sensors

MDPI

Article

Machine Learning and Swarm Optimization Algorithm in Temperature Compensation of Pressure Sensors

Hexing Wang [1,2,*] and Jia Li [1,2]

[1] Institute of Microelectronics of the Chinese Academy of Sciences, Beijing 100029, China
[2] University of Chinese Academy of Sciences, Beijing 100049, China
* Correspondence: wanghexing@ime.ac.cn

Abstract: The main temperature compensation method for MEMS piezoresistive pressure sensors is software compensation, which processes the sensor data using various algorithms to improve the output accuracy. However, there are few algorithms designed for sensors with specific ranges, most of which ignore the operating characteristics of the sensors themselves. In this paper, we propose three temperature compensation methods based on swarm optimization algorithms fused with machine learning for three different ranges of sensors and explore the partitioning ratio of the calibration dataset on Sensor A. The results show that different algorithms are suitable for pressure sensors of different ranges. An optimal compensation effect was achieved on Sensor A when the splitting ratio was 33.3%, where the zero-drift coefficient was 2.88 $\times$ 10^{-7}/°C and the sensitivity temperature coefficient was 4.52 $\times$ 10^{-6}/°C. The algorithms were compared with other algorithms in the literature to verify their superiority. The optimal segmentation ratio obtained from the experimental investigation is consistent with the sensor operating temperature interval and exhibits a strong innovation.

Keywords: pressure sensor; temperature compensation; differential evolution algorithm; quantum particle swarm optimization; dataset partitioning

check for updates

Citation: Wang, H.; Li, J. Machine Learning and Swarm Optimization Algorithm in Temperature Compensation of Pressure Sensors. *Sensors* **2022**, *22*, 8309. https://doi.org/10.3390/s22218309

Academic Editor: Roberto Teti

Received: 6 September 2022
Accepted: 26 October 2022
Published: 29 October 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

Micro-electron mechanical systems (MEMSs) typically consist of mass-fabricable power sources, micromachines, micro-actuators and channels with feature sizes mostly on the order of micrometers [1–3]. Pressure sensors are one of the most widely used types of MEMS sensors that convert pressure signals received from the outside world into electrical signals and signal output [4,5], which have been widely used in consumer electronics, clinical medicine, autonomous driving and other fields. Compared with traditional pressure sensors composed of metal strain gauges, MEMS pressure sensors are produced based on semiconductor materials and processing technology, such as silicon, which allows these sensors to have the advantages of high sensitivity, low cost, miniaturization and easy integration. According to the working principle of pressure sensors, MEMS pressure sensors can be divided into piezoresistive, piezoelectric, capacitive, resonant and other types [6–8]. However, the generation of a piezoresistive effect needs to interact with external physical quantities, leading to an offset in output accuracy in daily work, which will be affected by the environmental temperature, atmospheric humidity, external pressure and other factors. How to improve the output accuracy of piezoresistive pressure sensors has become one of the topics of widespread concern in science and industry.

Currently, after extensive exploration, two common methods have been established to compensate for the sensor output: hardware compensation and software compensation [9]. Hardware compensation is a process of improving the accuracy of compensation for different drift scenarios through circuit design [10]. The circuit structure of the hardware compensation method is relatively simple, which has been widely used in engineering

applications [11]. However, with the optimization and perfection of the packaging process, most of the sensors are put into use after the completion of the packaging, making it difficult to make subsequent adjustments after the completion of the hardware compensation circuit design. Furthermore, the change in the external temperature will have an irreversible influence on the components in the compensation circuit; thus, the error will tend to be significant with the growth of the service time [12]. Therefore, software compensation is gradually replacing hardware compensation in piezoresistive pressure sensors.

Software compensation is a process that involves improving the accuracy of piezoresistive pressure sensors by processing the collected output data as input by various algorithms [13]. According to the different types of algorithms, software compensation can be divided into parametric compensation and non-parametric compensation, based on intelligent optimization algorithms and machine learning [14]. Non-parametric compensation can solve the drawback that parametric algorithms cannot handle large-volume data, thus becoming the mainstream temperature compensation path [15–17]. The temperature compensation method based on numerical calculation has some limitations when the measurement accuracy of the pressure sensor is high. If the order of the fitting expression is higher than that of the least-squares fitting method for temperature compensation, an ill-conditioned problem will arise when solving the normal equation, meaning that a stable solution cannot be obtained. Machine learning models have powerful nonlinear approximation ability, overcoming some limitations of numerical calculation and providing a new choice for the temperature compensation of high-precision pressure sensors [18–20]. Suykens [21] introduced the least-squares support vector machine (LSSVM), which converts the inequality constraints in the traditional SVM to linear equations in the framework of regularization theory. Yao [22] optimized the wavelet neural network model and compensated the sensor temperature based on it. The results showed that the sensor temperature drift error can be optimized to 0.2%. Wang [23] proposed a temperature compensation method based on an improved cuckoo search optimizing a BP neural network for a multi-channel pressure scanner. The maximum full-scale error of all 32 channels is 0.02% FS (full-scale error) and below, which realizes its high-accuracy multi-point pressure measurement in a wide temperature range.

However, few algorithms are designed for sensors with specific ranges according to the existing research, and most methods ignore the operating characteristics of the sensors themselves [24–27]. This study considered this situation as an innovation point and proposes three temperature compensation algorithms, performing data acquisition and compensation for sensors with different ranges to investigate the range of the proposed algorithms, along with the optimal partitioning ratio. The results show that the intelligent optimization algorithms combined with the machine learning compensation method can improve the voltage output characteristics of the sensor and achieve higher compensation accuracy. Moreover, different compensation algorithms have different demands on the range of the sensor, which greatly improves the universality of the proposed algorithms. An innovative idea for algorithm compensation is provided through dataset segmentation based on the actual working environment temperature of the sensor.

2. Methods

2.1. Performance Parameters

Zero drift refers to the phenomenon in which the voltage across the bridge is not zero, primarily owing to the temperature change when the external stress of the pressure sensor is zero (i.e., at standard atmospheric pressure). The zero-drift coefficient is expressed as follows:

$$\alpha_0 = \frac{U_{0\max} - U_{0\min}}{U_{FS}(T_{\max} - T_{\min})} \quad (1)$$

where $T_{\max}$ and $T_{\min}$ denote the maximum and minimum values of the external temperature, respectively, when the sensor is working, and $U_{0\max}$ and $U_{0\min}$ denote the maximum and minimum values of the output voltages of the sensor under no-load operation. A

smaller zero-drift coefficient signifies that the output of the device is less significantly affected by the temperature and thus has superior output characteristics.

Sensitivity temperature drift refers to the phenomenon in which the voltage across the bridge increases nonlinearly as the input pressure increases when the operating ambient temperature rises or falls. The coefficient is defined as follows:

$$\alpha_s = \frac{U_{FS\max} - U_{FS\min}}{U_{FS\max}(T_{\max} - T_{\min})} \tag{2}$$

where U_{FS} represents the output-voltage values at different temperatures under full-load operation. A smaller sensitivity temperature coefficient signifies that the output of the device is less significantly affected by the temperature and has superior output characteristics.

In this study, the zero-drift coefficient and sensitivity temperature coefficient were used to reflect the output accuracy of the sensor.

