

Article

Linear Interval Approximation for Smart Sensors and IoT Devices

Marin B. Marinov ^{1,*} , Nikolay Nikolov ², Slav Dimitrov ², Todor Todorov ² , Yana Stoyanova ² and Georgi T. Nikolov ¹

¹ Faculty of Electronic Engineering and Technologies, Technical University of Sofia, 1756 Sofia, Bulgaria; gnikolov@tu-sofia.bg

² Faculty of Industrial Technology, Technical University of Sofia, 1756 Sofia, Bulgaria; nickn@tu-sofia.bg (N.N.); slav.dimitrov@gmx.net (S.D.); tst@tu-sofia.bg (T.T.); yast@tu-sofia.bg (Y.S.)

* Correspondence: mbm@tu-sofia.bg

Abstract: In this work, we introduce and use an innovative approach for adaptive piecewise linear interval approximation of sensor characteristics, which are differentiable functions. The aim is to obtain a discreet type of inverse sensor characteristic, with a predefined maximum approximation error, with minimization of the number of points defining the characteristic, which in turn is related to the possibilities for using microcontrollers with limited energy and memory resources. In this context, the results from the study indicate that to overcome the problems arising from the resource constraints of smart devices, appropriate “lightweight” algorithms are needed that allow efficient connectivity and intelligent management of the measurement processes. The method has two benefits: first, low-cost microcontrollers could be used for hardware implementation of the industrial sensor devices; second, the optimal subdivision of the measurement range reduces the space in the memory of the microcontroller necessary for storage of the parameters of the linearized characteristic. Although the discussed computational examples are aimed at building adaptive approximations for temperature sensors, the algorithm can easily be extended to many other sensor types and can improve the performance of resource-constrained devices. For prescribed maximum approximation error, the inverse sensor characteristic is found directly in the linearized form. Further advantages of the proposed approach are: (i) the maximum error under linearization of the inverse sensor characteristic at all intervals, except in the general case of the last one, is the same; (ii) the approach allows non-uniform distribution of maximum approximation error, i.e., different maximum approximation errors could be assigned to particular intervals; (iii) the approach allows the application to the general type of differentiable sensor characteristics with piecewise concave/convex properties.

Keywords: approximation; IoT; linearization techniques; piecewise approximation; resource constrained devices; smart sensors



Citation: Marinov, M.B.; Nikolov, N.; Dimitrov, S.; Todorov, T.; Stoyanova, Y.; Nikolov, G.T. Linear Interval Approximation for Smart Sensors and IoT Devices. *Sensors* **2022**, *22*, 949. <https://doi.org/10.3390/s22030949>

Academic Editor: Juan V. Capella

Received: 13 November 2021

Accepted: 21 January 2022

Published: 26 January 2022

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

1.1. Resource-Constrained Smart Sensor Devices—Definitions

In recent years, there has been a strong increase in the interest of scientists in smart, energy-saving sensor technologies. Many promising developments have been made in both sensor production and wireless communications. The implementation of complex sensor systems is already possible due to the high degree of miniaturization of the series of classical measurement methods. Research in the field of smart sensors and the Internet of Things (IoT) has been growing. Unfortunately, there is still a lack of clarity in the terminology used in the specialized literature [1].

1.1.1. Smart Sensors

It is often considered that the term ‘smart sensors’ was created along with the term IoT. However, IEEE introduced a definition of smart transducers (sensors and actuators) in

1997 in its IEEE 1451 standard [2,3]. For transducers to be smart, IEEE 1451 requires them to be supplemented with specific hardware components (memory/processor) and additional functionality. The Transducer Electronic Data Sheet (TEDS) stored in the memory, contains data related to identification, calibration, and correction, as well as manufacturing information.

1.1.2. IoT Devices

The emergence of IoT and the advances in the field of microcontrollers and the Internet allowed the industry to build smart devices integrating complex sensor and communication functions as well as signal processing algorithms.

The definitions of xconventional sensors, smart sensors according to the IEEE 1451 standard, and smart IoT sensors can be summarized as follows.

According to NIST, a sensor is a device that transforms physical, chemical, and biological parameters into electrical signals and converts them into analog and/or digital data. A smart sensor, according to the IEEE 1451 standard, adds functionality to the NIST definition: the device can store data for sensor identification, calibration, correction, manufacturer-related information, and serves as a communication interface or processor.

Additionally, at the next stage, the smart sensor/IoT includes additional computing resources to process and interpret data and make decisions locally. Thus “smart” functions are added successively to the conventional sensor [1,4].

1.2. Basic Resources of a Typical Smart Sensor/IoT Device and Their Limitations

Despite the nuances and differences in the definitions and characteristics of various devices, their main distinguishing feature is their limited resources.

A typical autonomously powered smart sensor/IoT device includes two basic groups of resources. *Hardware resources*: data storage, processing, and communication resources, as well as power supply. *Software resources*: operating systems, system software, preloaded applications, and lightweight services. The optimal distribution of these resources is crucial in most applications [5,6].

1.2.1. Limitations Based on Hardware

(1) Energy and computational constraints:

Usually, IoT devices are battery-powered and typically use low-power processors. Therefore, computationally extensive algorithms that require significant resources cannot be transferred directly to such devices.

(2) Memory constraints: IoT devices are made with limited RAM and Flash memory compared to conventional digital systems (e.g., desktop computers, laptops, etc.). They usually use mobile lightweight software tools and operating systems. Therefore, calculation schemes require approaches for efficient memory use. However, the traditional computational algorithms are not specifically designed for this purpose and that is why such algorithms cannot be used directly for IoT devices.

1.2.2. Limitations Based on Software

(1) Firmware constraints: The sensor–IoT operating systems that are embedded in the devices have thin stacks of network protocols.

(2) Flexibility: Remote reprogramming may not always be possible for devices, as the operating system may not be able to receive and integrate a new code.

1.3. Sensor Characteristics Linearization

1.3.1. Sensor Transfer Functions

A sensor device generates an output signal in response to a specific measurand/stimulus (input) by activating one or more physical effects (ergo, Seebeck’s effect, piezoresistive effect, etc.). The output from a sensor is usually an electrical quantity (e.g., voltage, current, frequency), denoted here by y , while the input, x , is some physical quantity of interest (e.g.,

temperature, chemical concentration, pressure). If the input–output function is time-invariant, it is commonly called a static transfer function/characteristic or simply a transfer function.

The sensors are parts of a measuring system. The main function of these systems is to derive the unknown value x from the measured value y . Thus, the measurement system shall employ an inverse sensor transfer function $x = x(y)$ to obtain (calculate) the value of the stimulus x [7].

In the context of the system (control) theory, the dependence $y = y(x)$ represents the nonlinear scalar-valued characteristic of the sensor while its inverse $x = x(y)$ represents the *inverse transfer function of the sensor* (Figure 1a).

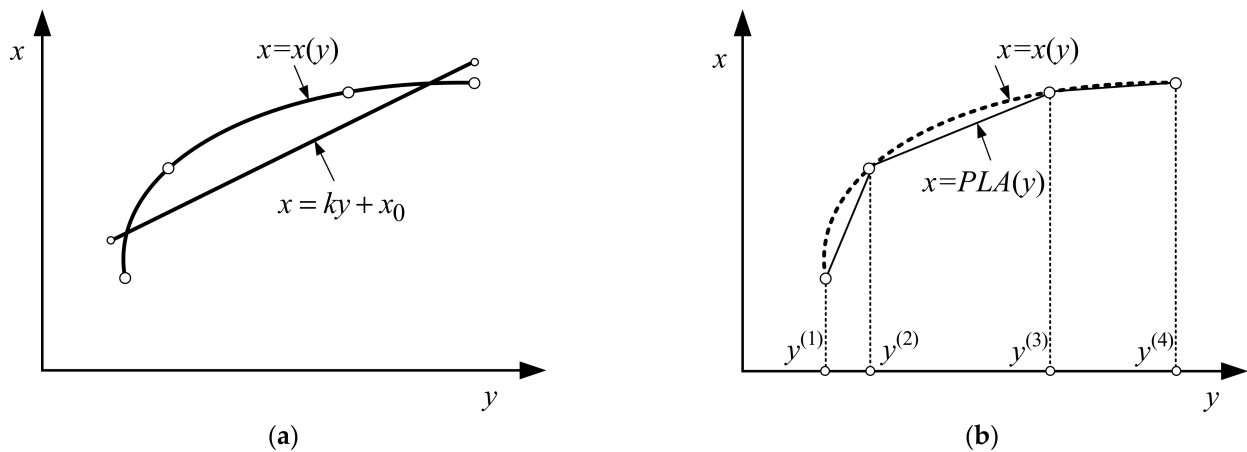


Figure 1. Graphical representation of inverse sensor transfer functions: (a) inverse sensor transfer function $x = x(y)$ and its linear approximation $x = ky + x_0$; (b) inverse sensor transfer function $x = x(y)$ and its piecewise linear approximation $x = PLA(y)$.

1.3.2. Approaches for Sensor Characteristics Linearization

Linearization is a major step in sensor signal processing. It can be considered as a nonlinearity correction between the output signal of the sensor and the associated measured quantity. The nonlinearity of the sensor can be reduced by using linearization schemes or a linearization algorithm [8]. Linearization methods can be grouped into three main classes:

- analog hardware linearization circuits;
- software-based algorithms for linearization;
- analog–digital mixed approaches [9].

An analog hardware method of linearization is performed by connecting a nonlinear analog circuit between the sensor and the ADC [10].

Software-based methods require the use of PCs or DSPs with significant computing resources [11]. Using these methods on low-cost controllers that support only simple mathematical operations is a major challenge due to the limited resources of the controller. Various software-based linearization methods are discussed in the literature. The most applied method is linearization based on the traditional look-up table (LUT), which can be implemented on any microcontroller [12]. In this method, the result of digital processing is used as an index in an array that stores the corrected data points [9,13].

