



**I. G. Tsmots<sup>1</sup>, V. M. Teslyuk<sup>1</sup>, V. G. Rabyk<sup>2</sup>, Yu. V. Opotyak<sup>1</sup>, Yu. V. Tsymbal<sup>1</sup>, O. V. Paziuk<sup>1</sup>**

<sup>1</sup> Lviv Polytechnic National University, Lviv, Ukraine

<sup>2</sup> Ivan Franko National University of Lviv, Lviv, Ukraine

## RECOVERY OF LOST NAVIGATION DATA IN MOBILE ROBOTIC PLATFORMS

Mobile robotic platforms (MRP) are increasingly used in various areas of human activity. When using them, an important task is to determine the spatial orientation, measure the parameters of the movement of the MRP, etc. One of the problems that arises in determining navigation data and other measured parameters is their loss at a specific period, for example, due to interference or the temporary loss of visibility of GNSS navigation satellites. However, the functioning of navigation components of onboard radio-electronic devices for measuring movement parameters and determining the spatial orientation of the MRP requires the availability of primary navigation information without loss and in real time. Therefore, it is necessary to recover lost navigation data, especially in the case of MRPs, using onboard facilities with limited computing performance. Modern algorithms for recovering lost navigation data are analyzed, and it is determined that the publications do not pay enough attention to implementing these algorithms, taking into account the limitations of embedded systems. For implementation in mobile robotic platforms, an algorithm using the principal component analysis (PCA) method was selected, which, with low computational complexity, provides sufficient accuracy of data recovery. Using the developed algorithm in mobile robotic platforms provides data processing on a computing platform with limited resources. It allows streaming data to be processed on the MRP's coordinates in real-time. Modern microcontrollers and systems on a chip (SoC) will enable you to solve the problem of recovering lost navigation data, taking into account restrictions on weight, dimensions, power consumption, etc. A structural diagram of a means for measuring motion parameters and determining spatial orientation for ground MRP has been developed. It is determined that the main components of the tool are a set of navigation sensors using a GPS/GNSS-based coordinate determination module. Data recovery tools have been created using the ESP32-C3 microcontroller, GNSS module type M10Q-5883, which contains a digital compass module QMC5883L and an accelerometer and gyroscope module MPU-6050. Debugging and testing the developed tools for recovering lost navigation data for MRPs have been performed. Analysis of the test results shows that the platform using the ESP32-C3 microcontroller provides data processing in 43 milliseconds. For the rate of GNSS data arrival at one measurement per second, this is enough to provide real-time mode.

**Keywords:** navigation sensors, motion parameters, measurement accuracy, microcontroller platform.

## Introduction

Mobile robotic platforms (MRPs) are increasingly used in various areas of human activity. When using them, an important task is determining the spatial orientation, measuring the parameters of the MRP movement, etc. One of the problems that arises in determining navigation data and other measured parameters is their loss at a specific time period, for example, due to interference, temporary loss of visibility of GNSS navigation satellites, etc. However, the functioning of navigation components of onboard radio-electronic means for measuring movement parameters and determining the spatial orientation of the MRP requires the availability of primary navigation information without loss

and in real time. Therefore, it is necessary to recover lost navigation data, especially in the case of MRP, using onboard facilities that have limited computing performance. Modern microcontrollers and systems on a chip (SoC) allow you to solve the problem of recovering lost navigation data, taking into account restrictions on weight, dimensions, power consumption, etc.

Therefore, an urgent task is to study methods and means for restoring lost navigation data for use in onboard radio-electronic systems for measuring MRP movement parameters, which take into account the requirements of a specific application and provide high technical and operational characteristics.

*The object of research* is the process of recovering lost navigation data for use in onboard radio-electronic means to determine the spatial orientation of MRP.

*The subject of the study* is methods and means of restoring lost navigation data for the implementation of onboard radio-electronic means for determining the spatial orientation of the MRP.

*The purpose of the work* is to develop means of recovering lost navigation data for the implementation of onboard radio-electronic systems to determine the spatial orientation of MRP in real time, taking into account limitations on computing performance.

To achieve this goal, the following *main objectives of the study* are defined:

- to analyze the structure of means for determining the spatial orientation of the MRP;
- to analyze modern algorithms for recovering lost navigation data in mobile robotic platforms;
- perform the choice of algorithms for implementation, which provides sufficient accuracy of data recovery and can be implemented on a computing platform with limited resources;
- to develop a structural diagram of a means for measuring motion parameters and determining spatial orientation for ground MRPs;
- determine the components of the lost navigation data recovery tool for the MRP;
- test the developed tools, evaluate their performance for lost data recovery tasks.

***Analysis of recent research and publications.*** In mobile robotic platforms [1–3], one essential element is providing navigation and obtaining navigation data in real time. Analysis of recent publications demonstrates attention to methods and means of recovering lost data. The papers show that measurement data is affected by several factors: unstable communication, environmental factors, synchronization problems, and unreliable nodes. For streaming data, the following are often used: Kalman filters for smoothing and prediction, interpolation methods based on the last known values, variants of the PCA method (streaming, incremental), and regression methods. Statistical methods (averaging, interpolation); regression models; CHM-based method, PCA method, or other methods with recovery based on correlation structure are used to recover missing data in batch data. [4] proposes a regression approach to solving the problem of missing data recovery based on Ito decomposition and the AdaBoost algorithm. Increasing the dimensionality of the input space due to the use of the second-degree Ito decomposition scheme and its high approximation properties made it possible to improve the accuracy of filling in the missing values.

The study [5] proposes an approach to recovering missing data based on the spatiotemporal correlation between nodes in the network, and missing data can be recovered using neighboring nodes using the ST-Hierarchical Long-Term Memory (ST-HLSTM) algorithm. [6] uses complete tensor-based data recovery models such as

Canonical Bayes Gaussian Polyadic Decomposition (BGCP), Extended Bayesian Tensor Factorization (BATF), and High-Precision Low-Rank Tensor Termination (HaLRTC) to recover missing data. In [7], the algorithm of Sobolev reconstruction in wireless sensor networks was investigated, which consistently recovers missing data even in situations of mass loss. A data recovery algorithm [8] based on attribute correlation and extremely randomized trees (ACET) is considered, in which the Spearman correlation coefficient is used to construct a correlation model between different attributes. The correlation model is used to select other attributes that have a strong correlation with that attribute, and then uses them to train highly randomized trees, and the lost data can be recovered by the trained model.

A modified stochastic gradient descent (SGD) algorithm for training linear models in the presence of missing data is discussed in the paper [9]. The algorithm does not require modeling of data distribution and works without assumptions about the distribution of input data. This makes it more versatile in application. The algorithm adapts to a different number of missing values in different features, ensuring correct model training and achieving the optimal convergence rate. When implementing the batch mode of the learning algorithm, new values of the weighting factors are calculated at each iteration, and in the stochastic mode, one training vector is randomly selected from the training sample at each iteration.

The paper [10] proposes a method for filling in the missing air pollution monitoring data, based on a neural network approach. After training the ANN with the selected architecture, the missing data is predicted in the network's operating mode. For this purpose, a non-iterative neural network [11] based on a model of geometric transformations was chosen. This neural-like neural network can operate in autoassociative (without a supervisor) and supervisor mode. In the autoassociative mode, learning is carried out using geometric Gramm – Schmidt transformations in  $n$ -dimensional space, where  $n$  is the number of features of the model. As a result of the transformations, an intermediate coordinate system is formed. Its direction coincides with the longest axis of the dispersion ellipsoid.

