Journal of Advanced Marine Engineering and Technology, Vol. 49, No. 6, pp. 560~569, 2025
J. Advanced Marine Engineering and Technology (JAMET)
https://doi.org/10.5916/jamet.2025.49.6.560

ISSN 2234-7925 (Print)
ISSN 2765-4796 (Online)
Original Paper

Calibration of low-grade inertial measurement unit sensors using real-coded genetic algorithms

Minh Thao Pham[1] · Kyoung-Kuk Yoon[†]

(Received December 16, 2025 ; Revised December 22, 2025 ; Accepted December 22, 2025)

Abstract: This paper presents a calibration process that implements a wide variety of solutions, including error modeling, platform design, algorithms, and measurement methods for estimating errors of a low-grade inertial measurement unit sensors (LIMUS), while ensuring cost savings for both instrumentation and computation. Error elements that include bias, non-orthogonality, misalignment, and scale factors are modeled as sensor error models (SEM). A do-it-yourself (DIY) integrated platform is capable of multi-position data collection that can reach up to 48 orientations for accelerometers (ACCE) and rotational velocity for gyroscopes (GYR), crafted using low-cost and available means. The local Earth's gravity is utilized to reference signals of ACCEs, while GYR' readings are compared to encoder samples. Then, real-coded genetic algorithms (R-CGA) are designed to estimate the SEMs. Root mean square residual (RMSR) between readings and reference signals is calculated and considered as the fitness function. Three different populations of 300, 500 and 1000 individuals are studied for evaluating the quality of error estimation. Additionally, analyzing the effect of individual selection probability in the selection operators provide an effective solution that reduces the number of initial individuals in the population and saves computing cost.

Keywords: Error estimation, Low-grade inertial measurement unit sensors, Real-coded genetic algorithms, Root mean square residual, Sensor error models

1. Introduction

Low-grade inertial measurement unit sensors (LIMUS) are widely used in many applications such as navigation systems, automobiles, and wearable devices. A LIMUS chip can integrate accelerometers (ACCE), gyroscopes (GYR) and many different types of sensors by micro-electro-mechanical systems (MEMS) technology while still ensuring a size reduction of a few millimeters. Furthermore, shock resistance, durability, and especially cost decrease are significantly improved **[1]**. However, LIMUSs have a common problem referred to as errors of imperfections in manufacturing and setting up. Users need to combine methodologies, specific instruments and algorithms for the identification and elimination of errors before use. This process is known as calibration. The calibration technique consists of the following steps: the initial step is to identify the LIMUS errors that must be eliminated, including bias, non-orthogonality, misalignment and scale factor. Then, modeling errors make the estimation of error parameters more convenient. Sensor error models (SEMs) are effective models **[2]-[4]** due to the complete inclusion of LIMUS error characteristics.

Secondly, reference signals are chosen based on sensor function. Generally, ACCEs measure the acceleration, hence the local Earth's gravity is considered an ideal and readily available reference signal for ACCE calibration. ACCE calibration can be basically implemented by placing ACCEs stably on a flat surface. ACCE readings are compared to the local Earth's gravity roughly 9.8m/s2. However, this method is limited in one orientation. For example, when a drone performs various tilt maneuvers, the local gravity of Earth acts on different orientations on its body coordinates. As a result, multi-position methods were developed to tackle one-position problem. The recommendation for the number of positions in ACCE calibration is over 21 positions **[5]**. The work in **[4]** implemented 24 orientations for a 3-axis ACCE. Other work in **[6]** points out that a multi-position data collection method shows better error detection ability specially misalignment than six-position calibration. Measuring multi-position

† Corresponding Author (ORCID: http://orcid.org/0000-0001-8612-9574): Professor, Division of Maritime AI & Cyber Security, Korea Maritime & Ocean University, 727, Taejong-ro, Yeongdo-gu, Busan 49112, Korea, E-mail: kkyoon@kmou.ac.kr, Tel: 051-410-4265

1 Ph. D. Candidate, Maritime AI & Cyber Security, National Korea Maritime & Ocean University, E-mail: sqrtthao96@g.kmou.ac.kr

readings requires specific instruments, such as a cube implemented for the six-position method in **[2]**. Many cost-effective platforms that common users could self-craft by available means, are provided in **[3][7][8]**. In case of GYRs, the Earth's rotational velocity is only employed for immediate and high-cost GYRs. Low-grade GYRs are incapable of sensing this slow velocity because of noise, drift and limited accuracy. Therefore, low-grade GYR calibration requires artificial rotational methods such as a turning table. They are easily adjustable for flexible position and high accuracy. However, their investment cost has soared substantially beyond user's budget. References **[9][10]** produce a variety of methods for calibration that do not require external signals. Although, the cost for experiment is less expensive, users should have an advanced levels of knowledge in computer science. To overcome these difficulties, users are encouraged to design platforms that are able for their budget while still ensuring technical criteria.

After the readings are ready, selecting the appropriate algorithm to estimate errors as much as possible is the key to success. The parameters of LIMUS error models were estimated by quasi-newton nonlinear optimization algorithm (Q-NNOA) and root mean square deviation function **[2]-[4]**. The Q-NNOA has the advantage of high computing speed and reduces dependency on cost function evaluation. But results sometimes converge to a local optimum rather than the global optimum **[11][12]**. SEMs are analyzed by Levenberg–Marquardt algorithm (L-MA) **[13][14]**. Robustness improves significantly, while there is a decrease in computing speed compared with the Q-NNOA. The results of **[11]** show that the noisier experimental data used, the more missing the global optimization. Genetic algorithms (GAs) are new candidates to tackle the mentioned disadvantage because of their excellent convergency ability **[15]-[18]**. The readings of ACCEs and GYRs are continuous values. It is difficult to implement them as binary digits because of the "Hamming Cliff" process **[19][20]**. Therefore, real-coded genetic algorithms (R-CGAs) were selected to fit our methodology of searching for optimal solutions in the domain of real numbers. In general, the core of the R-CGAs is similar to traditional GAs, with selection, crossover and mutation operators **[19]**. Many outstanding principles of operators have been published concerning crossover steps **[21]-[24]**, mutation steps **[25]-[29]**, and selection steps **[30][31]**.