The linearized sensor characteristics can greatly simplify the design, calibration, and measurement accuracy. This work elaborates further on the method presented in [14], used for the adaptive linear approximation of sensor characteristics.

In general, a scalar-valued inverse sensor transfer function is mathematically represented by a nonlinear regression model (e.g., exponential, power-law, or polynomial) constructed through least-squares error minimization (fit) of a statistically representative experimental data set. The identification of the inverse transfer function is often complicated and unreliable due to uncertainty in the choice of the analytic form of the function and

reductions in its parametrization. This of course leads to unwanted errors in the calibrated sensor response and hence should be avoided [15].

A way to avoid the uncertainty issues emerging from nonlinear regression-based identification of the inverse sensor response is provided by piecewise linearization (segmentation) of its transfer function. The latter is essentially a (simplified) polygonal approximation of $x = x(y)$ with algorithmic control of the approximation error (Figure 1b).

1.3.3. Piecewise Linear Approximation of Sensor Transfer Functions

Analog methods for linear approximation are still popular in engineering practice. However, thanks to advances in digital technology, the implementation of software methods can be completed easily, fast, at a low cost, and with a guarantee of high accuracy. Therefore, digital methods combined with software technologies solve linearization problems with better flexibility and efficiency.

Over the last two decades, different piecewise linear approximation (PLA) techniques have been proposed for the representation of sensor transfer functions. These techniques are reported in various application areas of sensor technology, such as data mining [16], data collection in wireless sensor networks [17,18], and activity recognition applications with wearable sensor nodes [19–21].

PLA algorithms for data volume reduction by approximating the signals collected in wireless sensor networks (WSN) are presented in [16,17]. These algorithms use buffers, and the calculation time depends on the length of the buffer. In [22], a fast alternative for time series segmentation is considered, in which the reduction in the calculation time significantly increases at the expense of the used memory. This is not an appropriate approach for resource-constrained architectures such as mobile sensor nodes. In contrast, the proposed approach has constant computation and memory complexity.

The SWAB (Sliding Window and Bottom-Up) algorithm is proposed in [16]. The approach is a combination of two steps, the first of which is offline segmentation. Based on this segmentation, in the second step, an online algorithm is created. An optimization of this algorithm, called mSWAB, is presented in [19]. A further improved implementation (called emSWAB) for limited resources, especially suitable for mobile sensor systems is presented in [20]. These approaches are based on linear interpolation, while the approach proposed in the paper is based on linear regression and can also be performed in offline and online modes.

In [23] another segmentation approach was introduced, based on polynomial least-squares approximation with polynomials of arbitrary order. From the presented experiments with this approach when first-order polynomials are used, results like the Sliding Window (SW) method are obtained in terms of the approximation error and the degree of data compression.

Another buffer-based PLA algorithm is presented in [24]. The proposed approach is a combination of either connected or disconnected segments. In contrast, the proposed algorithm for approximating the sensor characteristics uses connected segments, which is a more appropriate solution to be implemented in architectures with constraint memory and processing resources.

In addition to providing a reliable representation of transfer function, PLA approaches have two more attractive features:

- a. They create a simple and fast-to-identify characteristic shape of the transfer function (see [19,21,25]) that could be used for just-in-time classification purposes in large-scale control systems.
- b. They require a reduced amount of memory for data storage of the signal from the sensor device and hence possess significant economic potential for realization in smart sensors with reliable, low-cost microcontrollers. This is a crucial issue for two application domains: data mining when huge amounts of sensor data should be preprocessed under the constraint of short system-response times; and acquisition followed by wireless transmission of long-term sensor data.

1.4. Main Error Components of the Smart Sensor/IoT Device

As already mentioned, the goal of this study is to limit the memory footprint without violating system accuracy requirements. The basic structure of the smart sensor and its main components are shown in Figure 2. To achieve the required accuracy levels for a smart sensor, the main components of the compound must be considered. They can be divided into error components from the analog part δ_{AS} and components from the digital part δ_{DP} . Details are given in [26].

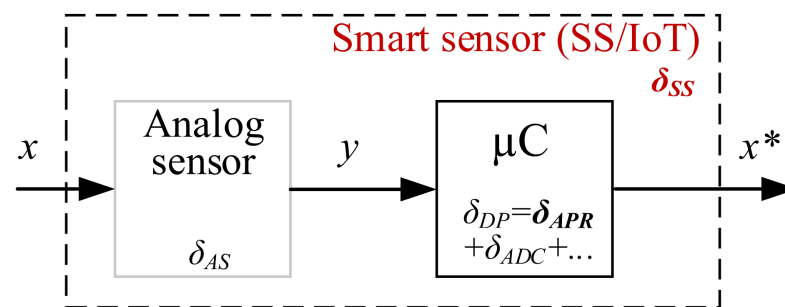


Figure 2. The basic structure and main error components of the smart sensor/IoT (after piecewise linear approximation and analog–digital conversion; x^* —microcontroller output value).

Here δ_{AS} —relative compound analog sensor error; δ_{DP} —relative compound digital processing error caused by the digital domain signal; $\delta_{APR} = \frac{\Delta^x}{x_{FS}}$ —relative approximation error; Δ^x —predefined absolute *approximation* error; x_{FS} —full scale of microcontroller output value x^* ; δ_{ADC} —relative analog–digital conversion error. Based on the considered serial structure, the total system error budget can be determined as:

$$\delta_{SS} = \sqrt{\delta_{AS}^2 + \delta_{DP}^2}.$$

The main components of the digital processing error δ_{DP} can also be attributed to δ_{ADC} and δ_{APR} . When an analog sensor and ADC are selected, the error levels δ_{AS} and δ_{ADC} cannot be affected. High-resolution ADCs are most often used in smart sensors and IoT implementation and therefore the δ_{ADC} the component is negligibly small which proves δ_{APR} to be a key component through which the total error levels can be managed, which will be discussed below.

To achieve typical goals in system design, where $\delta_{SS} \approx \delta_{AS}$, the aim is to have: $\delta_{AS} > (3 \div 10)\delta_{DP}$.

Published look-up tables, which are used in several applications, are often limited in the number of digits and this can be an additional source of rounding errors [26].

1.5. Proposed Approach

The proposed approach can be used for piecewise adaptive linearization of piecewise convex/concave sensor characteristics. As the name suggests, a typical piecewise convex/concave sensor response is characterized by a function consisting of a finite number of linked convex/concave segments with a strictly monotonous slope and hence a constant sign of the second derivative. Within the full measurement range of the sensor, the bounds of these segments can be identified from the condition of removing the second derivative, i.e., as the inflection points of the sensor characteristics.

The specificity of the proposed approach in comparison with the approach presented in [11] is illustrated in Figure 3.

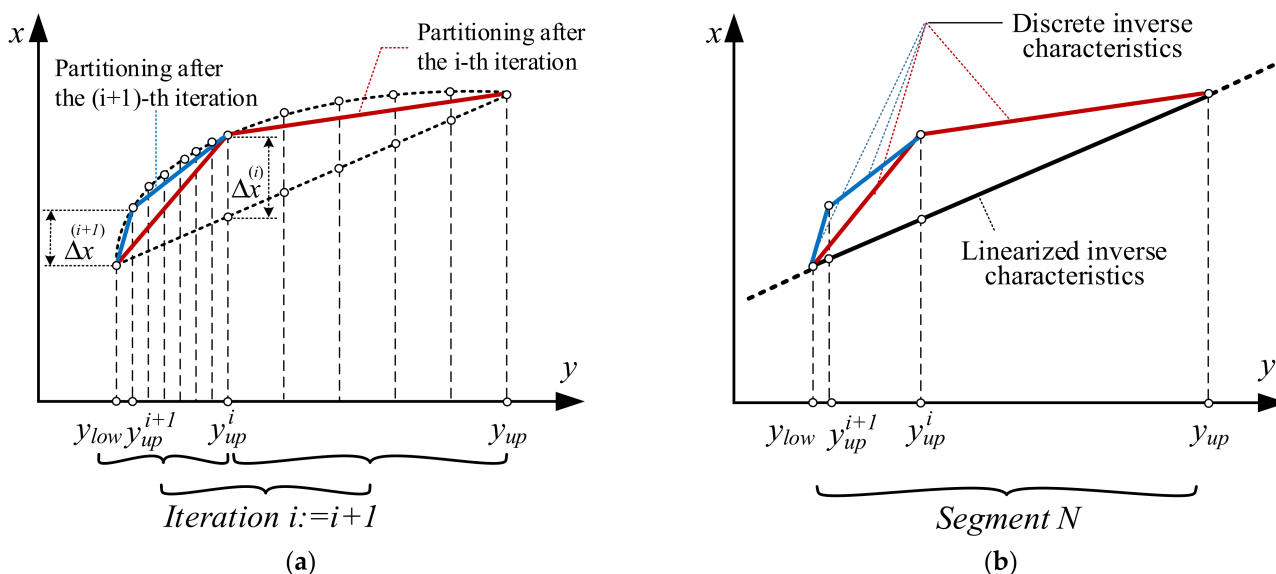


Figure 3. Graphical illustration of (a) the method in [11] and (b) the proposed approach.

An example of such a sensor characteristic is the relationship between resistance and temperature in the platinum temperature sensors set by the Callendar–van Dusen equation.

In [11,27,28], a linearization approach to sensor characteristics was developed in cases where the characteristic is a differentiable function. Its inverse feature is determined in a discrete form (freeform) based on an iterative algorithm with defined sensor characteristics and maximum approximation error. After that, based on the developed adaptive approach with a defined maximum approximation error, the discrete inverse sensor characteristic is replaced by a new discrete characteristic (also freeform), aimed at minimizing the number of points defining the characteristic, which in turn is related to the possibilities of using cheaper microcontrollers.

