IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 4 January 2023, accepted 24 January 2023, date of publication 30 January 2023, date of current version 2 February 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3240766

RESEARCH ARTICLE

Real-Time Temperature Drift Compensation Method of a MEMS Accelerometer Based on Deep GRU and Optimized Monarch Butterfly Algorithm

GUO GANG-QIANG, CHAI BO, CHENG RUI-CHU, AND WANG YUN-SHUANG
Xi'an Microelectronics Technology Institute, Xi'an 710054, China
Corresponding author: Guo Gang-Qiang (guogangqiang9352@163.com)

ABSTRACT In recent years, inertial sensors based on Micro-Electro-Mechanical Systems (MEMS) have become increasingly popular. They have been widely used in various fields due to their low cost, small size, and low power consumption. It seems that MEMS inertial sensors may eventually fully occupy the middle to lower end inertial navigation application market that traditional inertial sensors previously occupied. To realize the full potential of MEMS inertial sensors, one of the critical issues is their temperature drift. This paper first proposed a recursive model for real-time compensation. Then three algorithms were proposed, including a two-layer deep gated recurrent unit recurrent neural network (GRU-RNN, short GRU), deep GRU based on monarch butterfly algorithm (MBA), and deep GRU based on optimized monarch butterfly algorithm (OMBA). Each of these three algorithms is combined with the real-time compensation model to compensate for the temperature drift of a MEMS accelerometer. The experimental results proved the correctness of these three methods, and the MEMS accelerometer's temperature drift is compensated effectively. The results indicate that the deep GRU + OMBA shows the best performance for the temperature drift compensation combined with the real-time compensation model. After deep GRU + OMBA method compensation, the angle random walk, the bias instability, the rate random walk, and rate ramp of the MEMS accelerometer were improved from $4.97e^{-4}\ mg\ \cdot s^{\frac{1}{2}}$, $4.90e^{-4}\ mg$, $5.57e^{-5}\ mg/s^{\frac{1}{2}}$, $1.82e^{-6}\ mg\ /s$ to $3.90e^{-5}\ mg\ \cdot s^{\frac{1}{2}}$, $1.07e^{-5}\ mg$, $1.12e^{-6}\ mg/s^{\frac{1}{2}}$, $3.59e^{-8}\ mg\ /s$, respectively. Their percentage of improvement reaches 96.50% on average.

INDEX TERMS Deep GRU, MEMS accelerometer, real-time compensation, OMBA, temperature drift.

I. INTRODUCTION

Advances in the MEMS have made it possible to produce MEMS inertial sensors that measure angular velocity and acceleration. They have the advantages of small size, light weight, low cost, and low power consumption and have been widely used in aerospace, healthcare, gait analysis, sport science, activity recognition, and portable devices [1], [2], [3], [4], [5], [6], [7], [8], [9]. Since the mechanical structure of MEMS inertial sensors inevitably changes with heat, their performance changes drastically with temperature. Temperature effects greatly limit their applications. This paper mainly studies the temperature drift of MEMS accelerometers. Extensive literature studies improve the temperature characteristics of MEMS accelerometers. In general, there are two main ways to improve the temperature characteristics of MEMS accelerometers: hardware compensation methods and software compensation methods.

The hardware compensation method mainly includes circuit control or optimizing the structure of the MEMS accelerometer, which always needs extra cost, additional hardware, and increased power consumption and volume. Wang proposed two approaches for temperature compensation of high performance resonant MEMS accelerometers.

The associate editor coordinating the review of this manuscript and approving it for publication was Mauro Gaggero.

In the structural design, an electrostatic softening spring is utilized to isolate the external stress transmission, thereby improving the thermal robustness of the accelerometer. In the control circuit, a continuous ring-down technique is employed to attenuate measuring hysteresis, thereby improving the temperature compensation accuracy [10]. Liu proposed a temperature drift compensation method for high-precision capacitive accelerometers based on parasitic resistance [11]. Li proposed a new differential silicon substrate to significantly reduce the temperature drift of high-performance resonant accelerometers; in addition, its cross-interference from undesirable directions is also significantly reduced [12]. Tsai developed a CMOS-MEMS accelerometer whose mechanical structure is a stack of pure oxide layers. By limiting the distribution area of the metal layer and metal-oxide, the thermal deformation caused by the expansion coefficient mismatch is suppressed, thereby improving the sensitivity and reducing the temperature drift of the accelerometer [13]. Zotov proposed a silicon MEMS accelerometer with differential frequency-modulated. There are two identical tuning forks inside this accelerometer chip, their sensitive axes are in opposite directions, and the common mode error is eliminated by these two tuning forks, thus achieving continuous thermal compensation [14]. Jing integrates a temperature resonator in the chip to measure the temperature in the cavity of the silicon resonant accelerometer package so as to obtain a more accurate temperature in the cavity and improve the temperature compensation accuracy of the silicon resonant accelerometer [15].

The software compensation method is essentially a mathematical method that studies the relationship between the temperature and the output of the MEMS accelerometer. Then a temperature drift model is established, and compensation is completed through the model. Different from the hardware compensation methods, the software compensation methods always have the characteristics of simple structure, low cost, time saving, convenient implementation, and ease of improving accuracy. He analyzed the temperature dependence between thermal deformation and stiffness to develop temperature drift analysis models for bias and scaling factors. The model effectively improved the accuracy of a MEMS capacitive accelerometer [16]. Ruzza used low-order polynomials to compensate for the thermal effects of low-cost MEMS accelerometers [17]. Du proposed a real-time temperature compensation algorithm based on the linear relationship between the temperature of the MEMS accelerometer and its dynamic resonant frequency [18]. Han used a finite-element-method to complete the quantitative analysis of the temperature effect of the MEMS accelerometer. The temperature compensation of the MEMS accelerometer is achieved by simulating the deformation of the sensor chip [19]. Khankalantary used cubic spline interpolation to model the temperature dependence of the error coefficients to minimize temperature effects [20]. Yang proposed a simple mathematical model to obtain a more accurate and robust output of low-cost quartz accelerometers at high temperatures. The model takes into account both temperature and rolling sensitivity and has been successfully applied to temperature compensation of two low-cost quartz accelerometers [21]. Wang used an optimized back propagation neural network (BP NN) to eliminate zero-bias error and scale factor error associated with the temperature of MEMS resonant accelerometers [22].

In this paper, three methods are proposed to model and compensate for the temperature drift of a MEMS accelerometer. Its temperature drift can be viewed as a random time series, and an RNN is very good at processing sequential signals because its context nodes can remember previous information. GRU is a variant of ordinary RNN with fast operation speed and good accuracy. A two-layer deep GRU is employed for simulating the temperature drift of MEMS accelerometers more accurately. The MBA and OMBA are used to tune the parameters of the deep GRU to improve the generalization ability and robustness of the temperature drift model.

The remainder of this paper is organized as follows: the in-house-designed guidance navigation and control (GNC) Module, which integrated an Inertial Measurement Unit (IMU) and its temperature experiment are described in Section II; the temperature drift model and the compensation algorithm are shown in Section III; Section IV shows the data processing results and the analysis results; the conclusion is given in Section V.