2.2. Differential Evolution Algorithm

The differential evolution (DE) algorithm is optimized using a genetic algorithm, which simulates the biological evolution process. It is based on a real number encoding strategy and improves the global search ability by providing the variation probability, and its variation mode is simple [28,29]. In addition, it features a real-time adjustable search strategy and improves the running speed of the algorithm.

The core steps of the differential evolution algorithm are the same as those of a traditional genetic algorithm, including mutation, crossover and selection. Before performing the mutation operation, the population must be initialized as follows:

$$P(X) = \left\{ x_{j,i,g} \middle| j = 0, 1, \ldots D; i = 0, 1, \ldots, N_p; g = 0, 1, \ldots, g_{\max} \right\} \tag{3}$$

where X denotes the population individual, x indicates the individual gene, j indicates the chromosome number, i denotes the individual number, N_p indicates the population size, g corresponds to the number of generations in the population and $g_{\max}$ indicates the maximum number of iterations, which, in this study, is considered to be 50. The difference vector is obtained by calculating all the genes of two random individuals in the population and acting on the third individual to complete the mutation operation, given the mutation probability. The least-squares error signal function was chosen in this study, which can be expressed as follows:

$$E_k = \frac{1}{2} \sum_{j=1}^{o} \left\| {y_j}^{n\prime} - {y_j}^{n} \right\|^2. \tag{4}$$

where y denotes the input sample vector, and y' indicates the output vector after passing through the multilayer perceptron.

The initial weights and thresholds of the BP algorithm are randomly generated, and when the size of the dataset increases, they may fall into local minima during the training. An optimal solution search can be performed using the DE algorithm prior to initializing the parameters to enhance the ability of the BP algorithm to perform a global search. The flow of the temperature compensation algorithm proposed in this study is shown in Figure 1, and the steps are detailed below.

(1) Firstly, the BP network weights and thresholds were initialized, and then they were encoded with inputs for the DE algorithm.

(2) Then, variation, crossover and selection operations were performed to achieve population evolution, and the least-squares error signal function was selected for the fitness function in this study.

(3) After the iteration was completed, the optimal weights and thresholds were obtained and used as the initialization parameters of the BP network. The learning rate was set to 0.01, and the momentum coefficient was set to 0.8. The BP network structure consisted of one input layer, three implicit layers and one output layer.

(4) The algorithm was executed 10 times to eliminate the randomness error of the BP network and then averaged to obtain the final output-voltage results.

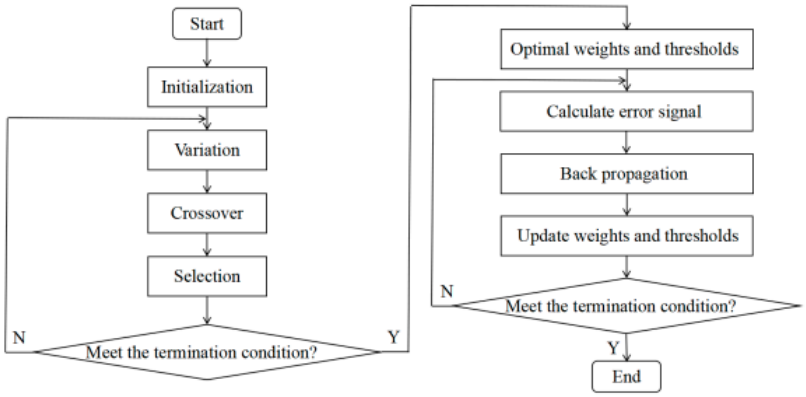


Figure 1. Temperature compensation flow using the DE algorithm.

2.3. Quantum Particle Swarm Optimization Algorithm

The quantum particle swarm optimization (QPSO) algorithm is optimized by PSO, where the particle displacement velocity is affected by both its historical and global optimal solutions [30,31]. In the PSO algorithm, the individual particle is affected by the local attractor, so the search result may fall into the local optimal solution and reduce the output accuracy of the algorithm. The QPSO algorithm retains the role of a local attractor and regards the search space as a quantum space. The position of particles appearing in space during each iteration follows the Schrodinger equation [32]. The Monte Carlo method can solve the probability of particles appearing in any position in space, and the solution results are as follows:

$$x_i^{g+1} = a_i^g \pm \frac{1}{2} L_i^g \ln\left(\frac{1}{r_i^{g+1}}\right) \tag{5}$$

$$L_i^g = \beta \left| C^g - x_i^g \right| \tag{6}$$

$$C^g = \frac{1}{N} \sum_{i=1}^{N} P_{ibest}^g \tag{7}$$

where L denotes the well length of the particle, C indicates the average of the optimal positions of all particles in space, N denotes the number of particles and β indicates the constriction factor. It can be seen that the coordinate points of the particles in the iteration are related to the average value of the optimal position of all particles in the QPSO algorithm, which weakens the potential energy effect of the local attractor and makes the algorithm more inclined to search for the overall optimal solution.

The algorithm can also improve the search efficiency for the overall optimal solution of the BP neural network. The flow of the temperature compensation algorithm is shown in Figure 2, and the steps are detailed below.

(1) The BP network weights and thresholds were initialized before they were encoded as inputs for the QPSO algorithm.

(2) The historical optimal value and global optimal value of particles were calculated, and then the optimal solution was found by comparing the fitness value after substituting in the fitness function.

(3) After the iteration was completed, the optimal weights and thresholds were obtained and used as the initialization parameters of the BP network. The structure of the BP network is the same as that of the DE algorithm.

(4) The algorithm was executed 10 times to eliminate the randomness error of the BP network and then averaged to obtain the final output-voltage results.

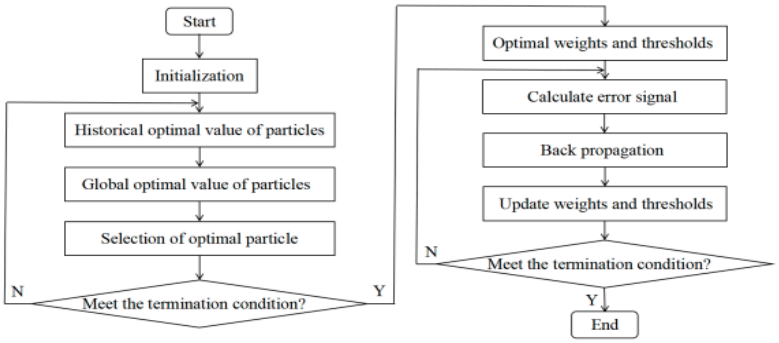


Figure 2. Temperature compensation flow using the QPSO algorithm.

2.4. Fruit Fly Optimization Algorithm

The fruit fly optimization algorithm (FOA) is an intelligent optimization algorithm that integrates the biological characteristics of fruit flies into the algorithm. The core idea is the colony characteristics of fruit flies, which can plan the foraging path of the group using the concentration of the surrounding food [33,34]. Global optimization is achieved through interaction between different groups.

Based on the characteristics of the FOA, it can be combined with the least-squares support vector machine (LSSVM), which uses the output error signal to replace the relaxation variables and optimizes the inequality relations in the constraint conditions, converting them into equality relations in order to reduce the algorithm complexity and speed up the model solving speed. The flow of the temperature compensation algorithm is shown in Figure 3, and the steps are detailed below.