With the specified sensor, the transfer characteristic is of the type $y = y(x)$ and the first task is to determine the inverse transfer sensor characteristic $x = x(y)$. In cases where the inverse function cannot be obtained in an explicit form, due to the nonlinear relation between $y(x)$ and x , using a discrete linear change of x , the corresponding values of $y(x)$ are obtained, which upon receipt of the inverse function results in a discrete relationship between $y(x)$ and x , but with a nonlinear discrete change of argument s . This problem can be solved by the splitting method [3], the Newton method [29], as well as by the iterative algorithm proposed in [11,30]. As a result, given a maximum approximation error, the inverse sensor characteristic is obtained in a discrete form $x_i = x_i(y_i), i = \overline{1, n}$, with a uniform changing step of the argument s .

The second task is related to the approximation of the obtained discrete inverse sensor characteristic in the form of $x_i = x_i(y_i), i = \overline{1, n}$, a new characteristic $x_j = x_j(y_j), j = \overline{1, k}$ with a specified maximum approximation error and minimization of k . One approach for solving the problem was developed in [2,11].

The purpose of the present work is to develop a new approach for solving the task of linearization of sensor characteristics (differentiable functions with a non-changing sign of curvature), which simultaneously solves the tasks of finding the inverse sensor characteristic and its linearization.

2. The Analytical Framework of the Proposed Approach for Linearization of Sensor Characteristics

Let a differentiable sensor characteristic of the type $y = y(x), x \in [x_{A_i}, x_B], i = \overline{1, n-1}$, be given. For each such sensor, the characteristic is fulfilled $y'(x) \neq 0$, as the function $y = y(x)$ is strictly monotonous ($y'(x) > 0$ or $y'(x) < 0$).

In the case considered here, if the sign of curvature remains unchanged, one of the following two conditions should be satisfied: $\text{sign}(y''(x)) > 0$ or $\text{sign}(y''(x)) < 0$.

Under linear approximation of the function within the specified range, the error $\Delta_i^y(x)$ is determined by

$$\Delta_i^y(x) = y(x) - \left(y_{A_i} + \frac{y_B - y_{A_i}}{x_B - x_{A_i}} (x - x_{A_i}) \right) = y(x) - y_{A_i} - k_i(x - x_{A_i}), \quad (1)$$

where $k_i = \frac{y_B - y_{A_i}}{x_B - x_{A_i}}$ is the slope of the straight line A_iB (Figure 4).

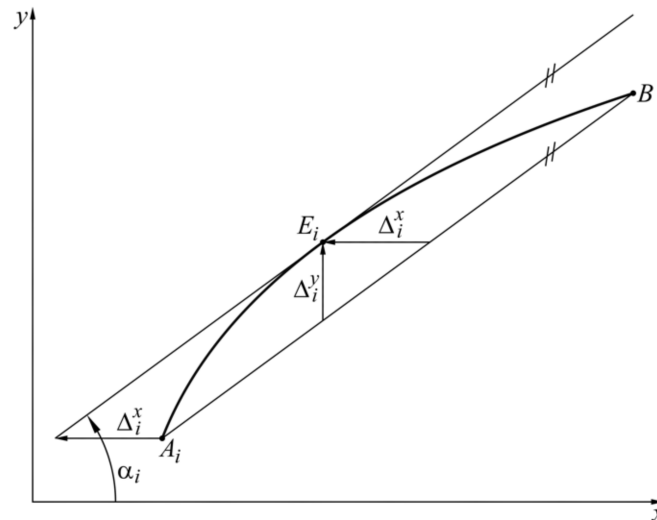


Figure 4. Graphical representation of the proposed approach.

From the necessary condition for extremum, $[\Delta_i^y(x_{E_i})]' = 0$ the following is obtained:

$$y'(x_{E_i}) - k_i = 0, \quad (2)$$

as the condition is also sufficient, $y''(x) \neq 0$.

The extremum always exists (Role Theorem [31]— $\Delta_i^y(x_{A_i}) = \Delta_i^y(x_B) = 0$), as at the same time it is unique—the function $(y'(x) - k_i)$ is monotonous and has different signs at the ends of the range.

With Equation (1), when $x = x_{E_i}$, the extremum value is obtained

$$\Delta_i^y(x_{E_i}) = y(x_{E_i}) - y_{A_i} - k_i(x_{E_i} - x_{A_i}).$$

The extremum of the error $\Delta_i^x(y)$ at a linear approximation of the inverse function $x(y)$ in the given range is determined by

$$\frac{\Delta_i^y(x_{E_i})}{\Delta_i^x(x_{E_i})} = -\tan\alpha_i = -k_i,$$

under which

$$\Delta_i^x(x_{E_i}) = \frac{\Delta_i^y(x_{E_i})}{-k_i} = x_{E_i} - x_{A_i} - \frac{y(x_{E_i}) - y_{A_i}}{k_i}. \quad (3)$$

In the context of this work, of practical interest is the following problem case—the maximum approximation error Δ^x is specified, the absolute value of which is less than the absolute value of the defined by Equation (3) error in the given range $[x_{A_i}, x_B]$. The goal is to find a subrange $[x_{A_i}, x_{A_{i+1}}]$, where $\Delta_i^x(x_{E_i}) = \Delta^x$ is fulfilled.

From Equations (2) and (3) it follows that

$$\Delta_i^x(x_{E_i}) = \Delta^x = x_{E_i} - x_{A_i} - \frac{y(x_{E_i}) - y_{A_i}}{k_i} = x_{E_i} - x_{A_i} - \frac{y(x_{E_i}) - y_{A_i}}{y'(x_{E_i})},$$

respectively,

$$y'(x_{E_i}) - \frac{y(x_{E_i}) - y_{A_i}}{x_{E_i} - x_{A_i} - \Delta^x} = 0. \quad (4)$$

The last equation defines x_{E_i} , then the right end $x_{A_{i+1}}$ of the range is determined by the equation

$$y(x_{A_{i+1}}) = y_{A_i} + y'(x_{E_i})(x_{A_{i+1}} - x_{A_i}). \quad (5)$$

3. Essence of the Proposed Approach for Linearization of Sensor Characteristics

Let a differentiable sensor characteristic in the form $y = y(x)$, $x \in [x_{A_i}, x_B]$, $i = \overline{1, n-1}$ be given as the sign of curvature does not change ($\text{sign}(y''(x)) > 0$ or $\text{sign}(y''(x)) < 0$). The aim is to obtain the inverse sensor characteristic $x(y)$ in the linearized form at a specified maximum approximation error Δ^x .

The approach is realized in the following way. With Equation (3) the extremum value of the Δ_1^x in the range $x \in [x_{A_1}, x_B]$ is determined. If $|\Delta_1^x| < |\Delta^x|$ then the function is approximated throughout the range with segment A_1B and the procedure ends. In the opposite case with Equations (4) and (5), the subrange $x \in [x_{A_1}, x_{A_2}]$, where $\Delta_1^x = \Delta^x$ is determined.

Then, through Equation (3), the extremum value of Δ_2^x in the range $x \in [x_{A_2}, x_B]$ is determined. If $|\Delta_2^x| < |\Delta^x|$, then the function is approximated throughout the range with the segment A_2B and the procedure ends, or otherwise through Equations (4) and (5) the subrange $x \in [x_{A_2}, x_{A_3}]$, where $\Delta_2^x = \Delta^x$, is determined, etc.

The inverse sensor characteristic $x(y)$ is approximated by the broken line $A_1A_2A_3 \dots A_{n-1}B \equiv A_n$, $A_i(y_{A_i}, x_{A_i})$, $i = \overline{1, n-1}$, $A_n(y_B, x_B)$, as

$$x(y) = x_{A_i} + (y - y_{A_i}) \frac{x_{A_{i+1}} - x_{A_i}}{y_{A_{i+1}} - y_{A_i}}, \quad y \in [y_{A_i}, y_{A_{i+1}}], \quad i = \overline{1, n-1}. \quad (6)$$

Generally, in all intervals, except for the last one, the maximum approximation errors are equal and equal to the maximum specified approximation error Δ^x . The error $\Delta_i^x(x)$ in each of the intervals is determined by the following equation:

$$\Delta_i^x(x) = x - x_{A_i} - (y(x) - y_{A_i}) \frac{x_{A_{i+1}} - x_{A_i}}{y_{A_{i+1}} - y_{A_i}}, \quad x \in [x_{A_i}, x_{A_{i+1}}], \quad i = \overline{1, n-1}. \quad (7)$$

Considering the next equation

$$\Delta_i^x[y(x)] = \Delta_i^x(x), \quad x \in [x_{A_i}, x_{A_{i+1}}], \quad i = \overline{1, n-1}, \quad (8)$$

and Equation (7) the error $\Delta_i^x(y)$ can be found.

Illustrative example: At a specified maximum approximation error Δ^x , find the linearized inverse characteristic of the characteristic $y(x) = a_1 + a_2(x - a_3)^2$, $x \in [x_A, x_B]$, when $\Delta^x = 2$, $a_1 = 2$, $a_2 = 0.02$, $a_3 = 2$, $x_A = 3$, $x_B = 51$.

Then $y(x) = 2 + 0.02(3 - 2)^2 = 2.02$, and $y'(x) = 0.04(x - 2)$, $y''(x) = 0.04$. So $y(x_{A_i}) = 2 + 0.02$.

From (4) follows

$$0.04(x_{E_i} - 2) - \frac{2 + 0.02(x_{E_i} - 2)^2 - 2.02}{x_{E_i} - 3 - 2} = 0,$$

respectively, $x_{E_i}^2 - 10x_{E_i} + 17 = 0$

The first of the two solutions 2.171572875 does not belong to the considered interval ($2.171572875 < x_{A_i} = 3$). Only the second solution 7.828427125 belongs to the considered interval. Therefore, $x_{E_i} = 7.828427125$.