The paper [12] investigates the problems associated with the incremental principal component analysis method (PCA). PCA updates are performed in real time and do not require all measurement (observation) data to be stored in memory. In real time, the values of the main components can change by leaps and bounds due to a change in the basis when new measurements are added, which complicates the interpretation of the results. To achieve smoothness of changes in the main components, a correction method is proposed, which takes into account changes in the matrix of transformations when adding new measurements. This is especially important for principal components with close eigenvalues. The correction method allows you to reduce jumps in the values of the main components, although it can affect the accuracy of calculations.

A detailed review of the methods of robust principal component analysis (RPCA) and its dynamic variant, robust subspace tracking (RST), is discussed in the paper [13]. RPCA aims to separate data into a low-level matrix (valid data) and a sparse matrix (outliers). This allows you to process data that contains anomalies or outliers efficiently. Methods enabling you to effectively track the variable subspace in streaming data in conditions of limited computing resources are considered. The paper presents the conditions under which the RPCA and RST algorithms provide accurate recovery of subspace and outliers, as well as their computational efficiency. An overview of the advantages and disadvantages of different approaches is provided, which helps choose the appropriate method for working with data containing emissions or having a variable structure.

The paper [14] discusses estimating parameters in linear dynamical systems with incomplete measurements or in the presence of missed measurement data. These parameters are the transition matrix of states  $A$ , the observation matrix  $C$ , the noise covariances  $Q$  and  $R$ , and the initial values. To solve this problem, combining the Kalman filter (for processing hidden states) with the EM algorithm (for evaluating parameters) is proposed. The Kalman filter very naturally recovers missed data. A forecast is performed for a specific time step. If there are no measurements for this step, the measurement correction step is skipped.

Paper [15] shows that the choice of the element base and navigation components for onboard equipment for measuring movement parameters and determining the spatial orientation of the MRP should be based on an integrated approach, which includes measurement methods and modern methods for evaluating the effectiveness of the use of the element base. However, as the analysis shows, the publications do not pay enough attention to the issues of recovering lost data from sensors, taking into account the limitations of embedded systems. Therefore, the problem of recovering lost navigation data in mobile robotic platforms remains relevant.

## Research results and their discussion

**Causes of missing data in datasets.** The loss of navigation data in mobile robotic platforms at specific time intervals is due to the temporary lack of visibility of GNSS navigation satellites, deliberate interference from electronic warfare (EW), etc. This complicates processing on navigation data only, since the accuracy of analysis, modeling, and interpretation of results can significantly reduce distortion of results.

The causes of missing data can vary and depend on technical and human factors. The main ones are:

1. Technical malfunctions. Malfunctions or failures of measuring instruments, communication breaks, or errors during data reading can lead to the loss of part of the measurement results. For example, sensors on the MRP may stop recording coordinates or acceleration due to the influence of electronic warfare equipment.

2. Physical conditions. Data collected in the field (meteorological, environmental, or biomedical observations) can be lost due to difficult operating conditions – precipitation, dust, electromagnetic interference, etc.

3. Limitations in software or data collection protocols. Incorrectly configured algorithms for reading, storing, or transmitting information may not provide for situations in which specific parameters have a null or non-standard value, resulting in ignoring or losing this data.

Depending on the nature and scale of such losses, missed values can significantly affect the quality of the analysis. The results may be inaccurate or even false if appropriate measures are not taken to detect, process, or recover them. This, in turn, can negatively affect decision-making and the effectiveness of systems based on this data.

**Basic methods for recovering missing data.** For many modern applications in science and technology, data is collected by streaming or batch methods. Streaming data is continuously received in real time and requires instant or sequential processing. Data may not be fully stored, but it is processed in real time with minimal latency and used for real-time monitoring and analysis. Examples of streaming data: data from the GPS module on the vehicle; data from an accelerometer or gyroscope on an autonomous robotic platform; real-time data from medical sensors (pulse, temperature, ECG).

Stream data is collected and processed in portions (batches). Processing is performed intermittently, or data is often stored before processing. Most often, batch data is used for statistical analysis and model training. Examples of batch data: files with recorded historical GPS tracks; log files stored hourly / daily and processed together; the results of a monthly survey or collection of electricity consumption indicators.

Streaming data carries information that changes over time and must be processed in a timely manner using devices with limited memory and hardware resources for decision-making. This is often combined with the problem of missing data, when not all data is measured (observed). To reveal these problems, the paper considers an example of the movement of an autonomous robotic platform, which receives data from the GPS module for orientation on the ground. GPS data is received with a specific error and may not be received due to the loss of signal from satellites. The task is to implement algorithms for recovering missed data from the GPS module, and their implementation on the ESP-32 microcontroller with limited memory. The paper discusses the algorithms of batch and incremental PCA, as well as the Kalman filter for recovering missed data.

**Method of the principal components analysis.** The PCA method with sliding window involves processing a fixed set of recent measurements with dimension  $N$ . At each time step, the window is updated: all measurement vectors are shifted (the 2nd measurement vector becomes the first, ..., the  $N$ th vector becomes the  $N-1$ st) and a new measurement vector is added to the end of the window. Let's assume that each of the measurement vectors has a

dimension  $m$  and the measurement data from the sensors arrive at a constant time interval. After  $N$  measurements, we get a matrix of measurements

$$X = \begin{bmatrix} x_{11} & x_{11} & \dots & x_{1m} \\ x_{21} & x_{22} & \dots & x_{2m} \\ \dots & \dots & \dots & \dots \\ x_{N1} & x_{N2} & \dots & x_{Nm} \end{bmatrix} \in \mathbb{R}^{N \times m}. \quad (1)$$

*Step 1.* Calculate the average for each of the columns of the  $X$  matrix.

$$\bar{x}_j = \frac{1}{N} \sum_{i=1}^N x_{ij}. \quad (2)$$

Where  $j = 1, \dots, m$ .

*Step 2.* Centering the measurement matrix  $X$ . Subtract the average values from each row of the measurement matrix.

$$X_C = \begin{bmatrix} x_{11} - \bar{x}_1 & x_{11} - \bar{x}_2 & \dots & x_{1m} - \bar{x}_m \\ x_{21} - \bar{x}_1 & x_{22} - \bar{x}_2 & \dots & x_{2m} - \bar{x}_m \\ \dots & \dots & \dots & \dots \\ x_{N1} - \bar{x}_1 & x_{N2} - \bar{x}_2 & \dots & x_{Nm} - \bar{x}_m \end{bmatrix}. \quad (3)$$

*Step 3.* Obtaining a covariance matrix:

$$Q = \frac{1}{N-1} X_C^t \cdot X_C \in \mathbb{R}^{m \times m}. \quad (4)$$

*Step 4.* Calculation of eigenvalues and eigenvectors of the covariance matrix  $Q$ . Let's denote the eigenvalues  $\lambda_i$ , and through  $\bar{w}_i$  – the orthogonal eigenvectors  $\bar{w}_i = (w_{i1}, w_{i2}, \dots, w_{im})$  of the matrix  $Q$ , which are related to each other by a dependence:

$$Q \cdot \bar{w}_i = \lambda_i \cdot \bar{w}_i, \quad i = 1, \dots, m. \quad (5)$$

The eigenvalues of the symmetric non-negative matrix  $Q$  are real and positive numbers. Let's order them in descending order  $\lambda_1 > \lambda_2 > \dots > \lambda_m$ . Similarly, we will place the eigenvectors corresponding to  $\lambda_i$ . Then the matrix  $W$  determines the linear transformation

$$\bar{y} = W \cdot \bar{x}, \quad (6)$$

where  $\bar{y}$  is the vector of the main components of the PCA. In other words, the PCA method performs the conversion of an input vector  $\bar{x} \in \mathbb{R}^m$  into an output vector  $\bar{y} \in \mathbb{R}^m$ .