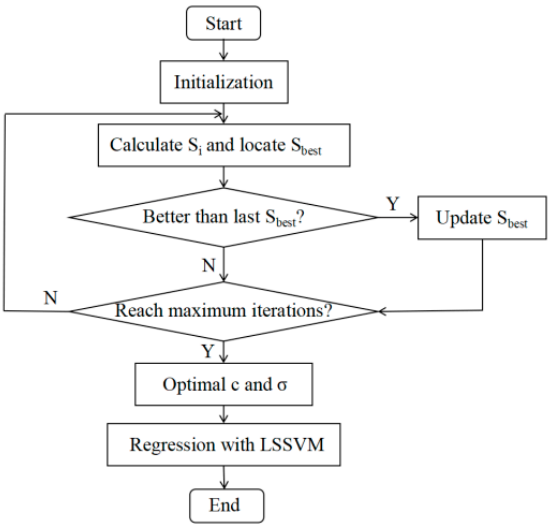


Figure 3. Temperature compensation flow using the FOA.

(1) The location of the fruit fly population was initialized using the Init function.

(2) The direction and distance of the fruit fly foraging were set, which can be expressed as follows:

$$\begin{cases} X(i,:) = X_{axis} + L_x \\ Y(i,:) = Y_{axis} + L_y \end{cases} \quad (8)$$

where X_{axis} and Y_{axis} denote the initial position of the fruit fly population, and L_x and L_y indicate the distances of horizontal and vertical movements.

(3) The distance formula D was used to calculate the distance of Drosophila from the initial position after flight, and the judgment value of the flavor concentration S was computed by taking the inverse, which are defined as follows:

$$\begin{cases} D(i,1) = \sqrt{X(i,1)^2 + Y(i,1)^2} \\ D(i,2) = \sqrt{X(i,2)^2 + Y(i,2)^2} \end{cases} \tag{9}$$

$$\begin{cases} S(i,1) = \frac{1}{D(i,1)} \\ S(i,2) = \frac{1}{D(i,2)} \end{cases} \tag{10}$$

(4) The value of the taste concentration was substituted into the fitness function to obtain the individual taste intensity S_i of the fruit fly.

(5) S_i and S_best were compared for different individual taste concentrations of fruit flies so that the initial position of fruit flies was updated. X_{axis} and Y_{axis} are defined as the penalty factor and kernel function parameters of the LSSVM, respectively.

(6) The optimal penalty factor and kernel function parameters were obtained through repeated iteration to proceed to the regression prediction.

(7) The algorithm was executed 10 times and then averaged to obtain the final output-voltage results.

2.5. Dataset Segmentation

When the scale of the calibration data is large, using all of them as the algorithm input will increase the complexity of the algorithm. Simultaneously, a small amount of data with large false fluctuations may reduce the compensation effect. Therefore, the dataset can be partitioned and divided into training and test sets based on different temperature intervals. If the temperature range is large during calibration, then some of the data within the operating temperature interval of the sensor can be used as the test set, whereas the remainder of the data can be used as the training set. This allows the piezoresistive sensor to determine the output compensation with high accuracy in the daily working environment temperature range. The splitting ratio is defined as follows:

$$\eta = \frac{\text{Test set data volume}}{\text{Test data volume}} \times 100\% \tag{11}$$

Different segmentation ratios can cause the output accuracy of the sensors to differ. The optimal segmentation ratio is explored in this study.

3. Experiments

3.1. Instruments and Equipment

Three sensors with different ranges were selected for data calibration in this experiment: MSPD700-ASO ("Sensor A"), CD0302-350KP-A ("Sensor B") and CD0302-70KP-A ("Sensor C"). The nonlinearity of each sensor was ±0.3% of the full-scale output. Since the packaged sensors were difficult to disassemble, the measurements were performed on bare pieces. The sensor models and their parameters are listed in Table 1.

Table 1. Sensor models and their parameters.

Sensor Model	Measurement Range (kPa)	Full-Scale Output (mV)	Operating Standard Temperature (°C)
Sensor A	700	100	20–100
Sensor B	350	90	
Sensor C	70	80	

The temperature test chamber is shown in Figure 4. The lower and upper limits of its operating temperature were −40 °C and 150 °C, which meet the requirements of

data calibration. The temperature test chamber could be heated and cooled by setting the expected temperature for data calibration.



Figure 4. Temperature test chamber.

The experimental platform is shown in Figure 5. The external pressure transformer acted on the wafer through the air gun, with the pressure ranging from 0 to 1 MPa, which can reduce the relative error of the actual pressure within 0.01%. To read the output signal, the probe should be placed in contact with the varistor of the wafer in the probe stand. The output data reading device embedded in the experimental platform must be set to constant current source mode in the experiment.

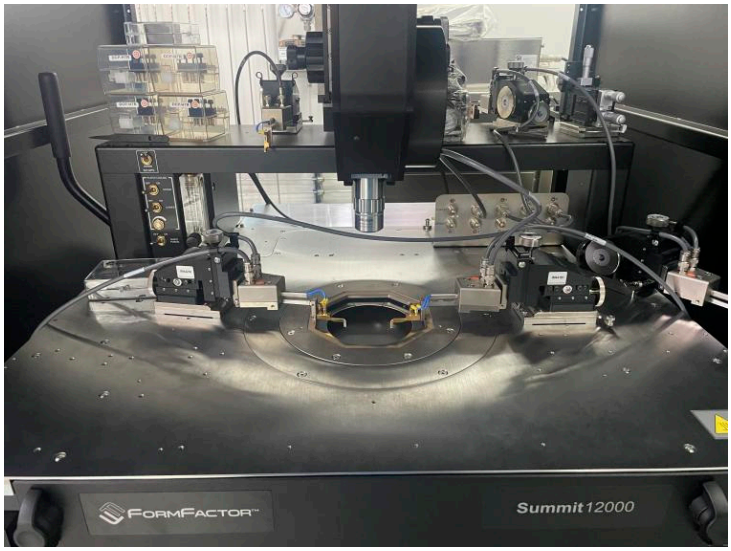


Figure 5. Experimental platform construction result.

The models and manufacturers of the instruments and equipment used in this experiment are shown in Table 2.

Table 2. Models and manufacturers of the instruments and equipment.

Instruments and Equipment	Model	Manufacturer
Sensor A	MSPD700-ASO	Memsensing Microsystems
Sensor B	CD0302-350KP-A	FATRI Technologies
Sensor C	CD0302-70KP-A	

Table 2. *Cont.*

Instruments and Equipment	Model	Manufacturer
Temperature test chamber	Systems att C-Serials	Angelantoni test technologies
Air gun	Cascade-DPP210	FormFactor
Experimental platform	Summit-12000	

3.2. Data Calibration Scheme

Data calibration methods can be divided into static and dynamic calibration. Static calibration follows a given temperature change step, which is suitable for small-scale data calibration. The output data in dynamic calibration are measured in real time during the process of temperature increase or decrease, which is appropriate for large-scale data calibration. In this study, the static calibration method was utilized to investigate the pressure range of the sensor to which the three algorithms apply at temperatures ranging from −20 to 100 °C, with a calibration step of 20 °C. Subsequently, the dynamic calibration method was used to investigate the optimal partitioning ratio of the dataset, with a temperature range from −20 to 150 °C and a calibration step of 10 °C. The specific steps of the static calibration were as follows.

