a set of measured data from a sensor output can be accomplished by high-degree polynomial interpolants where we should abstain from equally spaced data points. This may result in a problem in producing accurate interpolant over a wide range. To overcome this difficulty is to partition the wide range into subintervals $\alpha = x_1 < x_2 < \dots < x_n = \beta$, and then interpolate the function on each subinterval $[x_i, x_{i+1}]$ with a low-degree polynomial.

This is the piecewise polynomial interpolation idea. For linearization of the sensor outputs, we prefer linear approximations rather than polynomial fits.

Piecewise linear approximation is a technique of obtaining a function that fits a nonlinear objective function by adding binary or continuous variables and constraints to reformulate the original function [20]. The particular aim is to approximate a function of one variable in terms of sequential linear pieces. The successive data points connecting piecewise straight lines are called breakpoints.

Assume that a general nonlinear continuous function $f(x)$ of a single variable x , and x is within the interval $[\alpha, \beta]$. Call $y_i = f(x_i)$, $i = 1 : n$. The breakpoints of (x) in general are $\alpha = a_1 < a_2 < \dots < a_n = \beta$, and **Figure 4** indicates the piecewise linearization of $f(x)$ for breakpoints $[a_1, \dots, a_9]$. In the figure, the dots denote a measured data set. Usually, we use the measured data points as the breakpoints, $[\alpha, \beta] = [x_1, x_n]$. If we have a set of n data points $\{(x_1, y_1), \dots, (x_n, y_n)\}$, and connect these points with straight lines, then we obtain the graph of a piecewise linear function.

The significance in reformulating a consider that we have the piecewise as a linear function is to use a distinct variable for each segment. Consider that we have the piecewise linear function $f_k(x_k)$ in **Figure 4** that has eight line segments over the range of x_k values. We now have eight new variables $[x_{k1}, \dots, x_{k8}]$ and in general form, set

$$x_k = \sum_{i=1}^{n-1} x_{ki} \tag{9}$$

For each segment, the slope is

$$s_{ki} = \frac{y_{i+1} - y_i}{x_{i+1} - x_i}, \tag{10}$$

and the nonlinear function can be rewritten in terms of the slopes as

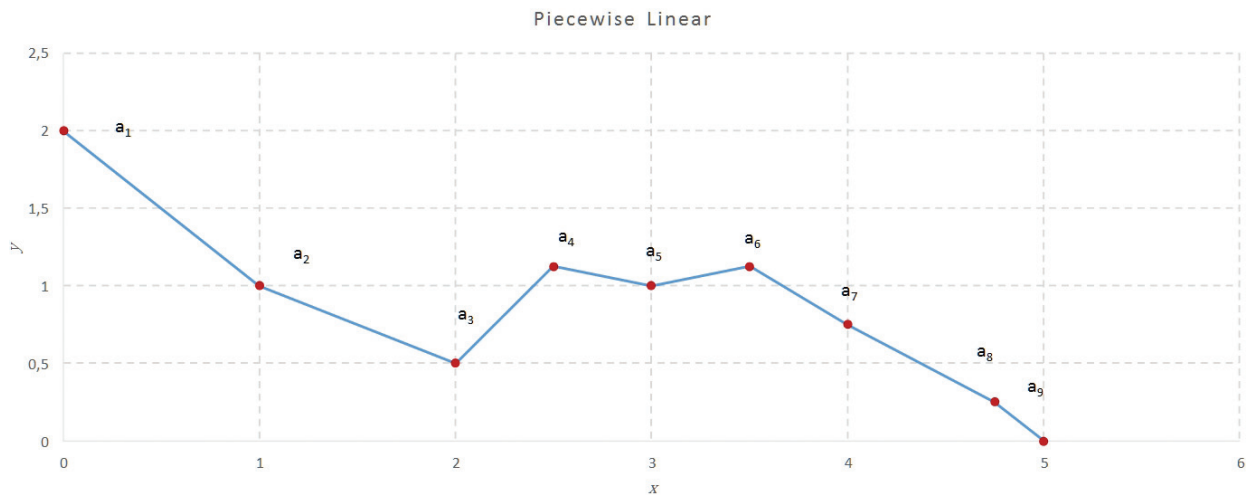


Figure 4. Piecewise linearization of the data set in **Table 1**.

$$f_k(x_k) = \sum_{i=1}^{n-1} s_{ki} x_{ki} \quad (11)$$

where $s_{k1} > s_{k2} > \dots > s_{k(n-1)}$ condition should be satisfied for $f_k(x_k)$ to be concave in order to accurately linearize the nonlinear function $f(x)$.

Unfortunately, there are piecewise linear functions that cannot be reformulated by linear programming as depicted above. Therefore, the nonlinear function (x) can be approximated by nonlinear programming algorithms.

You may find the piecewise linearization procedure in the nonlinear programming textbooks [21, 22], where a nonlinear function $f(x)$ can be approximately linearized over the interval $[\alpha, \beta]$ as

$$L(f(x)) = \sum_{i=1}^n f(a_i) w_i \quad (12)$$

$$x = \sum_{i=1}^n a_i w_i \quad (13)$$

The weights w_i are non-negative associated with the i th breakpoint as

$$\sum_{i=1}^n w_i = 1, \quad w_i \geq 0, \quad i = 1, 2, \dots, n \quad (14)$$

This mixed integer programming ensures the validity of the approximation by the conditions in [22].

The accuracy of the linear approximation significantly relates to the number of breakpoints where a greater number of breakpoints result in more accurate approximation. Unavoidably, adding a number of breakpoints results in a considerable growth in the memory size for storing these values. When linearization of the sensor outputs by a microcontroller is accomplished, the number of breakpoints should be considered as well as the accuracy.