From (5) follows

$$2 + 0.02(x_{A_{i+1}} - 2)^2 = 2.02 + 0.04(7.828427125 - 2)(x_{A_{i+1}} - 3),$$

Respectively,

$$x_{A_{i+1}}^2 - 15.65685425x_{A_{i+1}} + 37.97056275 = 0.$$

The first of the two solutions (3 and 12.65685425) corresponds to the starting point. Therefore, $x_{A_i} = 3$ and $x_{A_i} = 12.65685425$.

As a result of this approach, the inverse sensor characteristic $x(y)$ is approximated by the broken line $A_1A_2A_3B$, as shown in Table 1, where the coordinates of the points are given, also the corresponding intervals of the argument variation, the values of the argument in the extrema in the corresponding interval (as a result of the properties of the square parabola they are always in the middle of the interval) and the corresponding error Δ_i^x . The illustrative example is selected to enable good graphical visualization of the results shown in Figure 5.

Table 1. Results of the solution of the illustrative example.

A_1A_2	A_2A_3	A_3B
$A_1(3.000, 2.020)$	$A_2(12.657, 4.271)$	$A_3(30.314, 18.033)$
$A_2(12.657, 4.271)$	$A_3(30.314, 18.033)$	$B(51.000, 50.020)$
$x \in [3.000, 12.657]$	$x \in [12.657, 30.314]$	$x \in [30.314, 51.000]$
$x_{E_1} = 7.828$	$x_{E_2} = 21.485$	$x_{E_3} = 43.142$
$\Delta_1^x = 2$	$\Delta_2^x = 2$	$\Delta_3^x = 1.384$

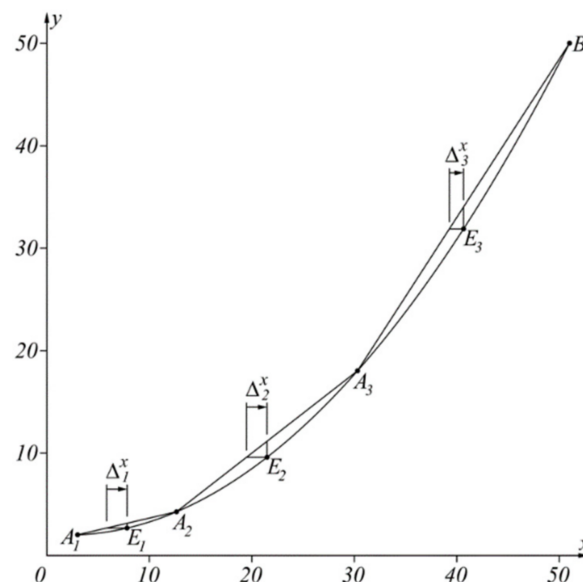


Figure 5. Graphical results of the solution of the illustrative example.

4. Linearization of the Inverse Sensor Characteristics of Temperature Sensors

The relation between the resistance and the temperature in the platinum temperature sensors is set by the equation of Callendar–van Dusen [32,33]. Using the proposed approach, the task of interval linearization of the inverse characteristic will be solved.

4.1. Linearization of the Inverse Sensor Characteristics of the Platinum Temperature Sensors in the Range $T \in [0, 661] ^\circ\text{C}$

The Callendar–van Dusen equation in the case under consideration is given by

$$R_T = R_0 \left(1 + AT + BT^2 \right), \quad (9)$$

where

$T, ^\circ\text{C}$ —temperature;

R_T —measured resistance at temperature T ;

R_0 —measured resistance at temperature $T = 0 ^\circ\text{C}$;

$A = 3.908310^{-3}$; $B = -5.77510^{-7}$ —coefficients according to ITS 90/IEC 60751 [34,35].

Under the denotations $R(T) = \frac{R_T}{R_0}$ and $a_3 = \frac{-A}{(2B)}$, Equation (9) takes the form

$$R(T) = 1 - \frac{A^2}{4B} + B \left[T + \frac{A}{2B} \right]^2 = 1 - \frac{A^2}{4B} + B(T - a_3)^2. \quad (10)$$

Differentiating (10) twice, concerning temperature T , leads to the first and second derivatives, respectively,

$$R'(T) = 2B(T - a_3), \quad (11)$$

$$R''(T) = 2B. \quad (12)$$

The graphs of function $R(T)$, its first derivative $R'(T)$, and its second derivative $R''(T)$ in the range $T \in [0, 661] ^\circ\text{C}$ defined by Equations (10)–(12), respectively, are shown in Figure 6.

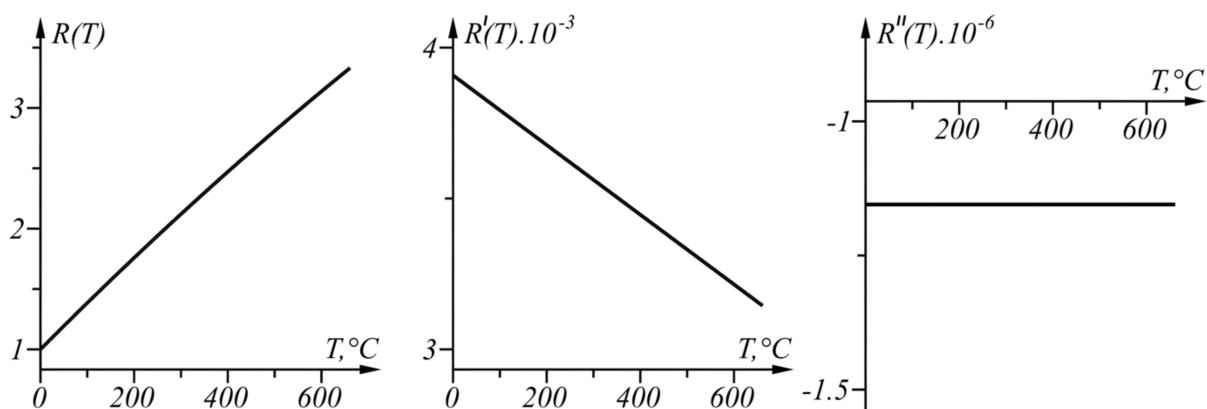


Figure 6. Graphs of the functions $R(T)$, $R'(T)$, and $R''(T)$ in the range $T \in [0, 661] ^\circ\text{C}$.

Since $R''(T)$ does not change its sign (the curve is concave in the considered range), the proposed approach for linear interval linearization can be applied because the function $T(R)$ can be found in the linearized form.

Equations (2)–(5) and (7), required for the approach, accordingly take the form:

$$R'(T_{E_i}) - k_i = 2B(T_{E_i} - a_3) - \frac{R_B - R_{A_i}}{T_B - T_{A_i}} = 0 \text{ or } T_{E_i} = \frac{T_{A_i} + T_B}{2}; \quad (13)$$

$$\begin{aligned} \Delta_i^T(T_{E_i}) &= \frac{\Delta_i^R(T_{E_i})}{-k_i} = T_{E_i} - T_{A_i} - \frac{R(T_{E_i}) - R_{A_i}}{k_i} = T_{E_i} - T_{A_i} - \frac{(R(T_{E_i}) - R_{A_i})(T_B - T_{A_i})}{R_B - R_{A_i}} \\ &= (T_B - T_{A_i}) \left(\frac{1}{2} - \frac{1}{4} \frac{T_B + 3T_{A_i} - 4a_3}{T_B + T_{A_i} - 2a_3} \right); \end{aligned} \quad (14)$$

$$T_{E_i} = \Delta^T + T_{A_i} + \sqrt{(\Delta^T + T_{A_i})^2 - T_{A_i}^2 - 2a_3 \Delta^T}; \quad (15)$$

$$T_{A_{i+1}} = 2T_{E_i} - T_{A_i}. \quad (16)$$

$$\Delta_i^T(T) = T - T_{A_i} - (R(T) - R_{A_i}) \frac{T_{A_{i+1}} - T_{A_i}}{R_{A_{i+1}} - R_{A_i}}, \quad x \in [T_{A_i}, T_{A_{i+1}}], \quad i = \overline{1, n-1}. \quad (17)$$

The numerical results from the application of the approach, when $\Delta^T = -0.375$ °C in the given temperature range $T_{A_1} = 0$ °C, $T_B = 661$ °C, are shown in Table 2 and in graphical form in Figure 7. The graph of the function $\Delta^T(T)$ from (17) is shown in Figure 8.

Table 2. Results in the range $T \in [0, 661]$ °C.

A_1A_2	A_2A_3	A_3A_4	A_4A_5
$A_1(0,1)$	$A_2(100.007, 1.385)$	$A_3(198.514, 1.753)$	$A_4(295.521, 2.105)$
$A_2(100.007, 1.385)$	$A_3(198.514, 1.753)$	$A_4(295.521, 2.105)$	$A_5(391.028, 2.440)$
$T \in [0, 100.007]$	$T \in [100.007, 198.514]$	$T \in [198.514, 295.521]$	$T \in [295.521, 391.028]$
$T_{E_1} = 50.0035$	$T_{E_2} = 149.261$	$T_{E_3} = 247.018$	$T_{E_4} = 343.275$
$\Delta_1^T = -0.375$	$\Delta_2^T = -0.375$	$\Delta_3^T = -0.375$	$\Delta_4^T = -0.375$
A_5A_6	A_6A_7	A_7A_8	
$A_5(391.028, 2.440)$	$A_6(485.035, 2.760)$	$A_7(577.543, 3.065)$	
$A_6(485.035, 2.760)$	$A_7(577.543, 3.065)$	$A_8 \equiv B(661, 3.331)$	
$T \in [391.028, 485.035]$	$T \in [485.035, 577.543]$	$T \in [577.543, 661]$	
$T_{E_5} = 438.032$	$T_{E_6} = 531.289$	$T_{E_7} = 619.271$	
$\Delta_5^T = -0.375$	$\Delta_6^T = -0.375$	$\Delta_7^T = -0.315$	