(1) The temperature test chamber was opened and left to stand for 2–3 min to allow its temperature to reach room temperature.
(2) The initial temperature (−20 °C) and the initial input pressure (vacuum) were set. The output data were read after the temperature test chamber reached the set temperature. The given pressure change step according to the measuring range of the pressure sensors considered the algorithm complexity and the minimum change interval of the pressure gun. Data calibration was carried out after the pressure was changed and stabilized.
(3) The operation in step (2) was repeated to carry out data calibration until the temperature rose to 100 °C and reduced to −20 °C.
(4) To evaluate the effectiveness of the temperature compensation, each sensor was calibrated three times with the same conditions, and its repeatability was tested.
(5) The output experimental calibration data were collated. For a specific sensor, the average output value of the same temperature in the process of heating and cooling was calculated as the output voltage at that temperature.

3.3. Data Calibration Results

Since the characteristics of the three sensors are basically the same, only the data calibration results of Sensor A are presented in this paper. The calibration results are shown in Table 3. There were eight pressure test points and seven temperature test points. U_p denotes the output voltage of the sensor. U_T indicates the output data of the temperature compensation resistor. Two-dimensional static calibration curves for Sensor A are shown in Figure 6. It can be observed from the figure that the degree of drift in the output curve increases as the input pressure increases.

Table 3. Calibration results of Sensor A.

T/°C	U_T/V	U_p/mV							
		0 kPa	100 kPa	200 kPa	300 kPa	400 kPa	500 kPa	600 kPa	700 kPa
−20	2.734	15.822	27.715	39.868	51.081	62.051	73.053	81.189	90.960
0	2.956	15.246	26.749	38.503	49.610	59.616	70.340	78.521	87.714
20	3.220	14.637	25.680	36.744	47.558	58.022	67.892	75.844	84.052
40	3.481	14.064	24.724	34.926	45.849	55.005	64.325	73.158	80.622
60	3.823	13.680	23.257	33.487	43.827	52.343	60.589	69.685	75.907
80	4.149	12.590	21.603	32.055	41.864	50.377	59.250	67.476	73.394
100	4.395	11.483	20.172	29.598	39.075	47.833	55.626	63.391	70.044

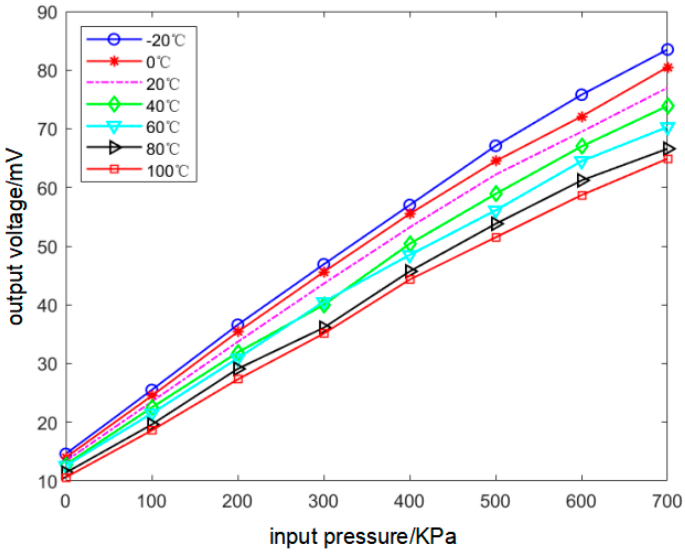


Figure 6. Output 2−D curves of Sensor A.

The zero−drift coefficients and sensitivity temperature coefficients of the three sensors are listed in Table 4. Sensor B had the best calibration data performance, whereas Sensor C had the worst performance, which can be attributed to the fact that the sensor die has been compensated in the hardware. However, the different models have different compensation accuracies, which in turn leads to a difference in the calibration data. The drift coefficients of the output voltages of the different sensors were large, which demonstrates the necessity for software compensation.

Table 4. Drift parameters of sensors before compensation.

Sensor Model	α_0 (°C)	α_s (°C)
Sensor A	4.03×10^{-4}	1.94×10^{-3}
Sensor B	3.99×10^{-4}	1.86×10^{-3}
Sensor C	4.04×10^{-4}	2.41×10^{-3}

3.4. Construction of Temperature Compensation Model

The temperature compensation model should be constructed first after the data calibration of the pressure sensor. From the description of the algorithms selected in this paper, it can be seen that different intelligent optimization algorithms are used to optimize the core parameters of the BP neural network and LSSVM. Therefore, the input of the BP neural network and LSSVM should be given to run the algorithm once the parameter optimization is completed.

As shown in Table 3, there were 56 calibration data (U_p) and 7 output data of the temperature compensation resistor (U_T). In the established model, the compensation pressure $U_{p'}$ is taken as the output result in order to make the calculation results universal. Therefore, U_p and U_T can be regarded as the input array of the algorithm for three different machine learning algorithms. The input array is normalized to accelerate the training speed of the BP neural network and LSSVM. When the machine learning algorithm with optimized parameters is finished running, the output compensation pressure results are also normalized. Therefore, the final compensation pressure data should be obtained after inverse normalization of the output results. This paper takes the output results of Sensor A compensated by the DE–BP algorithm shown in Table 5 as an example. The output results $U_{p'}$ under different calibration pressures were substituted into Equations (1) and (2) to calculate the compensated zero-drift coefficient and sensitivity temperature coefficient.

Table 5. Temperature compensation results of Sensor A based on the DE–BP algorithm.

T/°C	U_p/mV							
	0 kPa	**100 kPa**	**200 kPa**	**300 kPa**	**400 kPa**	**500 kPa**	**600 kPa**	**700 kPa**
−20	1.465	99.973	200.912	300.615	398.072	502.706	603.431	702.121
0	1.483	100.304	199.249	299.709	400.595	501.871	599.338	703.448
20	1.436	100.677	199.217	299.273	399.152	500.443	601.308	699.152
40	1.490	99.430	200.131	298.936	402.421	497.791	599.965	696.895
60	1.503	99.710	199.769	298.625	400.735	498.668	601.787	700.421
80	1.466	100.073	199.075	299.995	397.886	501.974	597.239	700.791
100	1.458	100.809	201.039	301.254	398.380	500.813	596.922	702.024

4. Selection and Optimization of Algorithm Parameters

4.1. Optimization of Parameters in DE

Before conducting temperature compensation based on the DE–BP algorithm, the variance probability M is optimized. In the early stage of the population iteration, the variation probability should be as high as possible to maintain the diversity of the population and prevent the global search performance from degrading toward a premature population maturity. In the late stage of the population iteration, an excessively high variation probability may cause the optimal solution to change once again, thereby increasing the search difficulty. Therefore, the probability of variation in the population is defined as follows:

$$\begin{cases} M = M_0 3^{\lambda} \\ \lambda = e^{-\frac{g}{g_{max}-g+1}+1} \end{cases} \quad (12)$$

where M_0 is the coefficient of variation.

