Magnetometer Calibration Using Genetic Algorithms

Egor L. Chekhov, Dmitry A. Antonov, Leonid A. Kolganov, Aleksey V. Savkin

Moscow Aviation Institute (MAI), Volokolamskoe Highway 4, 125993 Moscow, Russia

Abstract **– This paper describes the method for three-axis magnetometer calibration using genetic algorithms suitable for execution on industrial microcontrollers. Algorithm structure, description, simulation as well as field testing results are presented.**

Keywords **– calibration, genetic algorithms, magnetometer.**

1. Introduction

Remotely operated underwater vehicles (ROUVs) are widely used for industrial and scientific purposes, such as inspection of underwater pipelines, structural testing of offshore platforms, scientific studies and observations.

According to the Russian Federation state standard GOST R 56960-2016, unmanned underwater vehicles are classified as per Figure 1. (State standard, 2016).

ROUV spends underwater most of the time, which presents a number of challenges when it comes to navigation. The ROUV has to be able to obtain its orientation to stay stabilized when using video cameras for surveying or moving towards waypoints.

There exist a number of state-of-the-art methods for obtaining ROUV's orientation angles. For heavy class ROUVs inertial navigation systems (INS) are used, but for smaller ROUVs price and size of a typical INS are prohibitory.

DOI: 10.18421/TEM93-10
https://doi.org/10.18421/TEM93-10

Corresponding author: Dmitry A. Antonov,
Moscow Aviation Institute (MAI), Moscow, Russia.
Email: antonov_dm.a@mail.ru

Received: 16 April 2020.
Revised: 16 June 2020.
Accepted: 25 June 2020.
Published: 28 August 2020.

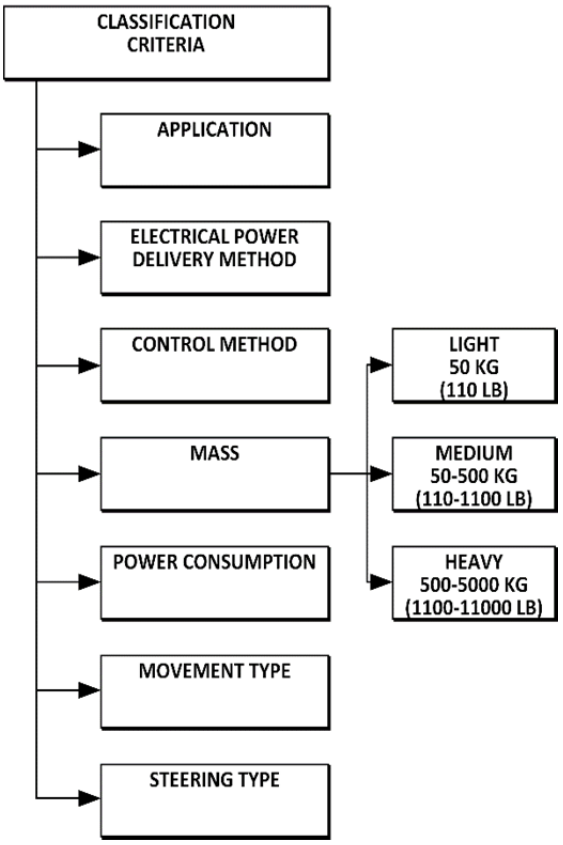

Figure 1. ROUV classification based on mass

For light and medium ROUVs compact attitude and heading reference systems (AHRS) are usually utilized. Such compact systems use magnetometer and accelerometer data to obtain heading which is then filtered using MEMS gyroscope readings.

Magnetometer-based heading sensors have the benefit of substantially smaller size and lower power consumption, but are prone to local disturbances in the magnetic field, which mostly originate from the vehicle itself, so they can be calibrated.

During development of AHRS for medium class ROUV (see Figure 2. and Figure 3.), magnetometer bias and gain calibration were implemented using well-known recursive least square (RLS) filter [1].

