sensors

MDPI

Article

Linear Interval Approximation of Sensor Characteristics with Inflection Points

Marin B. Marinov [1], Nikolay Nikolov [2,*], Slav Dimitrov [2], Borislav Ganev [1], Georgi T. Nikolov [1], Yana Stoyanova [2], Todor Todorov [2] and Lachezar Kochev [2]

[1] Faculty of Electronic Engineering and Technologies, Technical University of Sofia, 1756 Sofia, Bulgaria
[2] Faculty of Industrial Technology, Technical University of Sofia, 1756 Sofia, Bulgaria
* Correspondence: nickn@tu-sofia.bg

Abstract: The popularity of smart sensors and the Internet of Things (IoT) is growing in various fields and applications. Both collect and transfer data to networks. However, due to limited resources, deploying IoT in real-world applications can be challenging. Most of the algorithmic solutions proposed so far to address these challenges were based on linear interval approximations and were developed for resource-constrained microcontroller architectures, i.e., they need buffering of the sensor data and either have a runtime dependency on the segment length or require the sensor inverse response to be analytically known in advance. Our present work proposed a new algorithm for the piecewise-linear approximation of differentiable sensor characteristics with varying algebraic curvature, maintaining the low fixed computational complexity as well as reduced memory requirements, as demonstrated in a test concerning the linearization of the inverse sensor characteristic of type K thermocouple. As before, our error-minimization approach solved the two problems of finding the inverse sensor characteristic and its linearization simultaneously while minimizing the number of points needed to support the characteristic.

Keywords: approximation; graphical programming; Internet of Things; linearization techniques; measurement errors; smart sensors; sensor accuracy; thermocouples

Citation: Marinov, M.B.; Nikolov, N.; Dimitrov, S.; Ganev, B.; Nikolov, G.T.; Stoyanova, Y.; Todorov, T.; Kochev, L. Linear Interval Approximation of Sensor Characteristics with Inflection Points. *Sensors* **2023**, *23*, 2933. https://doi.org/10.3390/s23062933

Academic Editors: Yuping Wang, Xiao-Zhi Gao and Xingsi Xue

Received: 12 January 2023
Revised: 21 February 2023
Accepted: 3 March 2023
Published: 8 March 2023

1. Introduction (and Motivation)

1.1. Resource-Constrained Smart Sensor Devices and IoT

In recent years, there has been a growing interest in smart low-cost sensor technologies and IoT [1]. Many very promising developments were made in the field of both sensor technology and wireless communications. Research in the field of smart sensors (SS) and the Internet of Things (IoT) continues to grow [2,3]. Advances in microcontrollers and developments in the Internet have enabled the industry to create smart devices that efficiently integrate sophisticated sensing and communication functions, with primary signal-processing algorithms.

A key distinguishing feature of low-cost smart sensors and IoT devices is their *limited resources*. Typical self-powered smart sensors and IoT devices include two main resource groups.

Hardware resources: for data storing and computing, communication, and power.

Because IoT devices are battery-powered and use low-power processors, they cannot accommodate algorithms that require a large amount of computing power. Furthermore, IoT devices have limited memory compared to regular digital systems, requiring the use of lightweight mobile software and operating systems. As a result, the algorithms must be designed in a way that is efficient in terms of memory usage [4,5].

Software resources: operating systems, system software, and applications. The adaptive allocation of these resources is critical in most applications [6,7]. The remote reprogramming of devices is not always an option as the operating system may not be capable of accepting and integrating new codes.

1.2. The Main Error Components of Smart Sensors and IoT Devices

As already stated, the main objective of this study was to seek opportunities to allocate the limited resources of networked sensors and IoT adaptively and, in particular, to limit the amount of memory required without compromising the accuracy requirements of the devices. The structure of a smart sensor or an IoT device and its main components are shown in Figure 1.

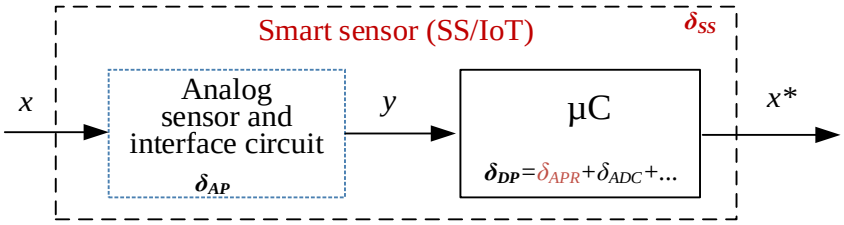

Figure 1. Block diagram and basic error components of smart sensor and IoT devices (x—input quantity, x*—microcontroller output).

For the devices to meet the desired accuracy standards, the major sources of error must be considered. These can be divided into δ_{AP} which are sourced from the analog portion and δ_{DP} which are sourced from the digital part. Further information can be found in [1] and [8].

Here, δ_{AP} is the relative compound error of the analog sensor and sensor interface, δ_{DP} is the relative compound error of the digital part, $\delta_{AP} = \frac{\Delta^x}{x_{FS}}$ is the relative approximation error, Δ^x is the predefined absolute approximation error; x_{FS} is the full scale (FS) of the microcontroller output value x*; and δ_{ADC} is the relative error of the analog-to-digital conversion. Given a serial device structure, the total error budget of the system can be determined as follows:

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

δ_{ADC} and δ_{APR} are the main error components of the δ_{DP} digital part. Once the analog sensor and the analog-to-digital converter (ADC) are selected in the device implementation, the error levels of δ_{AS} and δ_{ADC} cannot be affected. ADCs with high resolution are most commonly used in the implementation of smart sensors and IoT; therefore, the δ_{ADC} component is often negligible. This makes δ_{APR} an essential component by which the overall error rates can be controlled. To achieve typical accuracy levels in device design, where $\delta_{SS} \approx \delta_{AS}$, it is usually sufficient to achieve the condition $\delta_{AS} > (3 \div 10)\delta_{DP}$.

Published lookup tables, which are used in many applications, are often limited in the number of digits and this can be an additional source of rounding errors [8].

1.3. Sensor Characteristics Linearization Approaches

Linearization is an important part of the initial processing of sensor inputs. Nonlinearities in sensors can be reduced by either electronic linearization circuits or algorithms [9,10]. Linearization techniques can be divided into three main categories:

- Analog hardware linearization schemes;
- Linearization algorithms;
- Mixed hardware and software-based approaches [11].

Analog hardware linearization methods are typically performed by connecting an analog scheme between the sensor and the ADC [12].

In [13], the focus is on the cold-end compensation of thermocouples used in room temperature measurement applications. In [14], an approach to linearize the characteristics of a K-type thermocouple and thermistor used for its temperature compensation was shown. A reduction in the nonlinearity of about 100 times for the thermocouple and from 84% to 0.27% for the thermistor was demonstrated.

Software linearization techniques necessitate the deployment of (micro)computers or digital signal processors (DSPs) with powerful processing capabilities [15,16]. Imple-

menting these techniques on low-cost controllers that only have the capacity for limited computational tasks is very difficult due to the limited resources of the controller. Different software linearization methods were analyzed in the literature. One of the most frequently used methods was the lookup table (LUT)-based linearization, which can be implemented easily on any microcontroller [11,17].

This research built upon the technique outlined in [1,18], which was used to adaptively linearize sensor characteristics, make the design simpler, and improve the measurement accuracy of sensors and IoT devices that are resource-limited.

The identification of the inverse transfer function is often complicated due to the difficulty in selecting the correct analytic form of the function and the constraints in its parameterization. This can lead to inaccurate sensor responses, so it should be avoided. Generally, the scalar inverse sensor transfer function is modeled using a nonlinear regression model (e.g., polynomial, exponential, etc.) which is determined by minimizing the least squares error on a statistically representative set of data [19,20].

In [21], the use of a simple model based on a three-layer recurrent neural network (RNN) that considers a short history of the immediately preceding predictions (temperatures) together with the current measurement to predict the current output signal was shown. In [22], the use of popular sensor linearization techniques based on the multilinear model approach was shown to be popular due to the simplicity and transparency of local linear models. The paper presented a systematic data-driven approach that used the included angle method to determine optimal linear models. A fuzzy interpolation technique was then used to combine the linear models.

Linearization approaches can be also regarded as shape-preserving dimensionality-reducing methods [23]. In such methods, the inverse sensor characteristic is mapped into a polygonal shape, by using either distance minimizing embedding technique, or, whenever permissible, by range and accuracy requirements—a non-negative matrix factorization technique, as suggested in [24].

A general approach to reducing the uncertainty problems arising in nonlinear regression identification of sensor feedback can be provided by segmenting its transfer function.

Essentially, it implements a polygonal approximation of $x = x(y)$ with approximation error control. The proposed algorithmic control is an important aspect supporting adaptive resource allocation.

Piecewise linear approximation (PLA) for sensor data is a classical approach used in data compression. There are many other data compression approaches, such as piecewise aggregate approximation [25], discrete wavelet transform [26], discrete Fourier transform [27], Chebyshev polynomials [28], etc. However, PLA remains one of the most commonly used approaches for data compression, as claimed in [29,30].