This paper presents the calibration of 3-axis ACCEs and GYRs on the LIMUS according to the multi-position method that allow us to obtain a maximum of 48 orientations of ACCEs and 3-axis rotational velocity of GYRs. Bias, non-orthogonality, misalignment and scale factor are modeled as SEMs. Do-it-yourself (DIY) sets are designed for both ACCEs and GYRs. Then, R-CGAs are designed to evaluate value of fitness function, which is root of mean square residual (RMSR) minimization of ACCE and GYR readings and reference signals. The results are remarkable on convergency of 300, 500 and 1000 population size. Additionally, a solution to save computing cost of R-CGAs up to 70% of initial population is contributed by analyzing the effects of efficiency factor in selection operators.

Continuous sections of the paper are organized as follows. The elements of the modeling of LIMUS errors and the analysis of the operators of the R-CCAs for SEMs are demonstrated in Section 2. Section 3 explains how to collect the multi-position data from ACCEs and GYRs using a DIY equipment, and reference signals. The results of experiment and its discussion are given in Section 4. In the end, Section 5 highlights the main contributions of this paper.

2. Materials and Algorithms

2.1 Element Modeling of LIMUS Errors

The operation of LIMUS is affected by numerous errors caused by external sources such as temperature, vibration, and internal characteristics. In the laboratory, the external sources could be dismissed. As a result, the SEMs are formed by remaining components, which include bias, non-orthogonality, misalignment between LIMUS and instruments, and scale factors.

2.1.1 Error Elements

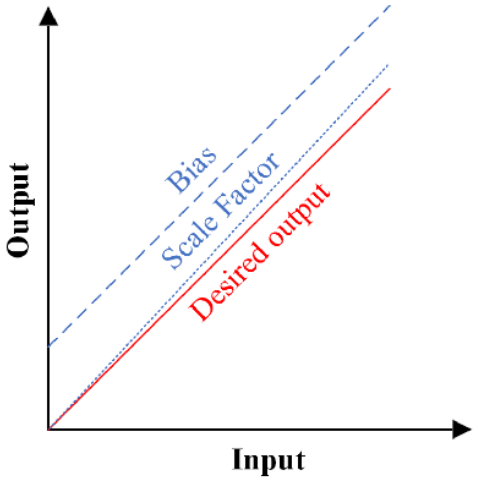

Figure 1: The effect of bias and scale factors on the output of LIMUS

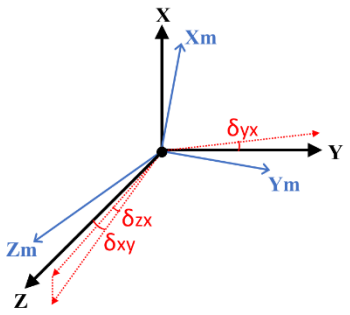

Figure 2: The effect of non-orthogonality and misalignment on the axis of LIMUS

Bias constitutes the majority of the total error in the LIMUSs **[31]**. Bias is an inherent error. It could be analyzed into two independent parts including static bias and varying bias. Its first term, as shown in **Figure 1**, is obtained by averaging input data from LIMUs when they are kept stable. The second term is varying bias, which contributes roughly 10% of total bias. The varying bias deviates significantly in response to temperature fluctuations. Eliminating bias can save over 10^{-1}m/s^2 for ACCEs and 100°/hr^{-1} for GYRs at consumer grade **[6][32]**.

The imperfection in manufacturing processes is the leading cause of non-orthogonality in each axis of LIMUS. Non-orthogonality affects all axes in any direction. An example of non-orthogonal errors is shown in **Figure 2** with red dotted axes. δ_{ij} is a non-orthogonal angle in i, and j axes.

In **Figure 2**, the Xm, Ym and Zm axes indicate the influence of misalignment, which is produced by manual attachment. Misalignment can be limited in a small range when manual correction methods are applied.

As illustrated in **Figure 1**, scale factors are determined as a slope, whose magnitude depends on the ratio of output-input data changes. Because scale factor amplifies practically all other errors, it is considered an important error type. The pattern of scale factor errors shifts from linear to nonlinear as the sensor's range increases **[33]**. In the case of LIMUS, scale factor magnitude can reach percentages instead of parts per million.

2.1.2 Modeling of LIMUS Errors

The errors of the LIMUS are modeled as SEMs to simplify algorithm implementation. The SEMs are defined uniformly in **[8]**, when a rotating instrument is used for both ACCEs and GYRs. In our model, the fixed-multi-position method is selected for ACCEs. The model of ACCE errors is illustrated in **Equation (1)**.

$$A = N_A S_A (R_A - B_A) = [a_x \quad a_y \quad a_z]^T,$$

$$N_A = \begin{bmatrix} 1 & 0 & 0 \\ \delta_{yx}^A & 1 & 0 \\ \delta_{zx}^A & \delta_{zy}^A & 1 \end{bmatrix}, S_A = \begin{bmatrix} s_x^A & 0 & 0 \\ 0 & s_y^A & 0 \\ 0 & 0 & s_z^A \end{bmatrix},$$

$$R_A = \begin{bmatrix} r_x^A & r_y^A & r_z^A \end{bmatrix}^T, B_A = \begin{bmatrix} b_x^A & b_y^A & b_z^A \end{bmatrix}^T \quad (1)$$

Here, A denotes the compensated vector obtained after calibrating the ACCEs. N_A stands for non-orthogonal matrix. Its elements are approximated as constants when non-orthogonal angles are small. S_A is a symmetric matrix of scale factors. R_A represents a raw vector of ACCEs. B_A denotes a bias vector.