3.3. Piecewise linear approximation with look-up table

When polynomial, hyperbolically tangent or exponential functions are linearized, computations can be accomplished by look-up tables avoiding the Taylor-series expansion or other polynomial calculations with excessive floating point operations. The given values of the data are stored digitally in a memory. The data points in the LUT can then be approximated by piecewise linear approximation method.

Nonlinear outputs of the sensors have a variant degree of nonlinearity through the operating range. Consequently, in utilizing the linear approximation method to reach a precise accuracy, the data points in the LUT can be selected that are not equidistant. Whenever the sensor output is more "linear," the distance between the data points can be increased, thus decreasing the

memory need for LUT. In the fixed step table, the step size must be matched to the most “nonlinear” part of the function throughout the LUT.

Look-up table usage for linearization of the sensor outputs provides more accurate results particularly for small-scale-embedded systems having limited size of memory. Nevertheless, it is not recommended to make use of look-up tables for sensor outputs with high resolution, since it needs larger memory than a small-scale-embedded system has.

3.4. Cubic spline method

The linear interpolation method for sensor linearization process exhibits distortions that result in corners on the plot of the function at the breakpoints. To obtain an interpolating function that has continuity around the breakpoints, cubic splines are used. On the other hand, they are just piecewise continuous, inferring that the third derivative is not continuous. If the sensor linearization process is susceptible to the evenness of derivatives higher than the second, cubic spline method is not preferred.

Consider the general nonlinear continuous function $f(x)$ of single variable x within the interval $[\alpha, \beta]$ with breakpoints $\alpha = a_1 < a_2 < \dots < a_n = \beta$. We have n breakpoints with $n - 1$ intervals. The cubic spline interpolation results in a piecewise continuous curve that traverses each breakpoint. For a set of n data points $\{(x_1, y_1), \dots, (x_n, y_n)\}$, and per interval, there is a unique cubic polynomial $s_i(x)$ with its particular factors as

$$s_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i \text{ for } x_i \in [x_i, x_{i+1}] \quad (15)$$

The cubic polynomial requires two conditions to match the values of the data set at each end of the intervals, which results in a piecewise continuous function on $[\alpha, \beta]$,

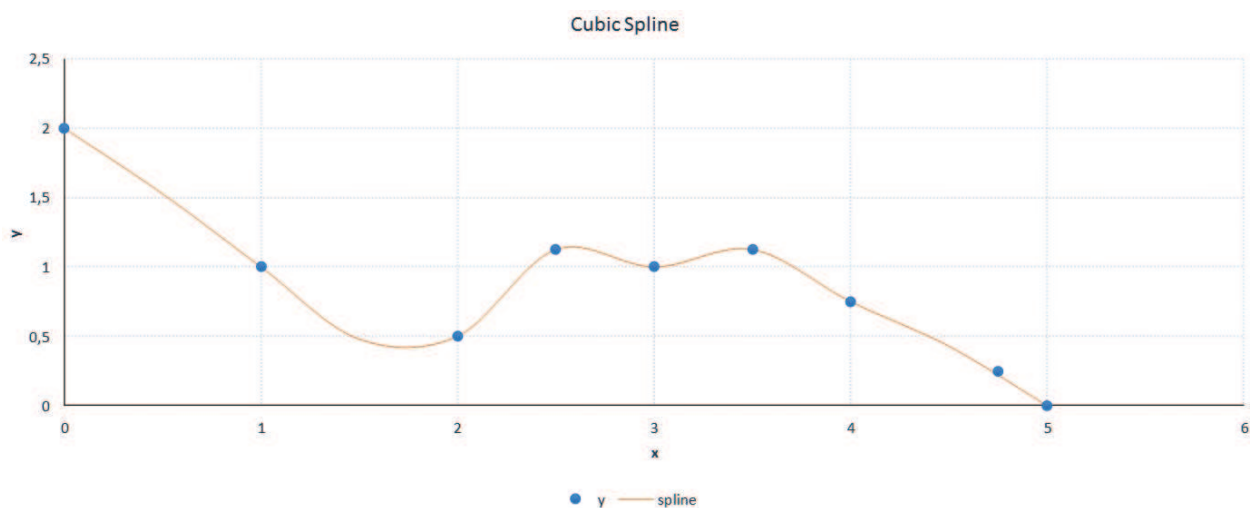


Figure 5. Cubic spline linearization of the data set in Table 1.

$$s_i(x_i) = y_i \text{ and } s_i(x_{i+1}) = y_{i+1} \quad (16)$$

Providing smoother interpolation, the first and the second derivatives must be continuous

$$s'_{i-1}(x_i) = s'_i(x_i), \quad s''_{i-1}(x_i) = s''_i(x_i). \quad (17)$$

The remaining boundary conditions to complete the cubic spline polynomial are as follows:

1. For natural cubic spline: $s''_0(x_0) = 0, \quad s''_{n-1}(x_n) = 0,$
2. For clamped cubic spline: $s''_0(x_0) = f'(x_0), \quad s''_{n-1}(x_n) = f'(x_n).$

The data in **Table 1** are now linearized by cubic spline interpolation method using a function added in Excel which is developed by SRS1 Software, LLC. **Figure 5** indicates the cubic spline linearization of $f(x)$. Compared to the linear interpolation, cubic spline approximation results in a smoother curve which has continuity around the breakpoints.

4. Sensor linearization

Sensor outputs are not as linear as we expect. Consequently, linearization by hardware or software has become one of the challenging parts in measurement systems.

There are a variety of temperature sensors in the market. Yet, numerous temperature sensors have nonlinear characteristics or temperature measuring ranges are quite narrow. While utilizing these sensors, outputs are compensated, and linearized using digital circuitry and software, resulting in “smart sensors.” These smart temperature sensors are thermocouples, thermistors, and resistive temperature detectors. The temperature sensors have low-voltage outputs, and nonlinear characteristic as mentioned in Section 2. Amplification and linearization of the voltage output of these temperature sensors is essential before utilizing them in industrial applications.