The approach dates back to the middle of the last century but, in recent years, the data compression problem became topical due to the widespread adoption of smart sensors and IoT devices. It was increasingly used wherever data acquisition devices limited local buffer space and communication bandwidth [31].

It is essential that data are compressed due to the restricted resources of data acquisition devices, including memory and communication capabilities. The main criteria for measuring compression quality are the magnitude of the approximation error and the number of line segments.

In optimizing the PLA results, two approaches are commonly used:

Setting a bound on the error Δ^x and minimizing the number of k segments;

Setting the number of k segments for which to construct a PLA with at most k segments that minimize the error Δ^x.

In the approach investigated in this paper, we set the delta error bound and minimized the PLA size [32].

1.4. Piecewise Linear Approximation of Sensor Characteristics with Inflex Points

In [15], an algorithm was formulated to linearize sensor characteristics when the characteristic was a differentiable function. This two-step algorithm involved an iterative process with a given sensor characteristic and a maximum approximation error. The result of this process was a discrete form (a broken line) of the inverse characteristic.

With a set sensor characteristic in the form $y = y(x)$ the first step is to determine the inverse sensory characteristic $x = x(y)$. The second step is related to the approximation of the received discrete inverse sensory characteristic in the species $x_i = x_i(y_i), i = \overline{1,n}$ with a new feature $x_j = x_j(y_j), \ \ j = \overline{1,k}$ given the maximum approximation error and minimization of k. An approach to solving the problem was developed in [15] and in [33].

In [1], a novel technique for linearizing the characteristics of sensors, which were represented by differentiable functions with a constant sign of curvature, was proposed. This approach simultaneously solved the issues of locating the inverse sensor characteristic and its linearization, as was exemplified by the resistance–temperature relationship of platinum temperature sensors that followed the Callendar–Van Dusen equation. The approach was characterized by the fact that when the maximum approximation error is set Δ^x in the linearization of the inverse sensing characteristic, the inverse sensing characteristic $x_i = x_i(y_i), i = \overline{1,n}$ is found directly in linearized form. The advantages of the developed approach are as follows:

- The approach is applied in intervals, and at each subsequent step (each subsequent interval) as a result of analogously solving the task under the new initial conditions, the desired solution is obtained directly, containing, in turn, the initial conditions for the next step;
- The maximum linearization error of the inverse sensor characteristic in all intervals is the same;
- The approach makes it possible to set a different maximum approximation error in each subsequent interval.

The proposed approach was tested in the linearization of the inverse sensor characteristic of Pt100 sensors, where the relationship between the resistance and the temperature was set by employing the Callendar–Van Dusen equation [34,35]. When the maximum approximation error is set $\Delta^T = -0.375$ °C in the interval $T \in [-200$ °C, 661 °C] the reverse sensory characteristic $T(R)$ (representing a broken line) is described by 11 points.

Thermocouples are simple and widely used sensor elements for temperature measurement. However, it is not easy to convert the voltage generated by the thermocouple to temperature with high accuracy for several reasons, the main ones being the low levels of useful signal and the non-linear relationship between temperature and voltage. The thermocouples also change their sensitivity depending on the temperature, as seen in Figure 2. This is why they were chosen for the tests in our study.

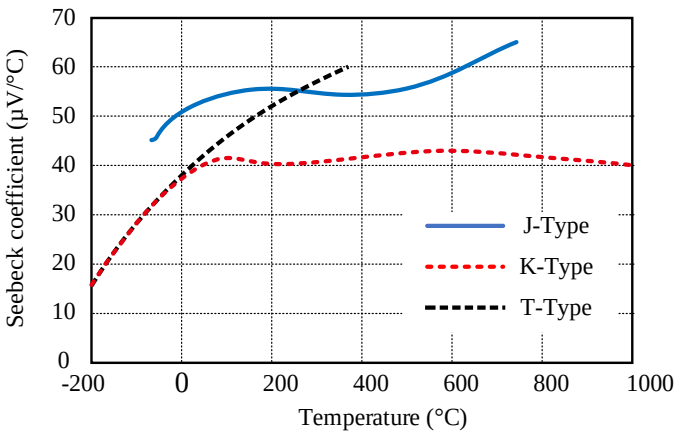

Figure 2. Temperature-dependent sensitivity change of three types of thermocouples (adopted from [36]). K- and J-type thermocouples were chosen for the tests in the study.

This approach for approximating sensor characteristics $y = y(x)$ with linear intervals can be used for features that contain inflection points, where the curvature changes in different subintervals (i.e., "concave" and "convex" functions). An example of this is type K thermocouples [37]. To do this, the approach must be applied separately to each of the subintervals in which the sensory characteristic is "concave" and separately to each of the subintervals in which the sensory characteristic is "convex". However, this would lead to the elimination of one of the main advantages of the approach developed in [1]—the maximum linearization error of the inverse sensor response in all intervals (except the last one) to be the same.

In the context of the above, this paper was devoted to the development of a generalized approach for linear point-interval approximation of sensor characteristics, representing differentiable functions, with inflection points present. The main advantage of this is that, similar to the approach developed in [1], the tasks of finding the inverse sensor characteristic and its linearization are solved simultaneously.

The proposed generalized approach for linear point-interval approximation of sensor characteristics applies to all sensory characteristics, representing differentiable functions. The essence of the approach is illustrated in Figure 3.

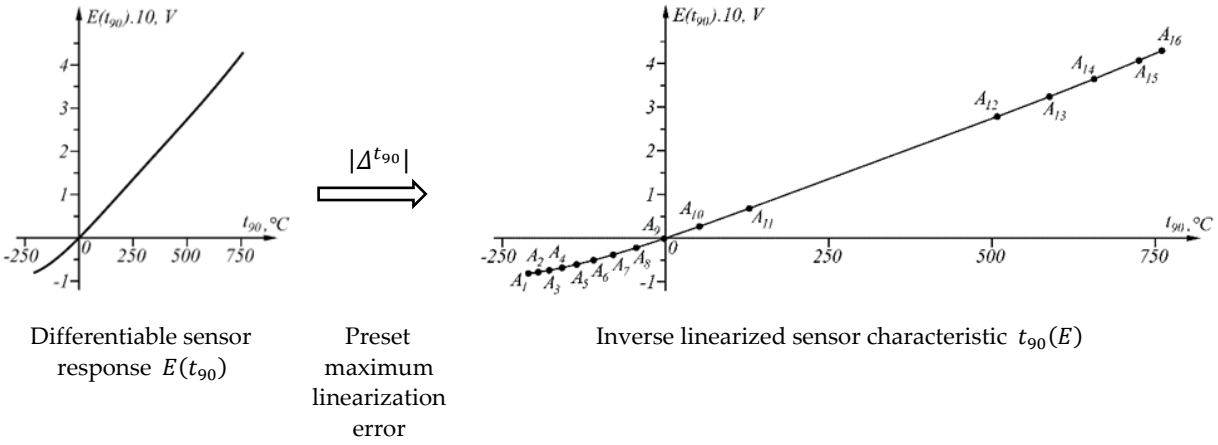

Figure 3. Graphical representation of the most important part of the proposed approach for linearization.

The preset maximum linearization error is the only parameter in the generalized approach for linear point-interval approximation of sensor characteristics, on which the number of segments of the broken line (inverse linearized sensor characteristic) depends. A smaller value of the parameter (the maximum linearization error) results in more segments. The particular value of this parameter is determined by the application of the sensor and the accuracy target set.

2. The Analytical Frame of the Proposed Approach

In [1], an approach for the linearization of sensor characteristics representing differentiable functions with an invariant sign of curvature was presented. In the present study, this approach will be further developed for cases where sensory features include inflection points.

Let a differentiable sensor characteristic be given in the form $y = y(x)$, $x \in [x_{A_i}, x_B]$, $i = \overline{1, n-1}$. The goal is to obtain the inverse sensor characteristic $x(y)$ in a linearized form with a set maximum approximation error Δ^x.

The approach is based on the following. We are looking for the closest to point A_i point E_{i_I} from the curve $y = y(x)$, such that tangent t_{i_I} to the curve at point E_{i_I} passes through the coordinate point $(x_{A_i} + \Delta^x, y_{A_i})$ (Figure 4).