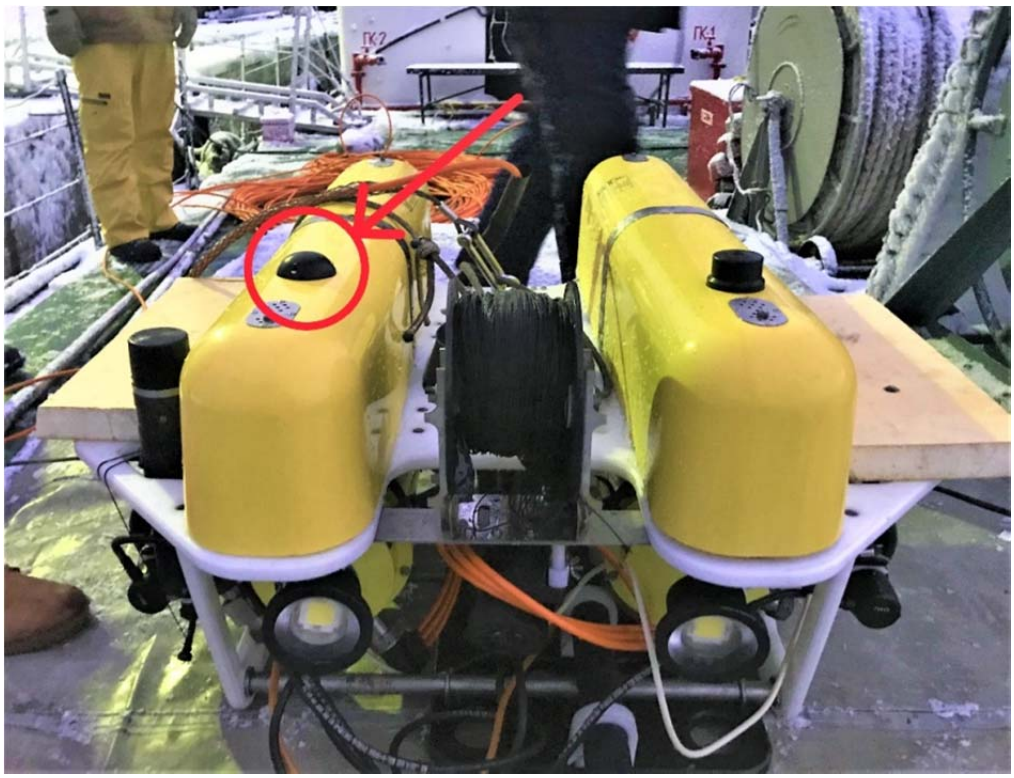

Figure 2. AHRS installed on ROUV

Figure 3. AHRS's housing and printed circuit board

Field testing revealed that hard and soft iron errors induced by ROUV's equipment and structural elements significantly impacted RLS filter ability to converge. As calibration should be performed with AHRS installed on ROUV, the dataset is skewed due to ROUV's limited ability to make full sphere rotations. Example of real data from a magnetometer during calibration is shown in Figure 4.