As the measurement range is the widest, and the linearity is poorer than most temperature sensors, thermocouples are in our scope of linearization practice. Due to their robustness to very high/very low temperatures, and oxidizing environments, they are preferred to other temperature sensors that melt with high temperature, or corrode with vapor. We present T/C compensation, and linearization with polynomial interpolation as a case study.

4.1. Theory of linearization

The development of a linearization block consists of several steps. We should first start with the nonlinear function $f(x)$, which characterizes the sensor output with respect to the input. This unknown function can be determined by a calibration process where multiple input/output (x and y) values are obtained experimentally. Using this set of data, $f(x)$ is then determined by one of the curve-fitting tools.

After obtaining the nonlinear function, $f(x)$, we can find the inverse function $f^{-1}(y)$ that is implemented in the linearization block as shown in **Figure 6**. In this part, we must make a

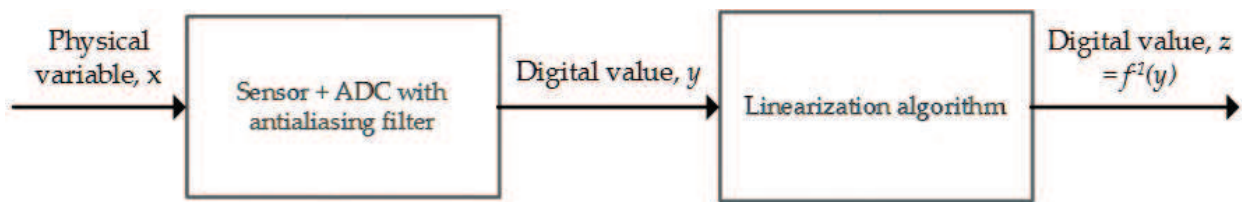


Figure 6. Sensor linearization procedure.

choice depending on the processor that we use. If our embedded measurement system is implemented on a 32-bit floating point controller, then the inverse function can be computed in real time. For 32-bit controllers, up to 13th degree polynomial interpolation can be executed with high accuracy, fast response, and minimum code size.

If the controller has an eight-bit processor with limited computing ability, then the best choice will be using a LUT stored in the memory. The function y is an n -bit integer at the output of an ADC, and it is used to fetch the inverse function $f^{-1}(y)$ in the memory. This is the procedure followed when using low-cost small-embedded systems for linearization process.

When we choose to use LUT, there will be limited source data set; therefore, an interpolation method must be used to compute the measured value. In this case, the interpolated value, $f^{-1}(y_m)$, can be calculated by using a piecewise linear interpolation method as

$$f^{-1}(y_m) \approx z_i + \frac{z_{i+1} - z_i}{y_{i+1} - y_i} \times (y_m - y_i) \quad (18)$$

where y_m is a measured variable between the points $[y_{i+1}, y_i]$ and $\frac{z_{i+1} - z_i}{y_{i+1} - y_i}$ is the slope of each segment. This procedure can be repeated for each interval between the breakpoints.

Similar procedure will be applied for cubic spline interpolation by using the equations given in Section 3.4.

Considering the required size of the memory for LUT entries and accuracy of the measurement, we know that for n -bit ADC, we need 2^n breakpoints for better accuracy. We have a contradiction whether to increase the number of breakpoints for better accuracy or to keep the number of LUT entries limited to decrease the memory size at the expense of accuracy. This decision will be given based on the circumstances. If we use a 32-bit controller with fast and high-computing ability, and large memory size, LUT size can be increased, but for an eight-bit controller it must be limited. In this case, we may limit the memory size by masking part of the bits of the digitized measurement without affecting the number of entries in the LUT. This can be considered only if the controller has a 16- or a 24-bit ADC, but limited memory size.

4.2. Thermocouple measurement setup and linearization algorithms

The most utilized thermocouples are types J, K, T, and E as given in **Table 2**. It can be interpreted from the table that the temperature ranges are the widest among other temperature sensors, but

Type	Composition	Temperature range
J	Iron vs Cu-Ni alloy	-210 to 1200°C
K	Ni-Cr alloy vs Ni-Al alloy	-270 to 1372°C
T	Cu vs Cu-Ni alloy	-270 to 400°C
E	Ni-Cr alloy vs Cu-Ni alloy	-270 to 1000°C

Table 2. Standard thermocouple types and their temperature ranges.

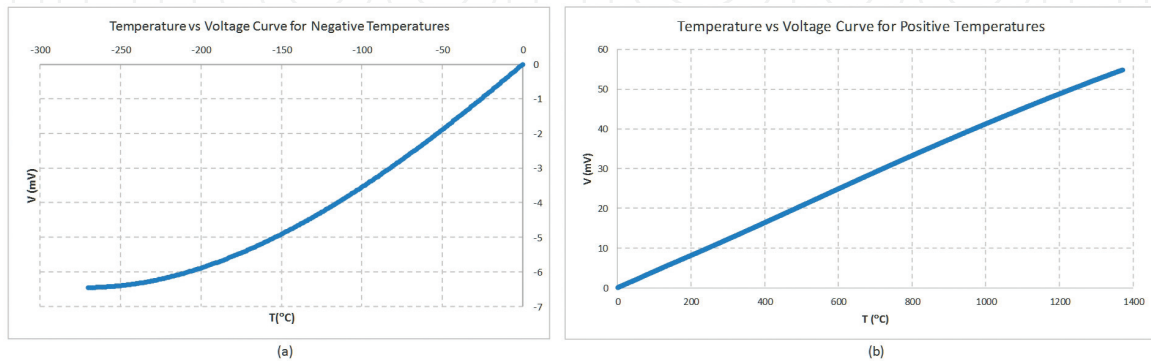


Figure 7. Characteristic curve of type K T/C (a) negative and (b) positive temperature range.

unfortunately the linearity is poor. We have selected type K T/C as its characteristic curve is considerably nonlinear, particularly in the negative temperature range as shown in **Figure 7**.