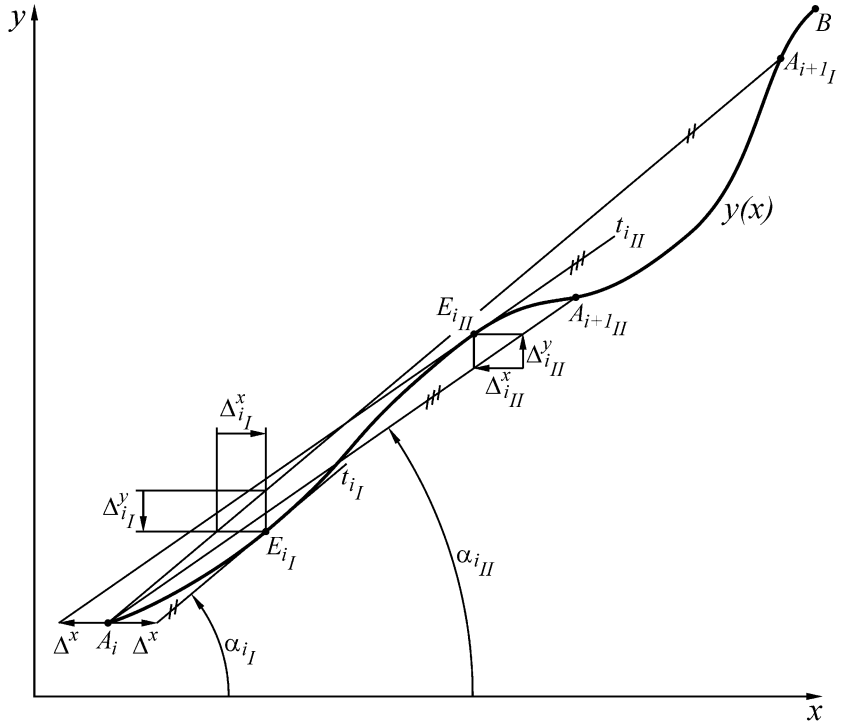

Figure 4. An example of mathematical processing of sensor characteristic.

Then, we search for the closest to point A_i point A_{i+1_I} from curve $y = y(x)$, being the point of intersection of curve $y = y(x)$ with the line passing through point A_i and parallel to the tangent t_{i_I}. Analogously, we look for the closest to point A_i point $E_{i_{II}}$ from curve $y = y(x)$, such that the tangent $t_{i_{II}}$ to the curve at point $E_{i_{II}}$ to pass through the coordinate point $\left(x_{A_i} - \Delta^x, y_{A_i}\right)$. Then, we search for the closest to point A_i point $A_{i+1_{II}}$ from curve $y = y(x)$, being the point of intersection of curve $y = y(x)$ with the line passing through point A_i and parallel to the tangent $t_{i_{II}}$ (Figure 4). The closer to the point A_i of the points A_{i+1_I} and $A_{i+1_{II}}$ is the endpoint A_{i+1} of the interval in which the maximum absolute value of the error is Δ^x when approximating the inverse sensor response $x(y)$ with a linear dependence (the segment A_iA_{i+1}).

If after some value of i, the interval $x \in \left[x_{A_i}, x_B\right]$ does not include an inflection point (the function $y = y(x)$ is "concave" or "convex" in the interval), then one of the two tangents t_{i_I} and $t_{i_{II}}$ does not exist, and the approach presented here is practically reduced to the approach presented in [1].

If $x_{A_{i+1}} > x_B$, then $A_{i+1} \equiv B$ and the procedure ends, as in the general case in this last interval, the maximum absolute error is less than the set Δ^x.

In the general case, when range $x \in \left[x_{A_i}, x_B\right]$ includes an inflection point, the two extremes $\Delta^x_{i_I}$ and $\Delta^x_{i_{II}}$ of error $\Delta^x_i(y)$ in the linear approximation of the inverse function $x(y)$ are determined by (Figure 4):

$$\begin{aligned} \Delta^x_{i_I}\left(x_{E_{i_I}}\right) &= \frac{\Delta^y_{i_I}\left(x_{E_{i_I}}\right)}{-k_{i_I}} = x_{E_{i_I}} - x_{A_i} - \frac{y\left(x_{E_{i_I}}\right) - y_{A_i}}{k_{i_I}}; \\ \Delta^x_{i_{II}}\left(x_{E_{i_{II}}}\right) &= \frac{\Delta^y_{i_I}\left(x_{E_{i_{II}}}\right)}{-k_{i_{II}}} = x_{\mathrm{E}_{i_{II}}} - x_{A_i} - \frac{y\left(x_{\mathrm{E}_{i_{II}}}\right) - y_{A_i}}{k_{i_{II}}}, \end{aligned} \tag{1}$$

where

$$\frac{\Delta_i^y\left(x_{E_{i_I}}\right)}{\Delta_i^x\left(x_{E_{i_I}}\right)} = -\tan\alpha_{i_I} = -k_{i_I};$$
$$\frac{\Delta_i^y\left(x_{E_{i_{II}}}\right)}{\Delta_i^x\left(x_{E_{i_{II}}}\right)} = -\tan\alpha_{i_{II}} = -k_{i_{II}}.$$

From the extremum necessary condition $[\Delta_{i_I}^y\left(x_{\mathrm{E}_{i_I}}\right)]' = 0$, respectively, $[\Delta_{i_{II}}^y\left(x_{\mathrm{E}_{i_{II}}}\right)]' = 0$, is obtained:

$$\begin{gathered} y'\left(x_{E_{i_I}}\right) - k_{i_I} = 0; \\ y'\left(x_{E_{i_{II}}}\right) - k_{i_{II}} = 0. \end{gathered} \tag{2}$$

From Equations (1) and (2) follows:

$$\Delta_{i_I}^x\left(x_{E_{i_I}}\right) = \Delta^x = x_{\mathrm{E}_{i_I}} - x_{A_i} - \frac{y\left(x_{\mathrm{E}_{i_I}}\right) - y_{A_i}}{k_{i_I}} = x_{\mathrm{E}_{i_I}} - x_{A_i} - \frac{y\left(x_{\mathrm{E}_{i_I}}\right) - y_{A_i}}{y'\left(x_{\mathrm{E}_{i_I}}\right)};$$

$$\Delta_{i_{II}}^x\left(x_{E_{i_I}}\right) = -\Delta^x = x_{E_{i_{II}}} - x_{A_i} - \frac{y\left(x_{E_{i_{II}}}\right) - y_{A_i}}{k_{i_{II}}} = x_{E_{i_{II}}} - x_{A_i} - \frac{y\left(x_{E_{i_{II}}}\right) - y_{A_i}}{y'\left(x_{E_{i_{II}}}\right)}.$$

respectively

$$\begin{gathered} y'\left(x_{E_{i_I}}\right) - \frac{y\left(x_{E_{i_I}}\right) - y_{A_i}}{x_{E_{i_I}} - x_{A_i} - \Delta^x} = 0; \\ y'\left(x_{E_{i_{II}}}\right) - \frac{y\left(x_{E_{i_{II}}}\right) - y_{A_i}}{x_{E_{i_{II}}} - x_{A_i} + \Delta^x} = 0. \end{gathered} \tag{3}$$

From the last equation, we determine the nearest points E_{i_I} and $E_{i_{II}}$ (in case there are more than one) to the point A_i.

From the equations

$$\begin{gathered} y\left(x_{A_{i+1_I}}\right) = y_{A_i} + y'\left(x_{E_{i_I}}\right)\left(x_{A_{i+1_I}} - x_{A_i}\right), \\ y\left(x_{A_{i+1_{II}}}\right) = y_{A_i} + y'\left(x_{E_{i_{II}}}\right)\left(x_{A_{i+1_{II}}} - x_{A_i}\right), \end{gathered} \tag{4}$$

$x_{A_{i+1_I}}$ and $x_{A_{i+1_{II}}}$ are determined, with the smaller of the two values representing $x_{A_{i+1}}$ (i.e., the point closest to the point A_i of the points A_{i+1_I} and $A_{i+1_{II}}$ being the endpoint A_{i+1} of the interval).

As a result of applying the approach, the sensory characteristic $y(x)$ is approximated by the broken line $A_1A_2A_3\ldots A_{n-1}B \equiv A_n$, $A_i\left(x_{A_i}, y_{A_i}\right)$, $i = \overline{1, n-1}$, $A_n(x_B, y_B)$.

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

$$\begin{gathered} x(y) = x_{A_i} + \left(y - y_{A_i}\right)\frac{x_{A_{i+1}} - x_{A_i}}{y_{A_{i+1}} - y_{A_i}}, \\ y \in \left[y_{A_i}, y_{A_{i+1}}\right],\ i = \overline{1, n-1}. \end{gathered} \tag{5}$$

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

$$\begin{gathered} \Delta_i^x(x) = x - x_{A_i} - \left(y(x) - y_{A_i}\right)\frac{x_{A_{i+1}} - x_{A_i}}{y_{A_{i+1}} - y_{A_i}}, \\ x \in \left[x_{A_i}, x_{A_{i+1}}\right],\ i = \overline{1, n-1}. \end{gathered} \tag{6}$$

The error $\Delta_i^x(y)$ can also be determined from Equation (6), because the function $y = y(x)$ is strictly monotonic (at any value of x matches a single corresponding value of $y(x)$)

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

3. Linearization of the Inverse Sensor Characteristic of Type K and Type J Thermocouples

3.1. Type K Segmentation in the Temperature Range $t_{90} \in [0, 1371.655\ ^\circ C]$

The thermoelectric voltage E in microvolts, as a function of temperature t_{90} in degrees Celsius, in the range $t_{90} \in [0,\ 1372\ ^\circ\mathrm{C}]$, is defined by:

$$E(t_{90}) = \sum_{i=0}^{9} c_i\ (t_{90})^i + \alpha_0 e^{\alpha_1 (t_{90} - 126.9686)^2},$$

where the coefficients c_i, α_0 and α_1 are specified in NIST ITS-90 [37].