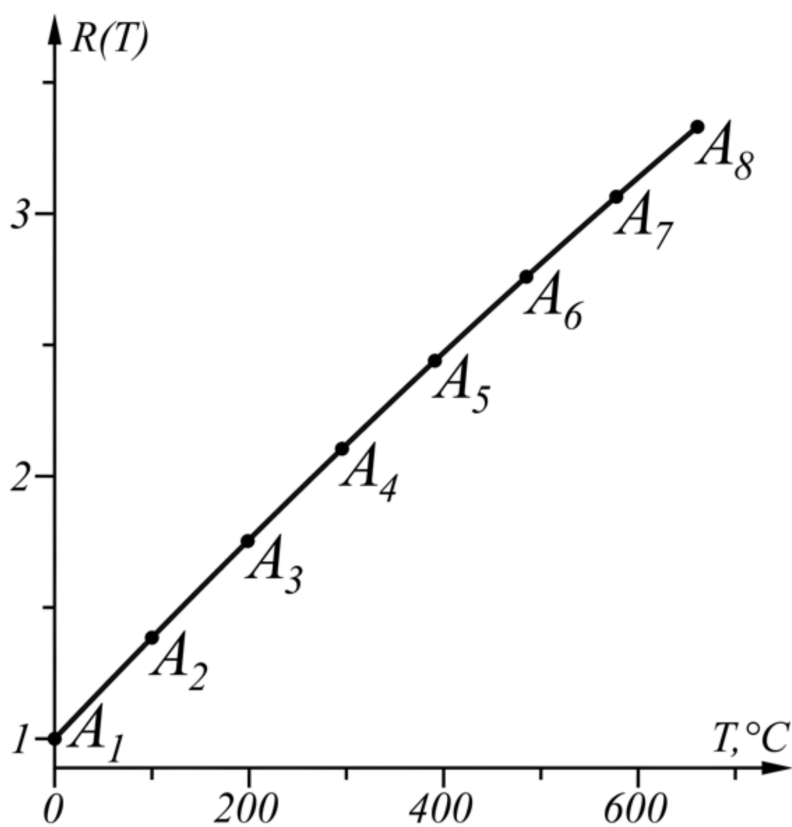


Figure 7. Graphical results of the solution in the range $T \in [0, 661]$ °C.

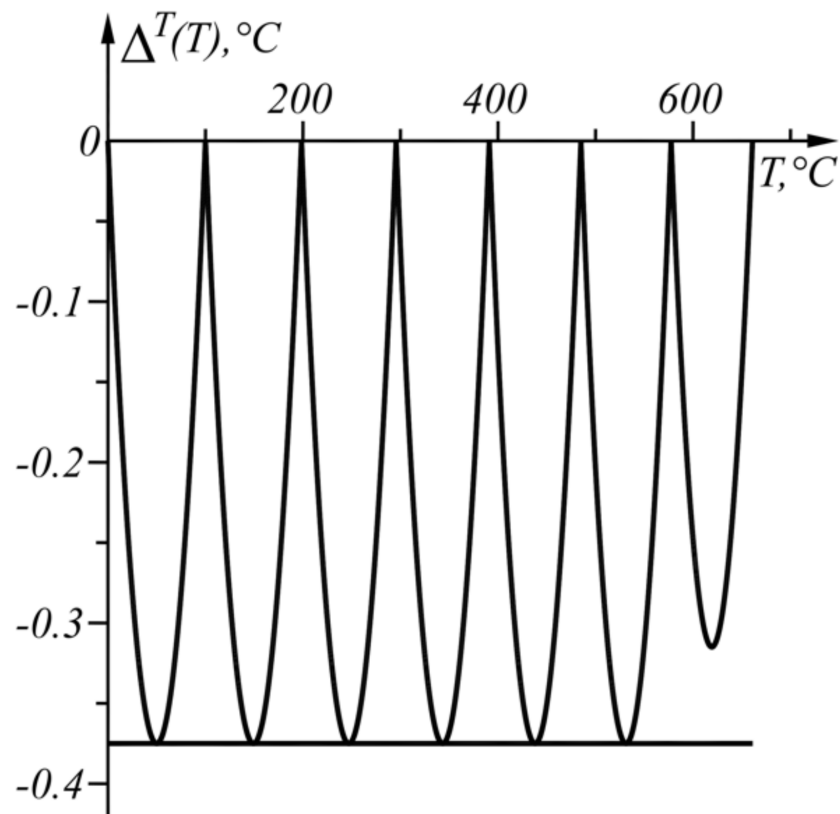


Figure 8. Graph of the function $\Delta^T(T)$ in the range $T \in [0, 661]$ °C.

4.2. Linearization of the Inverse Sensor Characteristics of the Platinum Temperature Sensors in the Range $T \in [-200, 0]$ °C

The Callendar–van Dusen equation when considering the case is given by

$$R_T = R_0 \left(1 + AT + BT^2 - 100CT^3 + CT^4 \right), \quad (18)$$

where

T , °C—temperature;

R_T —measured resistance at temperature T ;

R_0 —measured resistance at temperature $T = 0$ °C;

$A = 3.908310^{-3}$; $B = -5.77510^{-7}$; $C = -4.18310^{-12}$ —coefficients according to ITS 90/IEC 60751 [34].

Under the denotation, $R(T) = \frac{R_T}{R_0}$, the Equation (18) takes the form

$$R(T) = 1 + AT + BT^2 - 100CT^3 + CT^4. \quad (19)$$

Differentiating (19) twice, concerning the temperature T , leads to the first and second derivatives, respectively,

$$R'(T) = A + 2BT - 300CT^2 + 4CT^3; \quad (20)$$

$$R''(T) = 2B - 600CT + 12CT^2. \quad (21)$$

The graphs of $R(T)$, $R'(T)$, and $R''(T)$ are shown in Figure 9.

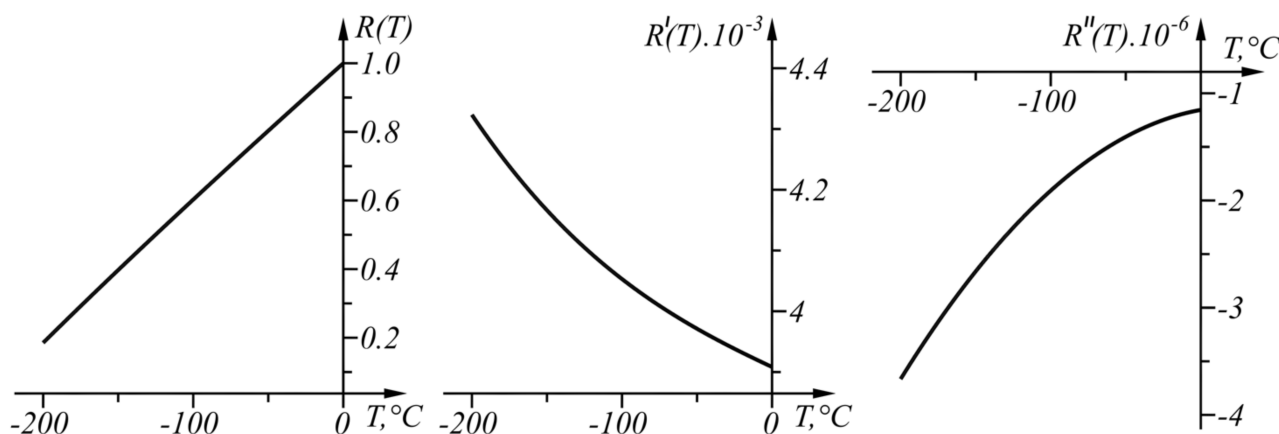


Figure 9. Graphs of the functions $R(T)$, $R'(T)$, and $R''(T)$ in the range $T \in [-200, 0]$ °C.

Since $R''(T)$ does not change its sign (the curve is concave in the range considered), the proposed approach for linear interval linearization can be applied because function $T(R)$ can be found in the linearized form.

Equations (2)–(5) needed for the implementation of the approach take the form, respectively,

$$A + 2BT_{E_i} - 300CT_{E_i}^2 + 4CT_{E_i}^3 - \frac{R_B - R_{A_i}}{T_B - T_{A_i}} = 0, \quad (22)$$

$$\Delta_i^T(T_{E_i}) = \frac{\Delta_i^R(T_{E_i})}{-k_i} = T_{E_i} - T_{A_i} - \frac{R(T_{E_i}) - R_{A_i}}{k_i} = T_{E_i} - T_{A_i} - (A(T_{E_i} - T_{A_i}) + B(T_{E_i}^2 - T_{A_i}^2) - 100C(T_{E_i}^3 - T_{A_i}^3) + C(T_{E_i}^4 - T_{A_i}^4)) \frac{T_B - T_{A_i}}{R_B - R_{A_i}}. \quad (23)$$

$$R'(T_{E_i}) - \frac{R(T_{E_i}) - R_{A_i}}{T_{E_i} - T_{A_i} - \Delta_i^T} = A + 2BT_{E_i} - 300CT_{E_i}^2 + 4CT_{E_i}^3 - \frac{A(T_{E_i} - T_{A_i}) + B(T_{E_i}^2 - T_{A_i}^2) - 100C(T_{E_i}^3 - T_{A_i}^3) + C(T_{E_i}^4 - T_{A_i}^4)}{T_{E_i} - T_{A_i} - \Delta_i^T} = 0. \quad (24)$$

$$1 + AT_{A_{i+1}} + BT_{A_{i+1}}^2 - 100CT_{A_{i+1}}^3 + CT_{A_{i+1}}^4 = 1 + AT_{A_i} + BT_{A_i}^2 - 100CT_{A_i}^3 + CT_{A_i}^4 + (A + 2BT_{E_i} - 300CT_{E_i}^2 + 4CT_{E_i}^3)(T_{A_{i+1}} - T_{A_i}). \quad (25)$$

The proposed approach will be applied when $\Delta^T = -0.375$ °C in the temperature range $T_{A_1} = 0$ °C, $T_B = -200$ °C.