It can be found that when the population is in the early stage of evolution, the probability of variation tends to $3M_0$, which increases the population diversity. When the individuals in the population evolve to a later stage, the probability of variation decreases to M_0, which is conducive to the retention of the optimal solution. In this study, experimental investigations were performed for the value of M_0. It takes the value interval of [0, 1], and the step size is 0.2. The performance results of the algorithm for different M_0 values are shown in Figure 7. From the figure, it can be observed that when M_0 was 0.6, the algorithm had the highest optimal solution search efficiency. Therefore, M_0 was set to 0.6.

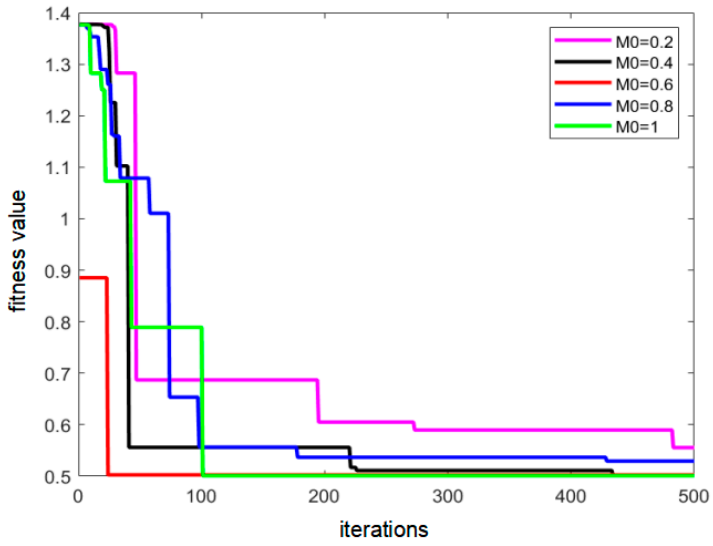


Figure 7. Schematic of algorithm performance under different M_0 values.

4.2. Selection of Parameters in QPSO

In the solution result of QPSO, the shrinkage factor is used. If it is a constant value, the global optimal solution may be missed in the later stage of the iteration, reducing the convergence speed of the algorithm. Therefore, it can be defined as a decreasing function to make the algorithm easier to converge in the later stage. In this paper, three definitions were adopted to define the decreasing function [35,36], which can be expressed as

$$\begin{cases} \beta_1 = k_1 + (k_1 - k_0)\frac{g_{max} - g}{g_{max}} \\ \beta_2 = (k_1 - k_0) \times (\frac{g}{g_{max}})^2 - (k_1 - k_0)(\frac{2g}{g_{max}}) + k_1 \\ \beta_3 = (k_0 - k_1) \times (\frac{g}{g_{max}})^2 + k_1 \end{cases} \quad (13)$$

where k_0 and k_1 are constants. In this paper, k_0 and k_1 were set to 0.5 and 1.

The curves of the three decreasing functions are shown in Figure 8. It can be seen that their algorithm performances are similar. In order to reduce the complexity of the algorithm, β_1 was chosen in this paper.

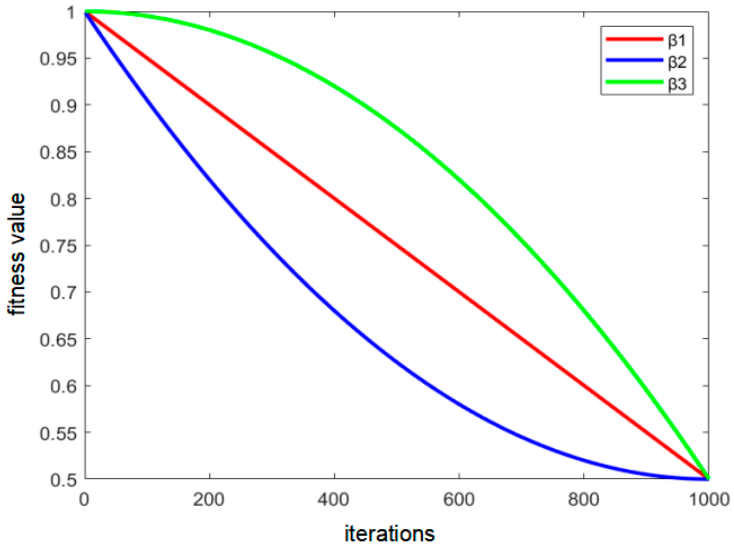


Figure 8. Diagram of the decreasing function.

4.3. Optimization of Parameters in FOA

In the sensor temperature compensation algorithm based on the Drosophila optimization algorithm and LSSVM, a random number function is usually used when Drosophila moves in random steps after initialization, which has some defects. The quality of the search differs in diverse periods of the population iteration. It is closer to the global optimal solution in the middle and later periods of the iteration. If the population does not obtain the global optimal solution at this time, the global optimal solution may be missed when the random search step is large. Provided that the random search step size is small at the beginning of the algorithm iteration, it will reduce the search efficiency of the optimal value and affect the output accuracy of the algorithm. Accordingly, this paper redefined the moving step as

$$\begin{cases} L = \lambda L_{max} + L_{min} \\ \lambda = e^{-\frac{g}{g_{max} - g + 1} + 1} \end{cases} \quad (14)$$

where L_{max} denotes the maximum value of the moving step, which is set to 5, L_{min} indicates the minimum value of the moving step, g corresponds to the current iteration number, g_{max} denotes the maximal number of iterations and p is the regulatory factor, which is usually set to 1.

This paper conducted an experimental study on the value of L_{min}, which varied from 0 to 0.02 under a step of 0.005. The performance results of the algorithm for different L_{min}

values are shown in Figure 9. It can be seen that the optimal solution search efficiency of the algorithm was the lowest when L_{min} was 0, and the optimal solution search efficiency of the algorithm was the highest when L_{min} was 0.02. Therefore, L_{min} was set to 0.02.

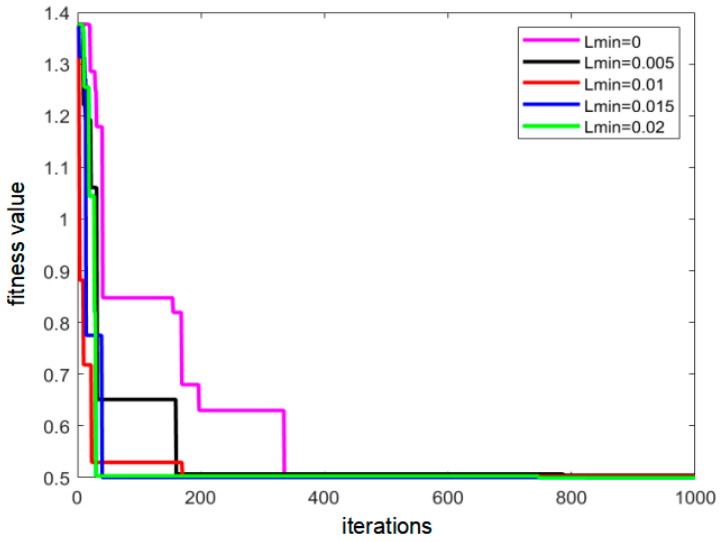


Figure 9. Schematic diagram of algorithm performance under different minimum L_{min} values.

5. Results and Discussion

5.1. Temperature Compensation Results Based on Calibration Data

The calibration data of different sensors were substituted into the different algorithms to perform temperature compensation, and the comparison of the results is shown in Table 6.