With the help of the proposed approach, the problem of interval linearization of the inverse characteristic of the sensory characteristic in question, including inflection points, will be solved. The graphs of the function $E(t_{90})$ and its first derivative $E'(t_{90})$ and the second derivative $E''(t_{90})$ in the interval $t_{90} \in [0, 1372\ ^\circ\mathrm{C}]$ are shown in Figure 5.

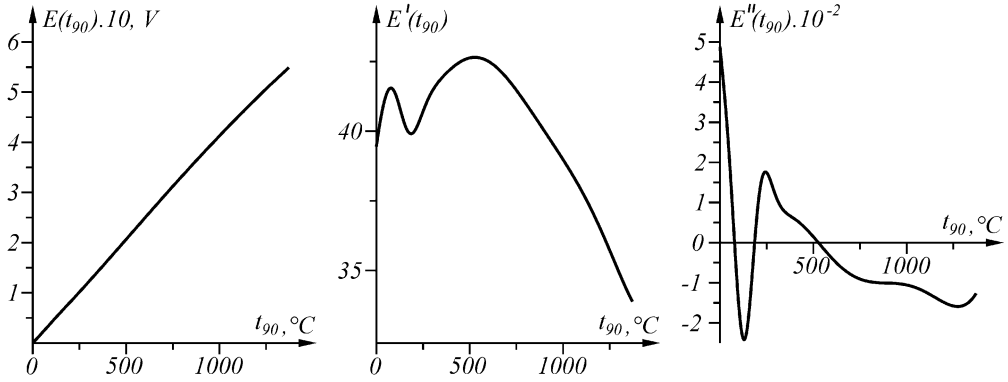

Figure 5. Functions E(t_90), its first derivative E′ (t_90) and second derivative E″ (t_90) in interval t_90 ∈ [0, 1372 °C].

The results of applying the approach at $\left|\Delta^{t_{90}}\right| = 0.04\ ^\circ$C in the temperature range $t_{90_{A_1}} = 0\ ^\circ$C, $t_{90_{A_n \equiv B}} = 1371.655\ ^\circ$C are shown numerically in Table 1 and in Figure 6. Function $\Delta_i^{t_{90}}(t_{90})$:

$$\Delta_i^{t_{90}}(t_{90}) = t_{90} - t_{90_{A_i}} - \left(E(t_{90}) - E_{A_i}\right)\frac{t_{90_{A_{i+1}}} - t_{90_{A_i}}}{E_{A_{i+1}} - E_{A_i}}, \quad t_{90} \in \left[t_{90_{A_i}}, t_{90_{A_{i+1}}}\right],\ i = \overline{1, n-1}. \tag{8}$$

is shown in Figure 7.

Table 1. Interval linearization results in a range $t_{90} \in [0,\ 1371.655\ ^\circ\mathrm{C}]$.

A_1A_2	A_2A_3	A_3A_4	A_4A_5
$A_1(0,\ 0.000)$ $A_2(16.882,\ 672.556)$	$A_2(16.882,\ 672.556)$ $A_3(35.972,\ 1446.874)$	$A_3(35.972,\ 1446.874)$ $A_4(60.200,\ 2444.756)$	$A_4(60.200,\ 2444.756)$ $A_5(108.378,\ 4442.207)$
$t_{90} \in [0,\ 16.882]$	$t_{90} \in [16.882,\ 35.972]$	$t_{90} \in [35.972,\ 60.200]$	$t_{90} \in [60.200,\ 108.378]$
$t_{90_{E_1}} = 8.312$	$t_{90_{E_2}} = 26.197$	$t_{90_{E_3}} = 47.351$	$t_{90_{E_4}} = 93.078$

Table 1. *Cont.*

$\Delta_1^{t_{90}} = 0.04$	$\Delta_2^{t_{90}} = 0.04$	$\Delta_3^{t_{90}} = 0.04$	$\Delta_4^{t_{90}} = -0.04$
A_5A_6 $A_5(108.378,\ 4442.207)$ $A_6(132.208,\ 5418.271)$	A_6A_7 $A_6(132.208,\ 5418.271)$ $A_7(156.708,\ 6408.103)$	A_7A_8 $A_7(156.708,\ 6408.103)$ $A_8(220.884,\ 8975.453)$	A_8A_9 $A_8(220.884,\ 8975.453)$ $A_9(248.358,\ 10,086.534)$
$t_{90} \in [108.378,\ 132.208]$	$t_{90} \in [132.208,\ 156.708]$	$t_{90} \in [156.708,\ 220.884]$	$t_{90} \in [220.884,\ 248.358]$
$t_{90_{E_5}} = 120.487$	$t_{90_{E_6}} = 144.107$	$t_{90_{E_7}} = 204.448$	$t_{90_{E_8}} = 234.769$
$\Delta_5^{t_{90}} = -0.04$	$\Delta_6^{t_{90}} = -0.04$	$\Delta_7^{t_{90}} = 0.04$	$\Delta_8^{t_{90}} = 0.04$
A_9A_{10} $A_9(248.358,\ 10,086.534)$ $A_{10}(276.647,\ 11,244.101)$	$A_{10}A_{11}$ $A_{10}(276.647,\ 11,244.101)$ $A_{11}(309.309,\ 12,594.865)$	$A_{11}A_{12}$ $A_{11}(309.309,\ 12,594.865)$ $A_{12}(348.251,\ 14,219.886)$	$A_{12}A_{13}$ $A_{12}(348.251,\ 14,219.886)$ $A_{13}(392.599,\ 16,084.688)$
$t_{90} \in [248.358,\ 276.647]$	$t_{90} \in [276.647,\ 309.309]$	$t_{90} \in [309.309,\ 348.251]$	$t_{90} \in [348.251,\ 392.599]$
$t_{90_{E_9}} = 262.283$	$t_{90_{E_{10}}} = 292.511$	$t_{90_{E_{11}}} = 328.276$	$t_{90_{E_{12}}} = 370.030$
$\Delta_9^{t_{90}} = 0.04$	$\Delta_{10}^{t_{90}} = 0.04$	$\Delta_{11}^{t_{90}} = 0.04$	$\Delta_{12}^{t_{90}} = 0.04$
$A_{13}A_{14}$ $A_{13}(392.599,\ 16,084.688)$ $A_{14}(443.288,\ 18,230.712)$	$A_{14}A_{15}$ $A_{14}(443.288,\ 18,230.712)$ $A_{15}(516.197,\ 21,334.882)$	$A_{15}A_{16}$ $A_{15}(516.197,\ 21,334.882)$ $A_{16}(602.521,\ 25,012.615)$	$A_{16}A_{17}$ $A_{16}(602.521,\ 25,012.615)$ $A_{17}(653.510,\ 27,173.147)$
$t_{90} \in [392.599,\ 443.288]$	$t_{90} \in [443.288,\ 516.197]$	$t_{90} \in [516.197,\ 602.521]$	$t_{90} \in [602.521,\ 653.510]$
$t_{90_{E_{13}}} = 417.202$	$t_{90_{E_{14}}} = 475.511$	$t_{90_{E_{15}}} = 567.800$	$t_{90_{E_{16}}} = 628.945$
$\Delta_{13}^{t_{90}} = 0.04$	$\Delta_{14}^{t_{90}} = 0.04$	$\Delta_{15}^{t_{90}} = -0.04$	$\Delta_{16}^{t_{90}} = -0.04$
$A_{17}A_{18}$ $A_{17}(653.510,\ 27,173.147)$ $A_{18}(697.047,\ 29,005.203)$	$A_{18}A_{19}$ $A_{18}(697.047,\ 29,005.203)$ $A_{19}(737.039,\ 30,675.189)$	$A_{19}A_{20}$ $A_{19}(737.039,\ 30,675.189)$ $A_{20}(775.070,\ 32,250.248)$	$A_{20}A_{21}$ $A_{20}(775.070,\ 32,250.248)$ $A_{21}(811.974,\ 33,765.632)$
$t_{90} \in [653.510,\ 697.047]$	$t_{90} \in [697.047,\ 737.039]$	$t_{90} \in [737.039,\ 775.070]$	$t_{90} \in [775.070,\ 811.974]$
$t_{90_{E_{17}}} = 675.665$	$t_{90_{E_{18}}} = 717.243$	$t_{90_{E_{19}}} = 756.163$	$t_{90_{E_{20}}} = 793.579$
$\Delta_{17}^{t_{90}} = -0.04$	$\Delta_{18}^{t_{90}} = -0.04$	$\Delta_{19}^{t_{90}} = -0.04$	$\Delta_{20}^{t_{90}} = -0.04$
$A_{21}A_{22}$ $A_{21}(811.974,\ 33,765.632)$ $A_{22}(848.239,\ 35,241.740)$	$A_{22}A_{23}$ $A_{22}(848.239,\ 35,241.740)$ $A_{23}(884.134,\ 36,689.927)$	$A_{23}A_{24}$ $A_{23}(884.134,\ 36,689.927)$ $A_{24}(919.767,\ 38,114.730)$	$A_{24}A_{25}$ $A_{24}(919.767,\ 38,114.730)$ $A_{25}(955.104,\ 39,514.991)$
$t_{90} \in [811.974,\ 848.239]$	$t_{90} \in [848.239,\ 884.134]$	$t_{90} \in [884.134,\ 919.767]$	$t_{90} \in [919.767,\ 955.104]$
$t_{90_{E_{21}}} = 830.133$	$t_{90_{E_{22}}} = 866.197$	$t_{90_{E_{23}}} = 901.959$	$t_{90_{E_{24}}} = 937.452$
$\Delta_{21}^{t_{90}} = -0.04$	$\Delta_{22}^{t_{90}} = -0.04$	$\Delta_{23}^{t_{90}} = -0.04$	$\Delta_{24}^{t_{90}} = -0.04$
$A_{25}A_{26}$ $A_{25}(955.104,\ 39,514.991)$ $A_{26}(989.996,\ 40,885.113)$	$A_{26}A_{27}$ $A_{26}(989.996,\ 40,885.113)$ $A_{27}(1024.231,\ 42,217.028)$	$A_{27}A_{28}$ $A_{27}(1024.231,\ 42,217.028)$ $A_{28}(1057.590,\ 43,502.686)$	$A_{28}A_{29}$ $A_{28}(1057.590,\ 43,502.686)$ $A_{29}(1089.912,\ 44,736.246)$
$t_{90} \in [955.104,\ 989.996]$	$t_{90} \in [989.996,\ 1024.231]$	$t_{90} \in [1024.231,\ 1057.590]$	$t_{90} \in [1057.590,\ 1089.912]$
$t_{90_{E_{25}}} = 972.582$	$t_{90_{E_{26}}} = 1007.164$	$t_{90_{E_{27}}} = 1040.978$	$t_{90_{E_{28}}} = 1073.829$
$\Delta_{25}^{t_{90}} = -0.04$	$\Delta_{26}^{t_{90}} = -0.04$	$\Delta_{27}^{t_{90}} = -0.04$	$\Delta_{28}^{t_{90}} = -0.04$
$A_{29}A_{30}$ $A_{29}(1089.912,\ 44,736.246)$ $A_{30}(1121.115,\ 45,915.218)$	$A_{30}A_{31}$ $A_{30}(1121.115,\ 45,915.218)$ $A_{31}(1151.210,\ 47,040.472)$	$A_{31}A_{32}$ $A_{31}(1151.210,\ 47,040.472)$ $A_{32}(1180.276,\ 48,115.572)$	$A_{32}A_{33}$ $A_{32}(1180.276,\ 48,115.572)$ $A_{33}(1208.447,\ 49,145.979)$
$t_{90} \in [1089.912,\ 1121.115]$	$t_{90} \in [1121.115,\ 1151.210]$	$t_{90} \in [1151.210,\ 1180.276]$	$t_{90} \in [1180.276,\ 1208.447]$
$t_{90_{E_{29}}} = 1105.594$	$t_{90_{E_{30}}} = 1136.239$	$t_{90_{E_{31}}} = 1165.810$	$t_{90_{E_{32}}} = 1194.416$
$\Delta_{29}^{t_{90}} = -0.04$	$\Delta_{30}^{t_{90}} = -0.04$	$\Delta_{31}^{t_{90}} = -0.04$	$\Delta_{32}^{t_{90}} = -0.04$