The first principal component  $\bar{y}_1$  is associated with  $\lambda_1$  by the eigenvector  $\bar{w}_1$ , defines the direction in the  $m$ -dimensional space in which the variance of the data is maximum. While the smallest principal component  $\bar{y}_m$  indicates the direction in which the variance is minimal. Each of the eigenvalues  $\lambda_i$  corresponds to the variance of the  $i$ -th principal component. To determine the eigenvalues and eigenvectors of the autocovariance matrix  $Q$ , the Jacobi method was used in this work.

*Step 5.* Recovery of missed data. Let's assume that at the beginning of the algorithm for recovering missing data,  $N$  measurements are known, i.e. the sliding window is completely filled. In this case, the missed data may appear only in the last line of the sliding window of dimension  $N$  and only the input vector  $x_N$  will need to be restored.

To recover missing data, you can use not all eigenvectors, but only the first  $k$  with the maximum fraction of variance. The contribution of each of the main components to the overall variation of the data is determined by the expression:

$$m_i = \frac{\lambda_i}{\sum_{i=1}^m \lambda_i}. \quad (7)$$

The inverse transformation consists of finding the vector  $\bar{x}$  from the expression

$$\bar{x} = W^t \bar{y}. \quad (8)$$

Let us assume that the vector  $x_N$  lies in the subspace described by the matrix  $W$  of eigenvectors. To restore it, we will use the expression

$$x_N \approx \bar{x} + W_k^t y_N, \quad (9)$$

where  $W_k^t$  is the matrix  $k$  of eigenvectors corresponding to the largest eigenvalues. Also, in the expression (9) decentration is performed when obtaining a vector  $x_N$ . Therefore, to restore the missing vector  $x_N$ , it is necessary to evaluate the vector of the principal components  $y_N$ . Let us define it as the mean of the vectors of the principal components  $y_1, y_2, \dots, y_{N-1}$ .

**Algorithm for recovering missed data based on the Kalman filter.** The Kalman filter is the theoretical basis for various recursive methods used in stochastic linear dynamical systems. The algorithm is based on the idea that the unknown state of a system can be estimated based on measurements obtained over time. The Kalman filter evaluates the state of a system in the format of a state space based on dynamical systems. The change of state from a point in time  $t_{k-1}$  to a moment in time  $t_k$  is described by a discrete equation (equation of dynamics)

$$x_k = Ax_{k-1} + Bu_k + w_k, \quad (10)$$

where  $x_k \in \mathbb{R}^n$  is the vector of the state of the system;  $A \in \mathbb{R}^{n \times n}$  – state transition matrix;  $u_k \in \mathbb{R}^l$  – input control vector;  $B \in \mathbb{R}^{n \times l}$  – input control matrix;  $w_k \in \mathbb{R}^n \sim N(0, Q)$  – process noise vector;  $Q \in \mathbb{R}^{n \times n}$  – process noise covariance matrix (determines the noise of the system and simulates perturbations that are not taken into account in the equation of motion).

Equation of measurement of a system (equation of statics)

$$z_k = Hx_k + v_k, \quad (11)$$

defines the relationship between the state of the system and the measurement at time  $t_k$ . In the expression (11)  $z_k \in \mathbb{R}^m$  – the vector of measurements;  $H \in \mathbb{R}^{m \times n}$  – matrix of observations;  $v_k \in \mathbb{R}^m \sim N(0, R)$  – measurement noise vector;  $R \in \mathbb{R}^{m \times m}$  – measurement noise covariance matrix (determines sensor measurement errors).

Each iteration of the Kalman filter algorithm consists of two steps: extrapolation and correction. During extrapolation, the estimate of the state  $\hat{x}_k^-$  based on the estimate  $\hat{x}_{k-1}^+$  and the covariance matrix of the estimate of errors  $P_k^-$  are calculated according to the expressions:

$$\hat{x}_k^- = A\hat{x}_{k-1}^+ + Bu_{k-1}, \quad (12)$$

$$P_k^- = AP_{k-1}^+ A^t + Q. \quad (13)$$

During the correction step, the Kalman matrix (gain matrix) is calculated

$$K_k = P_k^- H^t (HP_k^- H^t + R)^{-1}, \quad (14)$$

The updated state of the system and the updated error estimation covariance matrix are calculated according to the expressions:

$$\hat{x}_k^+ = \hat{x}_k^- + K_k y_k, \quad (15)$$

$$P_k^+ = (E - K_k) P_k^-, \quad (16)$$

where  $E$  is the identity matrix.

Consider a mathematical model that describes the movement of the MRP. The state vector at the step  $t_k$  looks like:

$$\mathbf{x}_k = [x_k, y_k, z_k, v_{x,k}, v_{y,k}, v_{z,k}]^t, \quad (17)$$

where  $x_k, y_k, z_k$  are the MRP coordinates,  $v_{x,k}, v_{y,k}, v_{z,k}$  are the components of the MRP velocity.

State transition matrix  $A$ :

$$A = \begin{bmatrix} 1 & 0 & 0 & T_s & 0 & 0 \\ 0 & 1 & 0 & 0 & T_s & 0 \\ 0 & 0 & 1 & 0 & 0 & T_s \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad (18)$$

where  $T_s$  is the sampling time. Measurement vector in step  $t_k$ :

$$\mathbf{z}_k = [x_k^m, y_k^m, z_k^m]^t, \quad (19)$$

where  $x_k^m, y_k^m, z_k^m$  are the coordinates of the object (longitude, latitude, altitude) obtained by GPS measurement.

Observation matrix:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}. \quad (20)$$

At the beginning of the Kalman filter algorithm, you need to perform the initialization of system state variables  $\hat{\mathbf{x}}_0^-$ , state error estimation covariance matrix  $P_0^-$ , process noise covariance matrix  $Q$ , measurement noise covariance matrix  $R$ .

System state initialization: the first coordinate measurement is assigned to state variables  $x_k, y_k, z_k$  and the velocity components  $v_{x,k}, v_{y,k}, v_{z,k}$  are set to zero. Since the measurement of GPS coordinates is carried out with an error of 2–5 m, the diagonal elements of the matrix  $P_0^-$  corresponding to the coordinates are given by values from this range, and the diagonal elements associated with the components of the velocity are given equal to zero (MRP is not movable). All non-diagonal elements of the matrix  $P_0^-$  are set to zero.

The initialization of the process noise covariance matrix  $Q$  and the measurement noise matrix  $R$  is performed based on data on the accuracy of the mathematical model of the system and measurement errors obtained from the sensors.

After that, a step-by-step algorithm for recovering measurement data based on the Kalman filter is performed:

1. Extrapolation of the MRP state taking into account the velocity.

2. If GPS measurement data is available, a correction based on the Kalman filter is performed, and the current assessment of the state variables is stored. If there are no measurement data, the correction is not performed, but only the results of the extrapolation of state variables are stored as their current estimate.