Using Equation (22), the value of the argument T'_{E_1} is determined, under which the function $\Delta_1^T(T_{E_1})$ (23) has an extremum in the range $T_B = -200$ °C $< T < T_{A_1} = 0$ °C.

The third power Equation (22) has two complex and one real root $T_{E_1} = -110.583$ °C. When $T_{E_1} = -110.583$ °C, $\Delta_1^T = -2.472$ °C is obtained through Equation (23). As $|\Delta_1^T| > |\Delta^T|$, Equations (24) and (25) determine the subrange $T \in [T_{A_2}, T_{A_1}]$ where $\Delta_1^T = \Delta^T$. The fourth power Equation (24) has two real and two complex roots as the real root $T_{E_1} = -47.977$ °C belongs to the range $[T_B, T_{A_1}]$. The Equation (25) is also a fourth power equation and has two real and two complex roots as one of the real roots is T_{A_1} and the other one is $T_{A_2} = -92.292$ °C.

Then the same steps are applied to the range $T_B = -200$ °C $< T < T_{A_2} = -92.292$ °C. The Equation (22) has two complex and one real root $T_{E_2} = -149.314$ °C. When $T_{E_2} = -149.314$ °C, $\Delta_2^T = -0.915$ °C is obtained by Equation (23). As $|\Delta_2^T| > |\Delta^T|$, Equations (24) and (25) define the subrange $T \in [T_{A_3}, T_{A_2}]$ where $\Delta_2^T = \Delta^T$. Equation (24) has two real and two complex roots as one of the real roots belongs to the range $[T_B, T_{A_2}]$ is $T_{E_2} = -130.207$ °C. Equation (25) has two real and two complex roots as one of the real roots is T_{A_2} and the other one is $T_{A_3} = -165.17$ °C.

Then, the same consecution is applied to the range $T_B = -200\text{ }^\circ\text{C} < T < T_{A_3} = -165.178\text{ }^\circ\text{C}$. The real root of Equation (22) is $T_{E_3} = -182.909\text{ }^\circ\text{C}$. When $T_{E_3} = -182.909\text{ }^\circ\text{C}$, Equation (23) $\Delta_{T_3} = -0.117\text{ }^\circ\text{C}$ is obtained. As $|\Delta_T| < |\Delta^T|$, the procedure ends.

The numerical results of the application of the approach when $\Delta^T = -0.375\text{ }^\circ\text{C}$ are shown in Table 3 and in graphical form in Figure 10. The graph of the function $\Delta^T(T)$ (17) is shown in Figure 11.

Table 3. Results in the range $T \in [-200, 0]\text{ }^\circ\text{C}$.

A_1A_2	A_2A_3	A_3A_4
$A_1(0, 1)$	$A_2(-92.292, 0.634)$	$A_3(-165.178, 0.334)$
$A_2(-92.292, 0.634)$	$A_3(-165.178, 0.334)$	$A_4 \equiv B(-200, 0.185)$
$T \in [-92.292, 0]$	$T \in [-165.178, -92.292]$	$T \in [-200, -165.178]$
$T_{E_1} = -47.977$	$T_{E_2} = -130.207$	$T_{E_3} = -182.909$
$\Delta_{T_1} = -0.375$	$\Delta_{T_2} = -0.375$	$\Delta_{T_3} = -0.117$

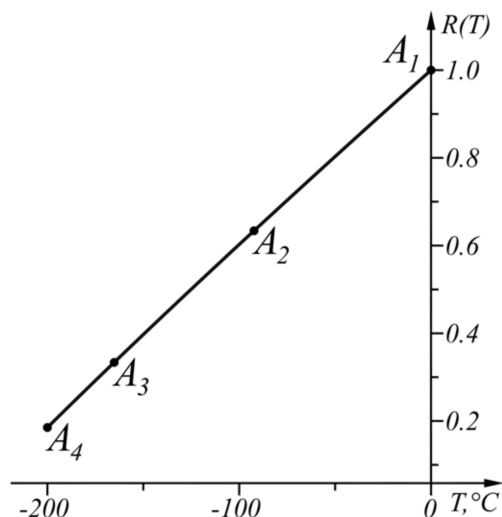


Figure 10. Graphical results of the solution in the range $T \in [-200, 0]\text{ }^\circ\text{C}$.

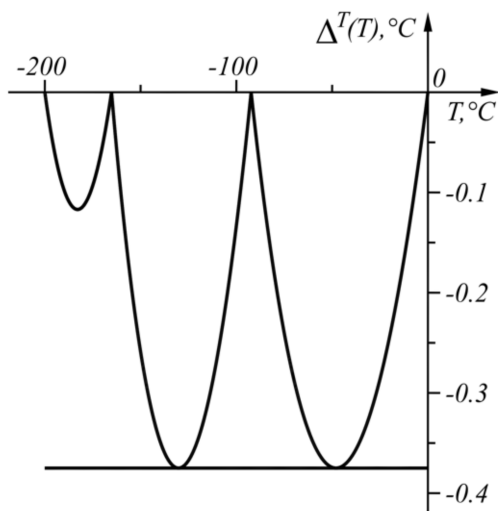


Figure 11. Graph of the function $\Delta^T(T)$ in the range $T \in [-200, 0]\text{ }^\circ\text{C}$.

4.3. Microcontroller Implementation of the Inverse Sensor Characteristics Linearization of Platinum Temperature Sensors in the Range $T \in [0, 661]^\circ\text{C}$

This design uses a Lolin 32 development board based on Espressif Systems' ESP32 microcontroller. Its main features include integrated Wi-Fi, flash memory, and USB communication, and is often used for IoT designs [36].

The linearization is performed by segmentation, in which the range is divided into the corresponding number of segments (from 1 to End) to guarantee the requirement for the maximum approximation error. The linearization algorithm for low-cost microcontrollers is presented in a pseudo-code form below (see Algorithm 1).

Algorithm 1 Linearization

Input: Measured RTD resistance, **R_T**, floating-point type

Output: Calculated temperature, **Temperature**, floating-point type

Initialization: defining the coordinates of the points bounding each interval $R_{a_1}, R_{a_2} \dots R_{a_{End}}$ and $T_{a_1}, T_{a_2} \dots T_{a_{End}}$

1: Determine the interval (n) in which the measured resistance is located, and whether it is in the respective temperature range

If $((R_T > R_{a_n}) \text{ and } (R_T \leq R_{a_{(n+1)}))$

2: Calculate the temperature according to the formula

Temperature = $(R_T - R_{a_n}) * ((T_{a_{(n+1)}} - T_{a_n}) / (R_{a_{(n+1)}} - R_{a_n})) + T_{a_n}$;

3: Return **Temperature**

The algorithm is implemented as a function in the Lolin D32 development system for the ESP32 microcontroller. When variables and constants are represented with a floating-point data type, the memory footprint of the function is 36 bytes of flash and 8 bytes of the system's RAM. The function is compiled with both Os and g3 optimizations—for size and debug information, respectively.

For comparison, in an implementation based on an optimized lookup table using an unsigned integer, the occupied program memory is about 3k bytes and 8 bytes of RAM [37].

4.4. An Illustrative Example of Linearization of the Inverse Sensor Characteristics of K-Type Thermocouples in the Temperature Range $T \in [-200, 0]^\circ\text{C}$ and Maximum Approximation Error $\Delta^T = 0.04^\circ\text{C}$

The thermoelectric voltage E in microvolts, as a function of temperature T in degrees Celsius, is defined by:

$$E(t) = \sum_1^{10} c_i T^i, \quad (26)$$

where the coefficients c_i are specified in NIST ITS-90 [38].

The graphs of function $E(T)$, its first derivative $E'(T)$, and its second derivative $E''(T)$ in the range, $T \in [-270, 0]^\circ\text{C}$, are shown in Figure 12.

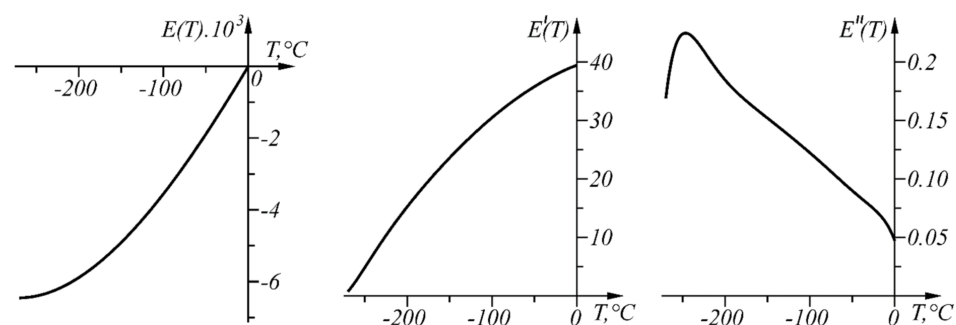


Figure 12. Graphs of the functions $E(T)$, $E'(T)$, and $E''(T)$ in the range $T \in [-270, 0]^\circ\text{C}$.

Since $E''(T)$ does not change sign (the curve is convex in the considered range), the proposed approach for linear interval linearization can be applied and the sensor characteristic can be determined in linearized form.

The numerical results from the application of the approach, when $\Delta^T = 0.04$ °C in the given temperature range $T_{A_1} = 0$ °C, $T_B = T_{A_{48}} = -200$ °C, are shown in Table 4 and in graphical form in Figure 13. The graph of the function $\Delta^T(T)$ is shown in Figure 14.

Table 4. Results in the range $T \in [-200, 0]$ °C.