Table 1. *Cont.*

$A_{33}A_{34}$ $A_{33}(1208.447,\ 49,145.979)$ $A_{34}(1235.896,\ 50,138.464)$	$A_{34}A_{35}$ $A_{34}(1235.896,\ 50,138.464)$ $A_{35}(1262.828,\ 51,100.866)$	$A_{35}A_{36}$ $A_{35}(1262.828,\ 51,100.866)$ $A_{36}(1289.490,\ 52,042.271)$	$A_{36}A_{37}$ $A_{36}(1289.490,\ 52,042.271)$ $A_{37}(1316.189,\ 52,973.750)$
$t_{90} \in [1208.447,\ 1235.896]$	$t_{90} \in [1235.896,\ 1262.828]$	$t_{90} \in [1262.828,\ 1289.490]$	$t_{90} \in [1289.490,\ 1316.189]$
$t_{90_{E_{33}}} = 1222.211$	$t_{90_{E_{34}}} = 1249.382$	$t_{90_{E_{35}}} = 1276.157$	$t_{90_{E_{36}}} = 1302.808$
$\Delta_{33}^{t_{90}} = -0.04$	$\Delta_{34}^{t_{90}} = -0.04$	$\Delta_{35}^{t_{90}} = -0.04$	$\Delta_{36}^{t_{90}} = -0.04$
$A_{37}A_{38}$ $A_{37}(1316.189,\ 52,973.750)$ $A_{38}(1343.352,\ 53,910.167)$	$A_{38}A_{39}$ $A_{38}(1343.352,\ 53,910.167)$ $A_{39}(1371.655,\ 54,874.662)$		
$t_{90} \in [1316.189,\ 1343.352]$	$t_{90} \in [1343.352,\ 1371.655]$		
$t_{90_{E_{37}}} = 1329.696$	$t_{90_{E_{38}}} = 1357.354$		
$\Delta_{37}^{t_{90}} = -0.04$	$\Delta_{38}^{t_{90}} = -0.04$		

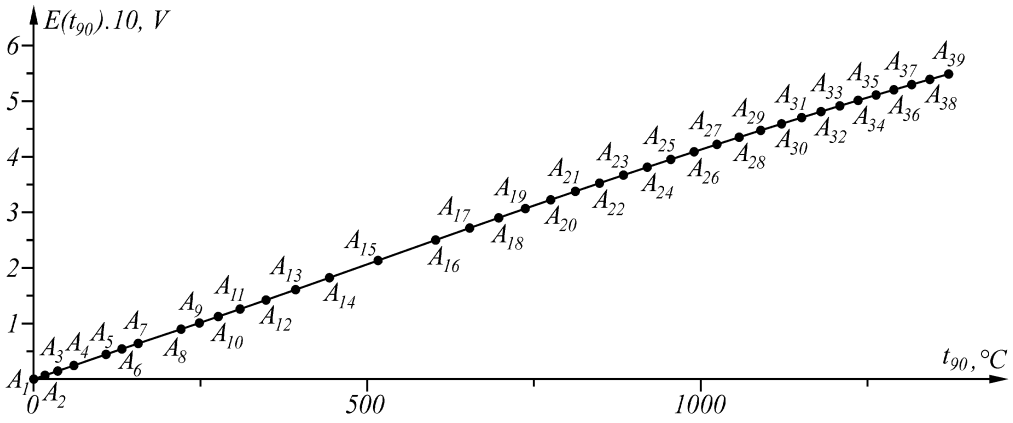

Figure 6. Segmentation of the characteristic in the temperature range $t_{90} \in [0,\ 1371.655\ ^{\circ}\mathrm{C}]$.

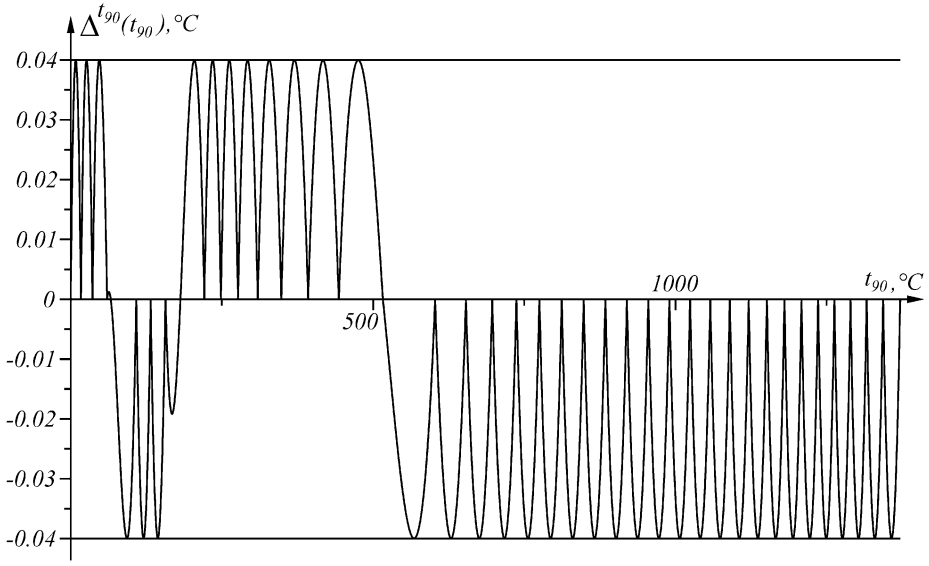

Figure 7. Graphical representation of the absolute error $\Delta_i^{t_{90}}(t_{90})$ in the temperature range $t_{90} \in [0,\ 1371.655\ ^{\circ}\mathrm{C}]$.

Error $\Delta_i^{t_{90}}(t_{90})$, defined as the difference between the actual and the measured value (8), is positive in the intervals in which the second derivative $E''(t_{90})$ of the sensory char-

acteristic $E(t_{90})$ is positive, and negative in the intervals in which the second derivative $E''(t_{90})$ of the sensory characteristic $E(t_{90})$ is negative (Figures 5 and 7).