The SEMs of GYRs are defined in **Equation (2)**. The bias vector, B_G, could be eliminated by taking average of GYR readings over a short period of time. Therefore, GYR error models are simplified, as demonstrated in **Equation (3)**.

$$G = M_G N_G S_G (R_G - B_G) \quad (2)$$

$$G = M_G N_G S_G R_{G-B} = \begin{bmatrix} g_{xx} & g_{xy} & g_{xz} \\ g_{yx} & g_{yy} & g_{yz} \\ g_{zx} & g_{zy} & g_{zz} \end{bmatrix},$$

$$M_G = \begin{bmatrix} 1 & m_{xy}^G & m_{zx}^G \\ m_{yx}^G & 1 & m_{yz}^G \\ m_{zx}^G & m_{zy}^G & 1 \end{bmatrix}, N_G = \begin{bmatrix} 1 & 0 & 0 \\ \delta_{yx}^G & 1 & 0 \\ \delta_{zx}^G & \delta_{zy}^G & 1 \end{bmatrix},$$

$$S_G = \begin{bmatrix} s_x^G & 0 & 0 \\ 0 & s_y^G & 0 \\ 0 & 0 & s_z^G \end{bmatrix}, R_G = \begin{bmatrix} r_{xx}^G & r_{xy}^G & r_{xz}^G \\ r_{yx}^G & r_{yy}^G & r_{yz}^G \\ r_{zx}^G & r_{zy}^G & r_{zz}^G \end{bmatrix} \quad (3)$$

Here, G denotes the compensated vector obtained after calibrating the GYRs. M_G, N_G, S_G, R_G and B_G are the misalignment matrix, the non-orthogonal matrix, the scale factor matrix, the raw readings vector, and the bias vector of GYRs, respectively. $R_{G\text{-}B}$ denotes the remaining output vector of GYRs after ignoring bias.

2.2 A Three-Dimension (3D) DIY Platform

Specialized instruments are ideal for performing LIMUS calibration. They are constructed from high-quality materials such as steel and high-accuracy joints. Some high-end equipment is equipped with sensors and electronic displays. Typically, tri-axis driven turntables reached 0.01mm of flatness. However, these devices are out of the reach of most users due to their high cost.

To calibrate LIMUS, a 3D-printed DIY platform provides a more reasonable method. This platform must meet certain calibration standards. For ACCEs, when the LIMUS is kept stable in a predefined position, the local gravity along each axis is measured.

Commonly, one axis is chosen as the origin for rotating around the other axes. **Figure 3** illustrates the 3D DIY platform that is used for LIMUS calibration. In **Figure 3**, each side of functional frame includes a 12-sided shaft allowing for 12-predefined positions. In case of rotating the shaft of the functional part around a single axis, for example, the Y-axis we take an additional 4 positions where the other two axes are aligned parallel and perpendicular to the horizontal plane. Thus, a total of 16 predefined positions is measured, as shown in **Figure 4**. Similarly, the 16-predefined-position experiment is conducted with the X-axis and Z-axis. As a result, 48 data samples of the LIMUS are ready. The misalignment between the functional frame and horizontal plane is less than one degree.

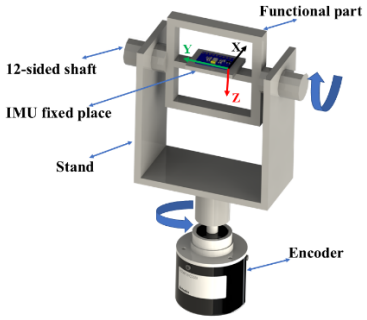

Figure 3: 3D DIY platform for the LIMUS calibration

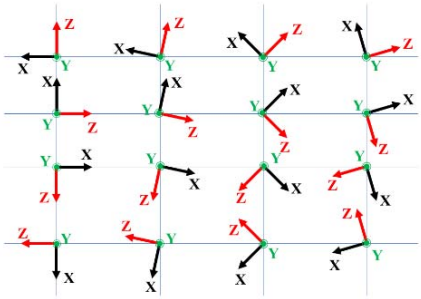

Figure 4: 16-pointed position along Y-axis of LIMUS

Table 1: Specifications of ACCE and GYR

Type	Features		Value
ACCE	Initial tolerance [%]		± 3
	Zero-G [mg]	X, Y	± 60
		Z	± 80
	Scale range [g]		± 2
	Cross-coupling [%]		± 2
GYR	Scale range [°/s]		± 2000
	Initial tolerance [°/s]		± 5
	Cross-coupling [%]		± 2
	Scale factor [%]		± 3

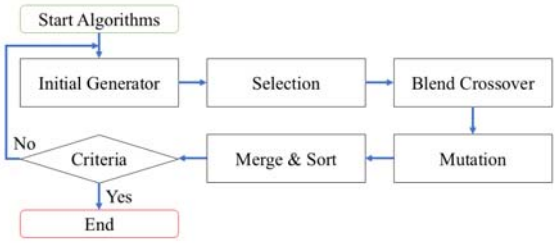

Figure 5: Flow chart of the R-CGAs

Table 2: Constraints of ACCE and GYR

Sensor	Element	Value
ACCE	Bias [mg]	± 80
	Scale Factor [%]	± 3
	Non-orthogonality [°]	± 5
GYR	Misalignment [%]	± 2
	Scale Factor [%]	± 3
	Non-orthogonality [°/s]	± 5