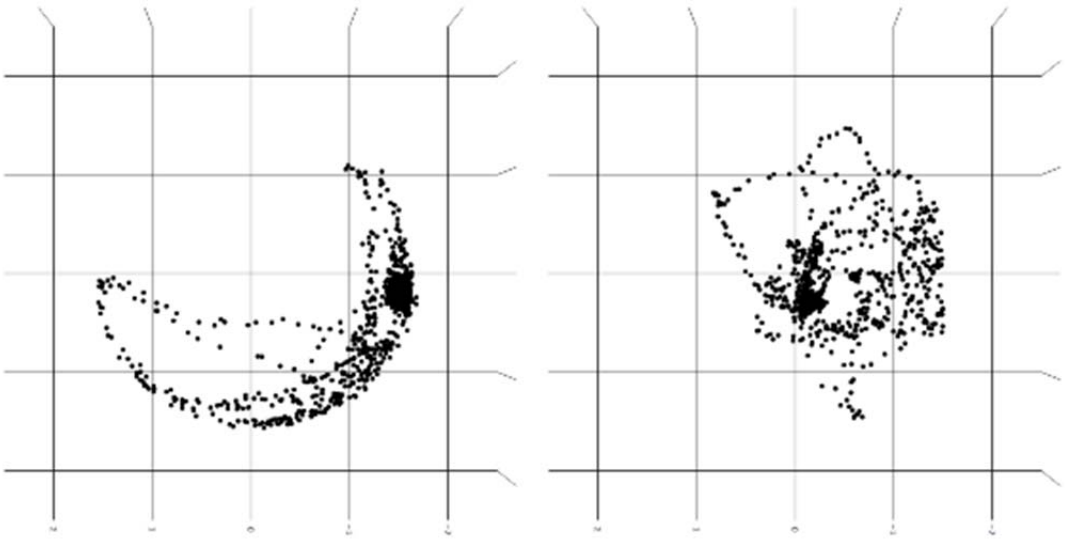

Figure 4. Real magnetometer calibration data

Multiple attempts to improve convergence have been taken, including:

- running multiple calibration cycles with incremental improvement of initial error estimation;
- adjusting regularization parameters to improve convergence;
- manually setting initial errors values.

Using all of the above methods allowed achieving 5 deg (RMS) accuracy, but it does not meet the 1 deg (RMS) design requirement.

The main reason for unsatisfactory RLS convergence is believed to be inadequate linearization of error model used in RLS. The model has to be linear in order to be parametrized with RLS, but originally an ellipsoid fitting problem is not linear and the original error model is linearized using substitution. Model used in RLS filter is derived as follows:

$$\vec{X}' = \vec{X} - K\vec{X} - \Delta\vec{X}$$

$$K = \begin{bmatrix} k_{00} & 0 & 0 \\ 0 & k_{11} & 0 \\ 0 & 0 & k_{22} \end{bmatrix}, \Delta\vec{X} = \begin{bmatrix} \Delta x_0 \\ \Delta x_1 \\ \Delta x_2 \end{bmatrix},$$

where

$\vec{X}$ are the measured values;

$\vec{X}'$ are the actual values;

$\Delta x_0, \Delta x_1, \Delta x_2$ are the bias errors;

k_{00}, k_{11}, k_{22} are the gain errors.

Since the earth magnetic field is constant when performing calibration, one can write:

$$(x_0(1-k_{00}) - \Delta x_0)^2 + (x_1(1-k_{11}) - \Delta x_1)^2 + (x_2(1-k_{22}) - \Delta x_2)^2 = 1$$

$$x_0^2(1-k_{00})^2 + \Delta x_0^2 - 2x_0(1-k_{00})\Delta x_0 + x_1^2(1-k_{11})^2 + \\ + \Delta x_1^2 - 2x_1(1-k_{11})\Delta x_1 + x_2^2(1-k_{22})^2 + \Delta x_2^2 - 2x_2(1-k_{22})\Delta x_2 = 1$$

$$-2x_0a_0 + x_0^2a_1 - 2x_1a_2 + x_1^2a_3 - 2x_2a_4 + a_5 = -x_2^2$$

$$a_0 = \Delta x_0 \frac{(1-k_{00})}{(1-k_{22})^2}, a_1 = \frac{(1-k_{00})^2}{(1-k_{22})^2}, a_2 = \Delta x_1 \frac{(1-k_{11})}{(1-k_{22})^2},$$
$$a_3 = \frac{(1-k_{11})^2}{(1-k_{22})^2}, a_4 = \Delta x_2 \frac{1}{(1-k_{22})}, a_5 = \frac{\Delta x_0^2 + \Delta x_1^2 + \Delta x_2^2 - 1}{(1-k_{22})^2},$$

where

$a_0, a_1, a_2, a_3, a_4, a_5$ are the coefficients of the linearized model.