A_1A_2	A_2A_3	A_3A_4	A_4A_5
$A_1(0, 0)$	$A_2(-14.596, -569.9626)$	$A_3(-27.626, -1066.926)$	$A_4(-39.819, -1520.303)$
$A_2(-14.596, -569.9626)$	$A_3(-27.626, -1066.926)$	$A_4(-39.819, -1520.303)$	$A_5(-51.342, -1937.358)$
$T \in [-14.596, 0]$	$T \in [-27.626, -14.596]$	$T \in [-39.819, -27.626]$	$T \in [-51.342, -39.819]$
$T_{E_1} = -7.494$	$T_{E_2} = -21.181929$	$T_{E_3} = -33.769$	$T_{E_4} = -45.620$
$\Delta^T = 0.04$	$\Delta^T = 0.04$	$\Delta^T = 0.04$	$\Delta^T = 0.04$
A_5A_6	A_6A_7	A_7A_8	A_8A_9
$A_5(-51.342, -1937.358)$	$A_6(-62.267, -2321.642)$	$A_7(-72.653, -2676.126)$	$A_8(-82.554, -3003.567)$
$A_6(-62.267, -2321.642)$	$A_7(-72.653, -2676.126)$	$A_8(-82.554, -3003.567)$	$A_9(-92.019, -3306.423)$
$T \in [-62.267, -51.342]$	$T \in [-72.653, -62.267]$	$T \in [-82.554, -72.653]$	$T \in [-92.019, -82.554]$
$T_{E_5} = -56.839$	$T_{E_6} = -67.489$	$T_{E_7} = -77.628$	$T_{E_8} = -87.308$
$\Delta^T = 0.04$	$\Delta^T = 0.04$	$\Delta^T = 0.04$	$\Delta^T = 0.04$
A_9A_{10}	$A_{10}A_{11}$	$A_{11}A_{12}$	$A_{12}A_{13}$
$A_9(-92.019, -3306.423)$	$A_{10}(-101.091, -3586.812)$	$A_{11}(-109.801, -3846.531)$	$A_{12}(-118.178, -4087.117)$
$A_{10}(-101.091, -3586.812)$	$A_{11}(-109.801, -3846.531)$	$A_{12}(-118.178, -4087.117)$	$A_{13}(-126.243, -4309.902)$
$T \in [-101.091, -92.019]$	$T \in [-109.801, -101.091]$	$T \in [-118.178, -109.801]$	$T \in [-126.243, -118.178]$
$T_{E_9} = -96.573$	$T_{E_{10}} = -105.461$	$T_{E_{11}} = -114.003$	$T_{E_{12}} = -122.223$
$\Delta^T = 0.04$	$\Delta^T = 0.04$	$\Delta^T = 0.04$	$\Delta^T = 0.04$
$A_{13}A_{14}$	$A_{14}A_{15}$	$A_{15}A_{16}$	$A_{16}A_{17}$
$A_{13}(-126.243, -4309.902)$	$A_{14}(-134.015, -4516.065)$	$A_{15}(-141.507, -4706.668)$	$A_{16}(-148.733, -4882.674)$
$A_{14}(-134.015, -4516.065)$	$A_{15}(-141.507, -4706.668)$	$A_{16}(-148.733, -4882.674)$	$A_{17}(-155.704, -5044.969)$
$T \in [-134.015, -126.243]$	$T \in [-141.507, -134.015]$	$T \in [-148.733, -141.507]$	$T \in [-155.704, -148.733]$
$T_{E_{13}} = -130.140$	$T_{E_{14}} = -137.770$	$T_{E_{15}} = -145.129$	$T_{E_{16}} = -152.227$
$\Delta^T = 0.04$	$\Delta^T = 0.04$	$\Delta^T = 0.04$	$\Delta^T = 0.04$
$A_{17}A_{18}$	$A_{18}A_{19}$	$A_{19}A_{20}$	$A_{20}A_{21}$
$A_{17}(-155.704, -5044.969)$	$A_{18}(-162.428, -5194.366)$	$A_{19}(-168.912, -5331.618)$	$A_{20}(-175.162, -5457.422)$
$A_{18}(-162.428, -5194.366)$	$A_{19}(-168.912, -5331.618)$	$A_{20}(-175.162, -5457.422)$	$A_{21}(-181.181, -5572.431)$
$T \in [-162.428, -155.704]$	$T \in [-168.912, -162.428]$	$T \in [-175.162, -168.912]$	$T \in [-181.181, -175.162]$
$T_{E_{17}} = -159.073$	$T_{E_{18}} = -165.677$	$T_{E_{19}} = -172.043$	$T_{E_{20}} = -178.177$
$\Delta^T = 0.04$	$\Delta^T = 0.04$	$\Delta^T = 0.04$	$\Delta^T = 0.04$
$A_{21}A_{22}$	$A_{22}A_{23}$	$A_{23}A_{24}$	$A_{24}A_{25}$
$A_{21}(-181.181, -5572.431)$	$A_{22}(-186.973, -5677.260)$	$A_{23}(-192.540, -5772.495)$	$A_{24}(-197.884, -5858.704)$
$A_{22}(-186.973, -5677.260)$	$A_{23}(-192.540, -5772.495)$	$A_{24}(-197.884, -5858.704)$	$A_{25} \equiv B(-200, -5891.404)$
$T \in [-186.973, -181.181]$	$T \in [-192.540, -186.973]$	$T \in [-197.884, -192.540]$	$T \in [-200, -197.884]$
$T_{E_{21}} = -184.083$	$T_{E_{22}} = -189.762$	$T_{E_{23}} = -195.217$	$T_{E_{24}} = -198.943$
$\Delta^T = 0.04$	$\Delta^T = 0.04$	$\Delta^T = 0.04$	$\Delta^T = 0.00666$

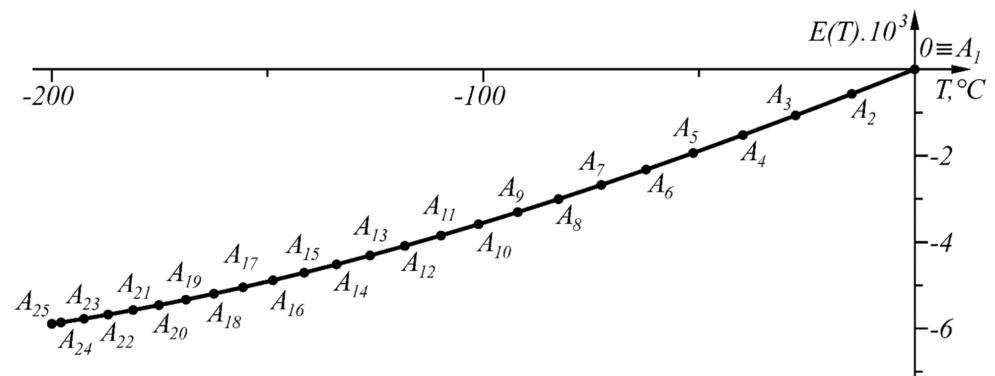


Figure 13. Graphical results of the solution in the range $T \in [-200, 0] \text{ } ^\circ\text{C}$.

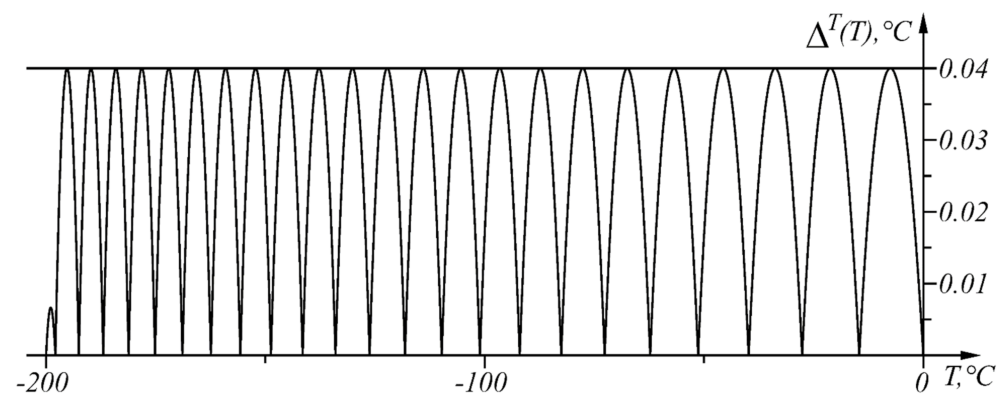


Figure 14. Graph of the function $\Delta^T(T)$ in the range $T \in [-200, 0] \text{ } ^\circ\text{C}$.

The linearization of the k-type thermocouple in the range $[-200, 0] \text{ } ^\circ\text{C}$ is implemented in the Lolin D32 development system for the ESP32 microcontroller. Variables and constants are represented by floating-point data type, the memory footprint of the function is about 145.3 kB, with activated size optimization “-Os”. This is about 10% of the system’s resources.

5. Conclusions

This paper presented an approach for linear interval approximation of sensor characteristics $y = y(x)$ where the characteristics are differentiable functions in which the sign of curvature does not change, i.e., they are either “concave” or “convex” functions. The aim is to obtain a discrete type of inverse sensor characteristic $x_i = x_i(y_i), i = \overline{1, n}$, with a predefined maximum approximation error Δ^x , with minimization of the number of points defining the characteristic, which in turn is related to the possibilities for using cheaper microcontrollers. The approach is characterized by the fact that when the maximum approximation error Δ^x under the linearization of the inverse sensor characteristic is given, the inverse sensor characteristic $x_i = x_i(y_i), i = \overline{1, n}$ is found directly in the linearized form. The advantages of the developed approach are as follows:

- the approach is applied at intervals, with each subsequent step (each successive interval) resulting in a similar solution of the problem under the new initial conditions to obtain the directly sought solution, which in turn contains the initial conditions for the next step;
- the maximum error under linearization of the inverse sensor characteristic at all intervals, except in the general case of the last one, is the same;
- the approach allows that different maximum approximation errors are set at each subsequent interval;
- the approach allows the application to general types of differentiable sensor characteristics with piecewise concave/convex properties.