II. RELATED WORK

There have been some investigations in the area of temperature drift compensation for MEMS accelerometers. For instance, Pan et al. [23] study the bias drift and scale factor drift compensation of a quartz flexible accelerometer after a quick turn-on and before the thermal balance is reached inside the system based on a wavelet neural network. They selected accelerometers at six different locations and 11 temperature sensing points in steps ten ℃ between −40 ℃ and 60 ℃ for their temperature experiments. Then a three-layer wavelet neural network and variable rate learning algorithm are employed to compensate for the accelerometer's bias and scale factor drift. Results show that the compensation scheme and algorithm increase efficiency and improve attitude precision. The selection of the temperature sensing points is one of the keys to achieving adequate compensation.

Li et al. [24] studied an improved BP NN based on genetic algorithm (GA) for temperature drift compensation of Quartzes Flexible accelerometer in different temperatures. They collected the accelerometer temperature and output voltage every 5 ℃ between −20 ℃ and 55 ℃ and used the accelerometer temperature as the input to BP NN and the accelerometer output as the output of BP NN. Then, each set of neural network weights and thresholds form the GA individual. After the GA operation(selection, crossover, and mutation), the individual with the best fitness is selected as the weight of the BP NN. The results show that this method

achieved a good effect and had fewer training steps and better fitting accuracy precision than the standard BP NN.

Yu et al. [25] proposed a temperature drift compensation method based on the artificial fish swarm (AFS) algorithm for a quartz flexible accelerometer. The AFS algorithm is inspired by the social behavior of fish in searching, swarming, and following, which can easily find the global optimum. In the experiment, the temperature controlled box's temperature is lowered to $-20\,^{\circ}\mathrm{C}$, then raised to $50\,^{\circ}\mathrm{C}$, then lowered to $-20\,^{\circ}\mathrm{C}$, and finally raised to $20\,^{\circ}\mathrm{C}$, and maintained at each temperature point for more than 4 hours. During the experiment, they collected the accelerometer temperature and output voltage and used them as the input and output of the stepwise regression method and the AFS algorithm. The results show that the drift instability of the accelerometer output is reduced from 160.2ug to 18.0ug after compensation by the AFS algorithm, and the compensation effect is better than the conventional stepwise regression method.

Zhu et al. [26] proposed a fusion algorithm to compensate for the temperature drift of a high-G mems accelerometer. This paper first established a temperature drift model based on a neural network and considered temperature, temperature variation rate, and temperature product term as the model input. And then, the experiment's output was dealt with an RBF neural network optimized by Kalman filter and genetic algorithm (RBF NN + GA + KF). After that, the output based on RBF NN + GA + KF is corrected by the output data of the temperature model. And at this time, the fusion algorithm is determined. The results show that after using the fusion algorithm, the temperature drift and noise characteristics of HGMA are well optimized.

Han et al. [27] proposed a BP NN model based on an adaptive genetic algorithm (AGA) to improve the accuracy of a JSD-1 capacitive accelerometers in rapidly changing temperature environments. In this paper, the ambient temperature and the accelerometer's output are used as inputs to the BP NN, where the accelerometer's output is influenced by its input acceleration and the ambient temperature. To compensate for the accelerometer temperature drift more accurately, GA is introduced to facilitate a more extensive search for optimal solutions in the solution space in this paper. The crossover and mutation probabilities of the standard GA are constant, which leads to a poor global search capability and a tendency to fall into local minima. To address this problem, this paper proposes a method for nonlinearly adjusting the crossover and mutation probabilities named AGA. During AGA execution, the average fitness of all group parameters is calculated, and the maximum fitness is found. Then, determine the order of GA execution based on the ratio of the average fitness to the maximum fitness. If the ratio is less than 0.7, the GA flow is selection, crossover, and mutation. Otherwise, the GA flow is selection, mutation, and crossover. The validation results show that AGA-BP has the best compensation for the JSD-1 capacitive accelerometers compared to GA-BP and multiple linear regression (MLR).

Wu et al. [28] studied the temperature characteristics and compensated for the temperature drift of a quartz flexible accelerometer in the cold start condition. This paper proposed first-order, second-order, and higher-order temperature drift compensation models based on temperature, temperature gradient, and time-related drift. And then, the particle swarm optimization (PSO) algorithm is employed to find the best model parameters. To demonstrate the feasibility and reliability of the proposed model, they experimented and compensated with the first-order compensation model as an example. The results show that the method proposed in this paper can flexibly compensate for the temperature drift and greatly reduce the output signal's standard deviation.

Qi et al. [29] simulated a MEMS accelerometer's structural deformation in diverse conditions to trace its temperature drift error (TDE). They deduced that the ambient temperature and its square, as well as the ambient temperature variation and its square, are the critical factor to TDE. Then, The TDE model was constructed based on the four factors mentioned above. Then, this paper introduced a BP NN based on PSO and GA (PSO + GA + BP NN) in its model parameter identification. GA is introduced to remove local optimums of BP NN. Further, PSO is utilized in GA to solve its probabilistic disorder to improve its TDE accuracy and real-time performance of the model to achieve temperature compensation. Next, the authors analyzed the heat conduction process between accelerometers and a thermal chamber and designed a temperature experiment. At last, the conventional and newly proposed models are compared based on the temperature experiment data. The result showed that the model proposed in this paper could estimate TDE more precisely and enhance the environmental adaptability of MEMS accelerometers.

The above analysis shows that the previous studies about temperature drift compensation of MEMS accelerometers have the following characteristics.

1) All of the literature uses single hidden layer neural networks for compensation.
2) All the literature does not consider the effect of the MEMS accelerometer historical output information on the current moment output.
3) The input parameters of all models in the literature are temperature or temperature-related terms.
4) The above literature's most comprehensive test temperature range is $-30\,^{\circ}\mathrm{C}$ to $60\,^{\circ}\mathrm{C}$.
5) In most of the literature, several temperature points in the test temperature range are selected for testing.
6) Most of the literature uses optimization algorithms to optimize the neural network parameters to find the optimal solution for the temperature compensation model.

With the continuous development of big data and computing power, deep learning methods have blossomed rapidly and have been widely utilized in various fields. Following this line of thinking, a deep learning approach for temperature drift compensation of MEMS accelerometers using deep GRU is proposed in this paper. This paper considers the effect of the accelerometer history output on the temperature

drift compensation model and uses GRU instead of a typical neural network to model the temperature drift of a MEMS accelerometer. This RNN variant has higher computational efficiency and accuracy. Besides, the last time's and current time's temperature drift outputs of the MEMS accelerometer are used as input and output of the model, respectively, to achieve recursion. Then, this paper has a broader temperature test range of $-40\,^{\circ}\mathrm{C}$ to $80\,^{\circ}\mathrm{C}$, and the chamber ramps up at a fixed rate, during which the MEMS accelerometer output is collected continuously. Moreover, this paper introduced the MBA to find the optimal weights of GRU, a swarm intelligence optimization algorithm proposed in recent years. Optimizing the MBA is to improve its speed and accuracy in finding the best weights.