This work uses a small module of motion processing unit (MPU) 9250, measuring 15mm×26mm, for LIMUS calibration. The MPU 9250 is a typical LIMUS and has been deployed in several applications, including 3D motion, wearable devices, and location services. Its design incorporates 3-degree-of-freedom (3-DOF) ACCEs, 3-DOF GYRs and 3-DOF magnetometers in a single chip. A digital Motion Processor allows for the fusion of all sensor data that updates accurate output. It supports the inter-integrated circuit (I^2C) and serial peripheral interface (SPI) communication up to 20MHz. The specs of ACCE and GYR in this experiment are collected from the datasheet of the MPU 9250 indicated in **Table 1**.

2.3 Construction of the R-CGAs

R-CGAs are powerful optimization tools for real numbers. Their operators enable the expansion of the search space by swapping a part of a gene in offspring or mutating them. **Figure 5** depicts the general flow chart of R-CGAs, which consists of six major steps: initial generator, selection, crossover, mutation, merge and sort. The details of these steps are illustrated in the following sections.

2.3.1 Problem

The primary goal of ACCE and GYR error model optimization is to minimize fitness functions, as demonstrated in **Equation (4)** and **(5)**. **Equations (6)** and **(7)** illustrate that the fitness functions represent the RMSR of the compensated matrix and the reference (gravity for ACCEs and rotational angle for GYRs).

$$Optimization(N_A, S_A, B_A) = Minimize(F_A) \quad (4)$$

$$F_A = \sqrt{\frac{\sum_{m=1}^{M}(\|A_m\|-\|G\|)^2}{m}} \quad (5)$$

$$Optimization(M_G, N_G, S_G) = Minimize(F_G) \quad (6)$$

$$F_G = \sqrt{\sum_{i=1}^{I}\sum_{j=1}^{J}\left(\|g_{ij}\| - \omega_{ij}\right)^2} \quad (7)$$

Where A_m denotes the compensated vector for each m attitude, m = *1, 2…, M*, M = *48*, G denotes gravity, g_{ij} and ω_{ij} are i, j elements of the rotated angle matrix of GYRs and platform, respectively.

Constraints are defined based on experimentally validated specifications provided by the manufacturer to ensure the feasibility of the proposed solution. The constraints of ACCEs and GYRs are shown in **Table 2** according to **Table 1**.

2.3.2 Initial Generator

The principle of R-CGAs is based on the natural evolution of the population, which is a set of individuals also called chromosomes. The number of chromosomes in a population, as calculated in **Equation (8)**, is determined by the user. Each chromosome corresponds to one gene chain. Traditionally, a gene is considered a binary digit (typically 0 or 1). Then, many types of encoders have been developed to allow interaction with multiple data categories such as R-CGAs. In this study, the elements of the ACCE and GYR chromosome matrices, chr_A and chr_G, are continuous numbers and are modified from **Equation (1)** and **(3)** to achieve **Equation (9)**.

$$pop = \begin{bmatrix} chr_1 \\ chr_2 \\ \vdots \\ chr_P \end{bmatrix} \quad (8)$$

Here, P is the total number of chromosomes, chr_i, in population, *pop*.

$$chr_A = \begin{bmatrix} \delta_{xy}^A & S_x^A & O_x^A \\ \delta_{zx}^A & S_y^A & O_y^A \\ \delta_{zy}^A & S_z^A & O_z^A \end{bmatrix},$$

$$chr_G = \begin{bmatrix} m_{xy}^G & m_{yx}^G & \delta_{xy}^G & S_x^G \\ m_{xz}^G & m_{zx}^G & \delta_{zx}^G & S_y^G \\ m_{yz}^G & m_{zy}^G & \delta_{zy}^G & S_z^G \end{bmatrix} \quad (9)$$

2.3.3 Selection

It is clear that chromosomes with better genes cause the value of the cost function to approach nearly zero during minimization. Accordingly, they should be given greater opportunities to produce better offspring. However, it is impossible to completely eliminate the weaker chromosomes, as this does not mean that all of their genes are useless. For this strategy, roulette wheels are utilized as selection operators, as shown in **Figure 6**, ensuring that weak chromosomes are also bred, but with low probabilities.

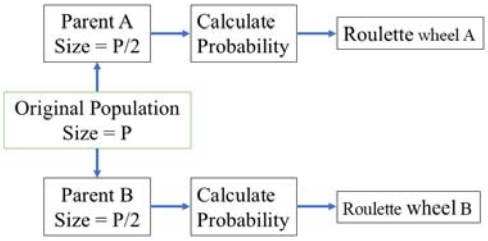

Figure 6: Selection principle based on roulette wheel

Figure 6 illustrates that the original population calculated in **Equation (8)** is slashed into two equal groups, including *A*, and *B*. Then, the probabilities of each parent in groups, ρ, are calculated by using **Equation (10)**.

$$\rho = e^{-\beta\frac{c_p}{\sum_p^P c_p}} \quad (10)$$

Here, c_p is the cost of the p^{th} element, p = *1, 2, …, P*, and β is efficiency factor.

Following that, they are placed in roulette wheels. *P/2* spins of the roulette wheel produce *P/2* pairs of parents. These pairs are used in crossover operators.