The proposed approach is verified through linearization of the inverse sensory characteristics of the Pt100 sensor and k-type thermocouple. The proposed approach to linear interval approximation of sensor characteristics $y = y(x)$, which are differentiable functions, can be also extended to sensor characteristics with inflection points (characteristics where the sign of curvature changes, in some subintervals they are “concave”, and in others, they are “convex” functions). A typical example of such a sensor characteristic is the Type K Thermocouple.

To overcome the problems arising from resource constraints, appropriate algorithms are needed that allow efficient connectivity and intelligent management of the measurement processes. Although the discussed computational examples are aimed at building adaptive approximations for temperature sensors, the algorithm can be applied easily to many other sensor types.

Author Contributions: Conceptualization, M.B.M., N.N. and S.D.; methodology, N.N., S.D. and T.T.; software, N.N., G.T.N. and Y.S.; investigation, M.B.M., N.N.; resources, G.T.N., N.N. and M.B.M.; writing—original draft preparation, N.N., M.B.M. and S.D.; writing—review and editing, M.B.M., N.N. and S.D.; visualization, N.N. and Y.S. All authors have read and agreed to the published version of the manuscript.

Funding: The APC was funded by the Research and Development sector at the Technical University of Sofia.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Acknowledgments: The authors would like to thank the Research and Development sector at the Technical University of Sofia for the financial support. The authors acknowledge the support of project BG05M2OP001-1.001-0008 “National Center for Mechatronics and Clean Technologies” Operational Program Executive Agency, Ministry of Education and Science, Bulgaria.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liu, X.; Baiocchi, O. A comparison of the definitions for smart sensors, smart objects, and Things in IoT. In Proceedings of the IEEE 7th Annual Information Technology Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, Canada, 13–15 October 2016.
2. NIST. Definitions—Sensors, National Institute of Standards and Technology. 2009. Available online: <http://www.nist.gov/el/isd/ieee/definitions.cfm> (accessed on 13 January 2022).
3. IEEE. The IEEE 1451.4 Standard for Smart Transducers. 2004. Available online: <https://standards.ieee.org/content/dam/ieee-standards/standards/web/documents/tutorials/1451d4.pdf> (accessed on 12 October 2021).
4. Oriwoh, E.; Conrad, M. Things’ in the Internet of Things: Towards a definition. *Int. J. Internet Things* **2015**, *4*, 1–5.
5. Hossain, M.M.; Fotouhi, M.; Hasan, R. *Towards an Analysis of Security Issues, Challenges, and Open Problems in the Internet of Things*; IEEE: New York, NY, USA, 2015.
6. Zahoor, S.; Mir, R. Resource management in pervasive Internet of Things: A survey. *J. King Saud Univ.—Comput. Inf. Sci.* **2018**, *33*, 921–935. [[CrossRef](#)]
7. Fraden, J. *Handbook of Modern Sensors*; Springer: Cham, Switzerland; Heidelberg, Germany; New York, NY, USA; Dordrecht, The Netherlands; London, UK, 2016.
8. Attari, M. Methods for linearization of non-linear sensors. In Proceedings of the CMMNI-4, Fourth Maghrebin Conference on Numerical Methods of Engineering, Algiers, Algeria; 1993; Volume 1, pp. 344–350.
9. Erdem, H. Implementation of software-based sensor linearization algorithms on low-cost microcontrollers. *ISA Trans.* **2010**, *49*, 552–558. [[CrossRef](#)] [[PubMed](#)]
10. Johnson, C. *Process Control Instrumentation Technology*, 8th ed.; Pearson Education Limited: London, UK, 2013.
11. Marinov, M.; Dimitrov, S.; Djamiykov, T.; Donscheva, M. An Adaptive Approach for Linearization of Temperature Sensor Characteristics. In Proceedings of the 27th International Spring Seminar on Electronics Technology, ISSE 2004, Bankya, Bulgaria, 13–16 May 2004.
12. Brignell, J. Software techniques for sensor compensation. *Sens. Actuators A Phys.* **1990**, *25*, 29–35. [[CrossRef](#)]
13. Flammini, A.; Marioli, D.; Taroni, A. Transducer output signal processing using an optimal look-up table in microcontroller-based systems. *Electron. Lett.* **2010**, *33*, 552–558.

14. Grützmacher, F.; Beichler, B.; Hein, A.; Kirste, T.; Haubelt, C. Time and Memory Efficient Online Piecewise Linear Approximation of Sensor Signals. *Sensors* **2018**, *18*, 1672. [[CrossRef](#)] [[PubMed](#)]
15. Islam, T.; Mukhopadhyay, S. Linearization of the sensors characteristics: A review. *Int. J. Smart Sens. Intell. Syst.* **2019**, *12*, 1–21. [[CrossRef](#)]
16. Keogh, E.; Chu, S.; Hart, D.; Pazzani, M. An online algorithm for segmenting time series. In Proceedings of the IEEE International Conference on Data Mining, ICDM2001, San Jose, CA, USA, 29 November–2 December 2001.
17. Liu, C.; Wu, K.; Pei, J. An energy-efficient data collection framework for wireless sensor networks by exploiting spatiotemporal correlation. *IEEE Trans. Parallel Distrib. Syst.* **2007**, *18*, 1010–1023. [[CrossRef](#)]
18. Pham, N.; Le, T.; Choo, H. Enhance exploring temporal correlation for data collection in WSNs. In Proceedings of the IEEE International Conference on Research, Innovation and Vision for the Future, RIVF 2008, Ho Chi Minh City, Vietnam, 13–17 July 2008.
19. Van Laerhoven, K.; Berlin, E.; Schiele, B. Enabling efficient time series analysis for wearable activity data. In Proceedings of the International Conference on Machine Learning and Applications, ICMLA'09, Miami Beach, FL, USA, 13–15 December 2009.
20. Berlin, E.; van Laerhoven, K. An on-line piecewise linear approximation technique for wireless sensor networks. In Proceedings of the 2010 IEEE 35th Conference on Local Computer Networks (LCN), Denver, CO, USA, 11–14 October 2010.
21. Berlin, E.; van Laerhoven, K. Detecting leisure activities with dense motif discovery. In Proceedings of the 2012 ACM Conference on Ubiquitous Computing, Pittsburgh, PA, USA, 5–8 September 2012.
22. Lemire, D. A better alternative to piecewise linear time-series segmentation. In Proceedings of the 2007 SIAM International Conference on Data Mining, Minneapolis, MN, USA, 26–28 April 2007.
23. Fuchs, E.; Gruber, T.; Nitschke, J.; Sick, B. Online segmentation of time series based on polynomial least-squares approximations. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *32*, 2232–2245. [[CrossRef](#)] [[PubMed](#)]
24. Luo, G.; Yi, K.; Cheng, S.; Li, Z.; Fan, W.; He, C.; Mu, Y. Piecewise linear approximation of streaming time-series data with max-error guarantees. In Proceedings of the 2015 IEEE 31st International Conference on Data Engineering (ICDE), Seoul, Korea, 13–17 April 2015.
25. Junker, H.; Amft, O.; Lukowicz, P.; Tröster, G. Gesture spotting with body-worn inertial sensors to detect user activities. *Pattern Recognit.* **2008**, *41*, 2010–2024. [[CrossRef](#)]
26. Chevrier, M. *TI Designs. Optimized Sensor Linearization for Thermocouple*; TIDUA11A; Texas Instruments Incorporated: Dallas, TX, USA, June 2015; Revised September 2015.
27. van der Horn, G.; Huijsing, J. *Integrated Smart Sensors: Design and Calibration*; Springer: New York, NY, USA, 2012.
28. Pereira, J.M.D.; Postolache, O.; Girao, P.M.B.S. PDF-Based Progressive Polynomial Calibration Method for Smart Sensors Linearization. *IEEE Trans. Instrum. Meas.* **2009**, *58*, 3245–3252. [[CrossRef](#)]
29. Kincaid, D.; Kincaid, D.; Cheney, E. *Numerical Analysis: Mathematics of Scientific Computing*; American Mathematical Society: Washington, DC, USA, 2009.
30. Šturcel, J.; Kamenský, M. Function approximation and digital linearization in sensor systems. *ATP J.* **2006**, *2*, 13–17.
31. Marsden, J.; Weinstein, A. *Calculus I*; Springer: New York, NY, USA, 2012.
32. Fluke. *How to Calibrate an RTD or Platinum Resistance Thermometer (PRT) AN*; Fluke Corporation, Hart Scientific Division: American Fork, UT, USA, 2009.
33. Cejer, M. *Resistive Temperature Detectors: An Alternative to Thermocouples for Precise, Repeatable Temperature Measurements*; Keithley Instruments, Inc.: Cleveland, OH, USA, 2004.
34. Yang, I.; Gam, K.; Joung, W. Modeling the calibration of the industrial platinum-resistance thermometers according to GUM. *Accredit. Qual. Assur.* **2013**, *18*, 359–362. [[CrossRef](#)]
35. Fericola, V.C.; Iacomini, L. Approximating the ITS-90 Temperature Scale with Industrial Platinum Resistance Thermometers. *Int. J. Thermophys.* **2008**, *29*, 1817–1827. [[CrossRef](#)]
36. Nabto. ESP32 for IoT: A Complete Guide. 2020. Available online: <https://www.nabto.com/guide-to-iot-esp-32/> (accessed on 13 January 2022).
37. Haney. Library for Accurate Pt100 RTD Ohms-to-Celsius Conversion. 2018. Available online: https://github.com/drhaneyp/pt100rtd/tree/master/examples/pt100_temperature (accessed on 13 January 2022).
38. Candela, G. *TI Designs. Isolated Loop Powered Thermocouple Transmitter*; TIDU449B; Texas Instruments Incorporated: Dallas, TX, USA, October 2014; Revised July 2016.