III. THE IN-HOUSE-DESIGNED GNC MODULE AND TEMPERATURE EXPERIMENT

A. THE IN-HOUSE-DESIGNED GNC MODULE

The in-house-designed GNC Module consists of multiple parts, including dual-core processor, AD, IMU, external interface, etc., and is packaged in three-dimensional form. The internal space of the GNC module is filled with special materials to make its components a whole, which effectively improves its impact resistance. Its surface is plated with metal to interconnect different parts in the same vertical direction. The GNC module can autonomously provide a full range of acceleration and angular velocity information through the IMU integrated in its internal space. By acquiring information from various sensors and peripherals from the outside, such as GPS information or geomagnetic field information, functions such as integrated navigation, flight control, electronic control combination, and steering gear control can be realized. Its size is 30 mm × 30 mm × 21.6 mm. The picture of the GNC Module is shown Fig 1.

FIGURE 1. In-house-designed GNC Module.

B. TEMPERATURE EXPERIMENT

We conducted a temperature test on the GNC Module to gain insight into its temperature characteristics. During the temperature test, the GNC Module was placed in a temperature controlled oven with a static base to ensure that its output was not affected by external vibrations, as shown in Fig 2. The temperature range was set from $-40\,^{\circ}\mathrm{C}$ to $80\,^{\circ}\mathrm{C}$ and the ramp rate was $1\,^{\circ}\mathrm{C}$/min. The temperature was initially held at $-40\,^{\circ}\mathrm{C}$ for 1 hour. Then, the oven was warmed up to $80\,^{\circ}\mathrm{C}$ and stay at $80\,^{\circ}\mathrm{C}$ for 1 hour. The entire temperature test lasted for 4 hours. At the beginning of the heating process, the computer outside the temperature controlled oven synchronously collected the output of the GNC Module with the frequency of 200Hz.

FIGURE 2. Temperature test equipment.

IV. REAL-TIME COMPENSATIOM MODEL

Many authors have considered temperature and other temperature-dependent quantities to model the temperature drift of MEMS accelerometers. The accelerometer temperature drifts compensation at the current moment is calculated after obtaining the current accelerometer temperature. But the current accelerometer temperature drift and the current accelerometer temperature usually output synchronously. Therefore, real-time compensation is not possible in this way. To solve this problem, we consider using a

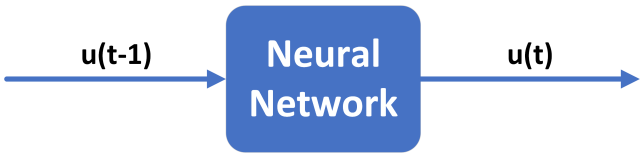

FIGURE 3. The framework of the real-time compensation model.

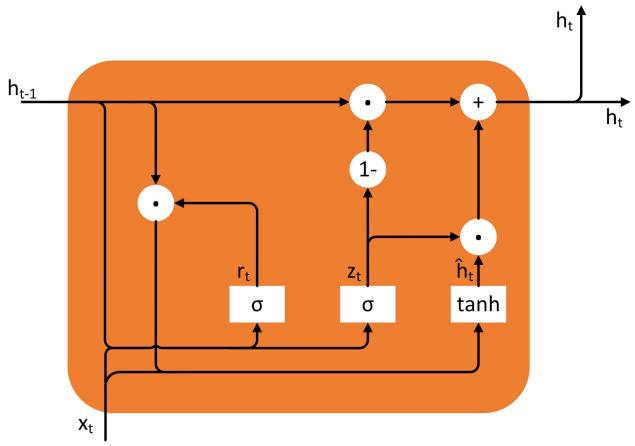

FIGURE 4. Basic structure of GRU workflow.

one-step ahead predictor to perform real-time compensation of the temperature drift of the MEMS accelerometer. That is, the temperature drift output at time $t-1$ ($u(t-1)$) is used as the model input, and the temperature drift output at time t ($u(t)$) is used as the model output simultaneously. Hence, the MEMS accelerometer readings of the temperature experiment from 1 to $M-1$ ($[u(1), u(2), \cdots, u(M-1)]$) and from 2 to M ($[u(2), u(3), \cdots, u(M)]$) are represented as the input and output data for the proposed model, respectively. That is, $X = [u(1), u(2), \cdots, u(M-1)]$ and $Y = [u(2), u(3), \cdots, u(M)]$. M is the total number of the MEMS accelerometer's temperature drift output data. The full real-time compensation model parameters are obtained from the training data set. The current moment output minus the predicted temperature drift of the last moment to achieve real-time compensation. The target model could be defined as:

$$Y(t) = u(t) = F[X(t)] = F[u(t-1)] = F[u(t-1)] \quad (1)$$

where $X(t)$ and $Y(t)$ are the input and output of the model at time t, respectively. $F[\cdot]$ is the target function to be trained. Fig 3 shows the framework of the proposed model.

V. ALGORITHM

A. GRU AND DEEP GRU

1) GRU

GRU is a variant of RNN proposed by Cho in 2014, which aims to make each recurrent unit adaptively capture dependencies in the data [30]. Fig 4 shows the basic architecture of GRU workflow, which is built with the "gate" structure. There are two gate units in the GRU to regulate the flow of information inside the unit, called "update gate" and "reset gate" [31].

The role of the update gate z_t is to control how much information can be brought into the current state from the previous state of the unit. The update gate is computed by expression (2).

$$z_t = \sigma(W_{zx}x_t + W_{zh}h_{t-1} + b_z) \quad (2)$$

where z_t is the output of update gate at time t; x_t is the input of GRU at time t; h_{t-1} is the output of GRU at time $t-1$; W_{zx} and W_{zh} are coefficient of x_t and h_{t-1},respectively; b_z is the threshold of update gate; $\sigma(\cdot)$ is the sigmoid funtion.

The reset gate enables the GRU to forget the previous computational state, making it unaffected by the previous state. The reset gate is computed by expression (3), which is computed similarly to the update gate.

$$r_t = \sigma(W_{rx}x_t + W_{rh}h_{t-1} + b_r) \quad (3)$$

where r_t is the output of reset gate at time t; x_t is the input of GRU at time t; h_{t-1} is the output of GRU at time $t-1$; W_{rx} and W_{rh} are coefficient of x_t and h_{t-1},respectively; b_r is the threshold of reset gate; $\sigma(\cdot)$ is the sigmoid funtion.

The candidate hidden state $\hat{h}_t$ is calculated as expression (4).

$$\hat{h}_t = tanh(W_{\hat{h}x}x_t + W_{\hat{h}h}(r_t \odot h_{t-1}) + b_{\hat{h}}) \quad (4)$$

where $\hat{h}_t$ is the candidate hidden state output of GRU at time t; x_t is the input of GRU at time t; h_{t-1} is the output of GRU at time $t-1$; r_t is the output of reset gate at time t; $\odot$ represents the point-wise multiplication operation; $W_{\hat{h}x}$ and $W_{\hat{h}h}$ are coefficient of x_t and $r_t \odot h_{t-1}$,respectively; $b_{\hat{h}}$ is the threshold; $tanh(\cdot)$ is the hyperbolic tangent function.