**Implementation of an algorithm for recovering lost navigation data for MRP.** Based on the PCA algorithm with a sliding window, the program `Restore_Missing_Data_PCA` is developed. The program is written in C for the ESP32 microcontroller. The specified microcontroller has sufficient computing resources for streaming processing (recovery) of data in real time.

The steps of the algorithm are as follows:

- receives the input sequence of GPS data (altitude, longitude, latitude);
- for each time step  $T_s$  forms a sliding window of a given size  $N$ ;
- calculates the covariance matrix for this window;
- calculates eigenvalues and eigenvectors of a symmetric covariance matrix using the Jacobi method;
- for a specified number of main components (this parameter is specified) recovers the lost data. It is executed if no input vector is received at the current time step.

To check the functioning of the program, measurement numbers with missing data were randomly generated. These data were restored and the error was calculated for them according to the expression

$$Err_i = \sqrt{\sum_{j=1}^3 (x_{i,j}^{rest} - x_{i,j}^m)^2}, \quad (21)$$

where is the restored  $x_{i,j}^{rest}$   $i$ -th input vector, the index  $x_{i,j}^m$  indicates the values of height, longitude, latitude, respectively, is the  $i$ -th input vector obtained from GPS. The study was performed to recover 6 input vectors from the  $N\_Point=50$  sample for different sliding window sizes  $N\_W=10, 20, 30, 40$ .

Below are the results of the program execution for the specified values and the obtained error values for the recovery of missed GPS data.

For a sliding window measuring 10 samples, the sliding window was formed 40 times, and the main components were found.

-- WINDOW\_SIZE = 10 --

| Index | Altitude | Longitude | Latitude  | Error |
|-------|----------|-----------|-----------|-------|
| ----- |          |           |           |       |
| 11    | 353.6    | 24.022163 | 49.827764 | 0.322 |
| 14    | 353.4    | 24.022201 | 49.827757 | 0.714 |
| 17    | 353.2    | 24.022238 | 49.827749 | 0.915 |
| 24    | 352.4    | 24.022353 | 49.827723 | 1.302 |
| 29    | 351.9    | 24.022427 | 49.827705 | 1.200 |
| 36    | 352.5    | 24.022533 | 49.827679 | 0.322 |

For a sliding window measuring 20 samples, the sliding window was formed 30 times, and the main components were found.

-- WINDOW\_SIZE = 20 --

| Index | Altitude | Longitude | Latitude  | Error |
|-------|----------|-----------|-----------|-------|
| ----- |          |           |           |       |
| 11    | 353.3    | 24.022205 | 49.827754 | 0.018 |
| 14    | 353.3    | 24.022205 | 49.827754 | 0.618 |
| 17    | 353.3    | 24.022205 | 49.827754 | 1.018 |
| 24    | 352.9    | 24.022274 | 49.827740 | 1.819 |
| 29    | 352.5    | 24.022341 | 49.827725 | 0.628 |
| 36    | 352.4    | 24.022443 | 49.827701 | 0.200 |

For a sliding window measuring 30 samples, the sliding window was formed 20 times, and the main components were found.

-- WINDOW\_SIZE = 30 --

| Index | Altitude | Longitude | Latitude  | Error |
|-------|----------|-----------|-----------|-------|
| 11    | 352.8    | 24.022279 | 49.827738 | 0.456 |
| 14    | 352.8    | 24.022279 | 49.827738 | 0.144 |
| 17    | 352.8    | 24.022279 | 49.827738 | 0.544 |
| 24    | 352.8    | 24.022279 | 49.827738 | 1.744 |
| 29    | 352.8    | 24.022279 | 49.827738 | 0.256 |
| 36    | 352.6    | 24.022378 | 49.827716 | 0.442 |

For a sliding window with a size of 40 samples, the sliding window was formed 10 times, and the main components were found.

-- WINDOW\_SIZE = 40 --

| Index | Altitude | Longitude | Latitude  | Error |
|-------|----------|-----------|-----------|-------|
| 11    | 352.8    | 24.022362 | 49.827718 | 0.526 |
| 14    | 352.8    | 24.022362 | 49.827718 | 0.074 |
| 17    | 352.8    | 24.022362 | 49.827718 | 0.474 |
| 24    | 352.8    | 24.022362 | 49.827718 | 1.674 |

**Table 1.** Results of testing the algorithm for recovering lost navigation data

| Parameters<br>(window size) | Minimum error | Maximum error | Processing Time<br>(Platform 1), msec | Processing Time<br>(Platform 2), msec |
|-----------------------------|---------------|---------------|---------------------------------------|---------------------------------------|
| 10                          | 0.322         | 1.302         | 3704 / 1457                           | 12                                    |
| 20                          | 0.018         | 1.819         | 4309 / 1654                           | 15                                    |
| 30                          | 0.144         | 1.744         | 4009 / 1519                           | 14                                    |
| 40                          | 0.326         | 1.674         | 2839 / 1065                           | 11                                    |

**Development and testing of tools for recovering lost navigation data in mobile robotic platforms.** To solve the problems of intelligent measurement of motion parameters and determination of the spatial orientation of ground MRPs, it is necessary to use a set of sensors, each of which measures a separate parameter of the MRP movement, and additional qualitative and quantitative values of motion parameters can be obtained by taking into account their change. Thus, each sensor must have separate computing tools for processing the received data, which, on the one hand, have the appropriate interfaces (SPI, I2C, Serial), and on the other hand, sufficient computing power to ensure interaction with sensors and implement the necessary data pre-processing in real time. To provide complex intelligent processing, computing resources of modern SoCs can be used.

The structural diagram of the means for measuring motion parameters and determining spatial orientation for ground MRPs is shown in Fig. 1. Its main components are:

- ESP32-C3 microcontroller;
- GNSS module type M10Q-5883;
- accelerometer and gyroscope module MPU-6050;
- QMC5883L digital compass module.

A 32-bit ESP32-C3 microcontroller developed by Espressif Systems is used for the collection and preliminary processing of navigation data. The main components are: a single-core RISC-V core operating at a frequency of up to 160 MHz; memory – 400 KB RAM, 4 MB flash memory; wired interfaces – GPIO, UART, SPI, I2C, I2S, PWM, ADC; wireless interfaces – Wi-Fi (2.4GHz 802.11 b/g/n) and Bluetooth 5.0 (LE). The specified microcontroller is used to

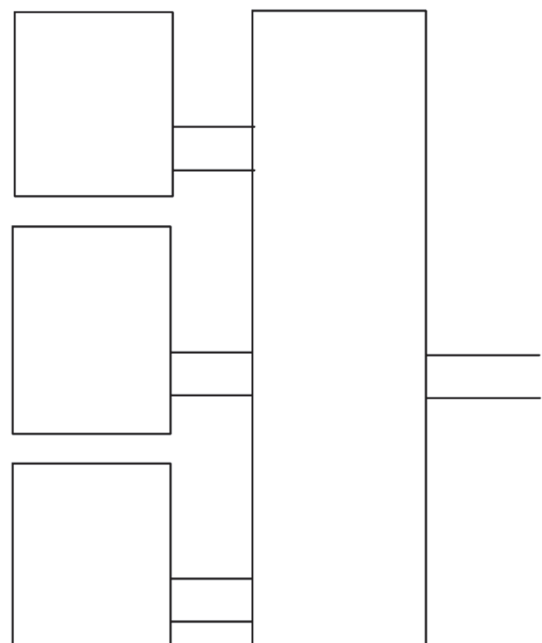
|    |       |           |           |       |
|----|-------|-----------|-----------|-------|
| 29 | 352.8 | 24.022362 | 49.827718 | 0.326 |
| 36 | 352.8 | 24.022362 | 49.827718 | 0.574 |

As the analysis shows, the errors in the recovery of missing GPS data for different sliding window sizes for the same input vectors differ insignificantly and range from 1.302 to 1.819 for maximum values.