2.3.4 Crossover

A pair of selected parents produces a couple of offspring, which inherit more or less of their parents' characteristics depending on the blend crossover operator, as illustrated in **Equation (11)**. Inheritance parameter, α, is the core of this operator. In general, α represents continuous values ranging from 0 to 1. To expand the exploration ability, a small amount, roughly 10% of α, is added, known as the expansion inheritance coefficient, α_e. This coefficient is defined in **Equation (12)**. The outcome of the crossover operator is offspring matrix with *P/2* rows and 2 columns, as illustrated in **Equation (13)**.

$$off_k^A = \alpha_e \odot pr_p^A + (1-\alpha_e)\odot pr_p^B,$$
$$off_k^B = \alpha_e \odot pr_p^B + (1-\alpha_e)\odot pr_p^A \quad (11)$$

$$\alpha_e \in [-0.1; 1.1] \subset \mathbb{R} \quad (12)$$

$$off = \begin{bmatrix} off_k^A & off_k^B \\ off_{k+1}^A & off_{k+1}^B \\ \vdots & \vdots \\ off_{P/2}^A & off_{P/2}^B \end{bmatrix} = \begin{bmatrix} off_i \\ off_{i+1} \\ \vdots \\ off_P \end{bmatrix} \quad (13)$$

Here, off_k^A and off_k^B denote the k^{th} pair of *A* and *B* offspring, pr_p^A and pr_p^B denote the p^{th} parent in the *A* and *B* groups in selection operators, respectively.

2.3.5 Mutation

Mutation operators are critical for maintaining the diversity of the chromosome's gene sequence. In a population, if a large number of chromosomes have comparable genetic sequences, the result may become a local minimum instead of the global minimum. Mutation operators ensure that distinct segments of individuals continue to survive.

The principle of mutation operators is demonstrated by the mutation coefficient, μ, and standard deviation, σ. The mutation coefficient defines the percentage of mutated genes. The value of this parameter was set to 0.1 in this research to avoid slowing convergence speed. The mutation operators process one offspring at a time, so it is necessary to rearrange the size of the offspring matrix from *P/2* rows and 2 columns to *P* rows and 1 column, as shown in **Equation (13)**. The offspring are then mutated using the mutation coefficient and the standard deviation, as given by **Equation (14)** to produce the mutant offspring matrix as calculated by using **Equation (15)**.

$$off_i^M = (1 + \mu \cdot \theta) \cdot off_i \tag{14}$$

$$off^M = \begin{bmatrix} off_i^M \\ off_{i+1}^M \\ \vdots \\ off_P^M \end{bmatrix} \tag{15}$$

Here, off^M denotes offspring matrix after mutating, and θ is continuous numbers generated by a standard normal distribution, as calculated in **Equation (16)**.

$$\theta \sim N(0, \sigma^2) \tag{16}$$

Figure 7 illustrates the principle of merge and sort operators. After following mutation steps, the total number of individuals in parent and offspring matrices has doubled from the original population. It is necessary to resize the population from *2P* to *P* individuals. The value of the cost function of each individual serves as a criterion for sorting and reduce the population size by half. After the k^{th} iteration, the *P* best chromosomes are prepared for the k^{th} + *1* iteration.

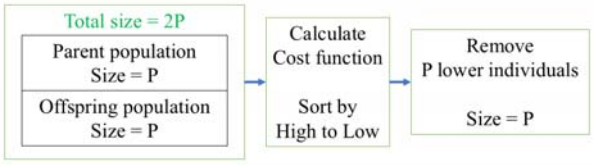

Figure 7: Merge and sort principle. Merging populations then sorting to remove redundant individuals

3. Experiment

3.1 ACCE Calibration

It is assumed that the temperature has no effect on LIMUS, as this experiment is conducted in a laboratory. A 3-DOF ACCE is attached to a DIY plastic frame, as shown in **Figure 3**. The fixed-multi-position method includes a total of 48-predefined positions. Each X, Y and Z axis is responsible for 16 attitudes by rotating 12-sided polygon shaft. An example of Y-axis rotation is shown in **Figure 4**. These DIY sets ensure that the 3-axis ACCE remains stable. The raw data of each attitude, after averaging, is compared to the norm of the gravity vector, which equals 1g (or 9.8m/s^2). There are 5 steps to collect the data from ACCEs as follows:

Step 1: Attach the MPU 9250 LIMUS to the frame.

Step 2: Power the system and hold steady for 10 seconds to neglect the effect of the warming-up process.

Step 3: Take 1000 samples of each attitude.

Step 4: Averaging the samples to limit noise and random walks.

Step 5: Remeasure the acceleration of 16 attitudes for X and Z axis.

3.2 GYR Calibration

Due to noise, low-grade GYRs are unable to sense the rotation velocity of the Earth. Hence, it is necessary to making rotation speed manually as references. The DIY platform is connected to an encoder, which is used for reading the rotational velocity. The encoder shaft is employed as the rotation axis. The process for GYR calibration is summarized as follows:

Step 1: Supply power to the GYR and wait for the system to stabilize.

Step 2: Fix the X-axis of the LIMUS to the encoder shaft.

Step 3: Read stable data of the 3-axis GYR in 10 seconds.

Step 4: Rotate the platform manually and measure the 3-axis output. The reference is obtained by calculating the response revolution of encoder.

Step 5: Similarly, implement steps 2 to 4 for Y and Z-axis.