4.2.1. Hardware

The suggested temperature measuring system consists of a type K T/C, a low-voltage micro-power amplifier, OPA333 with very low-offset voltage (max. 10 μ V) and near-zero drift over time, 10 k Ω thermistor for cold-junction compensation of T/C, a cost-efficient 32-bit microcontroller, and a serial port driver (**Figure 8**). This system uses an analog hardware- and software-mixed linearization approach.

The low-voltage output of the type K T/C is in the range of -6.458 to 54.886 mV corresponding to -270 to 1372°C input range, so this output is amplified by an external amplifier. Then, the amplified voltage is applied to the built-in ADC input. A low-pass filter is used for noise suppression across the T/C ends.

We preferred Arduino Due based on a 32-bit ARM core microcontroller, which has 16-channel 12-bit ADCs; USB, Universal Synchronous/Asynchronous Receiver/Transmitter (USART), Serial Peripheral Interface (SPI), and I2C compatible Two-wire Interface (TWI) serial communication ports; 512 KB of flash and 100 KB of SRAM memory size are sufficient to run the linearization process by polynomial calculations, or to store the data for look-up table.

The embedded temperature measuring system is built and the amplified output of the T/C is connected to an analog channel of the microcontroller. The 12-bit built-in ADC of the controller board is used to obtain digital values corresponding to the mV output of the T/C.

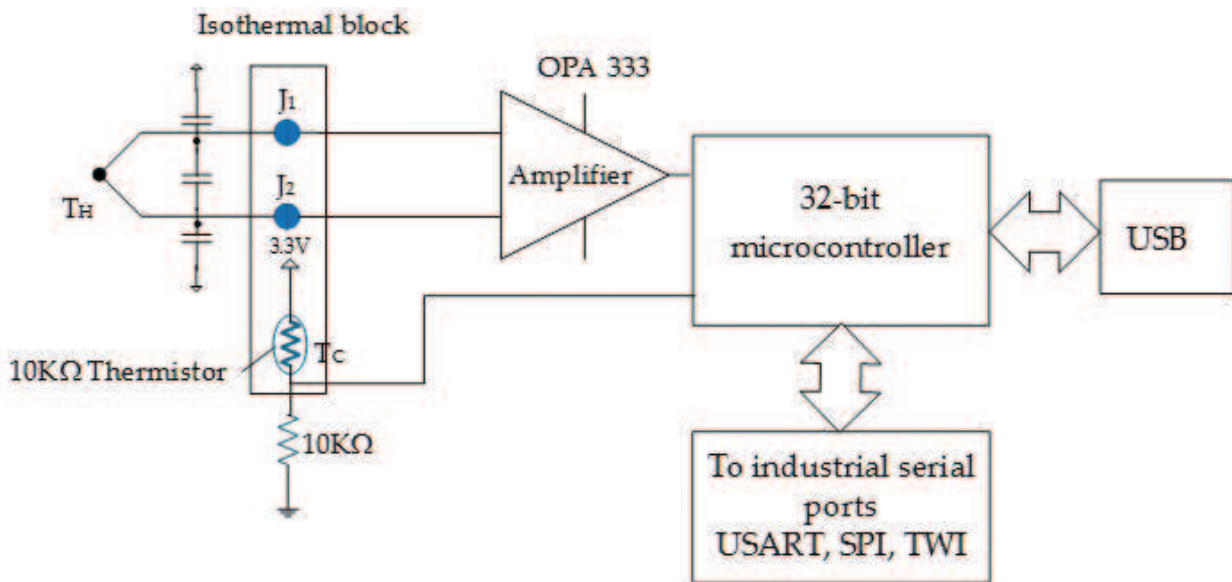


Figure 8. Temperature measuring system.

4.2.2. Software

Arduino Due board with ARM Cortex-M3 core can be programmed by its own integrated development environment (IDE) based on C/C++ programming language. The flowchart of the algorithm for the temperature measuring system is given in Figure 9.

The sixth degree polynomial obtained by Excel tool and the ninth degree polynomial [7] are used to reconstruct the inverse function of the T/C's characteristic function. For type K T/C, the following polynomial with the inverse coefficients given in Table 3 is used for calculating the temperature value corresponding to the measured voltage in millivolts:

$$t_{90} = \sum_1^n d_i E^i \quad (19)$$

4.3. Linearization algorithms

The model for the type K T/C is implemented by polynomial calculations in Eqs. (4) and (5) in Simulink using NIST ITS-90 Thermocouple Database [8]. First, piecewise linear interpolation is used with LUT to linearize the temperature sensor output signal, and the interpolation values along with the measured temperature values are generated at the outputs. The model for type K T/C measuring system is shown in Figure 10.

The first block "Type K Thermocouple Model" calculates the polynomial model for negative and positive temperature ranges separately. The second block "ADC input and hardware model" consists of analog scaling of the measured voltage output of the TC, an anti-aliasing filter, and ADC quantizer with a sample and a hold block. The final block is for software specifications for converting ADC values to temperature where 1D LUT is used with linear and cubic spline interpolation algorithm for linearization.