Linearized model requires additional conversions from linearized model coefficients $a_0, a_1, a_2, a_3, a_4, a_5$ to error model coefficients $\Delta x_0, \Delta x_1, \Delta x_2, k_{00}, k_{11}, k_{22}$, which in some cases can lead to division by zero. More complications arise when one tries to define initial values for RLS covariance matrix P. Even if error model coefficients uncertainties are known, linearized model uncertainties cannot be directly derived due to nonlinear dependencies.

After analyzing alternative approaches to error estimation, a genetic algorithm (GA) was chosen as a solution for the convergence problem.

There are several publications in which feature magnetometer calibration is using GA. In [2] the experiment was performed in open place far away from magnetic anomalies and power supply system. In [3] processing of real measurements was not performed at all, and in [4] experiment conditions are not clearly stated. In all of the described papers calibration was performed using PC, which is not always possible in practice, especially when magnetometer is already installed and is not readily accessible.

We present the method for three-axis magnetometer calibration using GA suitable for execution on industrial microcontrollers and capable to handle significant induced hard and soft iron errors.

2. Calibration Accuracy Requirements

The design goal of AHRS compass heading accuracy is 1 deg (RMS). During calibration using GA a magnetic field magnitude deviation from unity is minimized. Heading accuracy is translated to accuracy of the magnetic field magnitude as follows:

$$E_{RMS}(\psi) = 1\ deg \sim 0.018\ rad$$
$$E_{RMS}(|X|) = 0.018$$

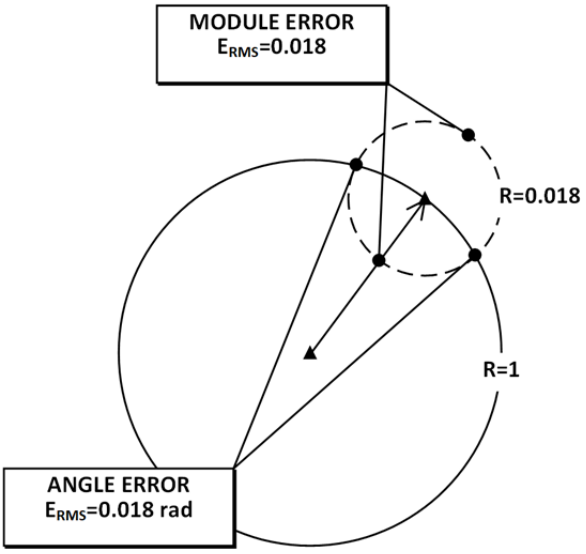

Figure 5. Magnetometer data after calibration

3. Magnetometer Error Model

For GA calibration misalignment errors were added to the magnetometer error model, which became the following:

$$\hat{\vec{X}} = \vec{X} - K\vec{X} - \Delta\vec{X}$$

$$K = \begin{bmatrix} k_{00} & 0 & 0 \\ k_{10} & k_{11} & 0 \\ k_{20} & k_{21} & k_{22} \end{bmatrix}, \Delta\vec{X} = \begin{bmatrix} \Delta x_0 \\ \Delta x_1 \\ \Delta x_2 \end{bmatrix},$$

where

$\vec{X}$ are the measured values;

$\vec{X}'$ are the actual values;

$\Delta x_0, \Delta x_1, \Delta x_2$ are the bias errors;

k_{00}, k_{11}, k_{22} are the gain errors;

k_{10}, k_{20}, k_{21} are the misalignment errors.

9 errors (bias, gain and misalignment) were encoded into solution vector:

$$\vec{S} = \{\Delta x_0, \Delta x_0, \Delta x_0, k_{00}, k_{10}, k_{11}, k_{20}, k_{21}, k_{22}\}$$

Where $\vec{S}$ is the GA solution vector.