Table 6. Comparison of temperature compensation results based on different algorithms.

Sensor Model		Non-Compensation	DE–BP	QPSO–BP	FOA–LSSVM
Sensor A	$\alpha_0/°C$	4.03×10^{-4}	7.94×10^{-7}	7.37×10^{-7}	7.51×10^{-7}
	$\alpha_s/°C$	1.94×10^{-3}	7.76×10^{-5}	7.92×10^{-5}	8.07×10^{-5}
Sensor B	$\alpha_0/°C$	3.99×10^{-4}	7.35×10^{-7}	8.06×10^{-7}	7.29×10^{-7}
	$\alpha_s/°C$	1.86×10^{-3}	7.13×10^{-5}	8.21×10^{-5}	7.29×10^{-5}
Sensor C	$\alpha_0/°C$	4.04×10^{-4}	5.93×10^{-7}	8.93×10^{-7}	8.98×10^{-7}
	$\alpha_s/°C$	2.41×10^{-3}	5.95×10^{-5}	8.78×10^{-5}	8.62×10^{-5}

It can be seen from the table that the output accuracy of the three sensors was significantly improved after the compensation of the different types of algorithms, which reflects the advantages of combining the simulation of biological population characteristics with machine learning. The QPSO–BP algorithm had the smallest zero-drift coefficient in Sensor A compensation. The FOA–LSSVM algorithm had the optimal zero-drift coefficient value in Sensor B compensation. The DE–BP algorithm had the best performance in terms of the sensitivity temperature coefficient of the three sensors after temperature compensation, as well as the best training effect on the zero-drift coefficient of Sensor C. These compensation results show that different intelligent optimization algorithms may have different temperature compensation results for different sensor calibration data, which is caused by the randomness of traditional machine learning. In this experiment, the DE–BP algorithm achieved a better compensation effect than the other two intelligent optimization algorithms, and Sensor A was suitable for more algorithms. Therefore, the DE–BP algorithm was used to segment the dataset for Sensor A, as shown in the following. The comparison of the compensation results of different sensors under different algorithms is visually shown in Figure 10.

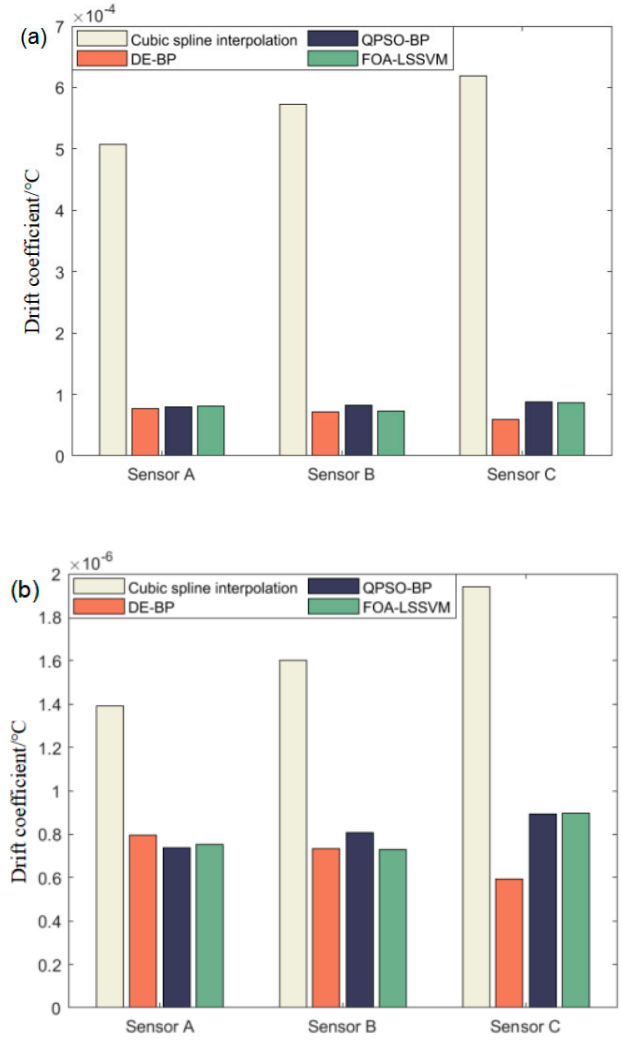


Figure 10. Comparison of different sensors: (**a**) zero-drift coefficient; (**b**) sensitivity temperature coefficient.

Table 7 intuitively shows the comparison of the zero-drift coefficient and sensitivity temperature coefficient between the proposed algorithms for parameter optimization and the temperature compensation algorithms presented in other studies. It can be seen that the three types of intelligent optimization algorithms that are combined with the machine learning framework obtained a higher output accuracy, and that the sensitivity temperature coefficients of three types of algorithms put forward in the paper are an order of magnitude lower compared with those of the other algorithms.

Table 7. Output accuracy of this paper's algorithms compared with other algorithms in the literature.

Algorithm	α_0 (/°C)	α_s (/°C)
Fuzzy neural network [37]	5.38×10^{-6}	7.24×10^{-4}
GSO–BP [38]	3.95×10^{-6}	1.82×10^{-4}
DE–SVM [39]	4.03×10^{-6}	1.03×10^{-4}
DE–BP	7.94×10^{-7}	7.76×10^{-5}
QPSO–BP	7.37×10^{-7}	7.92×10^{-5}
FOA–LSSVM	7.51×10^{-7}	8.07×10^{-5}

5.2. Temperature Compensation Results Based on Data Set Segmentation

In general, the typical operating environment temperature of the sensor is 20–100 °C; thus, the starting point of the test set data temperature was set to 20 °C, and the end point

was set to 100 °C. If the test set is defined as data within the range from 20−−70 °C, then the segmentation ratio is 33.3%, as calculated according to Equation (11). After the segmentation ratio was given, the DE–BP algorithm was used for temperature compensation. To eliminate the randomness error of the BP network, the algorithm was repeatedly executed 10 times to perform averaging. Table 8 lists the zero-drift coefficient and sensitivity temperature coefficient of the sensor with different splitting ratios. Evidently, when the split ratio was introduced, the sensor output accuracy was significantly improved, and the zero-drift coefficient was reduced by two orders of magnitude, while the sensitivity temperature coefficient was reduced by three orders of magnitude, compared with the results of the uncompensated Sensor A calibration data.

Table 8. Drift parameters of sensors at different segmentation ratios.

Segmentation Ratio	α_0 (/°C)	α_s (/°C)
0	3.93×10^{-4}	7.76×10^{-5}
5.60%	1.05×10^{-4}	2.74×10^{-5}
11.10%	8.41×10^{-5}	8.82×10^{-6}
16.70%	7.23×10^{-5}	7.13×10^{-6}
22.20%	6.44×10^{-5}	5.28×10^{-6}
27.80%	5.30×10^{-5}	4.52×10^{-6}
33.30%	4.29×10^{-5}	4.68×10^{-6}
38.90%	4.86×10^{-5}	4.97×10^{-6}
44.40%	5.91×10^{-5}	5.33×10^{-6}
50.00%	7.17×10^{-5}	7.49×10^{-6}

Figure 11 shows the trend of the sensor drift coefficient with the split ratio. Apparently, the zero-drift coefficient had a minimum value of 4.29×10^{-5}/°C at a split ratio of 33.3%. Correspondingly, the sensitivity temperature coefficient had a minimum value of 4.52×10^{-6}/°C at a split ratio of 27.8%. When the splitting ratio was [22.2%, 44.4%], the output sensitivity temperature coefficient changed slightly. Therefore, in this study, 33.3% was considered the optimal value of the splitting ratio of the sensor input data, i.e., the calibration data between 20 and 70 °C were used as the test set, whereas the remainder of the data were used as the training set, and the output accuracy of the sensor was optimal.