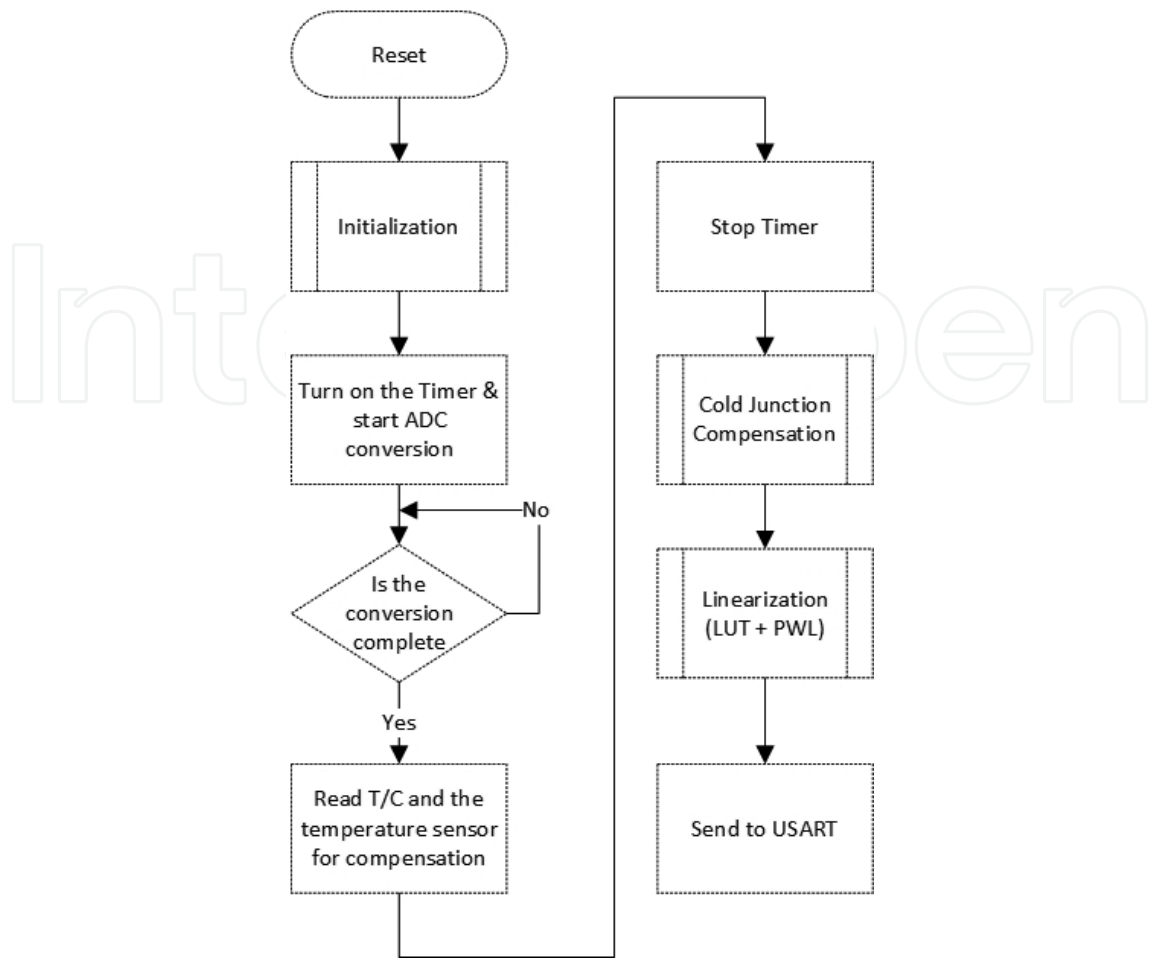


Figure 9. Flowchart of the developed software for the temperature measuring system.

Temperature range	-200 to 0°C	0 to 500°C	500 to 1372°C
Voltage range	-5.891 to 0	0 to 20.644	20.644 to 54.886
d_0	0.00E+00	0.00E+00	-1.32E+02
d_1	2.52E+01	2.51E+01	4.83E+01
d_2	-1.17E+00	7.86E-02	-1.65E+00
d_3	-1.08E+00	-2.50E-01	5.46E-02
d_4	-8.98E-01	8.32E-02	-9.65E-04
d_5	-3.73E-01	-1.23E-02	8.80E-06
d_6	-8.66E-02	9.80E-04	-3.11E-08
d_7	-1.05E-02	-4.41E-05	0.00E+00
d_8	-5.19E-04	1.06E-06	0.00E+00
d_9	0.00E+00	-1.05E-08	0.00E+00
Error range	-0.02 to 0.04	-0.05 to 0.04	-0.05 to 0.06

Table 3. Inverse coefficients for type K thermocouple.

**Temperature measurement system:
A chain of components from
physical environment to software**

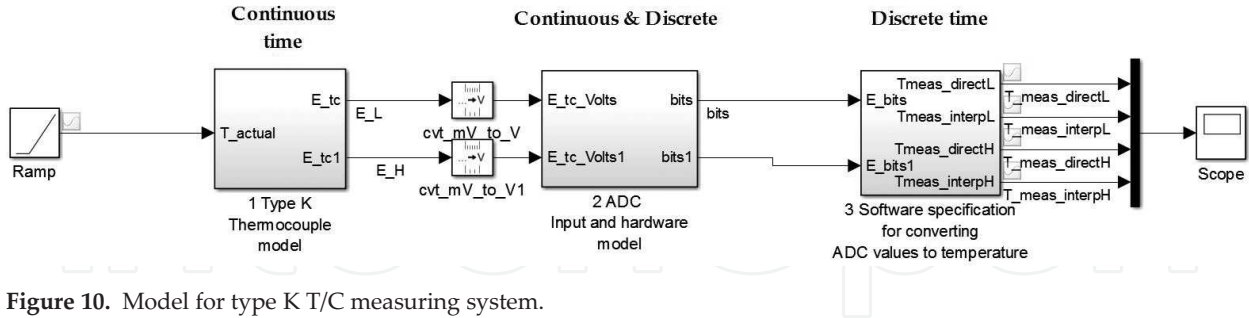


Figure 10. Model for type K T/C measuring system.