The output of GRU at time t is calculated as expression (5), which is a linear interpolation between the previous activation h_{t-1} and the candidate hidden state output $\hat{h}_t$.

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \hat{h}_t \quad (5)$$

The above analysis shows that x_t and h_t are the input and output, respectively. z_t, r_t, and $\hat{h}_t$ are intermediate variables in the calculation process. W_{zx}, W_{zh}, b_z, W_{rx}, W_{rh}, b_r, $W_{\hat{h}x}$, $W_{\hat{h}h}$, and $b_{\hat{h}}$ are the weights determined on the training data set. $\sigma(\cdot)$ and $tanh(\cdot)$ are activation functions.

2) DEEP GRU

Fig 4 shows a single-layer GRU and Fig 5 is a basic workflow of two-layer GRU. The x_t in the first layer GRU is the input of the real-time compensation model mentioned before. The output of the first layer GRU is passed to the second layer GRU and used as its input. The input vector goes through the two-layer GRU one by one to get the outputs. In general, a sequence GRU decides the output together. Its structure is shown in Fig 6.

Training deep GRU requires large datasets. The parameters of the entire deep GRU will be determined or learned during training.

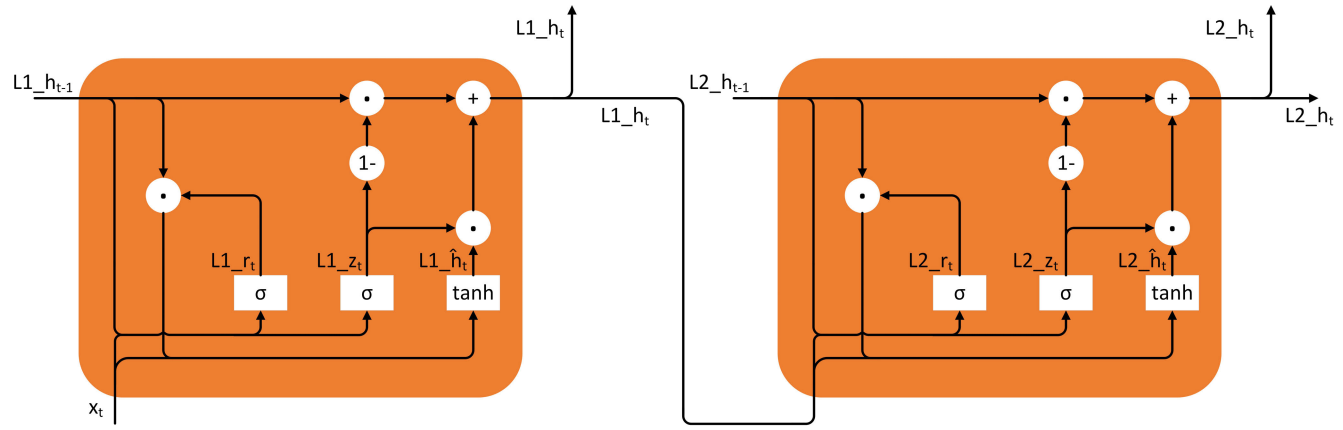

FIGURE 5. Workflow of two-layer GRU.

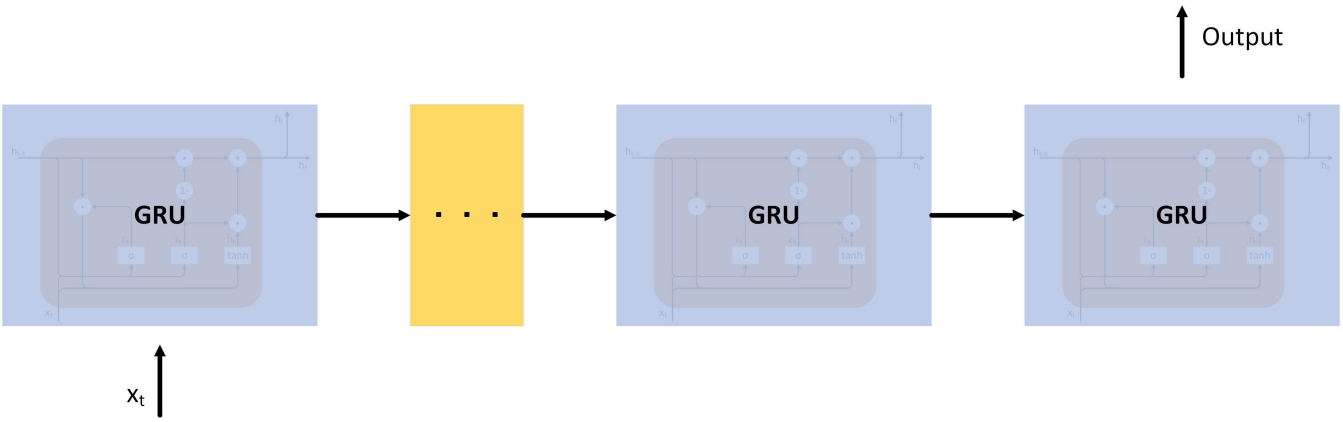

FIGURE 6. Basic structure of Deep GRU workflow.

B. MBA AND OMBA

1) MBA

The MBA is a swarm intelligence optimization algorithm proposed by Wang in 2015 [32]. It is a nature-inspired algorithm obtained by simplifying and idealizing the migration of monarch butterflies in eastern North America. In order to extract the optimization algorithm from the migration behavior of monarch butterflies, the migration behavior of monarch butterflies is simplified to the following rules.

1) The monarch butterfly population is divided into two parts, the subpopulation on Land1 and the subpopulation on Land2, which together constitute the entire monarch butterfly population. Each monarch butterfly is randomly distributed on Land1 or Land2.
2) Each child monarch butterfly individual is generated by the migration operator of the parent individual distributed on land1 or by the adjusting operator of the parent individual distributed on land2.
3) Total monarch butterfly population remains unchanged. If the fitness of the child individual is better than that of the parent, then replace the parent individual with the child individual. If the fitness of the child individual is not better than that of the parent, the child individual will be discarded, and the parent individual will remain unchanged and enter the next generation.
4) The monarch butterfly with the best fitness goes directly to the next generation, so as to ensure that the overall quality of the population will not decline.

The migration operator and adjusting operator for the monarch butterfly are shown below [32], [33].