Table 1 summarizes the test results.

Two computing platforms were used in the testing process: platform 1, based on the ESP32-C3/ESP32-WROOM microcontroller, and platform 2, Intel i7 7700. A total of 100 consecutive calculations were performed for each parameter value. As the analysis shows, even in the worst case, the recovery time of lost data using the ESP32-C3 microcontroller is 43 msec, which, in the case of one measurement per second for the real rate of GNSS data arrival, is enough to provide a real-time mode. The ESP32-WROOM-based computing platform has more than a twofold performance advantage.

process data from the specified navigation sensors, since it has the necessary I2C and UART interfaces.



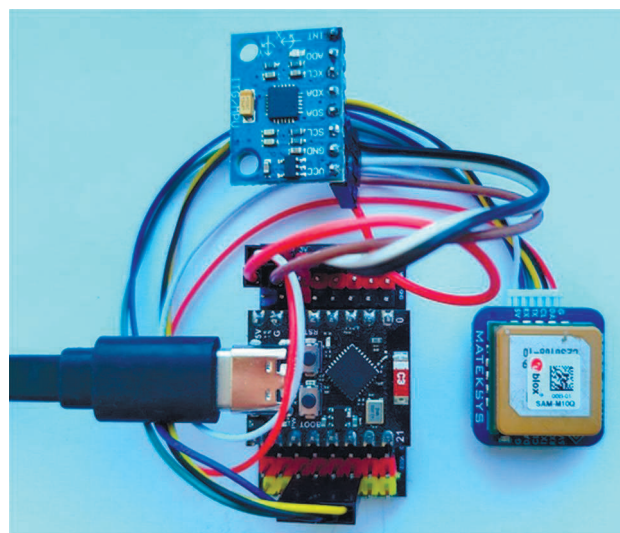
**Fig. 1.** Structural diagram of a means of measuring motion parameters and determining spatial orientation for ground-based MRPs

GNSS satellite positioning systems are used today to obtain operational information about the location of the

MRP. Several alternative systems are now actively used: GPS, BeiDou, Galileo, and Glonass. The M10Q-5883 GPS module adopts a GNSS module based on the u-blox M10 series chip and provides high sensitivity and data acquisition frequency for all L1 GNSS signals. It supports simultaneous reception of four GNSS (GPS, GLONASS, Galileo, and BeiDou). The large number of visible satellites allows the receiver to select the best signals to receive positioning data, which ensures the availability of positioning data, especially in challenging environments such as “urban canyons”. A u-blox Super-S (Super-Signal) technology provides high radio sensitivity and can improve dynamic positioning accuracy without line-of-sight conditions. The 15×15 mm square high-gain patch antenna provides the best balance between performance and small size. The I2C interface is used to operate the magnetometer QMC5883L. The direction and orientation in the MRP space are determined using a digital compass QMC5883L – 3-axis module for magnetic sensing at a low field level.

The GY-521 module has an integrated combined sensor that integrates a gyroscope and accelerometer with high-accuracy measurement of angular velocity and acceleration of the MPU-6050 type. High measurement accuracy is realized using a 16-bit ADC and built-in 1024-byte FIFO memory. An I2C interface is used for communication. Based on the MPU-6050, effective systems for stabilizing and tracking the movement of MRP are being built.

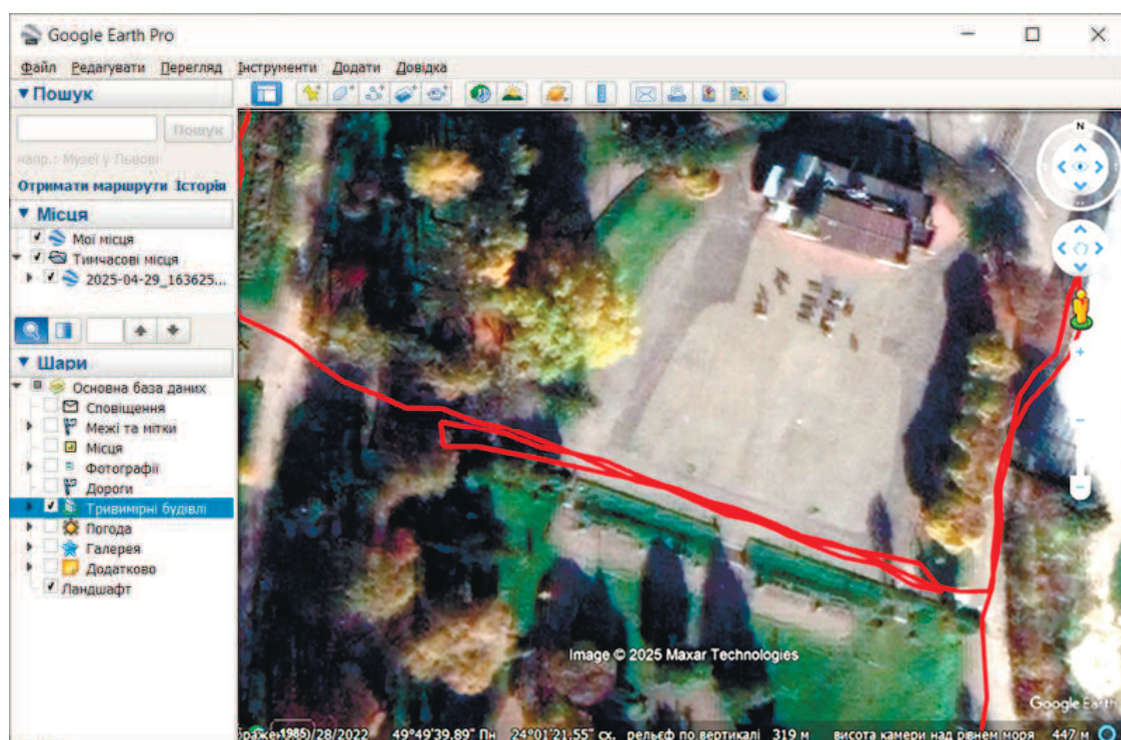
The general view of the means for measuring motion parameters and determining spatial orientation for ground-based MRP is shown in Fig. 2. The use of these modules and the ESP32 microcontroller ensured the possibility of its implementation in compact dimensions of 40×40 mm.



**Fig. 2.** General view of the means for measuring motion parameters and determining spatial orientation for ground-based MRCs

The module outputs a data stream via the NMEA-0183 protocol to the built-in UART. However, only geospatial coordinates are needed to position the MRP. Positioning tags \$GPGGA from the received NMEA data stream processed by the ESP32-C3 microcontroller to obtain longitude, latitude and altitude parameters. At the same time, data from the gyroscope / accelerometer is read. The obtained positioning data is transmitted to the MRP control microcomputer.

The developed module was used in real movement conditions to test the process of reproducing lost data. The Google Earth visualization of the resulting tracks is shown in Fig. 3.