4.4. Simulation results

The linearization algorithms are simulated for the most linear and most nonlinear parts of the thermocouple characteristic curve. Least-squares, linear, quadratic, and quartic spline polynomial interpolation methods are applied for linearization over the intervals -270 to 0°C and 0 – 300°C , and the simulation results are given in **Figure 11**.

The Simulink model shown in **Figure 10** has given the simulation results as in **Figure 12**.

We have 2^{12} breakpoints for the 12-bit ADC conversion for the most accurate result. Consequently, in the inverse function derivation part of the simulation block, the direct and interpolated temperature values traced each other closely with an error of $-9.15\text{E}-05$ to $8.61\text{E}-05^{\circ}\text{C}$ in the 0 – 1370°C range, and $-1.51\text{E}-05$ to $1.51\text{E}-05^{\circ}\text{C}$ in the -270 to 0°C range.

4.5. Experimental results

During the test procedure, the measurements are carried out by the type K T/C, and the temperature is increased by 10°C over the interval of -270 to 1372°C . The values are saved as text file that can be evaluated in Excel. The same measurements are made by an identical T/C using the Fluke-725 process calibrator.

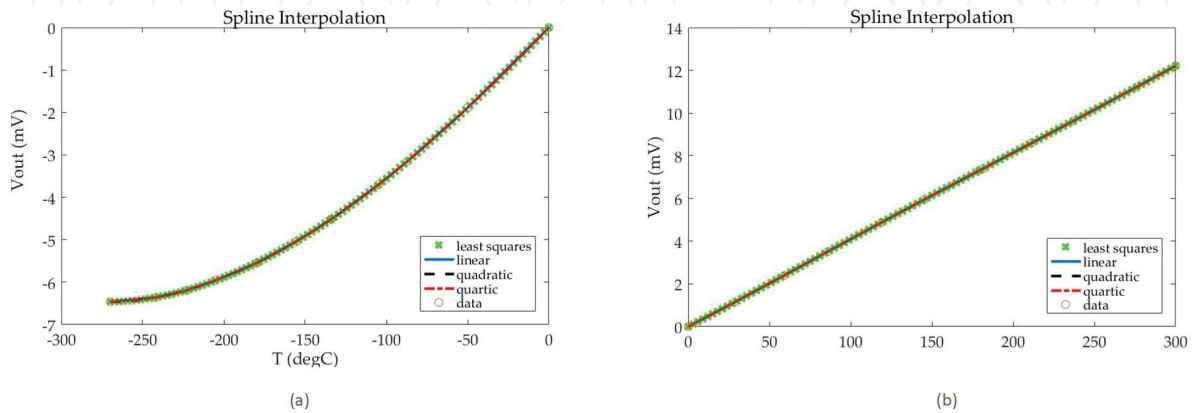


Figure 11. Spline polynomial interpolation results for (a) -270 to 0°C and (b) 0 – 300°C .

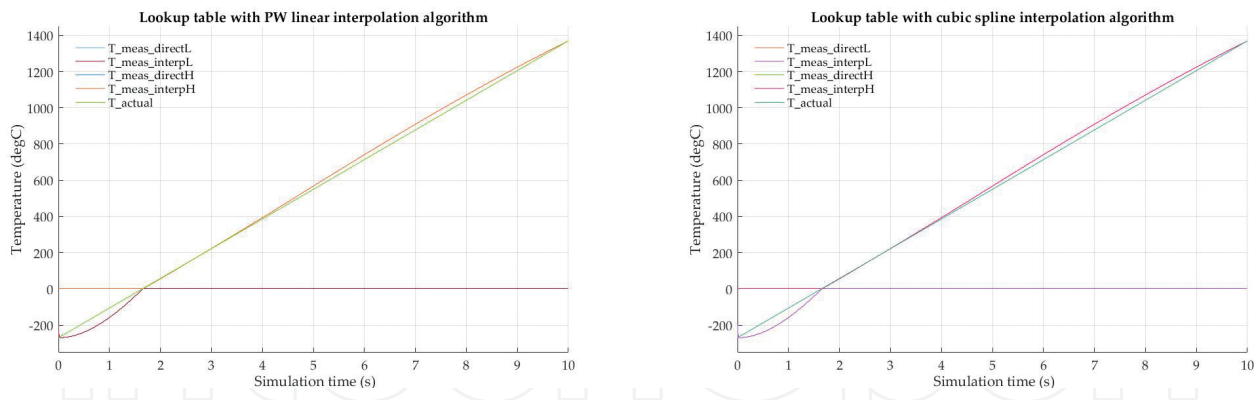


Figure 12. Linearization results for type K T/C by LUT with piecewise linear and cubic spline interpolation methods, respectively.

The obtained results are interpreted in terms of errors. The errors obtained from the piecewise linearization process with LUT for 28 breakpoints are given in **Figure 13**, and the error resulted in the positive temperature range is approximately -0.18 to 0.28°C . Compared to the linear function calculation error results declared by NIST ITS-90, -0.05 to 0.04°C for 0 – 500°C range, and -0.05 to 0.06°C for 500 – 1372°C range, the error range appears considerably higher than the “near-to-ideal” values, but piecewise linear approximation of the source data in the memory of the microcontroller optimizes the computation time.

The ninth degree polynomial inverse function is also applied to the measured T/C voltage over -200 to 1310°C range for the ranges described in **Table 3**. The measured values are incremented by about $21\ \mu\text{V}$ corresponding to 0.5°C . The results in **Figure 14** revealed that for the negative portion of the temperature measurements, the sixth degree polynomial “best fits” the function, whereas the ninth degree polynomial tracks the temperature values better than the sixth degree polynomial for the positive temperature range. So, we propose a mixed linearization method for type K T/C for the negative and positive portions of the full range.