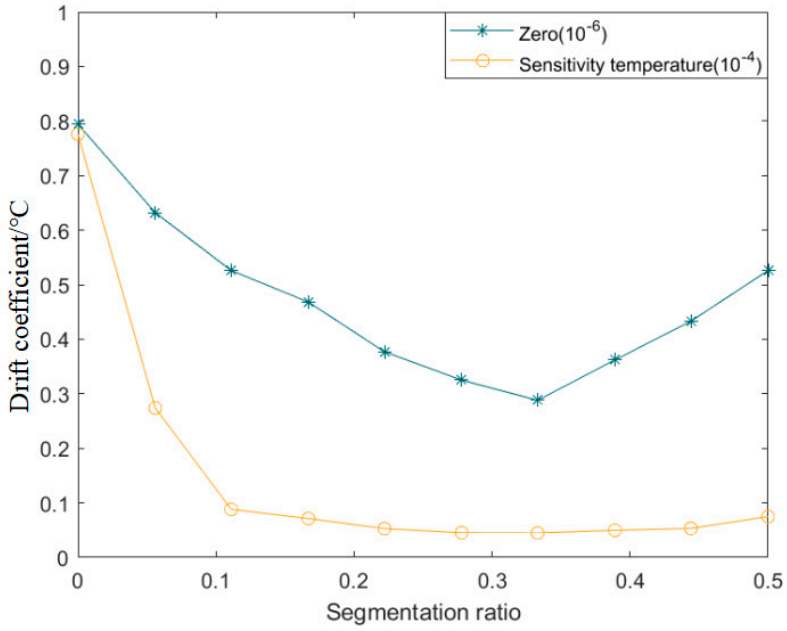


Figure 11. Trend of the sensor drift coefficient with different segmentation ratios.

6. Conclusions and Future Work

6.1. Conclusions

In this paper, we proposed three temperature compensation methods based on swarm optimization algorithms fused with machine learning for three different ranges of sensors and explored the partitioning ratio of the calibration dataset on Sensor A. The results show that the output accuracy of the three sensors was significantly improved after the compensation of different types of algorithms, which reflects the advantages of combining the simulation of biological population characteristics with machine learning. The optimal zero-drift coefficients of the three sensors were 7.37×10^{-7}, 7.29×10^{-7} and 5.93×10^{-7}, while the optimal sensitivity temperature coefficients of the three sensors were 7.76×10^{-5}, 7.13×10^{-5} and 5.95×10^{-5}. An optimal compensation effect was achieved on Sensor A when the splitting ratio was 33.3%, where the zero-drift coefficient was $2.88 \times 10^{-7}/^{\circ}\mathrm{C}$ and the sensitivity temperature coefficient was $4.52 \times 10^{-6}/^{\circ}\mathrm{C}$. A comparison with other algorithms in other studies revealed that the intelligent optimization algorithm framework designed in this study achieves a higher output accuracy with reductions in one and two orders of magnitude in the zero-drift and sensitivity temperature coefficients, respectively. The experimental results are highly instructive for the software compensation method for piezoresistive pressure sensors. In addition, the introduction of the optimal splitting ratio allows the sensor to determine the output compensation with high accuracy in its daily operating ambient-temperature interval, which is innovative.

6.2. Future Work

In this paper, the content of the experiment still needs to be optimized. For different sensors, even if every step in the process is the same, the output parameters of the products on the same pipeline will have a certain degree of minor error. Therefore, for sensors with the same pressure range, several different products of the same batch should be selected for calibration, and the output results should be processed with the mean or optimal value. Due to the limitation of cost, the data calibration of this part will be carried out later.

Author Contributions: Conceptualization, H.W. and J.L.; methodology, H.W.; software, H.W.; validation, H.W. and J.L.; formal analysis, H.W.; investigation, H.W.; resources, J.L.; data curation, H.W.; writing—original draft preparation, H.W.; writing—review and editing, J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Research and Development Program of China (2018YFB2002700); Strategic Pioneer Science and Technology Special Project (Class A) of the Chinese Academy of Science (XDA22020100); CAS (No. 201510280052XMXX201200019933); and CAS STS Program (2019T3015).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Pak, M.; Fernandez, F.V.; Dundar, G. A Novel Design Methodology for the Mixed-domain Optimization of a MEMS Accelerometer. *Integr. VSLI J.* **2018**, *6*, 314–321. [CrossRef]
2. Mukhopadhyay, S.C. Wearable Sensors for Human Activity Monitoring: A Review. *IEEE Sens. J.* **2014**, *15*, 1321–1330. [CrossRef]
3. Barzegar, M.; Blanks, S.; Timms, W. MEMS Technology and Applications in Geotechnical Monitoring: A Review. *Meas. Sci. Technol.* **2022**, *33*, 2–8. [CrossRef]
4. Hommel, M.; Knab, H.; Yousef, S. Reliability of Automotive and Consumer MEMS Sensors: An Overview. *Microelectron. Reliab.* **2021**, *39*, 126–128. [CrossRef]
5. Wang, X.; Guo, S.; Qu, H.; Song, M. Design of a Purely Mechanical Sensor-Controller Integrated System for Walking Assistance on an Ankle-Foot Exoskeleton. *Sensors* **2019**, *19*, 3196. [CrossRef] [PubMed]