4. Calibration Algorithm

Fitness function was defined and based on the assumption that earth's magnetic field is constant during calibration and its normalized magnitude is equal to 1, thus fitness function is defined:

$$F(\vec{S}, \vec{D}_0, \dots, \vec{D}_n) = -\sum_{i=0}^{n} \left(1 - \left|\vec{D}_i - K\vec{D}_i - \Delta\vec{X}\right|\right)$$

$$K = \begin{bmatrix} s_3 & 0 & 0 \\ s_4 & s_5 & 0 \\ s_6 & s_7 & s_8 \end{bmatrix}, \Delta\vec{X} = \begin{bmatrix} s_0 \\ s_1 \\ s_2 \end{bmatrix},$$

where

$\vec{D}_i$ is the magnetometer measurement;

F is the fitness function.

There are a lot of implementations of GAs. For this particular problem we chose to implement GA with elitism [5] selection strategy to guarantee convergence. The calibration algorithm was

implemented in C programming language and firstly tested on simulated dataset using consumer grade PC. It became apparent that GA is computationally demanding and requires optimization in order to execute on microcontroller [6], [7]. The following performance improvements were implemented:

- mutation step reduction was changed from linear ($h_i = h_0 - i \cdot a$) to hyperbolic; ($h_i = h_0 - h_0 \cdot i \cdot (i + a)^{-1}$),
- solution evaluation was performed on a random subset of original dataset allowing to reduce evaluation dataset for each step.

Mutation step reduction has a parameter (Δh and a) which defines how fast mutation step decreases [9]. Formula for calculating parameter a based on final required solution tolerance is:

$$t = h_0 - h_0 \cdot i_{max} \cdot (i_{max} + a)^{-1}$$
$$a = t \cdot i_{max} \cdot (h_0 - t)^{-1}$$

where
t is the target parameter tolerance;
h_0 is the initial step size;
i_{max} is the number of iterations;
a is the step decay speed.

For our case:

$$a = 0.01 \cdot 7500 \cdot (0.1 - 0.01)^{-1} = 1646$$

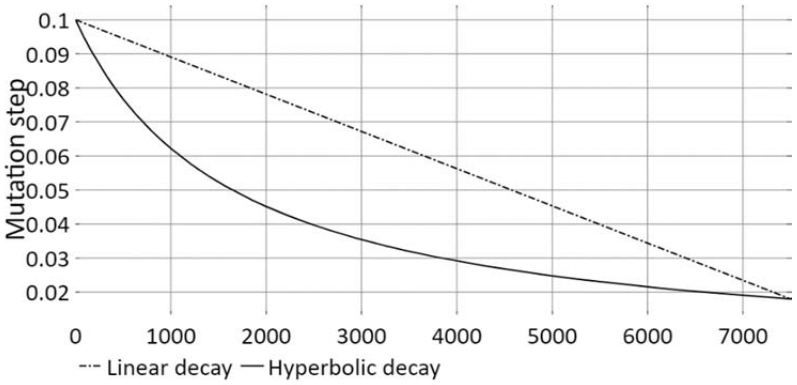

Figure 6. Mutation step reduction comparison

Data subset size parameter was manually optimized and based on repetitive testing. The resulting GA global parameters are presented in Table 1.

Table 1. GA global parameters

Parameter	Value
Solution size	9
Population size	32
Dataset size	4096
Random data subset size	64
Mutation probability	1/9
Crossover probability	0.8
Elite solutions	2
Initial mutation step	0.1
Iterations	7500

5. Algorithm Evaluation Using Simulated Data

To evaluate GA performance, magnetometer readings were generated. Algorithm for generating magnetometer readings is:

$$u_i = rand\left(-\frac{\pi}{2}, \frac{\pi}{2}\right), v_i = rand(-\pi, \pi),$$
$$\hat{\vec{X}}_i = \{\cos(u_i)\cos(v_i), \sin(u_i), \cos(u_i)\sin(v_i)\}$$

$$\vec{X}_i = \hat{\vec{X}}_i + K\hat{\vec{X}}_i + \Delta\vec{X} + \delta\vec{X}$$

$$K = \begin{bmatrix} 0.2 & 0 & 0 \\ 0 & 0.2 & 0 \\ 0 & 0 & 0.2 \end{bmatrix}, \Delta\vec{X} = \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \end{bmatrix},$$

where
$\vec{X}_i$ are the magnetometer readings;
$\hat{\vec{X}}_i$ are the undistorted magnetometer readings;
$\delta\vec{X}$ is the Gaussian noise with $m = 0, \sigma = 0.024$;
i is the number of generated sample;
$rand(a, b)$ is the random generator with uniform distribution on interval $[a, b]$.