Assume that there are a total of N monarch butterfly individuals in the population, each with n dimensions/elements. This monarch butterfly population can be represented as an expression (6). Each row of the matrix represents an individual monarch butterfly.

$$\begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N,1} & x_{N,2} & \cdots & x_{N,n} \end{bmatrix} \tag{6}$$

The monarch butterfly's population can be divided into two subpopulations on Land1 and Land2. The individuals in Land1 and Land2 can be expressed as an expression (7) and

an expression (8), respectively. And $N = N1 + N2$.

$$\begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1,1} & x_{N1,2} & \cdots & x_{N1,n} \end{bmatrix} \tag{7}$$

$$\begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N2,1} & x_{N2,2} & \cdots & x_{N2,n} \end{bmatrix} \tag{8}$$

a: MIGRATION OPERATOR

For individual i in the Land1 subpopulation, the migration operator is influenced by the location of other monarch butterflies on Land1 and Land2. The migration operator is used for information exchange within the monarch butterfly population to obtain better individuals. Information exchange can take place between individuals on land1 and Land2, or between individuals within Land1. The migration operator can be expressed as follows.

$$x_{i,k}^{t+1} = \begin{cases} x_{r1,k}^{t}, & if \quad r \le p \\ x_{r2,k}^{t}, & else \end{cases} \tag{9}$$

where $x_{i,k}^{t+1}$ is the kth dimension of ith individual in generation $t+1$ of the Land1 subpopulation; $r1$, $r2$ are individuals randomly selected on Land1 subpopulation and Land2 subpopulation, respectively; the parameter $r = rand * peri$, where $rand$ is a uniform random number on $[0, 1]$, $peri$ is usually a constant with a value of 1.2, representing migration period; p is an adjustable migration ratio, usually a constant with a value of $\frac{5}{12}$.

b: ADJUSTING OPERATOR

For individual i in the Land2 subpopulation, the adjusting operator needs to consider the global optimal individual of the previous generation, a random individual in subpopulation on Land2, and Lévy flight. The adjusting operator mainly considers the following three factors.

1) The influence of the global optimal individual on the overall population.
2) The effect of moving to a random individual on Land2.
3) The purpose of introducing the Lévy flight algorithm is to increase the diversity of the population and expand the search range for optimal individual.

The adjusting operator can be expressed as follows.

$$x_{i,k}^{t+1} = \begin{cases} x_{best,k}^{t}, & if \quad rand1 \le p \\ x_{r3,k}^{t}, & if \quad rand1 > p \cap rand2 \le BAR \\ x_{r3,k}^{t} + \alpha \times (dx - 0.5), & else \end{cases} \tag{10}$$

where $x_{i,k}^{t+1}$ is the kth dimension of ith individual in generation $t+1$ of the Land2 subpopulation; $x_{best,k}^{t}$ is the kth dimension of the global optimal individual in generation t; $r3$ is a individual randomly selected on Land2 subpopulation; $rand1$ and $rand2$ are uniform random numbers on $[0, 1]$; BAR is the adjusting rate of monarch butterflies, usually a constant with a value of $\frac{5}{12}$. α is the weighting factor; dx is a parameter calculated by Lévy flight.

The expression for α and dx can be given as (11) and (12) respectively.

$$\alpha = \frac{S_{max}}{t^2} \tag{11}$$

$$dx = Levy(x_{i,k}^{t}) \tag{12}$$

where S_{max} is max walk step of monarch butterfly, generally 1. $Levy(\cdot)$ stands for Lévy flight algorithm.

The constants in the algorithm are calculated based on the proportion of the time the monarch butterflies spend in various places and activities throughout the year, which is very realistic.

2) OMBA

According to the introduction of MBA, we need to compare the fitness of the child individuals with their parent individuals one by one and leave only one of the individuals in each group. Possibly, a particular group of parent individuals and child individuals has better fitness than another group of parent individuals and child individuals. However, each of the two groups discarded a worse individual within the group in this approach. To solve this problem, we proposed an optimization operator to optimize the MBA to improve its accuracy in finding the optimal individual. The optimization operator is to discard groups with poor fitness rather than discarding poor individuals in each group. The process of optimization operator is as follows. After executing the migration and adjusting operators, all individuals are sorted according to their fitness. The total population at this point is twice the initial population, including the parent and newly generated child populations. Then we select half of the population with better fitness to enter the next generation. Performing the migration operator and the adjusting operator increases the population's diversity. After selecting half of the population with better fitness, we use the optimal global individual to replace the one with the worst fitness among the selected individuals. And then, these individuals go into the next generation together.

According to the analysis mentioned above, the flow of the OMBA is as follows.

1) Initialize the population of N monarch butterfly individuals. Randomly select $N1$ individuals as a subpopulation on Land1 and $N2$ individuals as a subpopulation on Land2. Where $N = N1 + N2$.
2) Set the parameters of MBA such as migration period $peri$ (generally 1.2), the migration ratio p (generally $\frac{5}{12}$), butterfly adjusting rate BAR (generally $\frac{5}{12}$), the max walk step of Lévy flight S_{max} (generally 1), and the maximum generation G_{max}.
3) Use the migration operator and the adjusting operator to produce subpopulations on land 1 and 2, respectively. These two subpopulations form a new

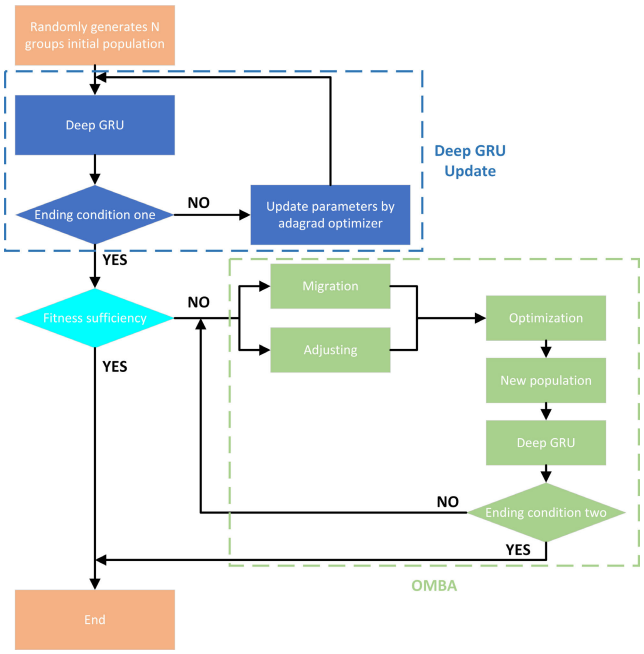

FIGURE 7. The flow chart of deep GRU + OMBA.

population, whose number is N. Then, calculate the fitness of the new population.

4) Sort the original population with the new population according to the fitness value and select half of the individuals with better fitness. The other half is discarded.
5) The individual with the worst fitness in the newly obtained population is replaced with the optimal global individual in the parent population.
6) Continue (3)-(5) when termination criteria is not met. After the algorithm is finished, the optimal individual is obtained.

C. FUSION ALGORITHM

The process of deep GRU + MBA is similar to that of deep GRU + OMBA. The difference is that OMBA has one more step of the optimization operator than MBA. Here, we present the fusion algorithm by the example of deep GRU + OMBA. The flow chart of the deep GRU + OMBA is shown in Fig 7 and its steps can be described as follows:

1) Randomly generates N groups of deep GRU parameters as initial values. Each group of parameters in the initial and subsequent deep GRU represents a set of weights for the deep GRU, a string of n elements, where n is the number of the trainable weight parameters. It is obtained by transforming the deep GRU weights into row vectors and stitching them together. The N groups of deep GRU parameters have the same form as an expression (6).
2) These N groups of initial values are used to train the deep GRU. Calculate the fitness corresponding to each group of parameters and continuously update the deep GRU parameters by adagrad optimizer.
3) Training the deep GRU until the ending condition one is met, that is, the accuracy of the deep GRU meets requirements or reaches preset cycles of the deep GRU training. If the accuracy meets requirements, finish training and export the optimal deep GRU parameters. Else the N sets of parameters of deep GRU constitute the initial population of the OMBA. The OMBA population consists of N individuals, each with n elements, in other words, n dimensions.
4) The N initial individuals obtained from the deep GRU is divided into two subpopulations on Land1 and Land2. Then migration operator, adjusting operator, and optimization operator of the OMBA are used to improve the accuracy of the deep GRU.
5) Repeat the last step until the ending condition two is met, that is, the accuracy of the deep GRU meets requirements or reaches preset cycles of the OMBA.
6) Stop training and export the optimal results.