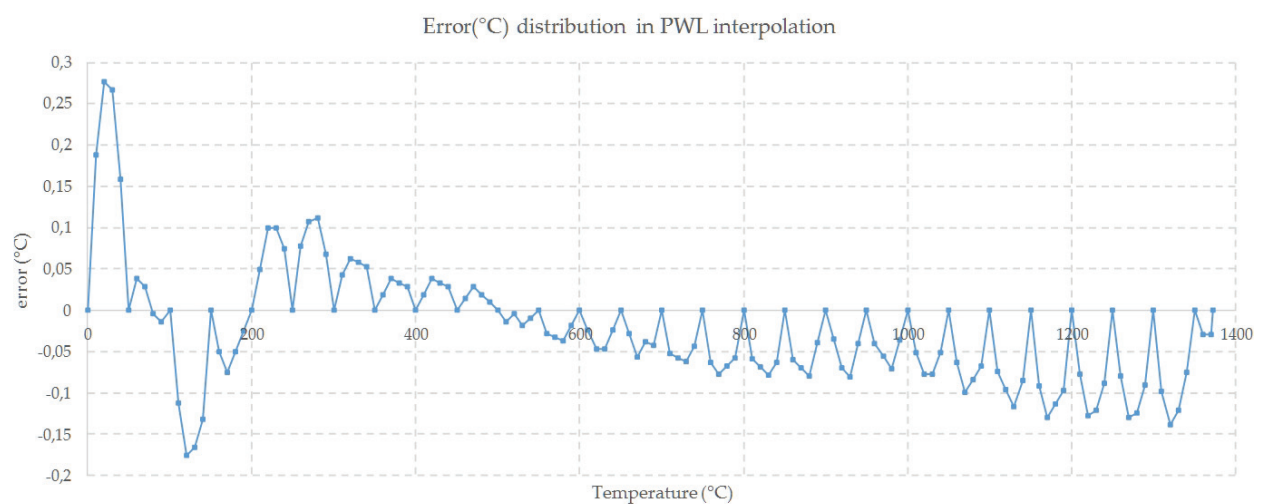


Figure 13. Error distribution for piecewise linear approximation method.

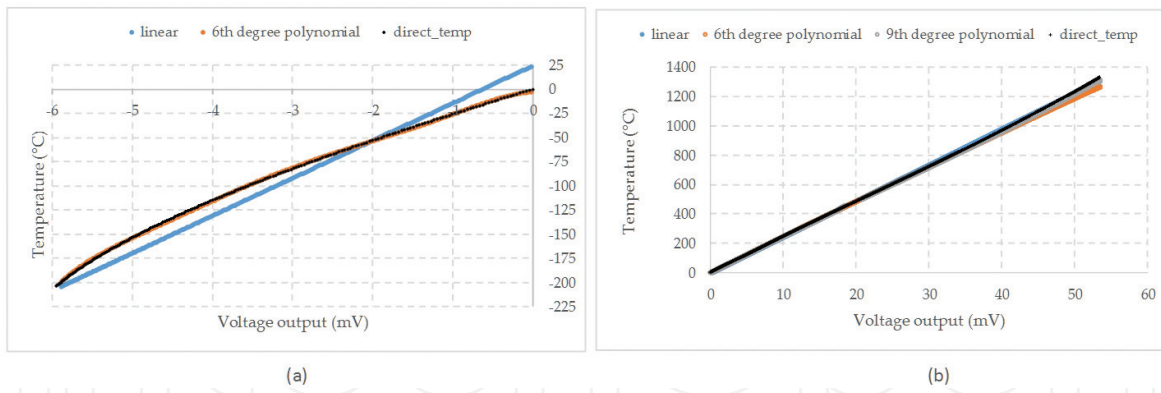


Figure 14. Linearization of T/C by polynomial for (a) negative and (b) positive temperature range.

5. Conclusion

In this chapter, we presented the measurement systems and linearization of sensors as well as the methods used in linearization. We presented the design and simulation process of a type K thermocouple as a case study. The linearization process was implemented with a 32-bit microcontroller. Type K thermocouple was connected to the Arduino Due controller board with an amplifier, and capacitors for the noise-suppression filter. Low-voltage output of the T/C, in the range of -6.458 to 54.886 mV, was amplified by an amplifier, and digitized by an internal 12-bit ADC. As the transmitters are low-power devices, a micro-power amplifier is utilized in the amplification of the output voltage of the T/C.

The cold-junction compensation of the thermocouple was realized by software using a thermistor fixed on an isothermal block. The actual type K thermocouple's voltage output in the nonlinear range was linearized by piecewise linear and polynomial interpolation methods, which were used to compute the temperature values between each breakpoint. Fewer number of breakpoints resulted in larger errors.

LUTs are used in a large range of applications including sensor linearization. The crucial points to consider are the number of LUT entries for better accuracy and the size of the LUT in design. Fewer number of LUT entries may cause serious errors in the measurement system, whereas larger number of entries may consume the memory of the controller.

In the polynomial approach for linearization, we propose 32-bit microcontrollers for better accuracy at the expense of cost compared to the small-embedded systems with limited computing ability and memory size. The calculation time for high-order polynomial equations may be too long for low-cost small-embedded systems with limited computational ability, but for implementation of the exact inverse sensor characteristic, the 32-bit controller Arduino Due is a reasonable solution.

Compared to the previous mixed signal works of the authors [11, 16], this case study offers a 32-bit ARM controller with high-computational ability and memory to run high-order polynomial interpolation method for linearization as well as higher accuracy in LUT with piecewise linearization. The ninth degree of polynomial was utilized in T/C linearization in [15], but the

range of linearization was limited to 0–200°C. We proposed a mixed polynomial linearization that best fits the negative and positive temperature ranges over the full range.