Calibration algorithms evaluation on generated dataset showed that standard RLS outperforms our approach by a small margin (Table 2.). It should be noted that GA algorithm allows misalignment errors estimation contrary to RLS 6 dimensional model [10], [8].

Table 2. Simulated dataset

Error component	RLS prediction error	GA prediction error
Bias $\Delta x_0, \Delta x_1, \Delta x_2$	0.001 0.001 0.001	0.025 0.023 0.015
Gain k_{00}, k_{11}, k_{22}	0.020 0.033 0.034	0.082 0.018 0.021

Error component	RLS prediction error	GA prediction error
Misalignment k_{10}, k_{20}, k_{21}	- - -	0.023 0.024 0.025
Fitness (mean) $\|1 - \|X\|\|$	0.031	0.039

6. Algorithm Evaluation Using Collected Data

Algorithms evaluation on real dataset showed that standard RLS failed to converge during the first run, which is clearly seen from Figure 7 meant for corrected values. Method using GA converged successfully with RMS error of 0.024 which is higher than the actual value due to magnetometer noise with standard deviation of 0.016. Corrected magnetometer values using GA are shown in Figure 8. Comparison of estimated error values are shown in Table 3.

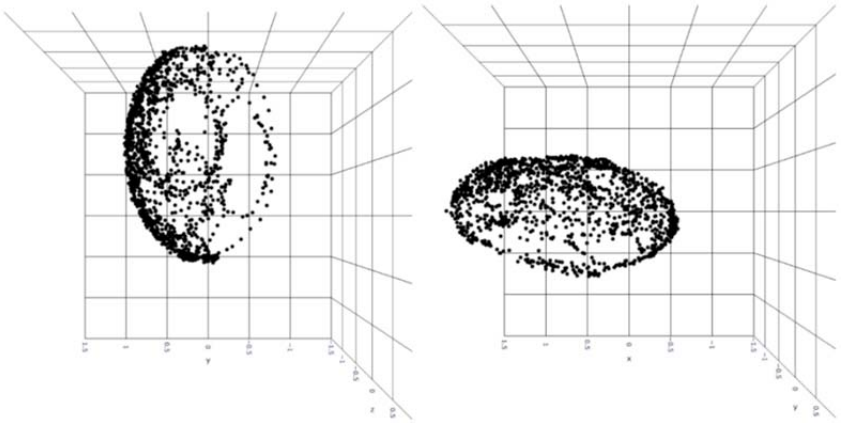

Figure 7. Magnetometer readings corrected with errors estimated by RLS

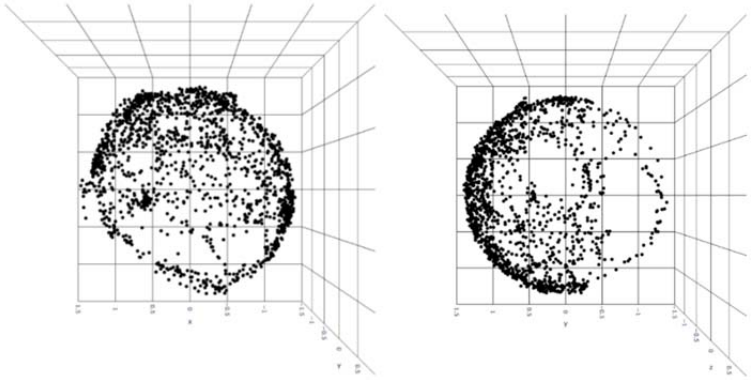

Figure 8. Magnetometer readings corrected with errors estimated by GA

Table 3. Real dataset

Error component	RLS prediction	GA prediction
Bias $\Delta x_0, \Delta x_1, \Delta x_2$	-3.564 -0.056 -0.417	-3.824 -0.098 -0.390
Gain k_{00}, k_{11}, k_{22}	1.300 0.575 0.276	1.277 1.303 1.440
Misalignment k_{10}, k_{20}, k_{21}	- - -	-0.003 -0.023 -0.025
Fitness (average) $\|1 - \|X\|\|$	0.326	0.024