6. Xu, N.; Tang, C.; Li, H.; Dai, F.; Ma, K.; Shao, J.; Wu, J. Excavation-Induced Microseismicity: Microseismic Monitoring and Numerical Simulation. *J. Zhejiang Univ. Sci. A* **2012**, *13*, 445–460. [CrossRef]
7. Li, X.; Liu, Q.; Pang, S.; Xu, K.; Tang, H.; Sun, C. High-Temperature Piezoresistive Pressure Sensor Based on Implantation of Oxygen into Silicon Wafer. *Sens. Actuators A Phys.* **2012**, *179*, 277–282. [CrossRef]
8. Huang, X.; Zhang, D. A High Sensitivity and High Linearity Pressure Sensor Based on a Peninsula-Structured Diaphragm for Low-Pressure Ranges. *Sens. Actuators A Phys.* **2014**, *216*, 176–189. [CrossRef]
9. Huang, M.; Zou, X.; Guo, Z. Research Status and Development of High Temperature and High Pressure Sensor. *Obs. Control Technol.* **2020**, *39*, 1–5. [CrossRef]
10. Joseph, R.; Ali, J.; Beth, L.; Nix, A.R. Review: Semiconductor Piezoresistance for Microsystems. *Proc. IEEE* **2009**, *97*, 513–552. [CrossRef]
11. Saoni, B.; Daniel, F.; Jordi, M. Characterization of CMOS-MEMS Resonant Pressure Sensors. *IEEE Sens. J.* **2017**, *17*, 6653–6661. [CrossRef]
12. Anh, V.; Zhang, X.; Zhu, B.; Li, X. The Development of a New Piezoresistive Pressure Sensor for Low Pressures. *IEEE Trans. Ind. Electron.* **2018**, *65*, 6487–6496. [CrossRef]
13. Jeong, T. Modeling of Sensor Behaviour for Improving Sensitivity and Performance. *Measurement* **2015**, *41*, 230–236. [CrossRef]
14. Bruyker, D.; Puers, R. Thermostatic Control for Temperature Compensation of a Silicon Pressure Sensor. *Sen. Actuators A Phys.* **2000**, *82*, 120–127. [CrossRef]
15. Lee, B.N.; Kim, K.N.; Park, H.D. Calibration and Temperature Compensation of Silicon Pressure Sensors Using Ion-implanted Trimming Resistors. *Sen. Actuators A Phys.* **2012**, *72*, 148–152. [CrossRef]
16. Guo, Z.; Lu, C.; Wang, Y.; Qi, S. Design and Experimental Research of a Temperature Compensation System for Silicon-on-Sapphire Pressure Sensors. *IEEE Sens. J.* **2017**, *36*, 709–715. [CrossRef]
17. Xia, D.; Chen, S.; Wang, S. Microgyroscope Temperature Effects and Compensation-control Methods. *Sensors* **2009**, *9*, 8349–8376. [CrossRef]
18. Ding, J.C.; Zhang, J.; Huang, W.Q.; Chen, S. Laser Gyro Temperature Compensation Using Modified RBFNN. *Sensors* **2014**, *14*, 18711–18727. [CrossRef]
19. Aryafar, M.; Hamedi, M.; Ganjeh, M.M. A novel temperature compensated piezoresistive pressure sensor. *Meas. J. Int. Meas. Confed.* **2015**, *63*, 25–29. [CrossRef]
20. Kirkpatrick, S.; Gelatt, C.D., Jr.; Vecchi, M.P. Optimization by simulated annealing. *Science* **1983**, *220*, 671–680. [CrossRef]
21. Suykens, J.A.K.; Vandewalle, J. Least squares support vector machine classifiers. *Neural Process. Lett.* **1999**, *9*, 293–300. [CrossRef]
22. Yao, M.; Xing, L.; Zhao, M. A Temperature Compensation Method of Pressure Sensor Based on Wavelet Neural Network. *Sens. Technol.* **2005**, *7*, 13–18. [CrossRef]
23. Wang, H.; Zeng, Q.H.; Zhang, Z.Y.; Wang, H.F. Research on Temperature Compensation of Multi-Channel Pressure Scanner Based on an Improved Cuckoo Search Optimizing a BP Nueral Network. *Micromachines* **2022**, *13*, 1351. [CrossRef] [PubMed]
24. Javidy, B.; Hatamlou, A.; Mirjalili, S.; Karmakar, R.; Kim, I.M. Ionsmotion Algorithm for Solving Optimization Problems. *Appl. Soft Comput.* **2015**, *32*, 72–79. [CrossRef]
25. Gong, Z.; Chen, K.; Zhou, X.; Wu, R. High-Sensitivity Fabry-Perot Interferometric Acoustic Sensor for Low-Frequency Acoustic Pressure Detections. *J. Lightwave Technol.* **2017**, *35*, 5276–5279. [CrossRef]
26. Pramanik, C.; Islam, T.; Saha, H. Temperature Compensation of Piezoresistive Micro-Machined Porous Silicon Pressure Sensor by ANN. *Microelectron. Reliab.* **2006**, *41*, 343–351. [CrossRef]
27. Stefano, D.; Patrice, R.; Giacomo, L.; Morimoto, A. MEMS Gyroscopes Based on Piezoresistive NEMS Detection of Drive and Sense Motion. *J. Microelectromech. Syst.* **2017**, *26*, 1389–1399. [CrossRef]
28. Kashyap, K.; Pathak, S.; Yadav, N.S. Logistic Map and Exponential Scaling Factor-Based Differential Evolution. *Int. J. Swarm Intell. Res.* **2022**, *13*, 1486–1492. [CrossRef]
29. Khaparde, A.R.; Alassery, F.; Kumar, A.; Alotaibi, Y.; Khalaf, O.I.; Pillai, S.; Alghamdi, S. Differential Evolution Algorithm with Hierarchical Fair Competition Model. *Intell. Autom. Soft Comput.* **2022**, *33*, 1045–1062. [CrossRef]
30. Fernandes, P.; Oliveira, R.; Neto, J. Trajectory Planning of Autonomous Mobile Robots Applying Particle Swarm Optimization Algorithm with Peaks of Diversity. *Appl. Soft Comput.* **2022**, *116*, 26–33. [CrossRef]
31. Fahad, S.; Yang, S.; Khan, S.U.; Khan, S.A.; Khan, R.A. A Hybrid Quantum Particle Swarm Optimization for Multimodal Electromagnetic Design Problems. *IEEE Access* **2022**, *10*, 72339–72347. [CrossRef]
32. Yang, J.; Cai, Y.; Tang, D.; Chen, W.; Hu, L. Memetic Quantum Optimization Algorithm with Levy Flight for High Dimension Function Optimization. *Appl. Intell.* **2022**, *17*, 622–635. [CrossRef]
33. Jiang, F.; Zhang, W.; Peng, Z. Multivariate Adaptive Step Fruit Fly Optimization Algorithm Optimized Generalized Regression Neural Network for Short-Term Power Load Forecasting. *Front. Environ. Sci.* **2022**, *53*, 7132–7140. [CrossRef]
34. Crawford, B.; Soto, R.; Mella, H.D.; Elortegui, C.; Palma, W.; Torres-Rojas, C.; Vasconcellos-Gaete, C.; Becerra, M.; Pena, J.; Misra, S. Binary Fruit Fly Swarm Algorithms for the Set Covering Problem. *Comput. Mater. Contin.* **2022**, *71*, 4295–4318. [CrossRef]
35. Li, Y.; Mu, W.S.; Zhu, X.Q.; Fu, Z.T. K-means Clustering Algorithm Based on Improved Quantum Particle Swarm Optimization and its Application. *Control Decis.* **2022**, *37*, 839–850. [CrossRef]
36. Shu, K.S.F.; Yi, Y.C. A Decreasing Inertia Weight Particle Swarm Optimizer. *Eng. Optim.* **2007**, *39*, 203–228. [CrossRef]

37. Song, Z.; Sun, Y.; Li, H. Optimal Design of Neural Network Based on Differential Evolution Algorithm. *Chongqing Norm. Univ. J.* **2022**, *39*, 79–89. [CrossRef]
38. Wang, H.; Fu, P.; Song, Y. Sensor Temperature Compensation Strategy Based On Firefly Optimization BP Neural Network Method. *Mech. Strength* **2020**, *42*, 109–114. [CrossRef]
39. Wen, C.; Wang, M.; Zhong, C. Research on Temperature Compensation of Silicon Piezoresistive Pressure Sensor Based on DE-SVM. *Chin. J. Sens. Actuators* **2019**, *32*, 1493–1498. [CrossRef]