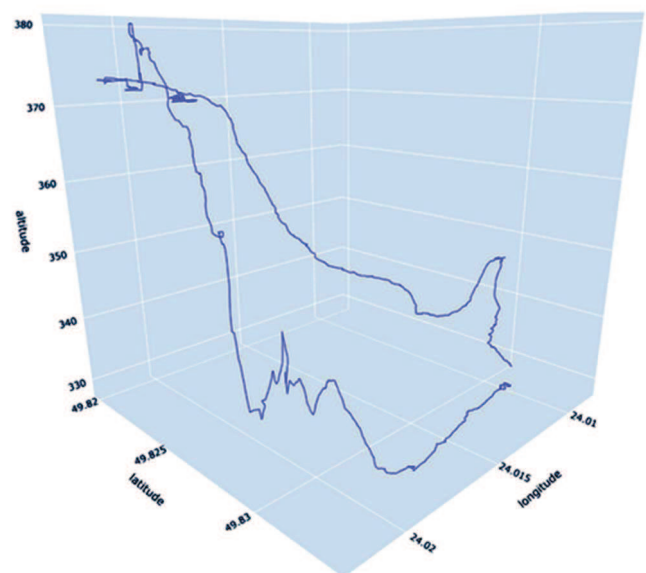
**Fig. 3.** Visualization of a track fragment in Google Earth using the developed module

The data received by the module was recorded by the smartphone using a specialized Serial USB Terminal application through a USB-UART adapter. This data has been converted to a KML file format for rendering in the Google Earth app on PC.

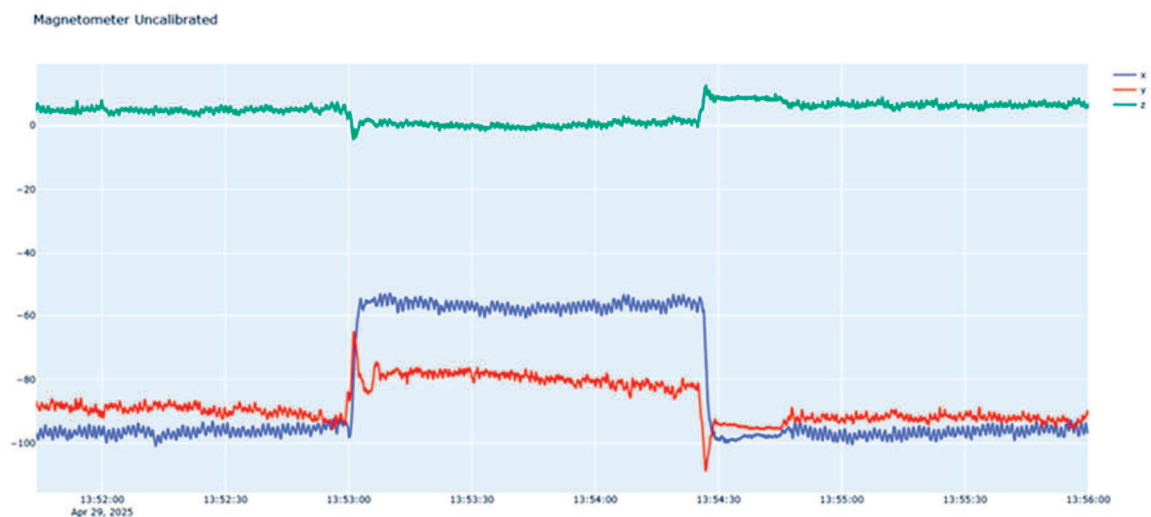
The data was stored in CSV format and then visualized using a Python program using the plotly, pandas, and scipy libraries. The altitude measurement data was further filtered to eliminate emissions (Fig. 4).

Visualization of data from the magnetometer module QMC5883L for one fragment of the track (Fig. 3) is shown in Fig. 5, and from the MPU-6050 gyroscope / accelerometer in Fig. 6. The measurement data was further filtered to eliminate emissions.

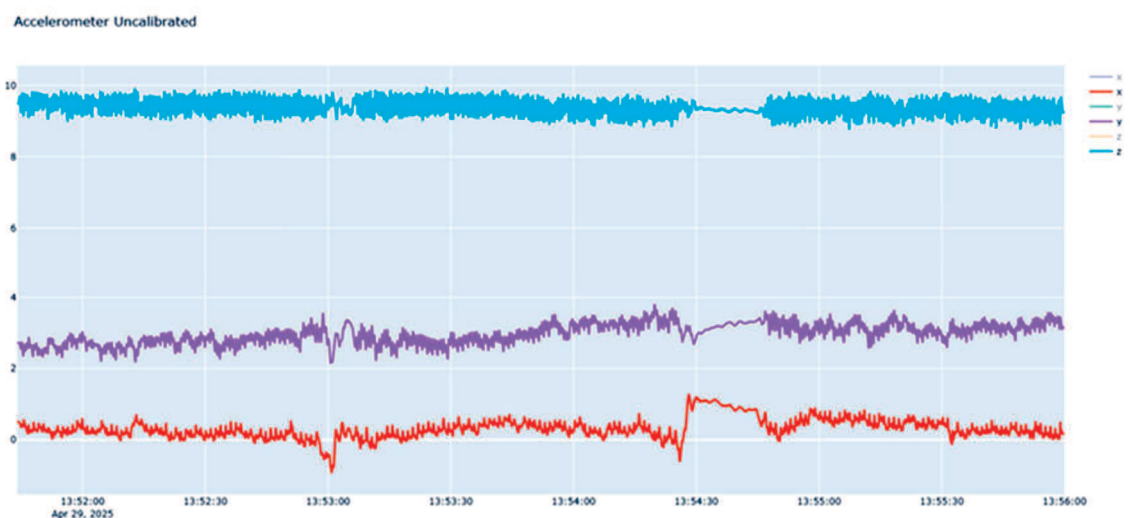
The data obtained from the M10Q-5883 GPS module, an electronic compass based on the QMC5883L magnetometer, and an accelerometer based on the MPU-6050 are pre-processed by the ESP32-C3 microcontroller and transmitted to the microcomputer through the UART interface for further processing in the MRP control system.



**Fig. 4.** Visualization in longitude-latitude-height coordinates of the received track using the developed module



**Fig. 5.** Visualization of QMC5883L magnetometer data



**Fig. 6.** Visualization of MPU-6050 gyroscope/accelerometer data

Additionally, the speed of execution of the developed algorithm for recovering lost data on various computing platforms was tested. Implementing the algorithm in the C

language provided a prompt transfer to several micro-controller and SoC platforms. Table 2 shows the results of testing the algorithm execution speed on different platforms.