7. Field Tests

After proving that magnetometer calibration using GA is implemented and configured correctly, calibration algorithm was ported to AHRS. AHRS utilizes 32-bit microcontroller and running at 100 Mhz. Running full calibration cycle on target processor takes 10 minutes to complete.

Finally, real calibration was performed and AHRS readings were compared to a reference compass (Figure 9.).

Figure 9. Reference compass used during field testing

Relative AHRS heading errors are shown in Table 4. In all cases AHRS heading error did not exceed 3 deg.

Table 4. Simulated dataset

Reference system heading	AHRS heading	AHRS error
350	350	+0
35	36	-1
80	77	+3
125	125	+0
170	167	+3
215	215	+0
260	257	+3
305	307	-2

GA calibration was used later during the mission (Figure 10.) and proved to be an improvement over RLS, because of its ability to reliably converge even with large induced errors. During operation AHRS heading accuracy proved to be sufficient for operator to navigate above and under water. Navigation to waypoints was performed using GNSS position and AHRS heading.

8. Conclusions

The method for three-axis magnetometer calibration using GA is presented. Algorithms were verified using simulated and collected magnetometer readings. Maximum heading error of 3 degrees is achieved during field tests.

There are still difficulties when it comes to utilizing GA in microcontrollers, as it consumes a significant amount of time (approx. 10 min), so further optimizations should be implemented.

Figure 10. ROUV during operation

Acknowledgments

The reported study was funded by RFBR according to the research project № 19-08-01223.

References

[1].Simon, D. (2006). *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons.

[2].Pang, X. L., Jiang, P., & Lin, C. S. (2015). The calibration method of three-axis magnetometer based on genetic algorithm. In *Applied Mechanics and Materials* (Vol. 722, pp. 373-378). Trans Tech Publications Ltd.

[3].Kim, E., & Bang, H. C. (2007, October). Bias estimation of magnetometer using genetic algorithm. In *2007 International Conference on Control, Automation and Systems* (pp. 195-198). IEEE.

[4].Sarcevic, P., Pletl, S., & Kincses, Z. (2014, September). Evolutionary algorithm based 9DOF sensor board calibration. In *2014 IEEE 12th International Symposium on Intelligent Systems and Informatics (SISY)* (pp. 187-192). IEEE.

[5].Obitko, M. (1998) *Introduction to genetic algorithms.* Retrieved from: https://www.obitko.com/tutorials/genetic-algorithms. [accessed: 05 February 2020].

[6].Troni, G., & Whitcomb, L. L. (2019). Field sensor bias calibration with angular-rate sensors: Theory and experimental evaluation with application to magnetometer calibration. *IEEE/ASME Transactions on Mechatronics*, *24*(4), 1698-1710.

[7].Yang, D., You, Z., Li, B., Duan, W., & Yuan, B. (2017). Complete tri-axis magnetometer calibration with a gyro auxiliary. *Sensors*, *17*(6), 1223.

[8]. Wu, Y., Zou, D., Liu, P., & Yu, W. (2017). Dynamic magnetometer calibration and alignment to inertial sensors by Kalman filtering. IEEE Transactions on Control Systems Technology, 26(2), 716-723.

[9].Riwanto, B. A., Tikka, T., Kestilä, A., & Praks, J. (2017). Particle swarm optimization with rotation axis fitting for magnetometer calibration. *IEEE Transactions on Aerospace and Electronic Systems*, *53*(2), 1009-1022.

[10]. Kok, M., & Schön, T. B. (2016). Magnetometer calibration using inertial sensors. *IEEE Sensors Journal*, *16*(14), 5679-5689.