4. Results and Discussion

In the calibration of ACCEs, the multi-predefined-position method is implemented with 48 orientations. The RMSR between the ACCE readings of each fixed orientation and local

Table 3: Estimated ACCE SEM by R-CGA and Q-NNOA. RMSR between ACCE and the local gravity before and after calibration

Type	Axis	Value	RMSR [g]	
			Before	After
Non-orthogonality [°]	yx	-0.00009	0.0793	$4.6e^{-4}$
	zx	-0.0015		
	zy	0.0028		
Scale factors [-]	x	0.9984		
	y	0.9985		
	z	0.9915		
Bias [g]	x	-0.1218		
	y	-0.0584		
	z	0.0197		

gravity is calculated to evaluate the quality of solutions. **Table 3** shows the results for ACCE error model elements, which are estimated by R-CGAs and Q-NNOA, as collected in **[4]**. It is obvious that Q-NNOA reacts quickly and accurately to static data. In the case of R-CGAs, the accuracy is equivalent to Q-NNOA with the RMSR roughly $4.6e^{-4}$g. However, the computing cost of the R-CGAs is higher than that of the Q-NNOA. The reason is that R-CGAs operate on the principle of natural evolution with complex operators. To achieve good solutions, a large population must be generated in the initial step. Therefore, reducing the initial population size and increasing specialized coefficients of operators are key to overcoming the high computational cost of R-CGAs.

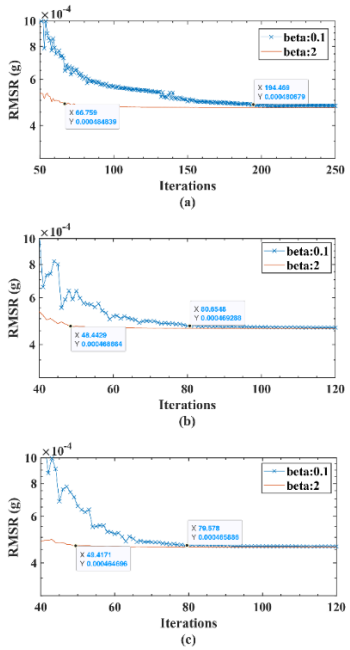

Figure 8: Effect of β on convergence on population size of (a) P = 300, (b) P = 500, and (c) P = 1000

Promoting the mutational process of genes might lead to unexpected results, such as non-convergence due to overstepping. The expansion parameter of the crossover operators is not a potential candidate for this strategy either. It should be kept below 0.15 to avoid repeating the same issues with the mutation operators. Selection operators act on the initial population without any changing impact to the genes. We increase mating opportunities for individuals with lower cost than the rest. This task was completed by assessing the impact of β on populations of 300, 500, and 1000 individuals, respectively, as shown in **Figure 8(a), (b), and (c)**. It can be clearly seen that the effect of β variation decreases with an increase in the number of individuals. In **Figure 8(a)**, the brown line corresponding to $\beta = 2$ reached the near optimum area at the 67th iteration, while the blue line ($\beta = 0.1$) takes 127 more iterations than the brown line. This gap narrows to approximately 32 and 30 iterations for populations of 500 and 1000 individuals in **Figure 8(b)**, and **(c)**. The reason is that the increase in the number of individuals leads to denser distribution. As a result, 70% of the initial population can be reduced from 1000 to 300 individuals when the efficiency coefficient is greater than 2.

Figure 9 shows the decrease in RMSR of population with 300 individuals over 500 iterations.

The ACCE error models estimated by R-CGAs are used to calibrate 48 orientations of ACCE readings. **Figure 10** shows that the deviation is dramatically reduced in all directions. The best value is found in the 6th orientation roughly $9.12e^{-5}$g. In contrast, the worst deviation is just $9.86e^{-4}$g in the 9th direction.

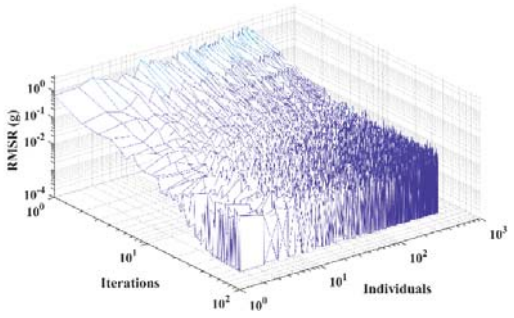

Figure 9: RMSR of population with P = 300 over 500 iterations

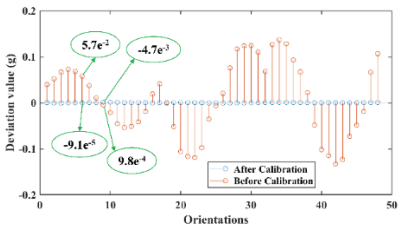

Figure 10: Deviation of 48 orientations before and after ACCE calibration

The 3D-printed DIY platform lacks automatic rotating parts. Additionally, the encoder shaft has a small resistance, so we used a soft wire to provide manual rotational force to the platform. The instability in force supply, imperfection of structure, and noise cause asymmetry for each axis in GYR test, especially at high rotation rates. It is necessary to perform the GYR readings in angular domain. The integral of the rotation rate over a period, which includes full error information, facilitates the estimation process. After taking the mean of the first 500 samples to remove bias, an interval where the GYR data is stable for each axis is selected.