VI. RESULTS AND ANALYSIS

The GNC module used in this paper was calibrated before the experiment. It is placed stationary in the temperature controlled oven. The sensitive axis of the MEMS accelerometer selected in this paper is perpendicular to the direction of local gravitational acceleration. A computer outside the temperature controlled oven collected its temperature drift output from the beginning of the heating process during the temperature experiment. The data output frequency is 200Hz, and the collecting process lasts more than two hours. So, we can get more than 1,440,000 data. This paper collected two sets of measurements of the MEMS accelerometers. The first set of measurements is used to train the model proposed in this paper to obtain its parameters, namely training data. In contrast, the second set of measurements is used as test data to verify the performance of the model trained with the first set of measurements, namely validation data.

The training of the proposed model is based on a CPU with six cores and six threads, and all the simulation and analysis for experiment data are carried out on Matlab. To remove noise caused by data fluctuations, we smooth the training data for 5s. The learning rate was set as 0.01. Figure 8 shows the simulation results of training the proposed model with the training data. As shown in Figure 8, all three methods fit the training data remarkably well, and we got three set parameters of the deep GRU.

The validation data is then used in these obtained models. To ensure that the validation data is as diverse as possible, we smooth the validation data for 0.01s. Figure 9 shows the fitting results for the validation data using the model parameters obtained earlier, and Figure 10 shows the compensation results for the validation data by the three methods. Intuitively all three methods work well. To quantify the compensation results of different methods, we introduce Allan variance analysis, the standard and most popular method for analyzing the performance of inertial sensors [26], [34], [35]. The Allan method is used to analyze the noise contained in the

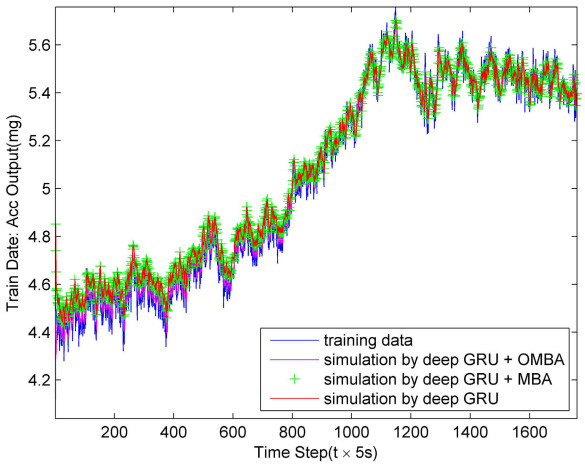

FIGURE 8. Simulation results on training data by the three methods.

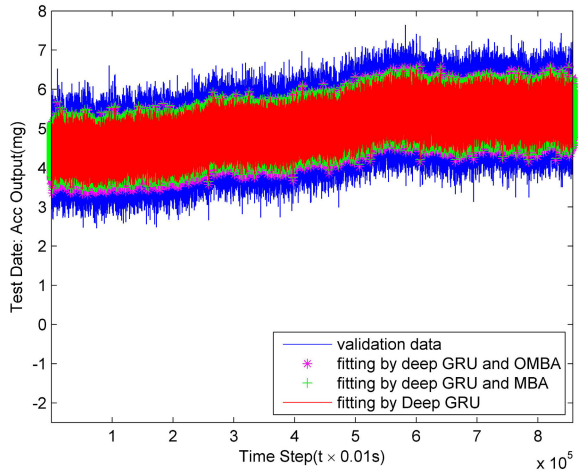

FIGURE 9. Fitting results on validation data by the three methods.

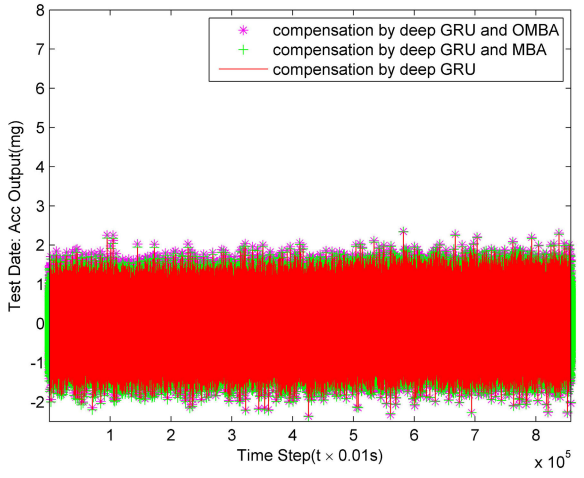

FIGURE 10. Compensation results on validation data by the three methods.

original signal output by the inertial sensor. It has five basic parameters, namely quantization noise (Q), angle random walk (N), bias instability (B), rate random walk (K), and rate ramp (R). Fig 11 shows the results of Allan variance analysis, which shows that the deep GRU+OMBA method has the best compensation effect. Analysis of the validation data shows no part of its Allen variance curve corresponding to quantization noise. Therefore, quantization noise was not considered in the quantitative analysis.

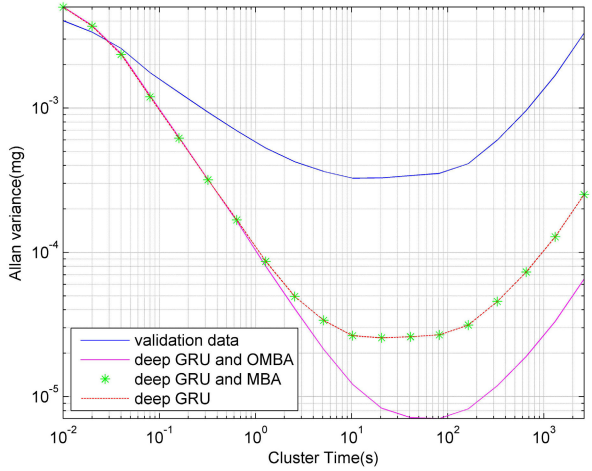

FIGURE 11. Allan variance analysis results after compensation of validation data by the three methods.

TABLE 1. Allan Variance Analysis for Different Compensation Methods.