3.2. Microcontroller and LabVIEW Implementation of the Inverse Sensor Characteristics Linearization Algorithm

This design uses an Adafruit ESP32 Feather V2 [38] development board based on Espressif Systems' ESP32 microcontroller [39]. The linearization is implemented by splitting the individual segments to ensure that the maximum predefined error requirements are satisfied. For this particular implementation, the coordinates of the points defining each interval were stored in two one-dimensional arrays each with 39 elements of type float (see Table 1). The input parameter of the implemented function is the generated voltage of the thermocouple, which must be compensated for the cold junction temperature. The variable to which its value is assigned is also of type float, with the voltage being given in μV. Following this, the determination of which of the 38 intervals the measured temperature fell into was made using the formula shown in Algorithm 1.

Algorithm 1. Linearization of a k-Type thermocouple

Algorithm K-Type Linearization
Input: Measured and compensated voltage, **Measured_Voltage**, floating point type
Output: Calculated temperature, **Temperature**, floating point type
Initialization: Define the points determining the coordinates of each interval:
U_set[1], U_set[2] . . . U_set[End] and T_set[1], T_set[2] . . . T_set[End]
1: Determine the interval, where the measured temperature is situated
If ((**Measured_Voltage** > U_set[n]) and (**Measured_Voltage** $\leq$ U_set[n+1]))
2: Calculate the temperature by the formula
Temperature = (Measured_Voltage − U_set[n])×((T_set[n+1] − T_set[n])/(U_set[n+1] − U_set[n])) + T_set[n];
3: Return **Temperature**

The implementation of the function for the particular example considered in Table 1 took 449 bytes of Flash memory and 4 bytes of RAM, with the size optimization set to -Os. Only 312 bytes were needed to define the two arrays, out of the 449 used.

The algorithm was also implemented as a virtual instrument in the LabVIEW programming environment (see Figure 8). The version used by the authors to validate the proposed virtual instrument was LabVIEW 2012, but it would work on any newer version. The input parameters were the generated voltage from the thermocouple (Measured Voltage) and the two arrays with the stored coordinate values for each point (U_set and T_set). The output parameters were the converted temperature and an array (output array) in which only one element will have a value other than the constant NaN, and this will be the converted temperature. For input voltages that are in the specified range, there will be only one case where the condition of the "case" structure will be "True". This is the case where the temperature needs to be calculated. In all other cases, the element with the constant "NaN" is added to the "Output array".

To prove the correct operation of the developed virtual instrument, values in the range $0 \div 54874.662$ μV with step 0.01 μV were fed to its input. Figure 9 shows the linearized output, the temperature calculated by approximate inverse functions giving temperature, t_{90}, as a function of the thermoelectric voltage and the difference between them.

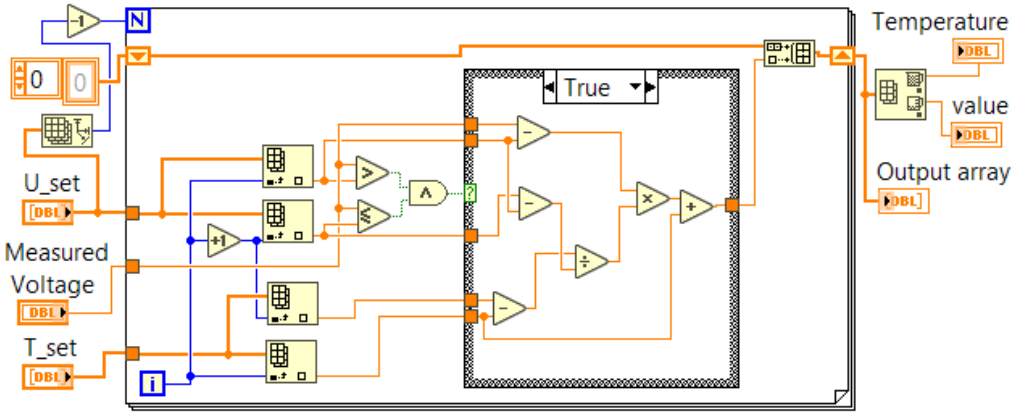

Figure 8. Implementation of the suggested algorithm in the graphical programming environment LabVIEW.

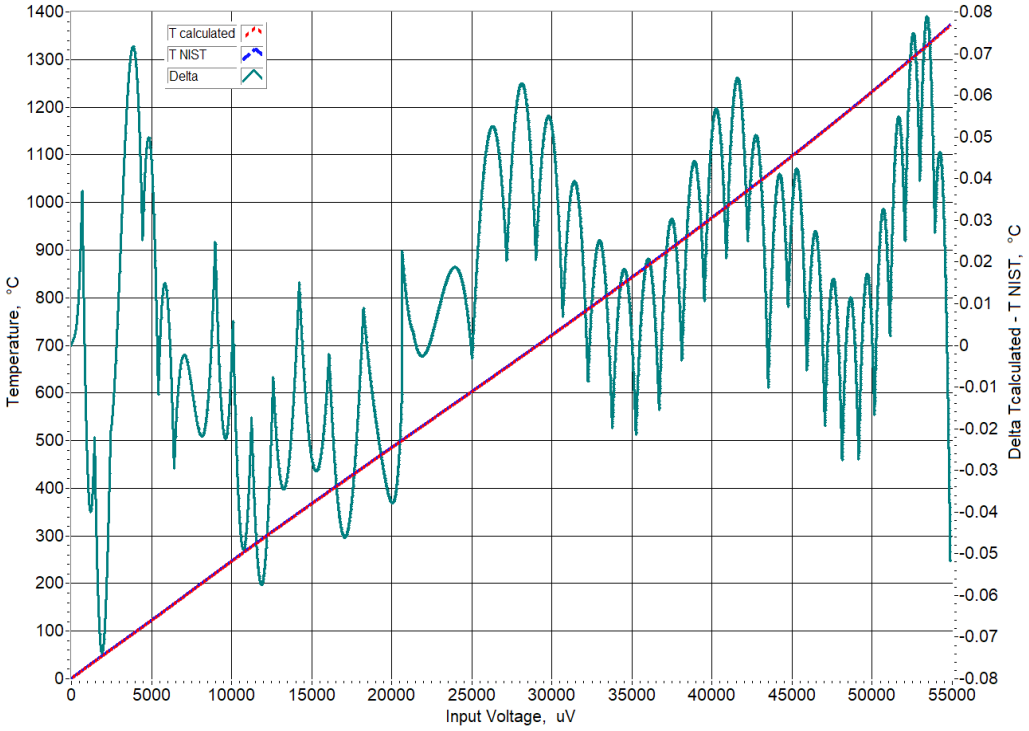

Figure 9. Linearized output in the range $t_{90} \in [0,\ 1371.655\ ^{\circ}\mathrm{C}]$.

We observed that the difference did not exceed the sum error of the approximation and the polynomial for the corresponding temperature range (Table 2).

Table 2. Error range for the temperature, t90, as a function of the thermoelectric voltage, in selected temperature and voltage ranges [37].

Temperature Range:	**−200 to 0 °C**	**0 to 500 °C**	**500 to 1 372 °C**
Voltage Range:	−5891 to 0 µV	0 to 20,644 µV	20 644 to 54,886 µV
Error Range:	0.04 °C to −0.02 °C	0.04 °C to −0.05 °C	0.06 °C to −0.05 °C

3.3. Linearization of the Inverse Characteristic of Type J Thermocouples in the Temperature Range $t_{90} \in [-210, 760\ ^\circ C]$

The thermoelectric voltage E in microvolts, as a function of temperature t_{90} in degrees Celsius, in the range $t_{90} \in [-210, 760\ ^\circ C]$ is defined by:

$$E(t_{90}) = \sum_{i=0}^{8} c_i\ (t_{90})^i, \tag{9}$$

where the coefficients c_i are specified in NIST ITS-90 [37].

With the help of the proposed approach, the problem of interval linearization of the inverse characteristic of the sensory characteristic in question, including inflection points, will be solved.

The graphs of the function $E(t_{90})$ and the first $E'(t_{90})$ and the second $E''(t_{90})$ its derivatives in the interval $t_{90} \in [-210, 760\ ^\circ C]$, are shown in Figure 10.

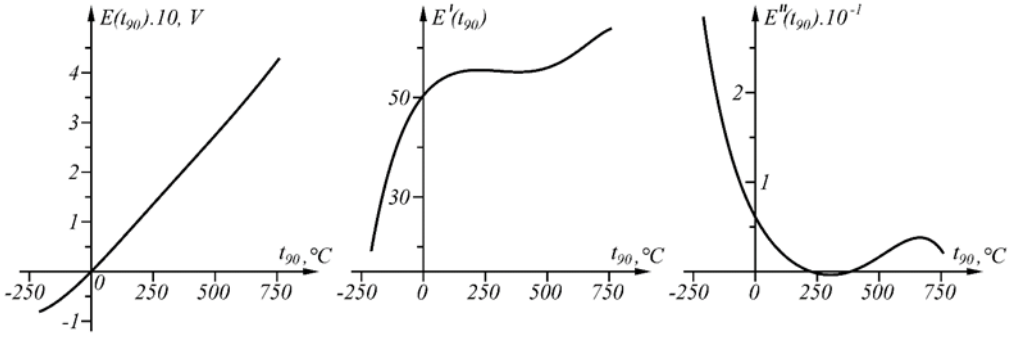

Figure 10. Functions $E(t_{90})$, its first derivative $E'(t_{90})$ and second derivative $E''(t_{90})$ in the interval $t_{90} \in [-210, 760\ ^\circ C]$.