**Table 2.** Execution speed of the developed algorithm on computing platforms

| Computing platform     | Calculation time, msec | Processor frequency, MHz | Performance indicator, sec·MHz |
|------------------------|------------------------|--------------------------|--------------------------------|
| Arduino UNO            | 9679                   | 16                       | 154.864                        |
| ESP8266                | 1458                   | 80                       | 116.64                         |
| ESP32-WROOM            | 381                    | 160                      | 60.96                          |
| ESP32-C3               | 968                    | 160                      | 154.88                         |
| Orange Pi Zero         | 78                     | 1000                     | 78                             |
| Raspberry Pi 4 Model B | 23                     | 1800                     | 41.4                           |
| Intel i7 7700          | 13                     | 3600                     | 46.8                           |

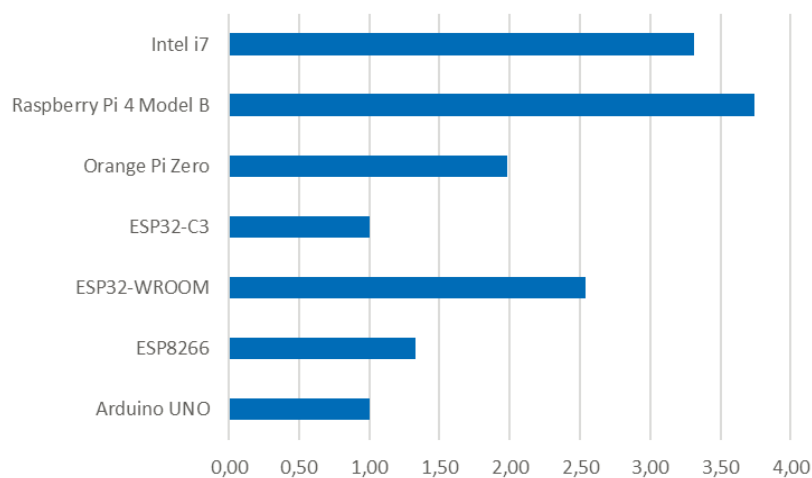
To compare the results, a total of 100 cycles of recovery of lost data were performed. The values of the recovery algorithm parameters and the data for processing on all platforms were the same. The algorithm was executed in one thread; parallelization was not used.

The efficiency indicator of the computing platform was calculated as the product of the calculation time by the frequency of the processor, taking into account the fact that the number of operations of the executed algorithm on each platform did not change, and the execution time is inversely proportional to the frequency of a particular processor. That

is, the indicator compares platforms' efficiency at the architecture level, or the number of cycles the processor needs to execute the algorithm.

For these reasons, the conditional performance factor of the platform was calculated (Fig. 7).

The conditional performance factor of the platform was calculated taking into account the data in Table 2 as the ratio of the efficiency indicator to the same indicator for the Arduino UNO platform. Obviously, for platforms using modern processors, such a conditional performance factor is higher, but not to such a large extent as one might hope.



**Fig. 7.** Conditional platform productivity factor

**Discussion of the results obtained.** Analysis of the problems of recovering lost navigation data in mobile robotic platforms requires solving the issue of choosing the appropriate algorithm and element base and components for such implementation. Based on the analysis, it is proposed to use the PCA main component analysis algorithm to implement the problem of recovering lost navigation data in the case of a mobile MRP, which involves processing a fixed set of last measurements with  $N$  samples with a sliding window and using only  $k$  eigenvectors that correspond to the largest eigenvalues. The use of the algorithm based on the Kalman filter for the recovery of missing data is promising, but further implementation on the microcontroller platform is required.

Components for creating tools for recovering lost navigation data for MRP are considered, and a set of navigation sensors using a GPS/GNSS-based coordinate determination module is defined. The created tools are implemented using the ESP32-C3 microcontroller, GNSS module type M10Q-5883, which contains a digital compass module QMC5883L and an accelerometer / gyroscope module MPU-6050. Software tools for the visualization of the accumulated data have been developed.

The development of tools for recovering lost navigation data for the MRP has been carried out. For the construction of hardware, the ESP32 microcontroller was used, and the performance of individual implementations of computing platforms was compared.

It is determined that the ESP32-WROOM platform has a more than twofold performance advantage over ESP32-C3. Testing the developed tools for recovering lost navigation data for MRP has been performed. Analysis of the test data shows that even in the worst case, the recovery time of lost data using the ESP32-C3 microcontroller is 43 msec, which, in the case of one measurement per second for the real rate of GNSS data arrival, is enough to provide a real-time mode.

*The scientific novelty of the research results* is that the algorithm for the recovery of lost navigation data, which is focused on use in mobile robotic platforms, which uses the method of principal components analysis PCA, which involves the processing of a fixed set of last measurements with  $N$  samples with a sliding window, has been improved and thus ensures the processing of data on a computing platform with limited resources.

*The practical significance of the research results* is the use of the developed algorithm for recovering lost navigation data in mobile robotic platforms, based on the principal components analysis PCA method, which provides data processing on a computing platform with limited resources and allows processing streaming data of MRP coordinates in real time.

## Conclusions

Modern algorithms for recovering lost navigation data in mobile robotic platforms have been analyzed, and an algorithm based on the principal component analysis method PCA has been selected for implementation, which, with low computational complexity, provides sufficient accuracy of data recovery and can be implemented on a computing platform with limited resources.

A structural diagram of a means for measuring motion parameters and determining spatial orientation for ground MRP has been developed. It is determined that the main components of recovering lost navigation data for MRP are a set of navigation sensors using a GPS/GNSS-based coordinate determination module. Data recovery tools have been created using the ESP32-C3 microcontroller, GNSS module type M10Q-5883, which contains a digital compass module QMC5883L and an accelerometer / gyroscope module MPU-6050.

Debugging and testing the developed tools for recovering lost navigation data for MRP have been performed. Analysis of the test results shows that the platform using the ESP32-C3 microcontroller provides data processing in 43 milliseconds. For the GNSS data arrival rate of one measurement / sec, this is sufficient to provide real-time mode.