Author details

Dilşad Engin^{1*} and Mustafa Engin²

*Address all correspondence to: dilsad.engin@ege.edu.tr

1 Ege Higher Vocational School, Control and Automation Technology Department, Ege University, Bornova, İzmir, Turkey

2 Ege Higher Vocational School, Electronic Technology Department, Ege University, Bornova, İzmir, Turkey

References

- [1] Considine D. M., editor. *Process Instruments and Controls Handbook*. Third Edition. New York: McGraw-Hill; 1985. 1766 p. DOI: 10.1002/aic.690330127
- [2] Gorbunov S. F., Tsy-pin B. V. Linearization of calibration characteristics of capacitance pressure sensors. *Measurement Techniques*. 2011;**53**(10):1113–1117. DOI: 10.1007/s11018-011-9626-y
- [3] Ilić D., Butorac J., Ferković L. Temperature measurements by means of NTC resistors and a two-parameter approximation curve. *Measurement*. 2008;**41**(3):294–299. DOI: <http://dx.doi.org/10.1016/j.measurement.2006.11.007>
- [4] Chen C. Evaluation of resistance–temperature calibration equations for NTC thermistors. *Measurement*. 2009;**42**(7):1103–1111. DOI: 10.1016/j.measurement.2009.04.004
- [5] Radetić R., Pavlov-Kagadejev M., Milivojević N. The analog linearization of Pt100 working characteristic. *Serbian Journal of Electrical Engineering*. 2015;**12**(3):345–357. DOI: 10.2298/SJEE1503345R
- [6] Zhu M. J. A thermometer based on diverse types thermocouples and resistance temperature detectors. *J. Shanghai Jiaotong Univ. (Sci.)*. 2015;**20**(1):93–100. DOI: 10.1007/s12204-015-1594-y
- [7] NIST 1999, 9.10 National Institute of Standards. NIST ITS-90 Thermocouple Database [Internet]. 12/21/1999 [Updated: 11/05/2014 15:16:48]. Available from: <http://srdata.nist.gov/its90/main/> [Accessed: 2016]
- [8] Aly A. A., Abo El-Lail A. S. A novel circuit for thermocouple signals linearization using AD converter. *International Journal of Information Technology and Computer Science*. 2013;**5**(3):56–60. DOI: 10.5815/ijitcs.2013.03.07

- [9] Santhosh K. V., Roy B. K. Soft calibration technique with SVM for intelligent temperature measurement using thermocouple. In: 2014 International Conference on Advances in Electronics Computers and Communications; 10–11 Oct. 2014; Bangalore, India. IEEE; 2015. p. 1–4. DOI: 10.1109/ICAEECC.2014.7002380
- [10] Danisman K., Dalkiran I., Celebi F. V. Design of a high precision temperature measurement system based on artificial neural network for different thermocouple types. *Measurement*. 2006;**39**(8):695–700. DOI: 10.1016/j.measurement.2006.03.015
- [11] Engin M. Linearization and maximizing performance of the thermocouple using embedded microcontroller. *Journal of Computational Intelligence and Electronic Systems*. March 2014;**3**(1):29–35. DOI: 10.1166/jcies.2014.1071
- [12] Wang D., Song L., Zhang Z. Contact temperature measurement system based on tungsten-rhenium thermocouple. In: 2010 International Conference on Computer Application and System Modeling (ICCASM 2010); 22–24 Oct. 2010; Shanxi, Taiyuan. IEEE; 2010. pp. V12-660–V12-663. DOI: 10.1109/ICCASM.2010.5622449
- [13] Gerstenhaber M., O'Sullivan M. Condition thermocouples before digitizing. *Test & Measurement World*. 2011;**31**:23–24.
- [14] Wei G., Wang X., Sun J. Signal processing method with cold junction compensation for thermocouple. In: 2009 IEEE Instrumentation and Measurement Technology Conference; 5–7 May 2009; Singapore. IEEE; 2009. pp. 1458–1462. DOI: 10.1109/IMTC.2009.5168685
- [15] Sarma U., Boruah, P. K. Design and development of a high precision thermocouple based smart industrial thermometer with on line linearisation and data logging feature. *Measurement*. 2010;**43**(10):1589–1594. DOI: 10.1016/j.measurement.2010.09.003
- [16] Engin M., Engin D. Compensation of thermocouple nonlinearities with embedded system. In: Scientific Committee, Editorial (MECO 2016), editors. 2016 5th Mediterranean Conference on Embedded Computing (MECO); 12–16 June 2016; Bar, Montenegro. IEEE; 2016. pp. 234–237. DOI: 10.1109/MECO.2016.7525749
- [17] Bengtsson L. E. Lookup table optimization for sensor linearization in small embedded systems. *Journal of Sensor Technology*. 2012;**2**(4):177–184. DOI: 10.4236/jst.2012.24025
- [18] Wang X., Wei G., Sun J. Free knot recursive B-spline for compensation of nonlinear smart sensors. *Measurement*. 2011;**44**(5):888–894. DOI: 10.1016/j.measurement.2011.02.009
- [19] Eds. Croarkin C. and Tobias P. /NIST/SEMATECH. e-Handbook of Statistical Methods [Internet]. 01.06.2003 [Updated: 30.10.2013]. Available from: <http://www.itl.nist.gov/div898/handbook/> [Accessed: 05.01.2017].

- [20] Lin M., Carlsson J. G., Ge D., Shi J., Tsai J. A review of piecewise linearization methods. *Mathematical Problems in Engineering*. 2013;**2013**:1–8. DOI: 10.1155/2013/101376
- [21] Hillier F. S., Lieberman G. J. *Introduction to Operations*. 7th ed. New York, NY, USA: McGraw-Hill; 1995. 1237 p. ISBN 0-07-232169-5
- [22] Taha H. A. *Operations Research*. 8th ed. Upper Saddle River, New Jersey, NJ, USA: Pearson, Prentice Hall; 2007. 838 p. ISBN 0-13-188923-0

IntechOpen

IntechOpen