Table 4: Deviation of GYR axes before and after calibration by R-CGAs and factorization algorithm

Axis	Before [°]	R-CGAs [°]	Factorization [°]
xx	7.4378	0.0101	0.4275
xy	1.7305	0.0034	-0.4037
xz	7.2812	0.000013	-10.8235
yx	4.537	-0.0016	0.5073
yy	9.9278	0.00081	-0.0364
yz	5.1652	0.00099	2.1457
zx	35.822	0.00097	10.8812
zy	3.0052	-0.0017	-2.3403
zz	6.1739	0.00081	-1.0616

Table 5: Estimated GYR Error Elements by R-CGAs

Type	Value		
M_G [-]	1	-0.0088	-0.039
	0.0081	1	-0.0137
	-0.00023	-0.0089	1
N_G [°]	1	0	0
	-0.0145	1	0
	-0.0496	0.0014	1
S_G [-]	0.9799	0	0
	0	0.9743	0
	0	0	0.9814

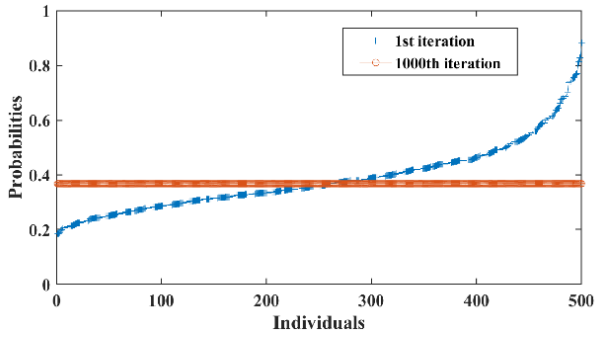

Figure 11: Probability distribution (see **Equation (10)**) of 500 individuals at the 1st and 1000th iteration

Table 4 shows that there is a significant deviation between the MPU 9250' GYRs and the encoder. The errors reached a maximum value of 35.822°. The results obtained from factorization algorithms based on **[4]** are not as efficient as those from R-CGAs. The maximum deviation of our R-CGAs is roughly 0.01°, while the value of the GYR calibration algorithms is over 10.88°. Generally, the Cholesky factorization method has difficulty with the data in this test because the elements in the raw matrix of GYRs vary significantly in magnitude. To overcome this difficulty, a powerful computing algorithm, such as R-CGAs, is needed. **Table 5** illustrates the GYR error elements estimated by R-CGAs.

Chromosomes exchange parts of their genes with others to create better solutions in the crossover operator. They become more similar around the optimum area. Consequently, the probability of selection is divided equally. **Figure 11** shows that the slope of the 1st iteration (blue line) becomes almost linear by 1000th iteration (orange line).

Conclusion

The purpose of this paper is to help common users reduce the cost of ACCE and GYR calibration. The cost of experimental instruments is reduced by designing 3D DIY sets to measure the 48 predefined positions of ACCEs and the tri-axial rotational velocity of GYRs on the same platform. SEMs, including bias, non-orthogonality, misalignment, and scale factors have been estimated by R-CGAs to eliminate errors as much as possible. The RMSR of ACCE calibration remains at $4.6e^{-4}$g. In the case of GYR calibration, the smallest deviation value is $1.3e^{-5}$°. The cost of computation is also decreased by cutting 70% of the initial population while remaining excellent convergence. Users will have additional choices related to measuring methods, platform design, and algorithms that are suitable for their backgrounds.

Acknowledgement

This research was supported by the Excellent Company Research Institute Development Program (ATC+) (No. 20017957) funded by the Ministry of Trade, Industry & Energy (MOTIE, Korea).

Author Contributions

Conceptualization, M. T. Pham and K.K Yoon; Methodology, M. T. Pham; Software, M. T. Pham; Validation, M. T. Pham and K.K Yoon; Formal Analysis, M. T. Pham; Investigation, M. T. Pham; Resources, M. T. Pham; Data Curation, M. T. Pham;

Writing—Original Draft Preparation, M. T. Pham; Writing—Review & Editing, K.K Yoon; Visualization, M. T. Pham; Supervision, K.K Yoon; Project Administration, K.K Yoon; Funding Acquisition, K.K Yoon.

References

[1] K. Borodacz, C. Szczepański, and S. Popowski, "Review and selection of commercially available IMU for a short time inertial navigation," Aircraft Engineering and Aerospace Technology, vol. 94, no. 1, pp. 45–59, 2022.

[2] J. Lv, A. A. Ravankar, Y. Kobayashi, and T. Emaru, "A method of low-cost IMU calibration and alignment," in 2016 IEEE/SICE International Symposium on System Integration (SII), pp. 373-378, 2016.

[3] U. Qureshi and F. Golnaraghi, "An algorithm for the in-field calibration of a MEMS IMU," IEEE Sensors Journal, vol. 17, no. 22, pp. 7479–7486, 2017.

[4] J. Rohac, M. Sipos, and J. Simanek, "Calibration of low-cost triaxial inertial sensors," IEEE Instrumentation & Measurement Magazine, vol. 18, no. 6, pp. 32–38, 2015.

[5] M. Sipos, P. Paces, J. Rohac, and P. Novacek, "Analyses of triaxial accelerometer calibration algorithms," IEEE Sensors Journal, vol. 12, no. 5, pp. 1157–1165, 2011.

[6] Z. F. Syed, P. Aggarwal, C. Goodall, X. Niu, and N. El-Sheimy, "A new multi-position calibration method for MEMS inertial navigation systems," Measurement Science and Technology, vol. 18, no. 7, pp. 1897–1907, 2007.

[7] J. Botero-Valencia, D. Marquez-Viloria, L. Castano-Londono, and L. Morantes-Guzmán, "A low-cost platform based on a robotic arm for parameters estimation of Inertial Measurement Units," Measurement, vol. 110, pp. 257–262, 2017.

[8] P. Garcia Mena, IMU Calibration platform, Bachelor's thesis, Universitat Politècnica de Catalunya, Spain, 2016.

[9] D. Tedaldi, A. Pretto, and E. Menegatti, "A robust and easy to implement method for IMU calibration without external equipments," in 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 3042–3049, 2014.