## References

- [1] Tsmots I. G., Opotyak Yu. V., Shtogrinets B. V., Dzyuba A. O., Oliynyk Yu. Yu. (2023). Basic structure of the system of neurofuzzy control of a group of mobile robotic platforms. *Ukrainian Journal of Information Technologies*, Vol. 5, No. 1, 77–85. <https://doi.org/10.23939/ujit2023.01.077>
- [2] Rejeb, A., Rejeb, K., Simske, S. J., & Treiblmaier, H. (2023). Drones for supply chain management and logistics: a review and research agenda. *International Journal of Logistics Research and Applications*, vol. 26, iss. 6, 708–731. DOI: 10.1080/13675567.2021.1981273,
- [3] Poulet, Guérin, F., and Guinand, F., (2021). Experimental and Simulation Platforms for Anonymous Robots Self-Localization, 29th Medit. Conf. on Control and Automation (MED), PUGLIA, Italy, 2021, 949–954. DOI: 10.1109/MED51440.2021.9480244
- [4] Izonin, I.; Kryvinska, N.; Tkachenko, R.; Zub, K. (2019). An approach towards missing data recovery within IoT smart system. *Procedia Comput. Sci. J.*, 155, 11–18.
- [5] Vedavalli, P., & Ch, D. (2023). A Deep Learning Based Data Recovery Approach for Missing and Erroneous Data of IoT Nodes. *Sensors*, 23(1), 170. <https://doi.org/10.3390/s23010170>
- [6] Gupta, G. P., Khandare, H. (2022). Missing Data Recovery Using Tensor Completion-Based Models for IoT-Based Air Quality Monitoring System. In: Karuppusamy, P., García Márquez, F. P., Nguyen, T. N. (eds) *Ubiquitous Intelligent Systems. ICUIS 2021. Smart Innovation, Systems and Technologies*, vol. 302. Springer, Singapore. [https://doi.org/10.1007/978-981-19-2541-2\\_33](https://doi.org/10.1007/978-981-19-2541-2_33)
- [7] Mondal, A., Das, M., Chatterjee, A., and Venkateswaran, P. (2020). Recovery of Missing Sensor Data by Reconstructing Time-varying Graph Signals, 30th European Signal Processing Conference (EUSIPCO), Belgrade, Serbia, pp. 2181–2185. DOI: 10.23919/EUSIPCO55093.2022.9909940
- [8] Cheng, H., Wu, L., Li, R. *et al.* (2021). Data recovery in wireless sensor networks based on attribute correlation and extremely randomized trees. *J Ambient Intell Human Comput* 12, 245–259. DOI: <https://doi.org/10.1007/s12652-019-01475-z>
- [9] Ghorbani, B., Krishnan, S., and Zou, J. (2021). Debiasing stochastic gradient descent to handle missing values. *Proc. 38th Int. Conf. Mach. Learn. (ICML)*, pp. 3748–3758.
- [10] Tkachenko, R., Mishchuk, O., Izonin, I., Kryvinska, N., and Stoliarchuk, R. (2019). A non-iterative neural-like framework for missing data imputation. *Procedia Computer Science*, vol. 155, pp. 319–326.
- [11] Tkachenko, R., Izonin, I. (2019). Model and principles for the implementation of neural-like structures based on geometric data transformations. *Advances in Computer Science for Engineering and Education*, Z. Hu, S. Petoukhov, I. Dychka, and M. He, Eds., Cham: Springer, 2019, pp. 578–587. DOI: 10.1007/978-3-319-91008-6\_58.
- [12] Lippi, V., Ceccarelli, G. (2019). Incremental principal component analysis: Exact implementation and continuity corrections. *Proceedings of the 16th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2019)*, Prague, Czech Republic, 2019, pp. 473–480. DOI: 10.5220/0007743604730480.
- [13] Vaswani, N., Narayanamurthy, P. (2018). Static and dynamic robust PCA and matrix completion: A review. *arXiv preprint arXiv:1803.00651*, 2018 [Online]. <https://arxiv.org/abs/1803.00651>.
- [14] Ghahramani, Z., Hinton, G. E., (1996). Parameter estimation for linear dynamical systems. University of Toronto Technical Report CRG-TR-96-2, 1996.
- [15] Tsmots, I., Skorokhoda, O., Tesliuk, T. & Rabyk, V. (2016). Designing features of hardware and software tools for intelligent processing of intensive data streams processing. *IEEE First International Conference on Data Streams and Processing, DSMP*, Lviv, pp. 332–335. DOI: 10.1109/DSMP.2016.7583570

## **ВІДНОВЛЕННЯ ВТРАЧЕНИХ НАВІГАЦІЙНИХ ДАНИХ У МОБІЛЬНИХ РОБОТОТЕХНІЧНИХ ПЛАТФОРМАХ**

Мобільні роботизовані платформи (МРП) все ширше застосовують у різних сферах людської діяльності. Важливим завданням під час їх використання є визначення просторової орієнтації, вимірювання параметрів руху МРП тощо. Однією із проблем, що виникає під час визначення навігаційних даних та інших вимірюваних параметрів, є їх втрата на певному часовому інтервалі, наприклад, через перешкоди або тимчасову втрату видимості навігаційних супутників GNSS. Однак для функціонування навігаційних компонентів бортових радіоелектронних пристроїв для вимірювання параметрів руху та визначення просторової орієнтації МРП потрібна наявність первинної навігаційної інформації без втрат та у режимі реального часу. Тому необхідно відновлювати втрачені навігаційні дані, особливо у випадку МРП, використовуючи бортові засоби, які мають обмежену обчислювальну продуктивність. Проаналізовано сучасні алгоритми відновлення втрачених навігаційних даних і визначено, що у публікаціях недостатньо уваги звернено на реалізацію вказаних алгоритмів з урахуванням обмежень вбудованих систем. У мобільних робототехнічних платформах на основі виконаного аналізу вибрано для реалізації алгоритм із використанням методу аналізу головних компонент (PCA), який за невеликої обчислювальної складності забезпечує достатню точність відновлення даних. Використання розробленого алгоритму у мобільних робототехнічних платформах забезпечує опрацювання даних на обчислювальній платформі з обмеженими ресурсами і дає змогу обробляти потокові дані про координати МРП у режимі реального часу. Сучасні мікроконтролери та системи на кристалі (SoC) дають можливість вирішити завдання відновлення втрачених навігаційних даних з урахуванням обмежень щодо маси, габаритів, енергоспоживання тощо. Розроблено структурну схему засобу вимірювання параметрів руху та визначення просторової орієнтації для наземних МРП. Визначено, що основними компонентами засобу є набір навігаційних давачів із використанням модуля визначення координат на основі GPS/GNSS. Створено засоби для відновлення даних із використанням мікроконтролера ESP32-C3, модуля GNSS типу M10Q-5883, який додатково містить модуль цифрового компаса QMC5883L та модуля акселерометра і гіроскопа MPU-6050. Виконано налагодження та тестування розроблених засобів відновлення втрачених навігаційних даних для МРП. Аналіз результатів тестування показує, що платформа з використанням мікроконтролера ESP32-C3 забезпечує опрацювання даних за 43 мс. Для темпу надходження даних GNSS одне вимірювання за секунду цього достатньо для забезпечення режиму реального часу.

**Ключові слова:** навігаційні давачі, параметри руху, точність вимірювання, мікроконтролерна платформа.

---

### **Інформація про авторів:**

**Цмоць Іван Григорович**, д-р техн. наук, професор, кафедра автоматизованих систем управління. **Email:** ivan.h.tsmots@lpnu.ua; <https://orcid.org/0000-0002-4033-8618>

**Теслюк Василь Миколайович**, д-р техн. наук, професор, кафедра автоматизованих систем управління. **Email:** vasyi.m.teslyuk@lpnu.ua; <https://orcid.org/0000-0002-5974-9310>

**Рабик Василь Григорович**, канд. техн. наук, доцент, кафедра радіофізики та комп'ютерних технологій. **Email:** vasyi.rabik@lnu.edu.ua; <https://orcid.org/0000-0003-2655-0812>

**Опотяк Юрій Володимирович**, канд. техн. наук, доцент, кафедра автоматизованих систем управління. **Email:** yurii.v.opotyak@lpnu.ua; <https://orcid.org/0000-0001-9889-4177>

**Цимбал Юрій Вікторович**, канд. техн. наук, доцент, кафедра автоматизованих систем управління. **Email:** yurii.v.tsymbal@lpnu.ua; <https://orcid.org/0000-0001-9119-6771>

**Пазюк Олександр Валерійович**, аспірант, кафедра автоматизованих систем управління, **Email:** oleksandr.v.paziuk@lpnu.ua; <https://orcid.org/0009-0004-2973-665X>

**Цитування за ДСТУ:** Цмоць І. Г., Теслюк В. М., Рабик В. Г., Опотяк Ю. В., Цимбал Ю. В., Пазюк О. В. Відновлення втрачених навігаційних даних в мобільних робототехнічних платформах. *Український журнал інформаційних технологій*. 2025, т. 7, № 1. С. 97–107.

**Citation APA:** Tsmots, I. G., Teslyuk, V. M., Rabyk, V. G., Opotyak, Y. V., Tsymbal, Y. V., & Paziuk, O. V. (2025). Recovery of lost navigation data in mobile robotic platforms. *Ukrainian Journal of Information Technology*, 7(1), 97–107. <https://doi.org/10.23939/ujit2025.01.097>