The results of applying the approach at $|\Delta^{t_{90}}| = 0.348\ ^\circ C$ in the temperature range $t_{90_{A_1}} = -210\ ^\circ C$, $t_{90_{A_n \equiv B}} = 760\ ^\circ C$ are shown numerically in Table 3 and graphically in Figure 11, and the graph of the function $\Delta_i^{t_{90}}(t_{90})$ is shown in Figure 12.

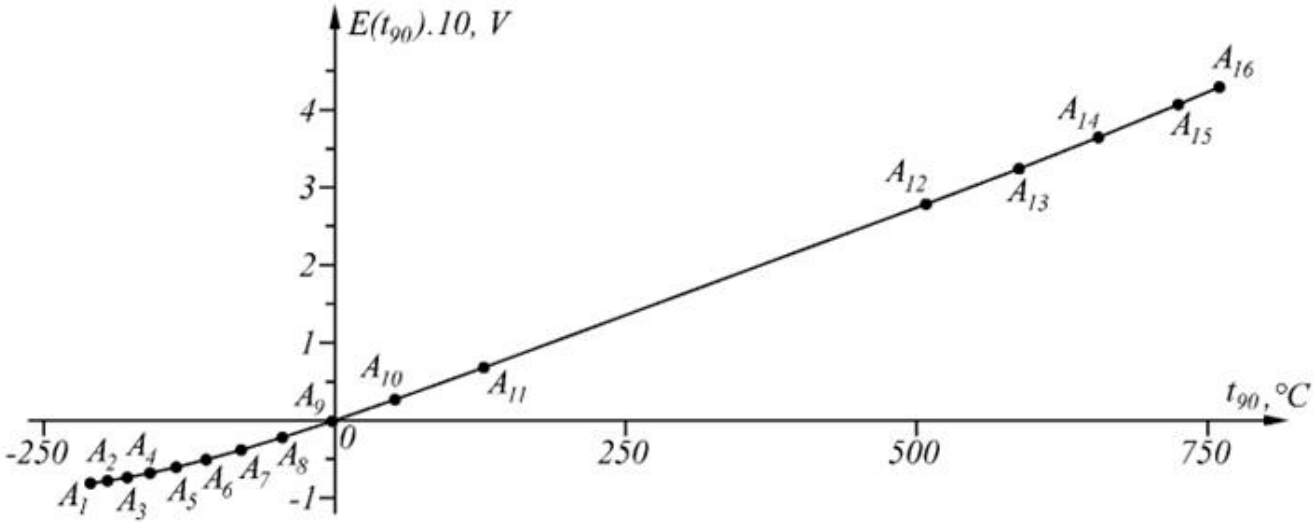

Figure 11. Segmentation in the temperature range $t_{90} \in [-210, 760\ ^\circ C]$.

Table 3. Interval linearization results in the range $t_{90} \in [-210, 760\ ^\circ\mathrm{C}]$.

A_1A_2 $A_1(-210,\ -8095.380)$ $A_2(-195.274,\ -7784.257)$	A_2A_3 $A_2(-195.274,\ -7784.257)$ $A_3(-178.302,\ -7356.617)$	A_3A_4 $A_3(-178.302,\ -7356.617)$ $A_4(-158.798,\ -6783.821)$	A_4A_5 $A_4(-158.798,\ -6783.821)$ $A_5(-136.378,\ -6031.463)$
$t_{90} \in [-210,\ -195.274]$	$t_{90} \in [-195.274,\ -178.302]$	$t_{90} \in [-178.302,\ -158.798]$	$t_{90} \in [-158.798,\ -136.378]$
$t_{90_{E_1}} = -202.698$	$t_{90_{E_2}} = -186.870$	$t_{90_{E_3}} = -168.660$	$t_{90_{E_4}} = -147.735$
$\Delta_1^{t_{90}} = 0.348$	$\Delta_2^{t_{90}} = 0.348$	$\Delta_3^{t_{90}} = 0.348$	$\Delta_4^{t_{90}} = 0.348$
A_5A_6 $A_5(-136.378,\ -6031.463)$ $A_6(-110.511,\ -5056.892)$	A_6A_7 $A_6(-110.511,\ -5056.892)$ $A_7(-80.438,\ -3804.623)$	A_7A_8 $A_7(-80.438,\ -3804.623)$ $A_8(-45.000,\ -2197.052)$	A_8A_9 $A_8(-45.000,\ -2197.052)$ $A_9(-2.242,\ -112.816)$
$t_{90} \in [-136.378,\ -110.511]$	$t_{90} \in [-110.511,\ -80.438]$	$t_{90} \in [-80.438,\ -45.000]$	$t_{90} \in [-45.000,\ -2.242]$
$t_{90_{E_5}} = -123.643$	$t_{90_{E_6}} = -95.749$	$t_{90_{E_7}} = -63.115$	$t_{90_{E_8}} = -24.233$
$\Delta_5^{t_{90}} = 0.348$	$\Delta_6^{t_{90}} = 0.348$	$\Delta_7^{t_{90}} = 0.348$	$\Delta_8^{t_{90}} = 0.348$
A_9A_{10} $A_9(-2.242,\ -112.816)$ $A_{10}(51.729,\ 2676.741)$	$A_{10}A_{11}$ $A_{10}(51.729,\ 2676.741)$ $A_{11}(128.045,\ 6801.389)$	$A_{11}A_{12}$ $A_{11}(128.045,\ 6801.389)$ $A_{12}(507.969,\ 27,839.325)$	$A_{12}A_{13}$ $A_{12}(507.969,\ 27,839.325)$ $A_{13}(587.888,\ 32,396.359)$
$t_{90} \in [-2.242,\ 51.729]$	$t_{90} \in [51.729,\ 128.045]$	$t_{90} \in [128.045,\ 507.969]$	$t_{90} \in [507.969,\ 587.888]$
$t_{90_{E_9}} = 23.657$	$t_{90_{E_{10}}} = 87.116$	$t_{90_{E_{11}}} = 448.839$	$t_{90_{E_{12}}} = 549.777$
$\Delta_9^{t_{90}} = 0.348$	$\Delta_{10}^{t_{90}} = 0.348$	$\Delta_{11}^{t_{90}} = 0.348$	$\Delta_{12}^{t_{90}} = 0.348$
$A_{13}A_{14}$ $A_{13}(587.888,\ 32,396.359)$ $A_{14}(656.077,\ 36,437.746)$	$A_{14}A_{15}$ $A_{14}(656.077,\ 36,437.746)$ $A_{15}(724.711,\ 40,678.242)$	$A_{15}A_{16}$ $A_{15}(724.711,\ 40,678.242)$ $A_{16}(760,\ 42,918.641)$	
$t_{90} \in [587.888,\ 656.077]$	$t_{90} \in [656.077,\ 724.711]$	$t_{90} \in [724.711,\ 760]$	
$t_{90_{E_{13}}} = 622.514$	$t_{90_{E_{14}}} = 689.931$	$t_{90_{E_{15}}} = 741.725$	
$\Delta_{13}^{t_{90}} = 0.348$	$\Delta_{14}^{t_{90}} = 0.348$	$\Delta_{15}^{t_{90}} = 0.065$	

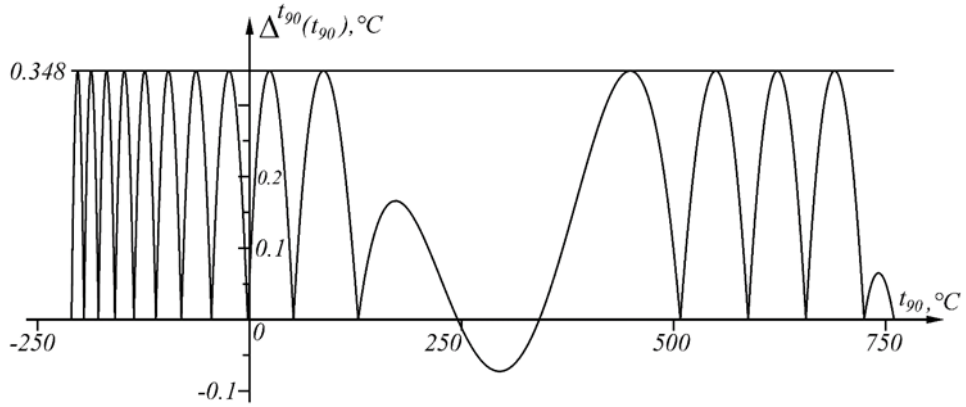

Figure 12. Graphical representation of the absolute error $\Delta_i^{t_{90}}(t_{90})$ in the range $t_{90} \in [-210, 760\ ^\circ\mathrm{C}]$.