[10] W. T. Fong, S. K. Ong, and A. Y. C. Nee, "Methods for in-field user calibration of an inertial measurement unit without external equipment," Measurement Science and Technology, vol. 19, no. 8, 085202, 2008.

[11] T. J. VanderNoot and I. Abrahams, "The use of genetic algorithms in the non-linear regression of immittance data," Journal of Electroanalytical Chemistry, vol. 448, no. 1, pp. 17–23, 1998.

[12] E. P. dos Santos, C. R. Xavier, P. Goldfeld, F. Dickstein, and R. Weber dos Santos, "Comparing genetic algorithms and Newton-like methods for the solution of the history matching problem," Computational Science–ICCS 2009: 9th International Conference Baton Rouge, pp. 377-386, 2009.

[13] K. S. Tee, M. Awad, A. Dehghani, D. Moser, and S. Zahedi, "Triaxial accelerometer static calibration," in Proceedings of the World Congress on Engineering, pp. 25-27, 2011.

[14] R. Zhang, F. Hoflinger, and L. M. Reind, "Calibration of an IMU using 3-D rotation platform," IEEE Sensors Journal, vol. 14, no. 6, pp. 1778–1787, 2014.

[15] I. Costa-Carrapiço, R. Raslan, and J. N. González, "A systematic review of genetic algorithm-based multi-objective optimisation for building retrofitting strategies towards energy efficiency," Energy and Buildings, vol. 210, p. 109690, 2020.

[16] A. Vukadinović, J. Radosavljević, A. Đorđević, M. Protić, and N. Petrović, "Multi-objective optimization of energy performance for a detached residential building with a sunspace using the NSGA-II genetic algorithm," Solar Energy, vol. 224, pp. 1426–1444, 2021.

[17] X. Shi, W. Long, Y. Li, D. Deng, and Y. Wei, "Research on the performance of multi-population genetic algorithms with different complex network structures," Soft Computing, vol. 24, no. 17, pp. 13441–13459, 2020.

[18] T. Alam, S. Qamar, A. Dixit, and M. Benaida, "Genetic algorithm: Reviews, implementations, and applications," International Journal of Engineering Pedagogy (iJEP), vol. 10, no. 6, pp. 57–77, 2020.

[19] K. Deep, K. P. Singh, M. L. Kansal, and C. Mohan, "A real coded genetic algorithm for solving integer and mixed integer optimization problems," Applied Mathematics and Computation, vol. 212, no. 2, pp. 505–518, 2009.

[20] R. A. Caruana and J. D. Schaffer, "Representation and hidden bias: Gray vs. binary coding for genetic algorithms," Machine learning proceeding, pp. 153–161, 1988.

[21] G. Singh, N. Gupta, and M. Khosravy, "New crossover operators for real coded genetic algorithm (RCGA)," 2015

International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS), pp. 135-140, 2015.

[22] Y. C. Chuang, C. T. Chen, and C. Hwang, "A real-coded genetic algorithm with a direction-based crossover operator," Information Sciences, vol. 305, pp. 320–348, 2015.

[23] Y. F. Jin, Z. Y. Yin, S. L. Shen, and D. M. Zhang, "A new hybrid real-coded genetic algorithm and its application to parameters identification of soils," Inverse Problems in Science and Engineering, vol. 25, no. 9, pp. 1343–1366, 2017.

[24] S. Picek and M. Golub, "Comparison of a crossover operator in binary-coded genetic algorithms," WSEAS Transactions on Computers, vol. 9, no. 9, pp. 1064–1073, 2010.

[25] P. H. Tang and M. H. Tseng, "Adaptive directed mutation for real-coded genetic algorithms," Applied Soft Computing, vol. 13, no. 1, pp. 600–614, 2013.

[26] A. Haghrah, M. Nazari-Heris, and B. Mohammadi-Ivatloo, "Solving combined heat and power economic dispatch problem using real coded genetic algorithm with improved Mühlenbein mutation," Applied Thermal Engineering, vol. 99, pp. 465–475, 2016.

[27] K. Deb and D. Deb, "Analysing mutation schemes for real-parameter genetic algorithms," International Journal of Artificial Intelligence and Soft Computing, vol. 4, no. 1, pp. 1–28, 2014.

[28] A. Haghrah, M. A. Nekoui, M. Nazari-Heris, and B. Mohammadi-Ivatloo, "An improved real-coded genetic algorithm with random walk based mutation for solving combined heat and power economic dispatch," Journal of Ambient Intelligence and Humanized Computing, vol. 12, no. 8, pp. 8561–8584, 2021.

[29] J. C. S. Garcia, H. Tanaka, N. Giannetti, Y. Sei, K. Saito, M. Houfuku, et al., "Multiobjective geometry optimization of microchannel heat exchanger using real-coded genetic algorithm," Applied Thermal Engineering, vol. 202, 117821, 2022.

[30] P. Subbaraj, R. Rengaraj, and S. Salivahanan, "Enhancement of self-adaptive real-coded genetic algorithm using Taguchi method for economic dispatch problem," Applied Soft Computing, vol. 11, no. 1, pp. 83–92, 2011.

[31] Y. Wan, M. Wang, Z. Ye, and X. Lai, "A feature selection method based on modified binary coded ant colony optimization algorithm," Applied Soft Computing, vol. 49, pp. 248–258, 2016.

[32] P. D. Groves, Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems, 2nd ed., United Kingdom: Artech House, 2013.

[33] X. Zhang, C. Zhou, F. Chao, C. M. Lin, L. Yang, C. Shang, et al., "Low-cost inertial measurement unit calibration with nonlinear scale factors," IEEE Transactions on Industrial Informatics, vol. 18, no. 2, pp. 1028–1038, 2022.