	$N(mg \cdot s^{\frac{1}{2}})$	$B(mg)$	$K(mg/s^{\frac{1}{2}})$	$R(mg/s)$
Raw	$4.97e^{-4}$	$4.90e^{-4}$	$5.57e^{-5}$	$1.82e^{-6}$
Deep GRU	$7.91e^{-5}$	$3.84e^{-5}$	$4.25e^{-6}$	$1.38e^{-7}$
Improvement(%)	84.08	92.16	92.37	92.39
Deep GRU + MBA	$7.91e^{-5}$	$3.84e^{-5}$	$4.25e^{-6}$	$1.38e^{-7}$
Improvement(%)	84.08	92.16	92.37	92.39
Deep GRU + OMBA	$3.90e^{-5}$	$1.07e^{-5}$	$1.12e^{-6}$	$3.59e^{-8}$
Improvement(%)	92.16	97.80	98.00	98.02

Table 1 shows the quantitative results and the percentage of improvement after compensation by the three methods. It proves that the proposed method offers the best performance. As seen from Table 1, after deep GRU compensation, the four Allen variance coefficients of the accelerometer are increased by 90.27% on average. The original MBA has little optimization effect in this application. After introducing the OMBA, the four Allan variance coefficient is improved further, and their percentage of improvement reaches 96.50% on average. This also proves the effectiveness of our optimization of the MBA.

VII. CONCLUSION

This paper proposed a real-time compensation model and a fusion algorithm for the temperature drift compensation of a MEMS accelerometer. It is a recursive compensation model. The temperature drift prediction for the next moment can be calculated recursively from the model and the temperature drift output for the current moment. For the temperature

drift compensation algorithm of the MEMS accelerometer, three algorithms are proposed: deep GRU, deep GRU + MBA, and deep GRU + OMBA. The test has proven that the proposed real-time compensation model combined with deep GRU + OMBA shows the best performance. First, the temperature drift accuracy is improved by the deep GRU and the real-time compensation model. Second, MBA is used to optimize the parameters of the deep GRU. Next, the optimization algorithms are used to optimize the MBA further to get more accuracy compensation, and the optimized MBA is called OMBA. Finally, a new fusion algorithm of deep GRU + OMBA is proposed. The Allan variance analysis is introduced to quantify the compensation effects of the three algorithms with the original data. Comparison of the original data with the compensation results using deep GRU + OMBA by Allan variance analysis, it shows that N from $4.97e^{-4}\ mg\cdot s^{\frac{1}{2}}$ to $3.90e^{-5}\ mg\cdot s^{\frac{1}{2}}$, B from $4.90e^{-4}\ mg$ to $1.07e^{-5}\ mg$, K from $5.57e^{-5}\ mg/s^{\frac{1}{2}}$ to $1.12e^{-6}\ mg/s^{\frac{1}{2}}$, and R from $1.82e^{-6}\ mg\ /s$ to $3.59e^{-8}\ mg\ /s$ and their percentage of improvement reaches 96.50% on average.

The main contributions of this paper are as follows:

1) A real-time compensation model is proposed in this paper.
2) The deep learning approach is introduced for the temperature drift compensation of MEMS accelerometers.
3) A swarm intelligence optimization algorithm, namely MBA, is introduced to find the optimal weights of GRU.
4) An optimization operator is proposed to optimize the MBA to improve its accuracy in finding the best weights.

The motivation of this paper was to explore a more accurate method for a MEMS accelerometer temperature drift modeling and compensation. The following problems need more discussion in the future.

1) Only a GNC module was employed in testing the proposed method. It is of significance for evaluating the performance of the deep GRU + OMBA method in different GNC modules since other GNC modules perform different temperature drift characteristics.
2) Both the training process and the validation process are based on the computer's CPU in this paper. Investigate ways to simplify the computational effort and then port the verification process to the GNC processor to enable online compensation.

REFERENCES

[1] W. Zhao, Y. Cheng, S. Zhao, X. Hu, Y. Rong, J. Duan, and J. Chen, "Navigation grade MEMS IMU for a satellite," *Micromachines*, vol. 12, no. 2, p. 151, Feb. 2021.

[2] G. de Alteriis, C. Conte, R. S. L. Moriello, and D. Accardo, "Use of consumer-grade MEMS inertial sensors for accurate attitude determination of drones," in *Proc. IEEE 7th Int. Workshop Metrol. Aerosp. (MetroAeroSpace)*, Jun. 2020, pp. 534–538.

[3] H.-C. Chang, Y.-L. Hsu, S.-C. Yang, J.-C. Lin, and Z.-H. Wu, "A wearable inertial measurement system with complementary filter for gait analysis of patients with stroke or Parkinson's disease," *IEEE Access*, vol. 4, pp. 8442–8453, 2016.

[4] S. Qiu, L. Liu, H. Zhao, Z. Wang, and Y. Jiang, "MEMS inertial sensors based gait analysis for rehabilitation assessment via multi-sensor fusion," *Micromachines*, vol. 9, no. 9, p. 442, 2018.

[5] M. Kos and I. Kramberger, "A wearable device and system for movement and biometric data acquisition for sports applications," *IEEE Access*, vol. 5, pp. 6411–6420, 2017.

[6] C. Russo, F. Mocera, and A. Somà, "MEMS sensors for sport engineer applications," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 1038, no. 1, Feb. 2021, Art. no. 012056.

[7] W. Johnston, M. O'Reilly, R. Argent, and B. Caulfield, "Reliability, validity and utility of inertial sensor systems for postural control assessment in sport science and medicine applications: A systematic review," *Sports Med.*, vol. 49, no. 5, pp. 783–818, May 2019.

[8] M. L. Hoang, M. Carratù, V. Paciello, and A. Pietrosanto, "Body temperature—Indoor condition monitor and activity recognition by MEMS accelerometer based on IoT-alert system for people in quarantine due to COVID-19," *Sensors*, vol. 21, no. 7, p. 2313, Mar. 2021.

[9] Z. Fu, G. Zhang, Y. Lin, Y. Liu, and J. Tan, "Calibration and compensation of inertial sensor errors in portable applications—A review," in *Proc. UKACC 11th Int. Conf. Control (CONTROL)*, Aug. 2016, pp. 1–4.

[10] Y. Wang, J. Zhang, Z. Yao, C. Lin, T. Zhou, Y. Su, and J. Zhao, "A MEMS resonant accelerometer with high performance of temperature based on electrostatic spring softening and continuous ring-down technique," *IEEE Sensors J.*, vol. 18, no. 17, pp. 7023–7031, Sep. 2018.

[11] Y. Liu and T. Ma, "Parasitic resistance-based high precision capacitive MEMS accelerometer phase shift and its usage for temperature compensation," *IEEE Sensors J.*, vol. 18, no. 2, pp. 629–634, Jan. 2018.

[12] B. Li, Y. Zhao, C. Li, R. Cheng, D. Sun, and S. Wang, "A differential resonant accelerometer with low cross-interference and temperature drift," *Sensors*, vol. 17, no. 1, p. 178, 2017.

[13] M.-H. Tsai, Y.-C. Liu, K.-C. Liang, and W. Fang, "Monolithic CMOS—MEMS pure oxide tri-axis accelerometers for temperature stabilization and performance enhancement," *J. Microelectromech. Syst.*, vol. 24, no. 6, pp. 1916–1927, Dec. 2015.