4. Conclusions

This paper presented a generalized approach for linear interval approximation of sensor characteristics $y = y(x)$, representing differentiable functions where the sign of the curvature changed, i.e., including inflection points. The goal was to obtain in a discrete form the inverse sensory characteristic in the species $x_i = x_i(y_i), i = \overline{1,n}$, at a pre-set maximum error Δ^x, while minimizing the number of points determining the characteristic. This approach enabled the utilization of low-cost microcontrollers.

The approach, similar to the approach presented in [1], was characterized by the fact that when the maximum approximation error is set Δ^x in the linearization of the inverse sensing characteristic, the inverse sensing characteristic $x_i = x_i(y_i), i = \overline{1,n}$ is found directly in linearized form. The advantages of the herein-developed approach are as follows:

- The approach is applied in intervals, and at each subsequent step (each subsequent interval), as a result of analogously solving the task under the new initial conditions, the desired solution is obtained directly, containing, in turn, the initial conditions for the next step;
- The maximum linearization error of the inverse response of the sensor in all but the last interval is the same;
- The approach makes it possible to set a different maximum predefined error bound in each subsequent interval.

The proposed generalized approach was tested in the linearization of the inverse sensor characteristic of Type K thermocouples [37]. When the maximum approximation error is set $\left|\Delta^{t_{90}}\right| = 0.04$ °C in interval $t_{90} \in [0, 1371.655$ °C], the inverse sensory characteristic $t_{90} = t_{90}(E)$ (representing a broken line) is described by 39 points (Figure 6).

The proposed generalized approach was also tested in the linearization of the inverse sensor characteristic of Type J thermocouples [39]. When the maximum approximation error is set $\left|\Delta^{t_{90}}\right| = 0.348$ °C in interval $t_{90} \in [-210, 760$ °C], the inverse sensor characteristic $t_{90} = t_{90}(E)$ (representing a broken line) is described by 16 points (Figure 11).

Author Contributions: Conceptualization, M.B.M., L.K., N.N. and S.D.; methodology, N.N., S.D., Y.S. and T.T., software, N.N., B.G., G.T.N. and Y.S.; investigation, N.N., M.B.M. and B.G.; resources, G.T.N., N.N., M.B.M. and T.T.; writing—original draft preparation, N.N., S.D. and M.B.M.; 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: This research was funded by Project BG05M2OP001-1.001-0008 "National Center for Mechatronics and Clean Technologies" Operational Program Executive Agency, Ministry of Education and Science, Bulgaria.

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.

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

References

1. 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. [CrossRef] [PubMed]
2. Li, Z.; Liu, K.; Su, Y.; Ma, Y. Adaptive resource allocation algorithm for internet of things with bandwidth constraint. *Trans. Tianjin Univ.* **2012**, *18*, 253–258. [CrossRef]
3. 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.
4. Oriwoh, E.; Conrad, M. 'Things' in the Internet of Things: Towards a definition. *Int. J. Internet Things* **2015**, *4*, 1–5.
5. 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]
6. Hossain, M.M.; Fotouhi, M.; Hasan, R. Towards an Analysis of Security Issues, Challenges, and Open Problems in the Internet of Things. In Proceedings of the 2015 IEEE World Congress on Services, New York, NY, USA, 27 June–2 July 2015.
7. 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]
8. Chevrier, M. *TI Designs. Optimized Sensor Linearization for Thermocouple. TIDUA11A*; Texas Instruments Incorporated: Dallas, TX, USA, 2015; (revised September 2015).
9. 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.
10. 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]
11. Erdem, H. Implementation of software-based sensor linearization algorithms on low-cost microcontrollers. *ISA Trans.* **2010**, *49*, 552–558. [CrossRef] [PubMed]

12. Johnson, C. *Process Control Instrumentation Technology*, 8th ed.; Pearson Education Limited: London, UK, 2013.
13. Lundström, H.; Mattsson, M. Modified Thermocouple Sensor and External Reference Junction Enhance Accuracy in Indoor Air Temperature Measurements. *Sensors* **2021**, *21*, 6577. [CrossRef] [PubMed]
14. Anandanatarajan, R.; Mangalanathan, U.; Gandhi, U. Linearization of Temperature Sensors (K-Type Thermocouple) Using Polynomial Non-Linear Regression Technique and an IoT-Based Data Logger Interface. *Exp. Tech.* **2022**. [CrossRef]
15. Marinov, M.; Dimitrov, S.; Djamiykov, T.; Dontscheva, 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.
16. Šturcel, J.; Kamenský, M. Function approximation and digital linearization in sensor systems. *ATP J.* **2006**, *1*, 13–17.
17. Flammini, A.; Marioli, D.; Taroni, A. Transducer output signal processing using an optimal look-up table in mi-crocontroller-based systems. *Electron. Lett.* **2010**, *33*, 552–558.
18. 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]
19. Islam, T.; Mukhopadhyay, S. Linearization of the sensors characteristics: A review. *Int. J. Smart Sens. Intell. Syst.* **2019**, *12*, 1–21. [CrossRef]
20. Van der Horn, G.; Huijsing, J. *Integrated Smart Sensors: Design and Calibration*; Springer: New York, NY, USA, 2012.
21. Ghosh, R.; Nag, S.; Gupta, R. A Software-based Linearization Technique for Thermocouples using Recurrent Neural Network. In Proceedings of the 2021 IEEE Mysore Sub Section International Conference (MysuruCon), Hassan, India, 24–25 October 2021; pp. 302–306.
22. Srinivasan, K.; Sarawade, P.D. An Included Angle-Based Multilinear Model Technique for Thermocouple Linearization. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 4412–4424. [CrossRef]
23. Berahmand, K.; Mohammadi, M.; Saberi-Movahed, F.; Li, Y.; Xu, Y. Graph Regularized Nonnegative Matrix Factorization for Community Detection in Attributed Networks. *IEEE Trans. Netw. Sci. Eng.* **2022**, *10*, 372–385. [CrossRef]
24. Nasiri, E.; Berahmand, K.; Li, Y. Robust graph regularization nonnegative matrix factorization for link prediction in attributed networks. *Multimed. Tools Appl.* **2022**, *82*, 3745–3768. [CrossRef]
25. Yi, B.K.; Faloutsos, C. Fast time sequence indexing for arbitrary *Lp* norms. In Proceedings of the International Conference on Very Large Data Bases, San Francisco, CA, USA, 10–14 September 2000.
26. Popivanov, I.; Miller, R. Similarity search over time-series data using wavelets. In Proceedings of the 18th International Conference on Data Engineering, San Jose, CA, USA, 26 February–1 March 2002.
27. Rafiei, D.; Mendelzon, A. Similarity-based queries for time series data. In Proceedings of the IEEE International Conference on Data Engineering, Sydney, Australia, 23 March–26 March 1999.
28. Cai, Y.; Ng, R. Indexing spatio-temporal trajectories with Chebyshev polynomials. In Proceedings of the ACM SIGMOD International Conference on Management of Data, New York, NY, USA, 13–18 June 2004.
29. Palpanas, T.; Vlachos, M.; Keogh, E.; Gunopulos, D.; Truppel, W. Online amnesic approximation of streaming time series. In Proceedings of the IEEE International Conference on Data Engineering, Boston, MA, USA, 2 April 2004.
30. Chen, Q.; Chen, L.; Lian, X.; Liu, Y.; Yu, J.X. Indexable PLA for efficient similarity search. In Proceedings of the International Conference on Very Large Data Bases, Vienna, Austria, 23–27 September 2007.
31. Cameron, S.H. Piece-Wise linear approximations, DTIC Document. *Tech. Note* **1966**. [CrossRef]
32. Luo, G.; Yi, K.; Cheng, S.W.; 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, Republic of Korea, 13–17 April 2015.
33. 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, 28 April 2007.
34. Haney, Library for Accurate Pt100 RTD Ohms-to-Celsius Conversion. 2018. Available online: https://github.com/drhaney/pt100rtd/tree/master/examples/pt100_temperature (accessed on 19 November 2021).
35. 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 2001.
36. Duff, M.; Towey, J. Two Ways to Measure Temperature Using Thermocouples Feature Simplicity, Accuracy, and Flexibility. *Analog. Dialogue Vols.* **2010**, 1–6.
37. Candela, G. *TI Designs. Isolated Loop Powered Thermocouple Transmitter. TIDU449B*; Texas Instruments Incorporated: Dallas, TX, USA, 2014; (revised July 2016).
38. Rembor, K. Adafruit ESP32 Feather V2. 1 December 2022. Available online: https://cdn-learn.adafruit.com/downloads/pdf/adafruit-esp32-feather-v2.pdf (accessed on 8 December 2022).
39. Systems, E. ESP32PICOMINI02 Datasheet. 2022. Available online: https://www.espressif.com/sites/default/files/documentation/esp32-pico-mini-02_datasheet_en.pdf (accessed on 8 December 2022).