[14] S. A. Zotov, B. R. Simon, A. A. Trusov, and A. M. Shkel, "High quality factor resonant MEMS accelerometer with continuous thermal compensation," *IEEE Sensors J.*, vol. 15, no. 9, pp. 5045–5052, Sep. 2015.

[15] Z. Jing, Q. Anping, S. Qin, B. You, and X. Guoming, "Research on temperature compensation method of silicon resonant accelerometer based on integrated temperature measurement resonator," in *Proc. 12th IEEE Int. Conf. Electron. Meas. Instrum. (ICEMI)*, vol. 3, Jul. 2015, pp. 1577–1581.

[16] J. He, J. Xie, X. He, L. Du, and W. Zhou, "Analytical study and compensation for temperature drifts of a bulk silicon MEMS capacitive accelerometer," *Sens. Actuators A, Phys.*, vol. 239, pp. 174–184, Mar. 2016.

[17] G. Ruzza, L. Guerriero, P. Revellino, and F. M. Guadagno, "Thermal compensation of low-cost MEMS accelerometers for tilt measurements," *Sensors*, vol. 18, no. 8, p. 2536, Aug. 2018.

[18] J. Du, Y. Guo, Y. Lin, X. Zheng, and Z. Jin, "A real-time temperature compensation algorithm for a force-rebalanced MEMS capacitive accelerometer based on resonant frequency," in *Proc. IEEE 12th Int. Conf. Nano/Micro Eng. Mol. Syst. (NEMS)*, Apr. 2017, pp. 214–217.

[19] D. Han, J. Bai, Q. Lu, S. Lou, X. Jiao, and G. Yang, "The analysis of temperature effect and temperature compensation of MOEMS accelerometer based on a grating interferometric cavity," *Proc. SPIE*, vol. 9960, pp. 288–294, Aug. 2016.

[20] S. Khankalantary, S. Ranjbaran, and S. Ebadollahi, "Simplification of calibration of low-cost MEMS accelerometer and its temperature compensation without accurate laboratory equipment," *Meas. Sci. Technol.*, vol. 32, no. 4, Apr. 2021, Art. no. 045102.

[21] W. Yang, B. Fang, Y. Y. Tang, and X. Qin, "A temperature compensation model for low cost quartz accelerometers and its application in tilt sensing," *Math. Problems Eng.*, vol. 2016, pp. 1–10, Aug. 2016.

[22] S. Wang, W. Zhu, Y. Shen, J. Ren, H. Gu, and X. Wei, "Temperature compensation for MEMS resonant accelerometer based on genetic algorithm optimized backpropagation neural network," *Sens. Actuators A, Phys.*, vol. 316, Dec. 2020, Art. no. 112393.

[23] Y. Pan, L. Li, C. Ren, and H. Luo, "Study on the compensation for a quartz accelerometer based on a wavelet neural network," *Meas. Sci. Technol.*, vol. 21, no. 10, Oct. 2010, Art. no. 105202.

[24] X. F. Li, D. H. Li, J. M. Gao, and M. S. Pang, "Temperature drift compensation algorithm based on BP and GA in quartzes flexible accelerometer," *Appl. Mech. Mater.*, vols. 249–250, pp. 95–99, Dec. 2012.

[25] X. D. Yu, J. L. Li, S. K. Yan, G. Wei, and G. Li, "Temperature drift compensation based on artificial fish swarm algorithm for quartz flexible accelerometer," *Appl. Mech. Mater.*, vols. 513–517, pp. 4030–4034, Feb. 2014.
[26] M. Zhu, L. Pang, Z. Xiao, C. Shen, H. Cao, Y. Shi, and J. Liu, "Temperature drift compensation for high-G MEMS accelerometer based on RBF NN improved method," *Appl. Sci.*, vol. 9, no. 4, p. 695, Feb. 2019.
[27] Z. Han, L. Hong, J. Meng, Y. Li, and Q. Gao, "Temperature drift modeling and compensation of capacitive accelerometer based on AGA-BP neural network," *Measurement*, vol. 164, Nov. 2020, Art. no. 108019.
[28] J. Wu, T. Huang, Z. Zhu, and K. Song, "Cold starting temperature time-related compensation model of inertial sensors based on particle swarm optimization algorithm," *Rev. Sci. Instrum.*, vol. 92, no. 6, Jun. 2021, Art. no. 065106.
[29] B. Qi, S. Shi, L. Zhao, and J. Cheng, "A novel temperature drift error precise estimation model for MEMS accelerometers using microstructure thermal analysis," *Micromachines*, vol. 13, no. 6, p. 835, May 2022.
[30] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder–decoder approaches," 2014, *arXiv:1409.1259*.
[31] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*.
[32] G.-G. Wang, S. Deb, and Z. Cui, "Monarch butterfly optimization," *Neural Comput. Appl.*, vol. 31, no. 7, pp. 1995–2014, 2019.
[33] Y. Feng, S. Deb, G.-G. Wang, and A. H. Alavi, "Monarch butterfly optimization: A comprehensive review," *Expert Syst. Appl.*, vol. 168, Apr. 2021, Art. no. 114418.
[34] Z. Y. Miao, F. Shen, D. J. Xu, C. M. Tian, and K. P. He, "Online estimation method of Allan variance coefficients for MEMS IMU," *J. Instrum.*, vol. 9, no. 9, Sep. 2014, Art. no. P09001.
[35] S. Chong, S. Rui, L. Jie, Z. Xiaoming, T. Jun, S. Yunbo, L. Jun, and C. Huiliang, "Temperature drift modeling of MEMS gyroscope based on genetic-Elman neural network," *Mech. Syst. Signal Process.*, vol. 72, pp. 897–905, May 2016.

GUO GANG-QIANG received the B.S. degree in electronic science and technology from Xidian University, Xi'an, China, in 2014, and the M.S. degree in computer science and technology from the Xi'an Microelectronic Technology Institute, Xi'an, in 2017, where he is currently pursuing the Ph.D. degree.

His research interests include microsystem integration technology, guidance, navigation and control technology, MEMS inertial sensors, and artificial intelligence.

CHAI BO received the B.S. degree in space flight engineering from Northwestern Polytechnical University, Xi'an, China, in 1987.

He is currently a Professor and the Chief Engineer with the Xi'an Microelectronic Technology Institute. His research interests include launch vehicle, strategic missile computer, tactical missile computer, microsystem integration technology, guidance, navigation and control technology, MEMS inertial sensors, and artificial intelligence.

CHENG RUI-CHU received the B.S. degree in communication engineering from Northwestern Polytechnical University, Xi'an, China, in 2019, and the M.S. degree in computer science and technology from the Xi'an Microelectronic Technology Institute, Xi'an, in 2022.

She is currently an Engineer with the Xi'an Microelectronic Technology Institute. Her research interests include microsystem integration technology, guidance, navigation and control technology, MEMS inertial sensors, and artificial intelligence.

WANG YUN-SHUANG received the B.S. degree in electronic and information engineering and the M.S. degree in electronic and communication engineering from Xidian University, Xi'an, China, in 2012 and 2015, respectively.

She is currently a Senior Engineer with the Xi'an Microelectronic Technology Institute. Her research interests include system integration, guidance, navigation and control technology, and digital signal processing.